The Visual Computer
https://doi.org/10.1007/s00371-024-03303-8

ORIGINAL ARTICLE

Shape generation via learning an adaptive multimodal prior

Xianglin Guo[1,2] · **Mingqiang Wei**[2]

Accepted: 20 January 2024

Abstract
Significant interest and progress have been drawn to the recent advancements in image creation using deep generative model, but the field of automatic three-dimensional shape creation is largely under-developed and inspires a great deal of research activity across a wide variety of disciplines. We add a new kind of previously named variational mixture of posteriors into the adversarial network using geometric data described as volumetric grids. Our main contribution is the introduction of a new type of prior called variational mixture of posteriors prior into the adversarial network, dubbed VAMPPRIOR- 3DGAN, in a mathematic principled way. Specifically, we leverage an encoder as a regularizer to penalize missing modes, while introduce a variational mixture of posterior prior as the latent variable distribution of GAN to dynamically and adaptively update its prior distribution. The key intuition behind this architecture is that the latent variables should retain information about the data to minimize the undue impact of the prior assumptions. This seemingly simple modification to the GAN framework is surprisingly effective and results in models which enable diversity in generated samples, although trained with limited data. Realistic 3D objects can be easily generated by sampling the VampPrior-3DGAN's latent probabilistic manifold. For validation, we apply our method on tasks from the fields of three-dimensional volumetric generation, reconstruction from a single RGB image and partial shape completion from a single perspective view, and show that it is on par with or outperforms the state-of-the-art approaches, both quantitatively and qualitatively.

Keywords Shape generation · Variational mixture of posteriors prior · GAN · Diversity

1 Introduction

The three-dimensional (3D) content creation involves coordinated work with artists, modelers, designers and animators. One of the key challenges of the industry is to create a seamless pipeline. The virtual reality, robotics and computer graphics industry understands this process to be time-consuming and cumbersome. Many collection of 3D models are created and published on online repositories such as SHAPENET [1] and MODELNET [2]. These online repositories contain large-scale useful information that details about textures, styles, structures and poses of object classes. This information can be helpful to designers in the modeling process. Thus, leveraging this information with tools can enforce data-driven constraints, providing completions of partially designed objects, or even through the synthesis of whole shape from an image or merely a random noise vector.

One of the key challenges faced by the current academic research is to develop algorithms that can understand, analyze and auto-generate 3D content. Generative models address this issue [3, 4]. Currently, one of the hottest topics in deep learning and computer vision is generative adversarial networks (GAN) [5–7] or variational auto-encoder (VAE) [8, 9] for shape generation. These generative models serve as a test-bed for high-quality representation learning, feature extraction and unsupervised recognition using probabilistic spaces and manifolds.

While deep generative model acts as a generic mathematical framework which is very flexible and provides immense expressive power, the performance of VAE or GAN-based methods still leaves much to be desired when faced with challenging conditions. A downside to the 3D-VAE [4] is that it uses direct mean squared error instead of an adversarial

✉ Mingqiang Wei
mqwei@nuaa.edu.cn

Xianglin Guo
x.guo@ahut.edu.cn

1 College of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243032, Anhui, China

2 Shenzhen Research Institute, Nanjing University of Aeronautics and Astronautics, Shenzhen 518038, Guangdong, China

Published online: 20 March 2024

network, so the network tends to produce unreliable reconstruction, corresponding to more blurry images in image generation. Moreover, choosing a too simplistic prior like the standard normal distribution is known to result in over-regularized models with only few active latent dimensions, as a result, with very poor hidden representations. 3D-GAN [3] is based upon the original GAN architecture and training approach, which is well known to suffer from instability. The coexistence of instability is that GAN can easily result in a problem called *missing mode*, that is, the generation of the network G in GAN will be easily confined to some modes, but not rich.

An important factor in the aforementioned problem is the lack of control on the discriminator during GAN's training. Inspired by the observation that the optimization objectives for supervised learning are more stable, we suggest adding a supervised signal as a regularizer on top of the target of the discriminator. In the recent literature [10–12], we have noted that prior distribution of latent variable in fact plays a crucial role in generative models. The selection of an appropriate prior for GAN is, however, not trivial. Specifically, choosing an appropriate prior depends on the following criteria:

- The prior should have an expressive distribution and is flexible enough to capture a-priori knowledge, e.g., by Gaussian mixture models (GMMs).
- The prior should retain information from real samples as much as possible to create a both richer and informative prior.

These insights helped us to introduce the variational mixture of posteriors prior (VampPrior) as the distribution of the latent variable. The multimodal nature benefits our prior to achieve superiority over many other simple priors in terms of training complexity and expressiveness. Specifically, in contrast to modeling prior directly as Gaussian mixture models (GMMs), the VampPrior consists of a mixture distribution with components given by variational posteriors conditioned on a set of learnable pseudo-inputs (Eq. 2). Importantly, the prior and posterior are coupled in VampPrior which *implicitly* incorporates the real data information into the generator network. This will simultaneously facilitate the variety and fidelity of generator-generated false samples. Therefore, the GANs can benefit dramatically from this novel prior. Moreover, incorporating pseudo-inputs into VampPrior prevents the GANs from bearing the risk of potential overfitting, which makes the model less expensive to train. Thus, such priors provide a good compromise between computational convenience and flexibility. Therefore, we suggest a novel generative model, called VampPrior- 3DGAN, integrating this concept with the above solution to compensate for the missing modes.

Contributions Our principal contributions are:

- We introduce the prior, named as variational mixture of posterior prior [11], as the distribution function of the latent variable in GAN. It can enrich the prior and encode more information of the real samples.
- Our encoder serving as a regularizer to penalize missing modes, thus, can improve GAN's training stability and sample qualities.
- We propose the VampPrior- 3DGAN which allows learning prior from data and thus modeling multimodality for 3D generation tasks. Our method can learn multimodal distribution and generate high-fidelity, diverse 3D shapes.
- We showcase that our models have favorable properties, like enjoying high compatibility in the network architecture from dynamic shape generation to image-to-shape reconstruction.

The rest of the paper is organized as follows. Section 2 describes some related studies. Section 3 details the proposed method. Section 4 presents and discusses the results and findings. Section 5 concludes the paper.

2 Related work

Shape generation There are two main schools of work on shape generation: (a) the native 3D school. This school is characterized by training directly on 3D datasets such as ShapeNet [1], and is based on 3D data from training to inference. Some interesting works are: 3D-GAN [3], GET3D [13], TextCraft [14] (which implements text conditioning), AutoSDF [15], MeshDiffusion [16], etc. Such methods tend to be fast and no problem at all in generating the categories present in the dataset. However, generating models that require 'imagination' remains challenging. (b) 2D upscaling school. Some approaches draw on the imaginative powers of 2D generative AI to drive the generation of 3D content. Work in this genre has recently made a lot of progress riding on breakthroughs in 2D deep generative models such as Imagen [17] and stable diffusion (SD [18]). OpenAI Point·E [19] takes text as input, generates an image using the 2D diffusion model GLIDE [20], and then generates a point cloud based on the input image using the 3D point cloud diffusion model. DreamFusion [21] generates multiple perspectives by a 2D generative model (e.g., Imagen [17]) and then reconstructs it with NeRF [22]. The authors came up with a GAN-like approach, where NeRF and Imagen iterate back and forth. The advantage is that there is more diversity. Magic3D [23] cleverly divides the reconstruction process into two steps: the first step uses only NeRF for rough shape generation, and the second step uses a differentiable rasterizer to refine.

Point cloud complementation The pioneering work PointNet [24, 25] has led to a boom in 3D vision that has generated many subsequent studies. PointNet directly inspires researchers to focus on learning global feature embeddings from point clouds for point cloud generation (PSG [26]) and point cloud complementation [27, 28]. However, predicting local details and thin shape structures remains a challenge. To address these challenges, research efforts [29–32] utilize multi-scale local point features to reconstruct complete point clouds with fine-grained geometric details. With the help of attention mechanism, some works have provided impressive complementation results [31, 32]. As a challenging conditional generation problem, point cloud complementation is still an open problem. In the last two years, the diffusion model (DDPM [33], stable diffusion [18]) has made many breakthroughs and has become a big hit in the field of 2D AIGC. In the field of 3D content generation, however, the diffusion model is just in the exploration stage. Luo & Hu [34] is the first to use DDPM for unconditional point cloud generation. Lyu [35] and Zhou et al. [36] further use conditional DDPM for point cloud completion. The major difference is that Zhou et.al. do not refine or upsample the coarse point cloud generated by DDPM like Zhaoyang Lyu does.

Single-view reconstruction Significant progress has been made in the field of single-view 3D reconstruction. The choice of representation is clearly critical to the quality of shape reconstruction. Volume-based methods [3, 37, 38] contain most of the work for single-view 3D reconstruction. Due to their memory consumption limitations, however, these methods lack the scalability needed to reconstruct high-resolution and detailed shapes. Point cloud-based methods [26, 39, 40] have a smaller memory footprint. But because point clouds lack topological connectivity information, they require post-processing to obtain shapes from these point clouds. Mesh-based methods [41–43] utilize connectivity information, but are greatly dependent on the underlying model they are deforming. In general, there is no direct way to change the topology of the underlying mesh during the reconstruction process to achieve better edge and mesh flow, which facilitates better shape quality. Methods based on implicit surfaces (level set [44], SDF [45], occupancy [46] and implicit fields [47]) have recently received increasing attention. This is due to their desirable properties in single-view 3D reconstruction. However, due to the inefficiency of sampling methods, implicit surface-based methods usually lead to over-smoothed reconstructions. It is worth noting that new implicit representations are constantly being proposed. One of the most promising representations is the neural radiance fields (aka NeRF [22, 48, 49]). In the near future, one will continue to witness various novel breakthroughs in NeRF research efforts, such as single-view 3D reconstruction [50] [51].

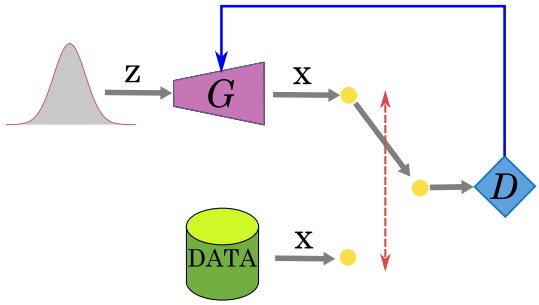

Fig. 1 Generative adversarial network

Regularization of latent space Another branch of related works, which perhaps more closely relates to our work, involves the regularization of GAN and the learning of a meaningfully structured latent space. [52] proposes two novel regularizer for the GAN training target: geometric metrics regularizer and mode regularizer. GM-GAN [53] incorporates a sparse prior-knowledge into the model, by sampling latent vectors using a multimodal probability distribution which better matches the sparse characteristics of the data space.

3 Methodology

In this section, we first provide two intuitions and then the corresponding solutions for our specific variant of 3D-GAN, dubbed VampPrior- 3DGAN, which mainly serves to address the mode collapse issue to improve samples' diversity.

Geometric intuition Canonically, the GAN Fig. 1 training procedure can be viewed as a non-cooperative two-player min–max game, in which the discriminator D attempts to distinguish real and generated examples, whereas the generator G tries to fool the discriminator by pushing the generated samples toward the direction of higher discrimination values.

We argue that training the discriminator D can be interpreted as training an evaluation metric on the sample space. Then, the generator G has to take advantage of the local gradient $\nabla \log D(G)$ provided by the discriminator to improve itself, namely to move toward the data manifold. The value function describing the GAN's min–max game can be formally formulated by

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{DATA}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1 - D \circ G(\mathbf{z}))] \quad (1)$$

where p_{DATA} is the data generating distribution, $\mathbf{z} \in \mathbb{R}^{d_{\mathbf{z}}}$ is a latent variable drawn from distribution $p(\mathbf{z})$ such as $\mathcal{N}(0; I)$ or $\mathcal{U}[-1; 1]$.

Upon comparison with the objective for the GAN generator, the optimization targets for supervised learning are more stable, from an optimization point of view. The difference

is clear: the optimization target for the GAN generator is a learned discriminator. While in supervised models, the optimization targets are distance functions with nice geometric properties. The latter usually provides much easier training gradients than the former.

These insights empower us to incorporate a supervised training signal as a regularizer on top of the discriminator target. Assume the generator $G(\mathbf{z}) : Z \rightarrow X$ generates samples by sampling first from a fixed prior distribution in space Z followed by a deterministic trainable transformation G into the sample space X. Together with G, we also jointly train an encoder $E(\mathbf{x}) : X \rightarrow Z$. Assume d is some similarity metric in the data space, we add $\mathbb{E}_{\mathbf{x}\sim p_{\text{DATA}}}[d(\mathbf{x}, G \circ E(\mathbf{x}))]$ as a metric regularizer.

The geometric motivation for this metric regularizer is straightforward. We are trying to match the generated manifold to the real data manifold by geometric distances, in addition to the gradient provided by the discriminator D. The idea of adding an encoder is equivalent to first training a point to point mapping $G(E(\mathbf{x}))$ between the two manifolds and then trying to minimize the expected distance between the points on these two manifolds.

In addition to the metric regularizer, we propose a prior regularizer intuition to further penalize missing modes.

Latent prior intuition In traditional GANs, the optimization target for the generator is the empirical sum $\sum_i \nabla \log D(G(\mathbf{z}_i))$. The missing mode problem is caused by the conjunction of two facts: (1) the areas near missing modes are rarely visited by the generator, by definition, thus providing very few examples to improve the generator around those areas [52], and (2) both missing modes and non-missing modes tend to correspond to a high value of D, because the generator is not perfect so that the discriminator can take strong decisions locally and acquire a high value of D even near non-missing modes. For most $\mathbf{z}$, the gradient of the discriminator $\nabla \log D(G(\mathbf{z}))$ implicitly pushes the density of the generator distribution toward the major mode. Only when $G(\mathbf{z})$ is very close to the minor mode can the generator get gradients to push itself toward this minor mode. However, it is possible that such $\mathbf{z}$ is of low or zero probability in the prior distribution $p_{\mathbf{z}}(\mathbf{z})$. We argue that this problem can be solved by two ways.

First solution: Increase the depth of generator network. We argue that the rationale why we would like to extend the depth of the generator is because the vanilla 3D-GAN [3], in essence, attempts to learn a mapping from a simplistic prior distribution $p_{\mathbf{z}}(\mathbf{z}) \sim \mathcal{N}(0, I)$ or $\mathcal{U}[-1; 1]$ to the complicated three-dimension data distribution. Such mapping requires a deep generator which can decode this single simplistic guassian (or uniform) to disentangle the underlying diverse modes or factors of variation within the real three-dimension data and encourage its samples' diversity. This, in turn, transfers into the requirement of large amounts of input data. However, when real three-dimension data is limited, yet originates from a diverse modality, increasing the network depth becomes infeasible. Furthermore, this also ends up in overfitting.

Second solution: Rather than raising the generator's depth, we instead recommend to enrich the prior distribution $p_{\mathbf{z}}(\mathbf{z})$ to strengthen the generator. Even if our central idea—utilizing a mixture model for latent variable— has been suggested in various papers mostly in the context of variational inference, for instance, GMVAE [10], yet, in the context of GANs, we have more considerations, i.e., at the top of a richer distribution, we would like our prior to take advantage of the information from the real samples. Concretely, if the prior is learned enough that to assign separate regions in the latent space to each datapoint, this effect should help the generator to decode a hidden representation to its corresponding voxel representation much easier. Combining this insight with the above approach meant to penalize the missing modes, we propose a hybrid architecture for the GAN objective, where we, in particular, propose the prior distribution of the latent vector of GAN as a variational mixture of posterior prior (VampPrior), which was first introduced by [11] to extend the VAE. The VampPrior consists of a mixture distribution with components given by variational posteriors conditioned on learnable pseudo-inputs:

$$p_\lambda(\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K} E(\mathbf{z} \mid \mathbf{u}_k), \tag{2}$$

where K is the number of components, and $\mathbf{u}_k$ is a parameterized vector referred as a *pseudo-input* which can be learned through backpropagation and can be thought of as hyperparameters of the prior, alongside parameters of the posterior ϕ, $\lambda = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k, \phi\}$.

Importantly, the VampPrior is multimodal. It makes the prior of GAN more expressive, thereby preventing the over-regularization of the prior. By incorporating pseudo-inputs, it prevents from potential overfitting once we pick $K \ll N$, which makes the model cost-effective to train. More specifically, the prior and posterior are coupled in VampPrior which implicitly incorporates the training data information into the generator network. Moreover, the learnable pseudo-inputs will fine-tune and tweak the prior to best suit the data distribution automatically.

Next, we describe this intuition more precisely.

3.1 VampPrior-3DGAN

In this section, we train a voxel-based VAE jointly with the GAN model. The model architecture is presented as Fig. 2.

Learned pseudo-input The previous observation suggests to prefer an expressive prior, so that the generator can easily decode a hidden representation on a voxel grid. In other

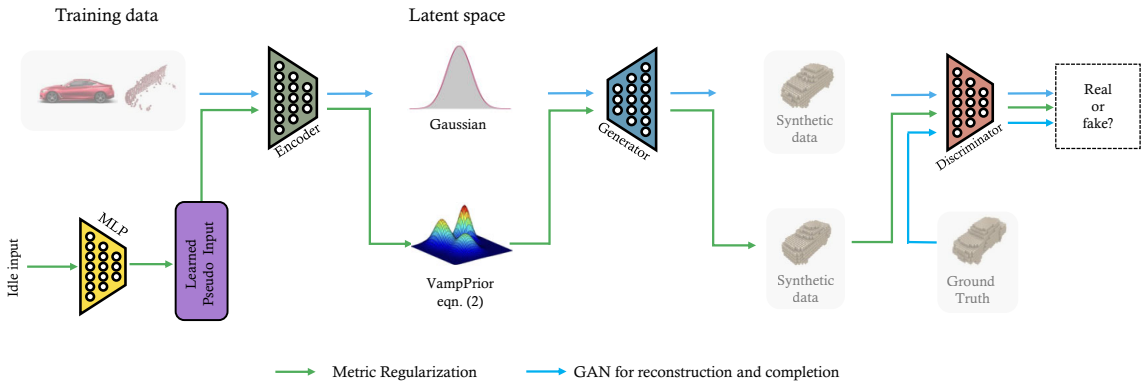

Fig. 2 Flowchart of VAMPPRIOR- 3DGAN for shape generation, completion and reconstruction

words, the encoder should be trained in order to have large variance. To achieve this effect, the VAMPPRIOR should be attracted by dissimilar pseudo-inputs and assigns separate regions within the latent space. Within this framework, we select the real samples as the weight of the deep neural network. A random noise vector $\mathbf{n}$ with the same dimension of $\mathbf{x}$ is then added to $\mathbf{x}$ element-wisely. Subsequently, we leverage the backpropagation procedure to tune these weights so as to learn these pseudo-inputs. The input of the deep neural network is an identity matrix with an order K, referred as idle-input. The schematic representation is in Fig. 3.

Loss Given a batch of training samples, we first pass these samples through encoder E and then reparameterize the output of the encoder to provide the input of the generator G. After jointly training the encoder and generator, we then dynamically sample from Eq. 2 on the latent space of encoder E. Sampling then from this dynamic and far more powerful mixture prior distribution in space Z, the generator G at this time can generate more diverse samples, which are then judged by the discriminator D. We then update the generator G and the discriminator D, alternately. The encoder and generator update their parameters by minimizing the following loss:

$$\begin{aligned}\mathcal{L}^{g}_{\text{gan}} &= \mathbb{E}_{\mathbf{x}\sim p_{\text{DATA}}}[\alpha_1 d(\mathbf{x}, G\circ E(\mathbf{x})) + \alpha_2 \log D(G\circ E(\mathbf{x}))] \\ &\quad - \mathbb{E}_{\mathbf{z}\sim p_\lambda(\mathbf{z})}[\log D(G(\mathbf{z}))], \qquad (3)\\ \mathcal{L}^{e}_{\text{vae}} &= \mathbb{E}_{\mathbf{x}\sim p_{\text{DATA}}}[\alpha_1 d(\mathbf{x}, G\circ E(\mathbf{x}))] \\ &\quad + \alpha_3 KL(p_\lambda(\mathbf{z}) \| E(\mathbf{z}\mid \mathbf{x})), \qquad (4)\end{aligned}$$

where $\mathbb{E}_{\mathbf{x}\sim p_{\text{DATA}}}[\log D(G\circ E(\mathbf{x}))]$ is the mode regularizer to encourage $G\circ E(\mathbf{x})$ to move toward a nearby mode of the data generating distribution and $KL(\cdot)$ is the KL divergence. α_1, α_2 and α_3 are the trade-off parameters controlling the fidelity and diversity of the fake samples. In this way, we can achieve fair probability mass distribution across different modes. The discriminator updates its parameters by minimizing the following loss:

$$\begin{aligned}\mathcal{L}^{d}_{\text{gan}} &= -\mathbb{E}_{\mathbf{x}\sim p_{\text{DATA}}}[\log(D(\mathbf{x})) + \log(1 - D(G\circ E(\mathbf{x})))] \\ &\quad - \mathbb{E}_{p_\lambda(\mathbf{z})}[1 - \log(D(G(\mathbf{z})))], \qquad (5)\end{aligned}$$

where $D(\cdot)$ is the probability of the input being a real volumetric shape, and $1 - D(\cdot)$ that of a synthetic one. The second component denotes the probability of the input being a synthetic shape generated by an encoder and generator network. Meanwhile, the third component denotes the one of the input being a synthetic shape generated directly from the VAMPPRIOR distribution. We implement D as a convolutional neural networks, whose last layer outputs the probability of the sample being a synthetic shape. For training this network, each mini-batch comprises of randomly sampled synthetic shape $\tilde{\mathbf{x}}$ from the VAMPPRIOR distribution $p_\lambda(\mathbf{z})$ and real shape $\mathbf{x}$. The target labels for the cross-entropy loss layer are 0 for every $\mathbf{x}_j$, and 1 for every $\tilde{\mathbf{x}}_i$. Then, the parameter of discriminator D for a mini-batch is updated by taking a stochastic gradient descent (SGD) step on the mini-batch loss gradient.

3.2 Shape reconstruction

We tackle the problem of reconstructing objects from RGB images in a novel training methodology. The reconstructive model enjoys the same architecture as used for shape generation (see Fig. 2 and Table 1). In contrast to shape generation, the input herein is an image. The encoder converts an image into a 200-dimensional vector of means and variances. We then dynamically fit a variational mixture of posterior model from the latent space of the encoder network to produce our noise vector. The latent vector is then passed through the generator network to generate a reconstructed object, which is then given to the discriminator to pass judgment on its validity.

The same generator and discriminator networks as used for shape generation (VAMPPRIOR-3DGAN) are implemented into this system and the encoder network is a simple

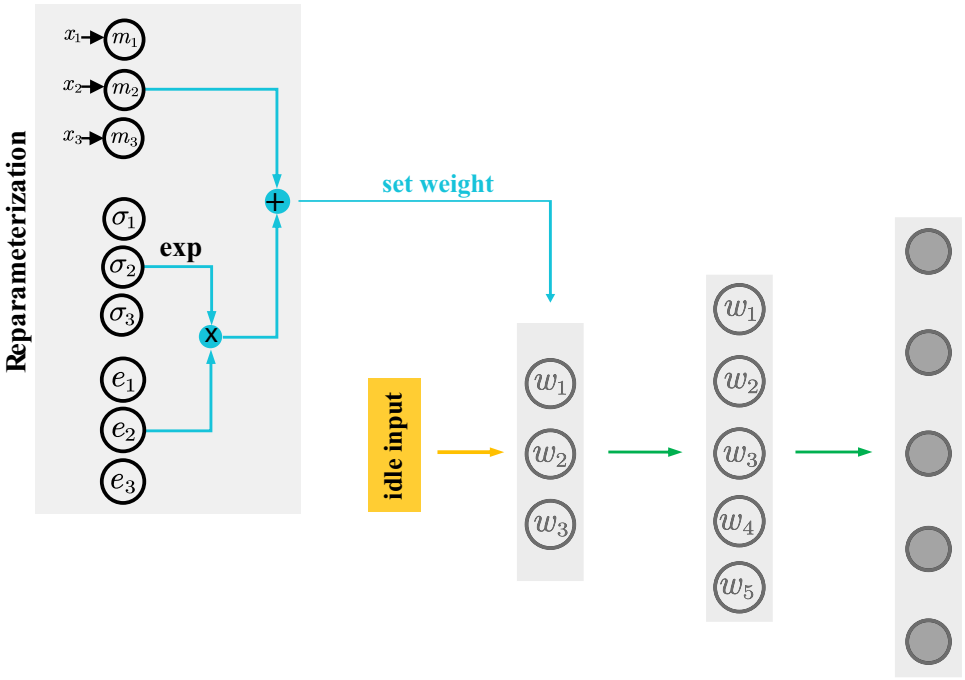

Fig. 3 Multilayer perception (MLP) for pseudo-inputs learning

five layer convolution neural network. During training, the discriminator and encoder networks are trained at every batch while the generator only learns every two batches. This last point is key to the integration of the systems, since if the encoder is not trained alongside the discriminator at every iteration, the system will not converge. This makes sense since both networks should learn similar features about the objects being created at approximately the same rate.

3.3 Partial shape completion

We discuss the issue of voxel form completion from sparse point clouds in this paper as well. This issue arises when only a single view of an individual object is given, or large parts of the object are occluded as in robotic applications. In order to form informed decisions (e.g., for path planning and navigation), it is of utmost importance to efficiently establish a representation of the environment which is as complete as possible. We accurately reconstruct an object's complete 3D shape and volume when presented with only a part of the object from a single perspective. We tackle this problem to show the generative power of our system, to highlight that our model is applicable to realistic robotic problems, and to demonstrate that our system is easily applicable to tasks involving reproducing 3D shapes from multiple input types.

4 Experiments

In this section, we verify that our novel architecture performs on par with or even better than the state-of-the-art generative framework. To assess the quality of our proposed neural generative model for 3D shapes, we conduct several extensive experiments. In Sect. 4.2, we investigate our model's ability to generate diverse samples. Following this, in Sect. 4.3, we test our model's ability to reconstruct real-world image, comparing our results to 3D-R2N2 [37] and NRSfM [55]. Finally, we demonstrate the shape completion from the output of a single perspective scan from a depth sensor.

4.1 Datasets

- **ModelNet** There are two variants of the ModelNet dataset, ModelNet10 and ModelNet40, introduced in [2], with 10 and 40 target classes, respectively. ModelNet10 has 3D shapes which are pre-aligned with the same pose across all categories. In contrast, ModelNet40 (which includes the shapes found in ModelNet10) features a variety of poses. In order to assess the ability of our model to handle 3D forms of great variety and complexity, we augment each class of ModelNet10 with a maximum number of 12 rotations while avoiding the risk of overfitting. For the shape completion task, we construct a synthetic dataset based on the ModelNet dataset, taking 15 random perspectives for each object in the ModelNet10 dataset. A test set of entirely unseen objects was held back for evaluation, examples of which, can be observed in the first rows of Figs. 12 and 13.
- **PASCAL 3D** The PASCAL 3D dataset is composed of the image from the PASCAL VOC 2012 dataset [56], augmented with 3D annotations using PASCAL3D+ [57]. PASCAL3D+ images exhibit much more variability com-

Table 1 Details of model architectures used in the experiments. The models were trained using Adam [54] optimizers. BN: Batch normalization, LR: Leaky ReLU, s2: stride 2

Experiments	Dataset	Optimizer	Arch.	
Generation	MODELNET40	Adam	Input	32x32x32x1.
	/CHAIR	1e-3 (gen)	Encoder	Conv with BN and LR:
		1e-3 (dis)		32x4x4x4(s2), 64x4x4x4(s2), 128x4x4x4(s2),
				256x4x4x4(s2), Flatten, FC128.
			Latents	128
			Gen	FC2048, Deconv with BN and ReLU:
				256x4x4x4(s2),128x4x4x4(s2),64x4x4x4(s2),
				32x4x4x4(s2). Tanh.
			Disc	Conv reverse of generator. LR. Sigmoid.
Completion	MODELNET10	Adam	Input	64x64x64x1.
	/BED	2.5e−3 (gen)	Encoder	Conv 32x4x4x4(s2), 64x4x4x4(s2),
		1e-4 (dis)		128x4x4x4(s2),256x4x4x4(s2),512x4x4x4(s2).
				FC4096. LR.
			Latents	40
			Gen	Deconv 512x4x4x4(s2), 256x4x4x4(s2),
				128x4x4x4(s2), 64x4x4x4(s2), 32x4x4x4(s2).
			Disc	Conv reverse of generator. LR. Sigmoid.
Reconstruction	PASCAL3D	Adam	Input	100x100x3.
		1e-3 (gen)	Encoder	Conv 32x4x4(s2), 64x4x4(s2), 128x4x4(s2),
		2e-4 (dis)		256x4x4(s2), FC2048. ReLU.
			Latents	200
			Gen	Deconv 256x4x4(s2), 128x4x4(s2),
				64x4x4(s2), 32x4x4(s2). Tanh.
			Disc	Conv reverse of generator. LR. Sigmoid.

pared to the current 3D datasets, and on average there are over 3,000 object instances per category. We voxelize the 3D CAD models with resolution $32 \times 32 \times 32$ and the same training and testing splits as NRSfM [55], which is also used to conduct real-world image reconstruction. Note that only pre-processing techniques applied were image cropping and padding with 0-intensity pixels to create final samples of resolution 100×100.

- **SYNTHETIC DATASET** A new synthetic dataset of images and 3D models that we created solely serves to train and validate our models from a single RGB image for three-dimensional object reconstruction. The dataset was directly obtained from the online SHAPENET repository [1]. It consisted of six object classes, rendered in front of backgrounds images from the SUN dataset [58] and mantled with random textures from the Describable Textures Dataset [59]. Each RGB image is accompanied with its ground truth 3D model from the SHAPENET repository, with $64 \times 64 \times 64$ voxel resolution. To test our models, we also collected the IKEA dataset from the Google 3D Warehouse which consists of a set of 800 images rendering a large collection of objects, and their corresponding object models. These objects fall into six categories, namely, beds, bookcases, chairs, desks, sofas and tables, and are evaluated in accordance with resolution $64 \times 64 \times 64$. The dataset presents a strong evaluation tool for heavily occluded images in realistic scenes, using only the constraint that the object is centered within the image.

4.2 Evaluating shape generation and learning

To examine our model's ability to generate high-resolution 3D shapes with realistic details, we design a task that involves shape generation and shape interpolation. We add Gaussian noise to the learned latent codes on test data taken from MODELNET and then use our model to generate "unseen" samples that are diverse to the input voxel.

It can be noted that the suggested VAMPPRIOR- 3DGAN demonstrates the ability to transition between two objects smoothly. Our findings on shape generation are illustrated in Fig. 4. We further compare to previous state-of-the-art results in shape generation, which are depicted in Fig. 5). Figure 6

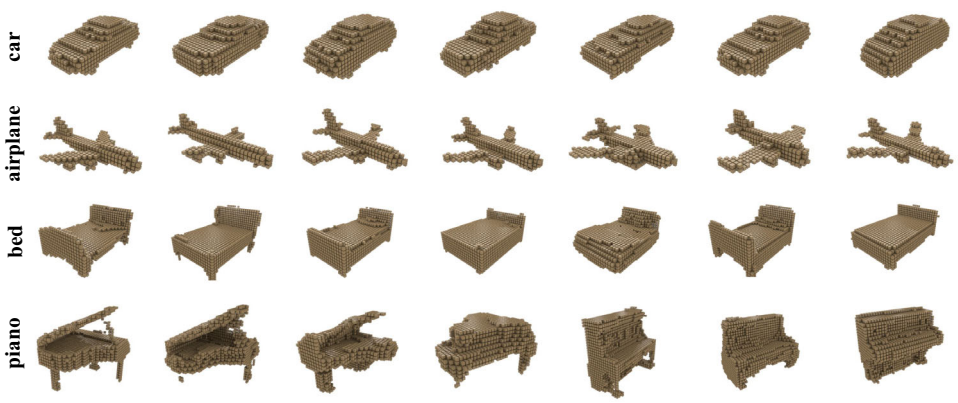

Fig. 4 Shape generation results by our VAMPPRIOR- 3DGAN model on MODELNET40. The picture is best viewed in color on screen

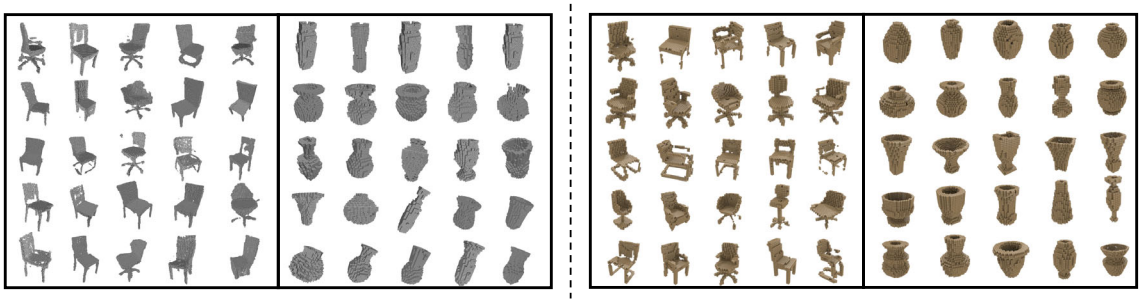

Fig. 5 Visualization comparison of diversity between 3D-GAN (left) and VAMPPRIOR- 3DGAN (right). The picture is best viewed in color

Table 2 Comparing sample diversity by the augmented inception-score values for baseline 3D-GAN and VAMPPRIOR-3DGAN across the 5 categories of MODELNET40 dataset

Method	AIRPLANE	CAR	CHAIR	SOFA	VASE	Mean
3D- GAN	2.72±.20	2.06±.09	2.22±.23	2.12±.55	2.16±.15	2.15±.25
OURS	2.78±.02	2.31±.02	1.27±.01	3.63±.14	2.00±.05	2.28±.62

Larger scores are better. The entries represent score's mean value and standard deviation for the category

shows the results of our shape interpolation experiment, from both within-class and across-class perspectives.

For our system of VAMPPRIOR- 3DGAN framework, the choice of the number of pseudo-inputs (Eq. 2), denoted by $N_{pseudoInput}$, is made empirically—more complicated data distributions require more pseudo-inputs. Larger value of $N_{pseudoInput}$ helps model with relatively increased diversity. Nevertheless, increasing $N_{pseudoInput}$ also increases memory requirements. Our experiments indicate that increasing $N_{pseudoInput}$ beyond a point has little to no effect on the model capacity since the VAMPPRIOR tends to 'crowd' and become redundant. We use a $N_{pseudoInput}$ between 50 and 100 for our experiments.

In order to quantitatively characterize the diversity of generated voxel samples in our experiments, we design an augmented version of the inception score, a measure which has been found to correlate well with human evaluation, for different experiments instead of human annotators. We describe this score next.

Augmented inception score The inception score was considered as a good assessment for sample quality:

$$\exp(\mathbb{E}_x[KL(p(y \mid x) \| p(y))]), \tag{6}$$

where x denotes one sample, $p(y \mid x)$ is the softmax output of a trained classifier of the labels, and $p(y)$ is the overall label distribution of generated samples. The intuition behind this score is that a strong classifier usually has a high confidence for good samples. However, it is desirable to have diversity within voxel samples of a particular category. To characterize this diversity, we use a cross-entropy style score $-p(y \mid x_i) \log(p(y \mid x_j))$ where x_js are samples of the same class as x_i as per the outputs of the trained inception model. We incorporate this cross-entropy style term into the original inception-score formulation and define the augmented inception score as a KL divergence:

$$\exp(\mathbb{E}_{x_i}[\mathbb{E}_{x_j}[KL(p(y \mid x_i) \| p(y \mid x_j))]]). \tag{7}$$

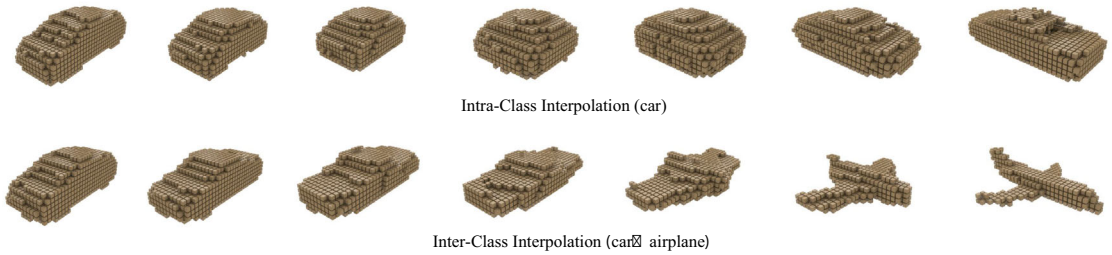

Fig. 6 Continuous morphing of output shapes achieved by linear interpolation of shape vectors. The picture is best viewed in color

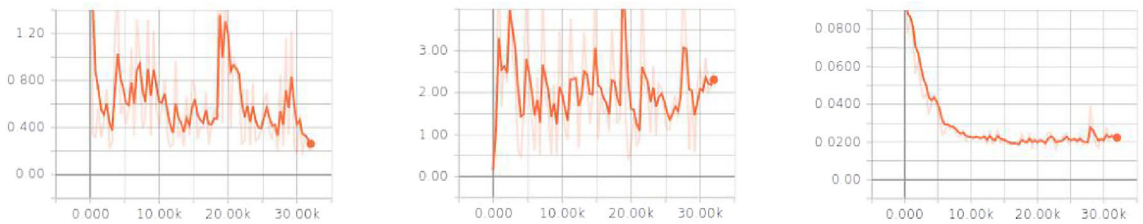

Fig. 7 A graph depicting the discrimination (left), generation (mid) and reconstruction loss (right) at each iteration, while training the VAMPPRIOR-3DGAN system on the MODELNET10 bed dataset

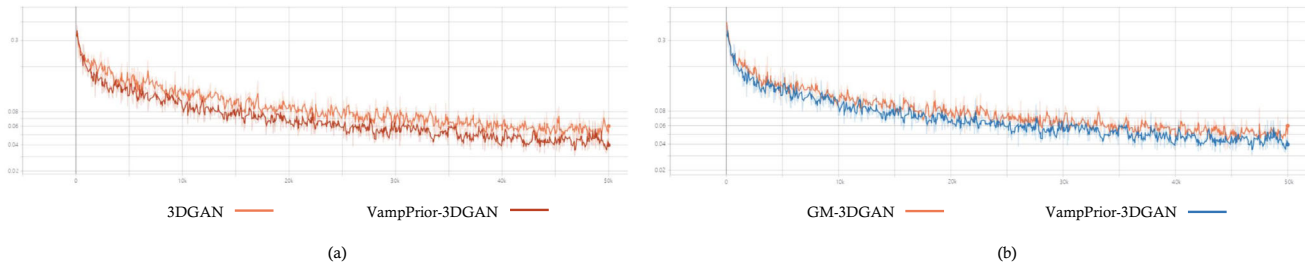

Fig. 8 Comparison of reconstruction loss on MODELNET10 chair dataset with different prior

Essentially, this augmented inception score can be viewed as a proxy for measuring intra-class sample diversity along with the sample quality. In our experiments, we report the augmented inception score on a per-class basis and a combined score averaged over all classes.

For evaluation, we first pretrained a 3D inception network, that is generalized from image inception network, on the training set of MODELNET40. The 3D inception network is a four-layer CNN classifier. After pretraining, the last layer of the inception network is then fine-tuned by transfer learning and then applied to compute the augmented inception scores for the generated samples. We selected five categories—airplane, car, chair, sofa and vase. Note that, during training, we augment the dataset using the rotated version of the voxels. We compare the generated results of 3D-GAN and VAMPPRIOR- 3DGAN. Figure 5 shows the samples generated by 3D-GAN and VAMPPRIOR- 3DGAN, respectively. During this case, our samples are visibly better, and arise from a more stable training procedure. The samples generated by our framework also exhibit larger diversity, visibly and consistent with augmented inception scores additionally (Table 2).

For the MODELNET10 dataset, we will assume that the data generating distribution may be approximated with 10 dominant modes, here we define the term "mode" as a connected component of the data manifold. We first examined whether our system was able to generate objects from a distribution consisting of just one object, but set in 12 different orientations from the MODELNET10 dataset. The value of $N_{pseudoInput}$ is set to 50. This task was clearly successful, since it produced sufficiently varied objects of high quality. This can be observed in Fig. 4. It was possible to track the quality of the objects using the reconstruction and generation loss, and this can be viewed in Fig. 7. Figure 8 showcases a comparison of the reconstruction loss curve with 3D-GAN. One can see in the figure that our method VampPrior-3DGAN converges faster and the reconstruction error is at least 0.04 smaller than that of 3D-GAN. Moreover, in order to see more clearly the effectiveness of the prior we introduced, we further replace vampprior with mixture of Gaussian (dubbed GM-3DGAN), and from the figure one can see the difference

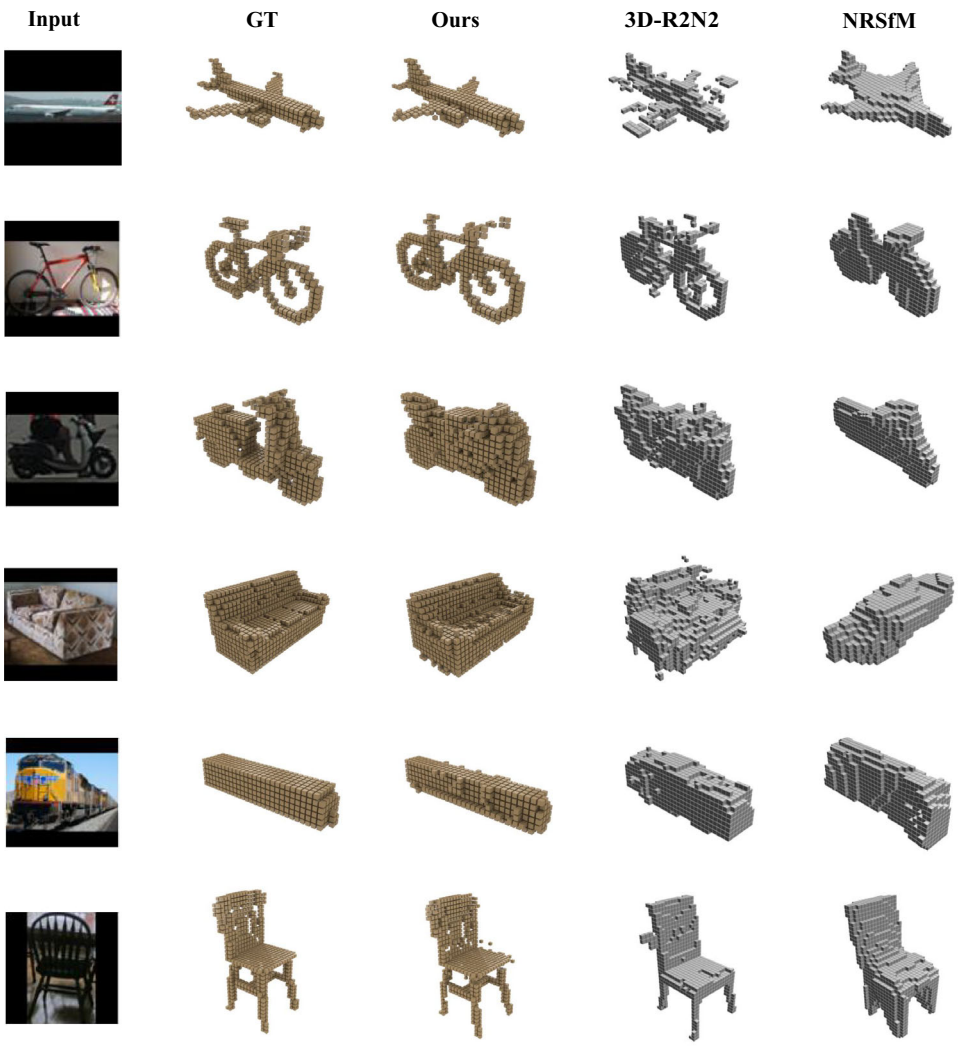

Fig. 9 Reconstruction samples for PASCAL3D from the separately trained VAMPPRIOR- 3DGAN

Table 3 Per-category voxel predictive performance on PASCAL 3D, as measured by IoU

Method	Aero	Bike	Boat	Bus	Car	Chair	Mbike	Sofa	Train	TV	Mean
3D- R2N2 [LSTM- 1]	0.472	0.330	0.466	0.677	0.579	0.203	0.474	0.251	0.518	0.438	0.456
3D- R2N2 [RES3D- GRU- 3]	0.544	0.499	**0.560**	0.816	0.699	0.280	0.649	0.332	0.672	**0.574**	0.571
NRSfM	0.298	0.144	0.188	0.501	0.472	0.234	0.361	0.149	0.249	0.492	0.318
OURS JOINTLY	0.514	0.269	0.327	0.558	0.633	0.199	0.301	0.173	0.402	0.337	0.432
OURS SEPARATELY	**0.645**	**0.671**	0.554	**0.856**	**0.786**	**0.304**	**0.656**	**0.623**	**0.798**	0.454	**0.619**

Bolding indicates that our method achieves best results

Table 4 Average precision scores on the IKEA dataset

Method	Bed	Bookcase	Chair	Desk	Sofa	Table	Mean
ALEXNET- FC8 [60]	29.5	17.3	20.4	19.7	38.8	16.0	23.6
ALEXNET- CONV4 [60]	38.2	26.6	31.4	26.6	69.3	19.1	35.2
T- L NETWORK [60]	56.3	30.2	32.9	25.8	71.7	23.3	40.1
3D- VAEGAN JOINTLY [3]	49.1	31.9	42.6	34.8	79.8	33.1	41.2
3D- VAEGAN SEPARATELY	63.2	46.3	47.2	40.7	78.8	42.3	53.1
OURS SEPARATELY	**78.6**	**52.2**	**57.3**	**51.7**	**83.1**	**52.9**	**62.1**

Bolding indicates that our method achieves best results

Table 5 Comparison of computational efficiency, model size and IoU for single-view 3D reconstruction on the ShapeNet testing set. FIT: Forward inference time

Methods	#Parameters (M)	Memory (MB)	FIT (ms)	Training time (hours)	IoU (%)
3D-R2N2 [37]	35.97	1407	73.1	169	56.1
OGN [38]	12.46	793	37.9	192	59.8
PSG [26]	178.32	956	85.6	78	64.2
Pix2Vox [61]	114.24	2729	9.9	25	66.2
Ours	22.16	117	8.4	21	65.0

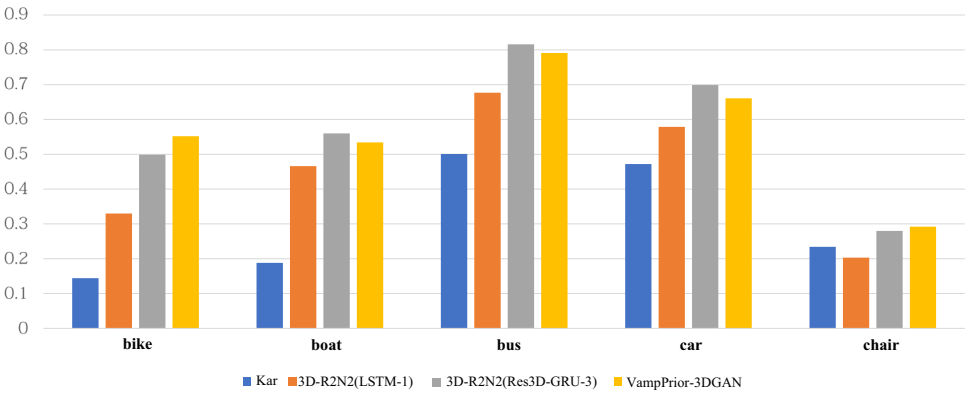

Fig. 10 IoU metric on the SHAPENET subset

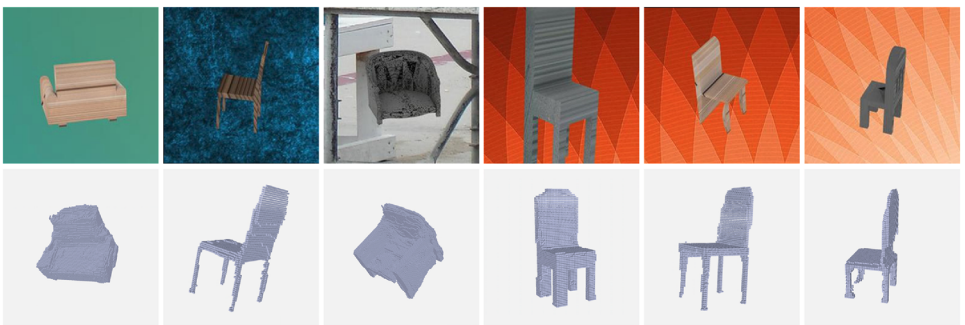

Fig. 11 Sample reconstruction results from single image using the VAMPPRIOR- 3DGAN model, from a distribution consisting of the chair class from the SHAPENET Core dataset. In the 1st row is the RGB input, in the 2nd is the reconstruction of our method. The picture is best viewed in color on screen

in reconstruction quality due to the two priors. Our vampprior is on average 0.02 lower than the GM prior.

4.3 Evaluating shape reconstruction from single image

Another application of the proposed VAMPPRIOR- 3DGAN is single-image shape reconstruction. This is a challenging problem, forcing our model to deal with real-world images under a variety of lighting conditions and resolutions. Furthermore, there are many instances of model occlusion as well as different color gradings.

Metrics We use two metrics to evaluate the performance of 3D reconstruction. The first metric is voxel Intersection-over-Union (IoU) between a predicted voxel grid and its ground truth. It is formally defined as follows:

$$\text{IoU} = \frac{\sum_{ijk}\left[I(y'_{ijk} > p) * I(y_{ijk})\right]}{\sum_{ijk}\left[I\left(I(y'_{ijk} > p) + I(y_{ijk})\right)\right]}, \tag{8}$$

where $I(\cdot)$ is an indicator function, (i, j, k) is the index of voxel in three dimensions, y'_{ijk} is the predicted value at the (i, j, k) voxel, y_{ijk} is the ground truth value at (i, j, k), and p is the threshold for voxelization. The higher the IoU value, the better the reconstruction of a 3D model. We also report the average precision loss as a secondary metric. Higher values indicate higher confidence reconstructions.

To test our model on this application, we use the PASCAL3D dataset and utilize the same exact training and testing splits from [55]. We compare our results with those reported for recent approaches, including the NRSfM [55] and 3D-R2N2 [37] models. Note that these also used the exact same experimental configurations as we did.

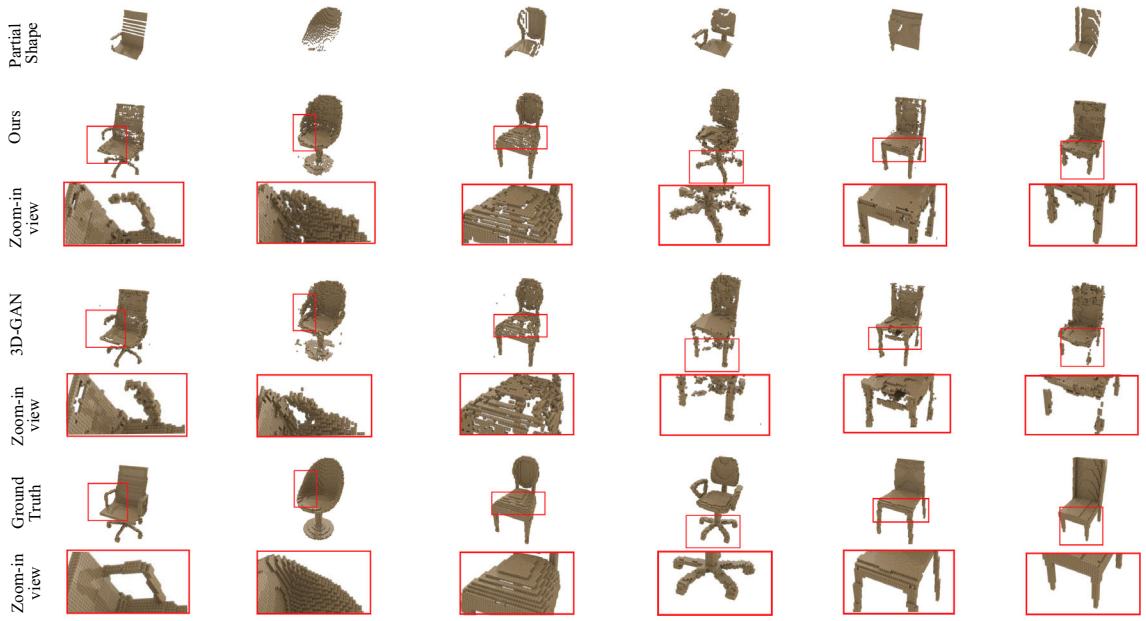

Fig. 12 First row: Sample synthetic single perspective kinetic scans created by the authors, produced from the chair class of the MODELNET10 dataset. Second row: The corresponding shape completion result of our VAMPPRIOR- 3DGAN framework. Third row: The corresponding shape completion result of 3D-GAN. Final row: The corresponding ground truth volumetric grid. The picture is best viewed in color on screen

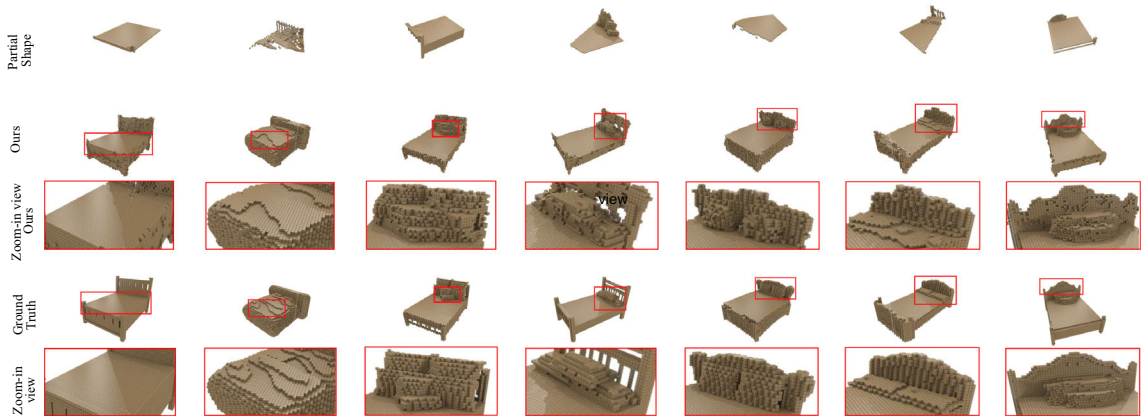

Fig. 13 First row: Sample synthetic single perspective kinetic scans created by the authors, produced from the bed class of the MODELNET10 dataset. Second row: The corresponding shape completion result of our VAMPPRIOR- 3DGAN framework. Third row: The corresponding ground truth volumetric grid. The picture is best viewed in color on screen

For this task, we train our model in two different ways: (1) jointly on all categories, and (2) separately on each category. In Fig. 9, we observe better reconstructions from the separately trained VAMPPRIOR- 3DGAN when compared to previous work. Unlike the NRSfM, our model does not require any segmentation, pose information or keypoints. In addition, our model is trained from scratch while the 3D-R2N2 is pretrained using the ShapeNet dataset. However, the jointly trained VAMPPRIOR- 3DGAN did not outperform the 3D-R2N2, which is also jointly trained. The performance gap is due to the fact that the 3D-R2N2 is specifically designed for image reconstruction and employs a residual network to help the model learn richer semantic features.

Quantitatively, we compare our model to the NRSfM and two versions of 3D-R2N2, one with an LSTM structure and another with a deep residual network. The IoU results are shown in Table 3. Observe that our jointly trained model performs comparably to the 3D-R2N2 LSTM variant while

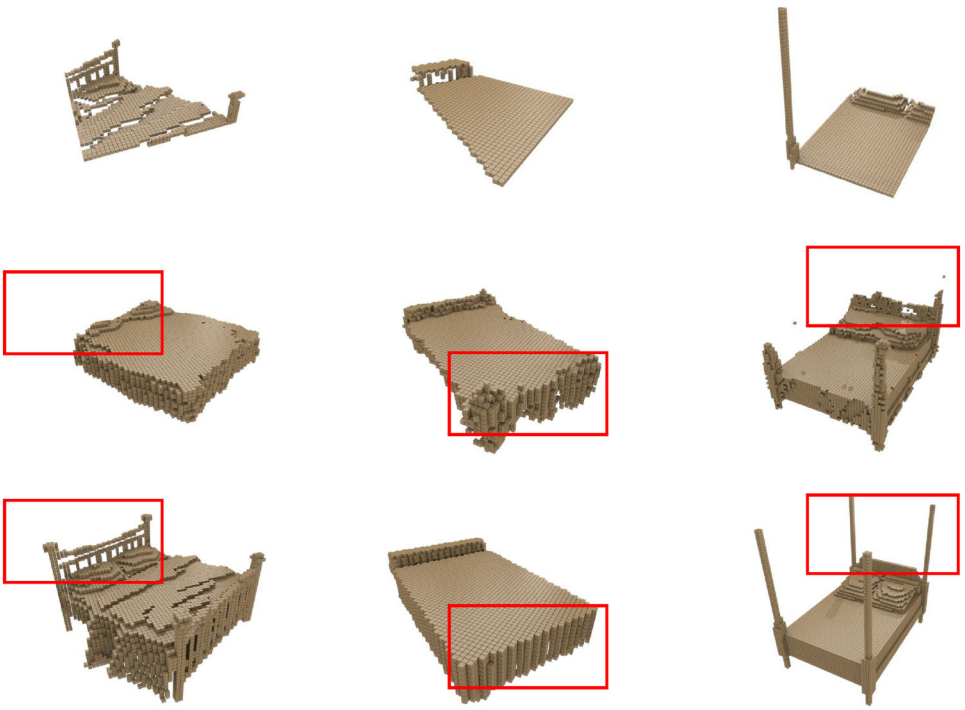

Fig. 14 Failure cases in shape completion on bed category of ModelNet10 dataset. The picture is best viewed in color on screen

the separately trained version surpasses the 3D-R2N2 ResNet structure in 8 out of 10 categories, half of them by a wide margin.

In this experiment, we trained networks with our method on the task of single-image 3D reconstruction. This was a task also performed by the 3D-GAN system [3], and, therefore, provides a fair and quantitative basis for a comparison of the two methods. The result evaluated on the IKEA dataset is illustrated in Table 4. Our system consistently outperforms the original 3D-GAN system and several other previous approaches, with a mean average precision of 62.1% across all classes. Figure 10 showcases the IoU metric results with a comparison of the start-of-the-art methods on the ShapeNet subset. Figure 11 illustrates the example reconstruction results using our method.

Table 5 showcases the numbers of parameters, model size, forward inference time, training time and IoU of different methods. Although our model underperforms Pix2Vox by about 1.2 in the IoU metric, there is an 30% reduction in parameters in our method compared to 3D-R2N2. In order to make a fair comparison, the running times are obtained on the same PC with an NVIDIA GTX 1080 Ti GPU. Our method is about nine times faster in forward inference than 3D-R2N2 in single-view reconstruction.

4.4 Shape completion

The task of recovering 3D shape completion of artifacts from the output of single perspective scan from a depth sensor is added to a voxel-encoded variant of our VampPrior-3DGAN system. Two models were produced: the first trained on chair and bed objects and the second on all the objects in the ModelNet10 dataset. The experiment are clearly quite successful and the examples of recovered objects from the test set can be viewed in the second row of Figs. 12 and 13. In addition, Fig. 14 also indicates several failure cases on bed category of ModelNet10. This is mostly due to the fact that the sample form is far away from the data distribution, and the shape itself is very complex.

5 Conclusion and future work

In this paper, we introduced a novel GAN-based deep generative model, VampPrior- 3DGAN, with a powerful prior as the mixture of variational posterior. Our model is successful in shape generation from complex multimodal distribution involving multiple distinct classes. We demonstrate that the models produced by our system can learn the distributions involving multiple object classes in multiple orientations. In addition, we explain how our method can be smoothly extended to single-image reconstruction, without alternative network architecture. We demonstrate this model's generative power by recovering three-dimensional objects from a single image with different light, background and texture. We achieve state-of-the-art performance on the synthetic dataset. Finally, we again show the system's generative power by successfully applying it to object completion from single perspective scan. While we have focused on dense regular-grid-based shape generation and binary occupancy maps in

this paper, it is straightforward to extend the framework to scene generation and reconstruction from scene image/depth map.

Acknowledgements This work was supported by the Key Project Fund of College Research Program of Anhui Provincial Department of Education (No. 2023AH051129), the Shenzhen Science and Technology Program (No. JCYJ20220818103401003, No. JCYJ20220530172403007) and the Guangdong Basic and Applied Basic Research Foundation (No. 2022A1515010170).

Declarations

Conflict of interest All authors disclosed no relevant potential conflict of interest relationships.

References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
2. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: IEEE/CVF CVPR, pp. 1912–1920 (2015)
3. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: NeurIPS, pp. 82–90 (2016)
4. Brock, A., Lim, T., Ritchie, J.M., Weston, N.J.: Generative and discriminative voxel modeling with convolutional neural networks. In: NeurIPS: 3D Deep Learning, pp. 1–9 (2016)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS, pp. 2672–2680 (2014)
6. Wang, Y., Dai, B., Hua, G., Aston, J., Wipf, D.: Recurrent variational autoencoders for learning nonlinear generative models in the presence of outliers. IEEE J. Sel. Topics Signal Process. **12**, 1615–1627 (2018)
7. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: An overview. IEEE Signal Process. Mag. **35**, 53–65 (2018)
8. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. Stat **1050**, 1–14 (2014)
9. Lim, K., Jiang, X., Yi, C.: Deep clustering with variational autoencoder. IEEE Signal Process. Lett. **27**, 231–235 (2020)
10. Dilokthanakul, N., Mediano, P.A., Garnelo, M., Lee, M.C., Salimbeni, H., Arulkumaran, K., Shanahan, M.: Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648 (2016)
11. Tomczak, J.M., Welling, M.: Vae with a vampprior. AISTATS **2018**, 1214–1223 (2018)
12. Gu, S., Bao, J., Chen, D., Wen, F.: Priorgan: Real data prior for generative adversarial nets. arXiv preprint arXiv:2006.16990 (2020)
13. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned from images. In: NeurIPS, vol. 35, pp. 31841–31854 (2022)
14. Sanghi, A., Fu, R., Liu, V., Willis, K., Shayani, H., Khasahmadi, A.H., Sridhar, S., Ritchie, D.: Textcraft: Zero-shot generation of high-fidelity and diverse shapes from text. arXiv preprint arXiv:2211.01427 (2022)
15. Mittal, P., Cheng, Y.-C., Singh, M., Tulsiani, S.: Autosdf: Shape priors for 3d completion, reconstruction and generation. In: IEEE/CVF CVPR, pp. 306–315 (2022)
16. Liu, Z., Feng, Y., Black, M.J., Nowrouzezahrai, D., Paull, L., Liu, W.: Meshdiffusion: Score-based generative 3D mesh modeling. In: The Eleventh International Conference on Learning Representations, pp. 1–26 (2023)
17. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS, vol. 35, pp. 36479–36494 (2022)
18. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE/CVF CVPR, pp. 10684–10695 (2022)
19. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv preprint arXiv:2212.08751 (2022)
20. Nichol, A.Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: International Conference on Machine Learning, pp. 16784–16804 (2022)
21. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: The Eleventh International Conference on Learning Representations, pp. 1–20 (2022)
22. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European Conference on Computer Vision, pp. 405–421 (2020)
23. Lin, C.-H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.-Y., Lin, T.-Y.: Magic3d: High-resolution text-to-3d content creation. In: IEEE/CVF CVPR, pp. 300–309 (2023)
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE/CVF CVPR, pp. 652–660 (2017)
25. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS, vol. 30, pp. 1–14 (2017)
26. Fan, H., Su, H., Guibas, L.: A point set generation network for 3d object reconstruction from a single image. In: IEEE/CVF CVPR, pp. 2463–2471 (2017)
27. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: 2018 International Conference on 3D Vision (3DV), pp. 728–737 (2018)
28. Tchapmi, L.P., Kosaraju, V., Rezatofighi, H., Reid, I., Savarese, S.: Topnet: Structural point cloud decoder. In: IEEE/CVF CVPR, pp. 383–392 (2019)
29. Pan, L.: Ecg: Edge-aware point cloud completion with graph convolution. In: IEEE Robotics and Automation Letters, vol. 5, pp. 4392–4398 (2020)
30. Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., Sun, W.: Grnet: Gridding residual network for dense point cloud completion. In: European Conference on Computer Vision, pp. 365–381 (2020)
31. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: IEEE/CVF ICCV, pp. 12498–12507 (2021)
32. Pan, L., Chen, X., Cai, Z., Zhang, J., Zhao, H., Yi, S., Liu, Z.: Variational relational point completion network. In: IEEE/CVF CVPR, pp. 8524–8533 (2021)
33. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS **33**, 6840–6851 (2020)
34. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: IEEE/CVF CVPR, pp. 2837–2845 (2021)
35. Lyu, Z., Kong, Z., Xudong, X., Pan, L., Lin, D.: A conditional point diffusion-refinement paradigm for 3d point cloud completion. In: International Conference on Learning Representations, pp. 1–24 (2021)

36. Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: IEEE/CVF ICCV, pp. 5826–5835 (2021)
37. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European Conference on Computer Vision, pp. 628–644 (2016)
38. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: IEEE/CVF ICCV, pp. 2107–2115 (2017)
39. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: NeurIPS, pp. 1–11 (2018)
40. Mandikal, P., Babu, R.V.: Dense 3d point cloud reconstruction using a deep pyramid network. In: IEEE Winter Conference on Applications of Computer Vision, pp. 1–9 (2019)
41. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: IEEE/CVF CVPR, pp. 216–224 (2018)
42. Deprelle, T., Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: Learning elementary structures for 3d shape generation and matching. In: NeurIPS, vol. 32, pp. 7435–7445 (2019)
43. Li, X., Liu, S., Kim, K., De Mello, S., Jampani, V., Yang, M.-H., Kautz, J.: Self-supervised single-view 3d reconstruction via semantic consistency. In: European Conference on Computer Vision, pp. 677–693 (2020)
44. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.P.: Deep level sets: Implicit surface representations for 3d shape inference. arXiv preprint arXiv:1901.06802 (2019)
45. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE/CVF CVPR, pp. 165–174 (2019)
46. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE/CVF CVPR, pp. 4460–4470 (2019)
47. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: IEEE/CVF CVPR, pp. 5939–5948 (2019)
48. Chen, J., Lyu, J., Wang, Y.: Neuraleditor: Editing neural radiance fields via manipulating point clouds. In: IEEE/CVF CVPR, pp. 12439–12448 (2023)
49. Huang, S., Gojcic, Z., Wang, Z., Williams, F., Kasten, Y., Fidler, S., Schindler, K., Litany, O.: Neural lidar fields for novel view synthesis. arXiv preprint arXiv:2305.01643 (2023)
50. Trevithick, A., Chan, M., Stengel, M., Chan, E., Liu, C., Yu, Z., Khamis, S., Chandraker, M., Ramamoorthi, R., Nagano, K.: Real-time radiance fields for single-image portrait view synthesis. ACM Trans. Gr. (TOG) **42**, 1–15 (2023)
51. Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., Chen, D.: Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. arXiv preprint arXiv:2303.14184 (2023)
52. Che, T., Li, Y., Jacob, A., Bengio, Y., Li, W.: Mode regularized generative adversarial networks. In: International Conference on Learning Representations, pp. 1–13 (2016)
53. Ben-Yosef, M., Weinshall, D.: Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images. arXiv preprint arXiv:1808.10356 (2018)
54. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR arxiv:1412.6980 (2015)
55. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: IEEE/CVF CVPR, pp. 1966–1974 (2015)
56. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
57. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: IEEE Winter Conference on Applications of Computer Vision, pp. 75–82 (2014)
58. Xiao, J., Ehinger, K.A., Hays, J., Torralba, A., Oliva, A.: Sun database: Exploring a large collection of scene categories. Int. J. Comput. Vis. **119**, 3–22 (2016)
59. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: IEEE/CVF CVPR, pp. 3606–3613 (2014)
60. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: European Conference on Computer Vision, pp. 484–499 (2016)
61. Xie, H., Yao, H., Sun, X., Zhou, S., Zhang, S.: Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In: IEEE/CVF ICCV, pp. 2690–2698 (2019)

Xianglin Guo is currently an assistant professor at the school of Computer Science and Technology, Anhui University of Technology. He received the Ph.D degree with the College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), China. His current research interests include machine learning and computer vision.

Mingqiang Wei received his Ph.D degree (2014) in Computer Science and Engineering from the Chinese University of Hong Kong (CUHK). He is a professor at the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), China. Before joining NUAA, he served as an assistant professor at Hefei University of Technology, and a postdoctoral fellow at CUHK. He was the receiver of CUHK Young Scholar Thesis Awards 2014. His research interests focus on computer graphics, computer vision and computer-aided art design. He serves IEEE Transactions on Multimedia (IS: Point Cloud Processing and Understanding), and The Visual Computer (IS: Deep Learning for 3D Segmentation) as a guest editor.