Lift3D: Synthesize 3D Training Data
by Lifting 2D GAN to 3D Generative Radiance Field

Leheng Li[1*]   Qing Lian[2]   Luozhou Wang[1]   Ningning Ma[3]   Ying-Cong Chen[1,2†]
[1]HKUST(GZ)   [2]HKUST   [3]NIO Autonomous Driving

Abstract

This work explores the use of 3D generative models to synthesize training data for 3D vision tasks. The key requirements of the generative models are that the generated data should be photorealistic to match the real-world scenarios, and the corresponding 3D attributes should be aligned with given sampling labels. However, we find that the recent NeRF-based 3D GANs hardly meet the above requirements due to their designed generation pipeline and the lack of explicit 3D supervision. In this work, we propose Lift3D, an inverted 2D-to-3D generation framework to achieve the data generation objectives. Lift3D has several merits compared to prior methods: (1) Unlike previous 3D GANs that the output resolution is fixed after training, Lift3D can generalize to any camera intrinsic with higher resolution and photorealistic output. (2) By lifting well-disentangled 2D GAN to 3D object NeRF, Lift3D provides explicit 3D information of generated objects, thus offering accurate 3D annotations for downstream tasks. We evaluate the effectiveness of our framework by augmenting autonomous driving datasets. Experimental results demonstrate that our data generation framework can effectively improve the performance of 3D object detectors. Code: len-li.github.io/lift3d-web
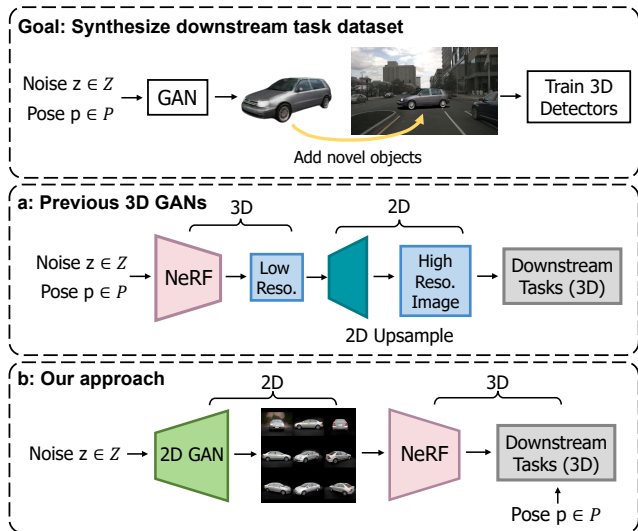

Figure 1. Our goal is to generate novel objects and use them to augment existing datasets. **(a)** Previous 3D GANs (*e.g.*, [28, 45]) rely on a 2D upsampler to ease the training of 3D generative radiance field, while struggle a trade-off between high-resolution synthesis and 3D consistency. **(b)** Our 2D-to-3D lifting process disentangles the 3D generation from generative image synthesis, leading to arbitrary rendering resolution and object pose sampling for downstream tasks.

1. Introduction

It is well known that the training of current deep learning models requires a large amount of labeled data. However, collecting and labeling the training data is often expensive and time-consuming. This problem is especially critical when the data is hard to annotate. For example, it is difficult for humans to annotate 3D bounding boxes using a 2D image due to the inherent ill-posed 3D-2D projection (3D bounding boxes are usually annotated using LiDAR point clouds).

To alleviate this problem, a promising direction is to use synthetic data to train our models. For example, data generation that can be conveniently performed using 3D graphics

engines offers incredible convenience for visual perception tasks. Several such simulated datasets have been created in recent years [3, 14, 30, 31, 42]. These datasets have been used successfully to train networks for perception tasks such as semantic segmentation and object detection. However, these datasets are expensive to generate, requiring specialists to model specific objects and environments in detail. Such datasets also tend to have a large domain gap from real-world ones.

With the development of Generative Adversarial Networks (GAN) [18], researchers have paid increasing attention to utilize GANs to replace graphics engines for synthesizing training data. For example, BigDatasetGAN [23] utilizes condition GAN to generate classification datasets via

---

*Work done during an internship at NIO Autonomous Driving.
†Corresponding author.

conditioning the generation process on the category labels. SSOD [27] designs a GAN-based generator to synthesize images with 2D bounding boxes annotation for the object detection task. In this paper, we explore the use of 3D GANs to synthesize datasets with 3D-related annotations, which is valuable but rarely explored.

Neural radiance field (NeRF) [26] based 3D GANs [6,28], which display photorealistic synthesis and 3D controllable property, is a natural choice to synthesize 3D-related training data. However, our experimental results show that they struggle to keep high-resolution outputs and geometry-consistency results by relying on a 2D upsampler. Furthermore, the generated images are not well aligned with the given 3D pose, due to the lack of explicit 3D consistency regularization. This misalignment would introduce large label noise in the dataset, limiting the performance in downstream tasks. In addition to these findings, the camera parameters are fixed after training, making them challenging to align the output resolution with arbitrary downstream data.

In this paper, we propose Lift3D, a new paradigm for synthesizing 3D training data by lifting pretrained 2D GAN to 3D generative radiance field. Compared with the 3D GANs that rely on a 2D upsampler, we invert the generation pipeline into 2D-to-3D rather than 3D-to-2D to achieve higher-resolution synthesis. As depicted in Fig. 1, we first take advantage of a well-disentangled 2D GAN to generate multi-view images with corresponding pseudo pose annotation. The multi-view images are then lifted to 3D representation with NeRF reconstruction. In particular, by distilling from pretrained 2D GAN, lift3D achieves high-quality synthesis that is comparable to SOTA 2D generative models. By decoupling the 3D generation from generative image synthesis, Lift3D can generate images that are tightly aligned with the sampling label. Finally, getting rid of 2D upsamplers, Lift3D can synthesize images in any resolution by accumulating single-ray evaluation. With these properties, we can leverage the generated objects to augment existing dataset with enhanced quantity and diversity.

To validate the effectiveness of our data generation framework, we conduct experiments on image-based 3D object detection tasks with KITTI [16] and nuScenes [4] datasets. Our framework outperforms the best prior data augmentation method [24] with significantly better 3D detection accuracy. Furthermore, even without any labeled data, it achieves promising results in an unsupervised manner. Our contributions are summarized as follows:

- We provide the first exploration of using 3D GAN to synthesize 3D training data, which opens up a new possibility that adapts NeRF's powerful capabilities of novel view synthesis to benefit downstream tasks in 3D.

- To synthesize datasets with high-resolution images and accurate 3D labels, we propose Lift3D, an inverted 2D-
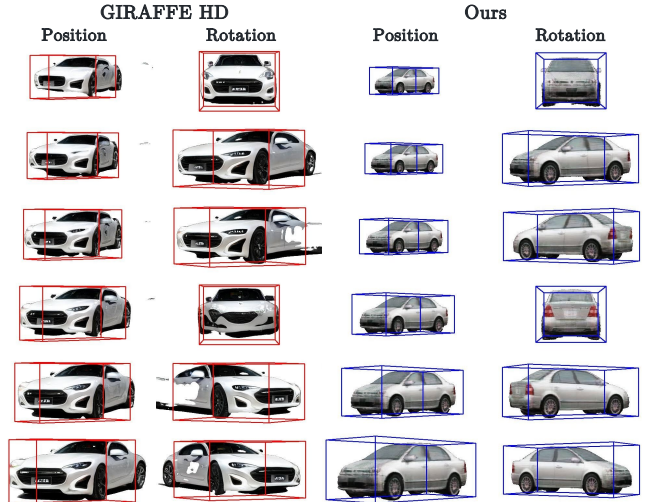

Figure 2. We compare our generation result with GIRAFFE HD [45]. We zoom in or rotate the sampled 3D box to control the generation of models. The rotation of the 3D box introduces artifacts to images generated by GIRAFFE HD. All images are plotted with sampled 3D bounding boxes.

to-3D data generation framework that disentangles 3D generation from generative image synthesis.

- Our experimental results demonstrate that the synthesized training data can improve image-based 3D detectors across different settings and datasets.

2. Related Work

Data Generation for Downstream Tasks   Benefiting from low data acquisition costs, learning from synthesized data is an attractive way to scale up the training data. Several studies like [1, 3, 14, 31] leverage graphic engines to synthesize training data without human annotation. However, they rely on pre-built 3D assets to mimic the world, which is also a non-negligible effort in the whole pipeline.

Without any burden of collecting 3D assets, generative models can also be considered as a neural rendering alternative to graphics engines. For example, BigDatasetGAN [23] generates classification datasets by conditioning the generation process on the class labels. SSOD [27] samples dataset with 2D bounding boxes via generative image synthesis. Our method goes further, utilizing 3D GAN to generate training data with 3D annotation, greatly reducing labeling effort in 3D data.

3D-aware Generative Image Synthesis   Recently, Generative Adversarial Networks (GANs) [18] have made great progress in generating high-resolution photorealistic 2D images. One natural extension of the 2D GANs is to endow their 3D controllable ability as 2D images are projections of the 3D world. To provide the 3D-aware ability, recent

work [6, 28] leverages neural implicit representation (*i.e.*, NeRF [26]), a 3D rendering module, to represent the generator of GANs. In these methods, the training of NeRF module is via the adversarial discriminator between the generated and real images. Due to a lack of 3D supervision, the 3D consistency between the view angle and object position with generated images is not explicitly regularized, making the objects can not align well with provided 3D information as depicted in Fig. 2. Furthermore, the high computation cost of volumetric rendering in the generation process also makes these methods suffer from scaling issues (*e.g.*, resolution). To alleviate this issue, several methods [5, 12, 19, 28, 29, 45] choose to leverage a separate 2D upsampler to produce a high-resolution image from a low-resolution feature grid rendered from NeRF (Fig. 1 (a)). Although this method achieves high-resolution synthesis, the underlying multi-view consistency is non-satisfactory as shown in Fig. 2.

Interpreting the latent representation of GANs [20, 32] has benefited a body of work disentangling various factors of generated objects in a 3D-controllable manner, *e.g.*, viewpoint and shape [25, 33, 50]. This indicates we can tune 2D GANs into 3D-aware generators by interpreting different factors of disentangled latents. Owing to this powerful latent representations of 2D GANs, we can achieve both high-resolution photorealistic image synthesis and rough 3D controllable property. For example, GANverse3D [50] manually annotates the pose labels of StyleGAN2 latents and distills the 3D-aware generation to a mesh-based rendering module. Shi et al. [33] disentangle and distill the 3D information from StyleGAN2 to 3D-aware face generation. Although they achieve 3D controllability of GAN, they haven't tried to convey the 3D-aware ability to benefit downstream tasks.

Data Augmentation in Object Detection Data augmentation is an effective technique for improving the performance of object detection. Although several data augmentation methods have yielded impressive gains for 2D tasks, they are hardly adapted to 3D vision tasks due to the violation of geometric relationship (*i.e.*, when manipulating the object in the image domain, it's non-trivial to obtain the corresponding bounding boxes in the 3D world.) To alleviate this issue, recent work [10, 24, 49] designs geometry-aware copy-paste methods that fix the relationship among the viewing angle, 2D and 3D position to obtain the corresponding 3D ground truth. However, these methods are limited to generating objects with fixed viewing angles and positions. They are also restricted to use pre-collected asset banks.

In this work, we design a generative-based data generation method for 3D object detection, in which an unlimited amount of objects with arbitrary position and rotation can be generated for training. Based on our method, the diversity of the training data can be effectively enhanced, leading to better coverage of long-tail scenarios in the safety-critical driving domain.

3. Method

3.1. Overview

We visualize the generation and augmentation pipelines in Fig. 3. Our method aims to generate novel objects with corresponding 3D bounding boxes annotation and compose them into existing scenes for training. Without loss of generality, we adopt monocular 3D object detection as the downstream task to evaluate the effectiveness of our method.

3.2. 3D GANs

Recently, the community has witnessed an explosion in the 3D generative radiance field [5,12,19,28,29,44,45,51,52]. However, as revealed in [19, 36], none of them can preserve strict multi-view consistency, partially on account of the usage of a 2D upsampler and lack of explicit 3D supervision. Fig. 2 shows the generation result of GIRAFFE HD [45], the SOTA 3D generative network on Car. When we render objects in fixed rotation and increasing depth, the objects gradually lose details during zoom-out. In the meantime, during sampling in fixed depth and varying rotation, the images break multi-view consistency as the objects only fit the 3D box in a single view. These unsatisfactory results prevent their application in downstream tasks.

To benefit downstream tasks with 3D generative synthesis, we design a method that **(i)** is able to generate high-resolution photorealistic images, **(ii)** its generated images are consistent with given sampling label, **(iii)** capable of generating images in any wanted distribution (not restrict to object-centric views).

The general idea of our method is to switch the order of generation process to 2D-3D rather than 3D-2D. We first introduce our efficient interpretation of StyleGAN2 that serves for dense multi-view synthesis in Sec. 3.3. Then, we detail the proposed lifting process that provides high-resolution synthesis and accurate 3D annotation in Sec. 3.4.

3.3. StyleGAN as a Proxy 3D Generator

To achieve **(i)**, we propose to incorporate powerful Style-GAN2 [21] to serve as a proxy 3D generator. Our goal is to leverage pretrained StyleGAN2 to densely sample multi-view images with pose labels. Note that other GANs may also work, but we choose [21] for its most photorealistic synthesis.

We draw aspiration from GANSpace [20] and SeFa [32] to disentangle StyleGAN2. Specifically, StyleGAN2 is a synthesis network that maps a latent code $\mathbf{z} \in \mathbb{R}^{512}$ sampled from a Gaussian distribution $\mathbf{Z} \in \mathcal{N}(0, \mathbf{I})$ into a real image. The latent $\mathbf{z}$ is first transformed to $\mathbf{w}^* = (w_1^*, w_2^*, ..., w_{16}^*) \in \mathbb{R}^{16 \times 512}$ through 16 affine transformations. Then $\mathbf{w}^*$ is serve as conditional information to modulate different layers of the synthesis network. GANSpace [20] found that different layers in $\mathbf{w}^*$ control different image attributes. For example,
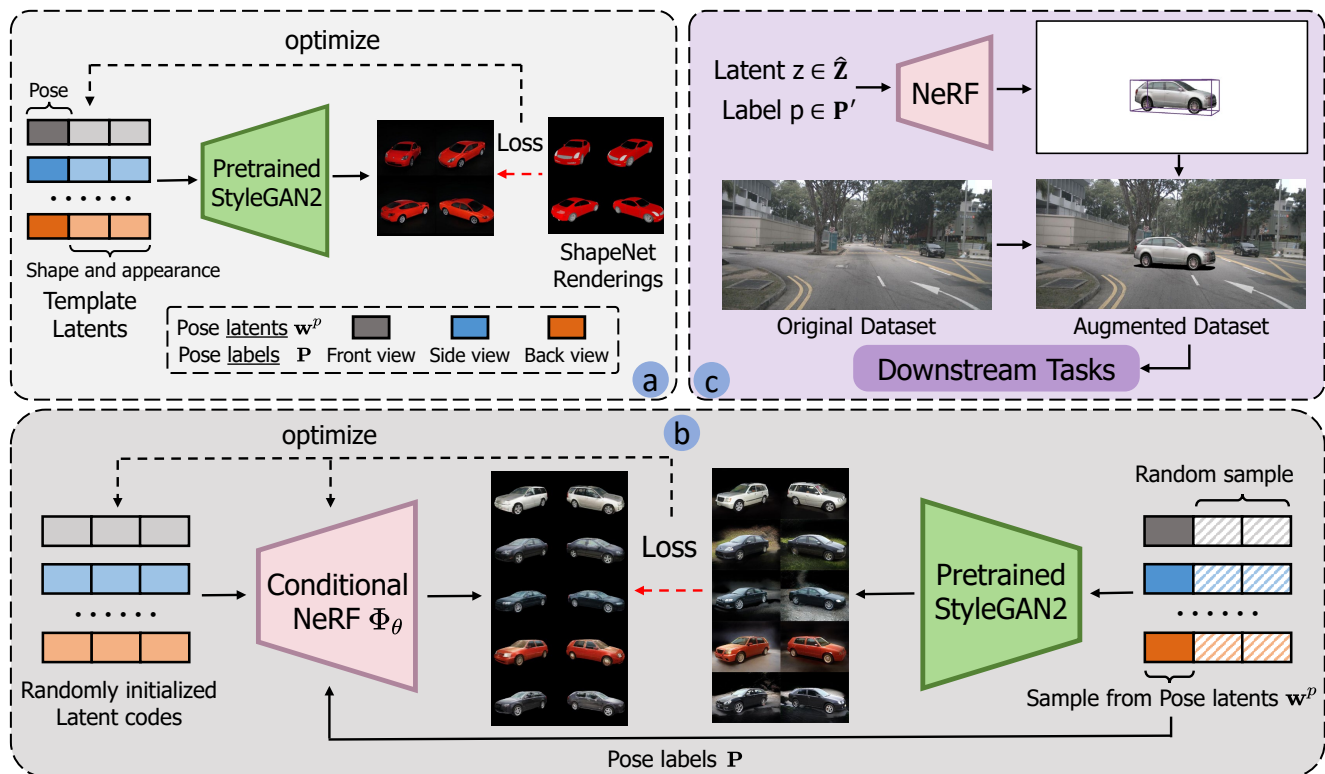
Figure 3. **Lift3D** is a 3D generation framework that enables training data generation by providing photorealistic synthesis and precise 3D annotation. The framework comprises three key modules: **(a)** Interpretation of StyleGAN2: We obtain latent-label pairs with the guidance of the ShapeNet renderings. **(b)** Lifting process: We optimize a shared conditional NeRF and randomly initialized latent codes to lift pretrained StyleGAN2 to 3D object NeRF. **(c)** Training data generation: By sampling latents $\mathbf{z}$, we can augment the existing dataset by adding novel objects with label p.

styles in the early layers adjust the camera viewpoint, while styles in the intermediate and higher layers influence shape, texture, and background. That's to say, if we sample a new latent but keep the early layers unchanged, we generate images of an object with a different shape and appearance but depicted in the same viewpoint. This motivates us to transfer high-quality StyleGAN2 to 3D-aware generators without retraining 3D GANs.

Following GANSpace [20], we empirically set the first 8 layers of latents as *pose* latents (denoted as $\mathbf{w}^p$), the remaining 8 layers of latents as *shape and appearance* latents. Fig. 3 (a) shows how we obtain latent-label pairs $\mathbf{w}^p$ and $\mathbf{P}$ for the lifting process. Since graphics-based rendered images naturally provide pose labels during rendering, we randomly select a 3D car model from ShapeNet [7], and render the model under 200 different viewpoints $\mathbf{P}$, then use optimization-based GAN inversion method [43] to find the corresponding template latents. Thus, the first 8 layers of template latents are associated with meaningful pose information $\mathbf{P}$ (*e.g.*, the pose latent $\mathbf{w}^p$ in darker gray represents a rendering from the front viewpoint). We next illustrate the usage of these 200 latent-label pairs $\mathbf{w}^p$ and $\mathbf{P}$ in Sec. 3.4.

3.4. Lifting 2D Images to 3D Radiance Field

Directly applying a 3D GAN in downstream tasks is nontrivial as discussed in Sec. 3.2. In this work, we propose a method that distills 3D knowledge from pretrained Style-GAN2: lifting generated images to 3D representation. As depicted in Fig. 3 (b), we jointly optimize a shared conditional NeRF and latent codes to lift StyleGAN2-generated multi-view images to 3D object NeRF. Detailed network structures are described below.

Generating 2D Data $(\mathbf{I}, \mathbf{P})$ As shown in the right side of Fig. 3 (b), we leverage the above disentangled StyleGAN2 to sample images $\mathbf{I}$ with pose label $\mathbf{P}$ for the lifting process. During sampling, the first 8 layers of latents are sampled from the pose latents $\mathbf{w}^p$. The remaining 8 layers of latents are randomly sampled from Gaussian distribution $\mathbf{Z}$ for diverse synthesis.

Incorporating Priors via Global Optimization $\hat{\mathbf{Z}}, \Phi_\theta$ A naive method of lifting multi-view images to 3D assets is to reconstruct each object with an isolated NeRF. However, due to the randomness of GAN sampling and imperfect GAN interpretation, the generated images and pose labels are noisy,

or even sometimes corrupted, hindering the reconstruction of a single object. This method also neglects the inductive bias in a certain category. To address the above issues, we model the lifting process as Generative Latent Optimization (GLO) [2], in which each object is assigned a corresponding latent vector. We optimize a shared NeRF $\Phi_\theta(z, P)$ with parameter θ and randomly initialized latent codes $\mathbf{z} \in \mathbf{Z}$ for all objects at the same time, thereby granting NeRF generator to learn the shared shape and appearance prior within the same category. The formula can be written as follows:

$$\hat{\mathbf{Z}}, \hat{\theta} = \arg\min_{\mathbf{z}, \theta} \mathcal{L}(\mathbf{I}, \Phi_\theta(\mathbf{z}, \mathbf{P})), \tag{1}$$

where $\hat{\mathbf{Z}}$ represents the concatenation of all latent codes, $\mathbf{P}$ is the pose label obtained from GAN interpretation, $\mathcal{L}$ is our loss function that will be introduced in Sec. 3.6.

Efficient 3D Representation Φ_θ We represent our NeRF network Φ_θ with a tri-plane [5] representation for efficient feature encoding. Specifically, we follow [5,21] and use a 2D convolution neural network to map constant input to three axis-aligned orthogonal feature planes with the condition of latent code $\mathbf{z}$. For any sampled 3D point $x \in \mathbb{R}^3$ of NeRF, we query its feature vector by projecting it onto each of the three feature planes, retrieving the corresponding three feature vectors via bilinear interpolation, then summing the three as the final feature vector. To further incorporate global information, we then feed the final feature vector to a single layer SIREN-based [35] MLP that conditions on $\mathbf{z}$ to output the density and RGB value.

Volumetric Rendering The expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with nearest and furthest sampling bounds t_n and t_f is:

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \tag{2}$$

$$\text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right). \tag{3}$$

The function $T(t)$ denotes the accumulated transmittance along the ray from t_n to t.

We also predict the binary mask of objects $\hat{\mathbf{M}}$ by the accumulated transmittance of the furthest sampling point to extract the foreground pixels.

$$\hat{\mathbf{M}} = \begin{cases} 1 & T(t_f) \geq 0.5 \\ 0 & T(t_f) < 0.5. \end{cases} \tag{4}$$

3.5. Sample and Composition

During sampling, a significant advantage of our method is that: we can synthesize any-resolution images by accumulating single-ray evaluation without relying on upsamplers. Fig. 3 (c) illustrates our sampling procedure.

Figure 4. Disentanglement of our NeRF generation. By distilling knowledge from pretrained StyleGAN2, our 3D generation framework directly inherits the well-disentangled property to generate the shape-appearance disentangled images.

Sample Position Our goal is to augment the existing dataset by populating novel objects with any wanted pose $\mathbf{P}'$. To place objects, a simple way is to uniformly sample positions on road planes, as camera height information can be extracted from calibration. However, this method does not consider relations between objects and environment layout *e.g.*, vehicles don't appear inside a building. To mitigate this problem, we use a pretrained semantic segmentation model HRNet [37] to extract drivable areas in the 2D image plane. Then we convert the perspective view into the bird's-eye-view (BEV) through Inverse Perspective Mapping (IPM). During sampling, objects that are outside the drivable area will be filtered out. Detailed parametrization of $\mathbf{P}'$ can be found in Supp. Material.

Shadow Generation As a non-negligible consequence of lighting interactions, shadows play a vital role in both visual quality [10] and downstream tasks [13, 41]. Similar to GeoSim [10], we cast a pre-computed shadow map at the bottom of each object.

Composition After generating the foreground images, alpha blending is utilized to maintain the visual quality of the object border when composing the foreground and background images. Specifically, we compute the output image as $I_1 \times M + I_2 \times (1 - M)$ where I_1 and I_2 are the background and foreground images, respectively. We apply a Gaussian filter to foreground mask M to smooth out the edges of the pasted objects.

3.6. Model Learning

In this section, we detail the learning objective of the lifting process. Given N pairs of multi-view images with

pose annotations $\{I_i, P_i\}_{i=1}^N$, we jointly optimize latents and NeRF generator parameters to minimize the following loss:

$$\mathcal{L} = \sum_{i=1}^{N} (\mathcal{L}_{RGB} + \lambda_{IoU} \mathcal{L}_{IoU} + \lambda_{perc} \mathcal{L}_{perc}), \quad (5)$$

where $\mathcal{L}_{RGB}$ is photometric loss, $\mathcal{L}_{IoU}$ is IoU loss, $\mathcal{L}_{perc}$ is perceptual loss, and λ_* are loss weights.

Photometric Loss   In order to optimize both latents and NeRF generator to faithfully reconstruct input objects, we encourage the pixels of the output image $\Phi_\theta(\mathbf{z_i}, P_i)$ to exactly match the pixels of the input image I_i. If both have shape $C \times H \times W$, then the pixel loss is defined as

$$\mathcal{L}_{RGB} = \frac{1}{CHW} |I_i - \Phi_\theta(\mathbf{z_i}, P_i)|, \quad (6)$$

where $|\cdot|$ denotes the L_1 loss. $\mathbf{z_i}$ and P_i are object latents and pose labels, respectively.

Perceptual Loss   We use perceptual loss to preserve high-level feature reconstruction. The loss network *VGG* is the 16-layer VGG network [34] pretrained on ImageNet [11].

$$\mathcal{L}_{perc} = \frac{1}{CHW} |VGG(I_i) - VGG(\hat{I}_i)|. \quad (7)$$

IoU Loss   In addition, we use IoU loss to enforce silhouette consistency.

$$\mathcal{L}_{IoU} = 1 - \frac{\hat{M} \cap M^*}{\hat{M} \cup M^*}, \quad (8)$$

where $\hat{M}$ is the binary mask predicted by volumetric rendering, M^* is the ground truth binary mask predicted by pretrained segmentation model PointRend [22].

4. Experiment

In this section, we evaluate the benefits of our 3D generative synthesis as data augmentation for the downstream monocular 3D object detection task on the KITTI [17] and nuScenes datasets [4].

4.1. Experimental Setup

KITTI [17]   KITTI dataset is one of the most famous autonomous driving datasets, which contains 7,481 training and test frames with 80,256 annotated 3D bounding boxes. Following [8, 9], we split the training data into the training and validation subsets and evaluate the effectiveness of our proposed method on the validation set.

nuScenes [4]   nuScenes is a large-scale autonomous driving dataset. This dataset is collected using 6 surrounded-view cameras that cover the full 360° field of view around the ego-vehicle. Compared with the KITTI dataset, nuScenes dataset contains 7x as many 3D bounding boxes in the urban scenes.

Sample Depth →

(a) GIRAFFE

(b) GIRAFFE HD

(c) Ours

Figure 5. Qualitative image comparisons between GIRAFFE [28], GIRAFFE HD [45] and Ours. We render objects with randomly sampled rotation while increasing the depth magnitude.

Evaluation Metrics   In the KITTI dataset, we follow the official protocol [17] and adopt the $AP|_{40}$ evaluation metrics on 3D bounding box estimation tasks. The evaluation is conducted separately based on the difficulty levels (Easy, Moderate, and Hard). In the nuScenes dataset, we adopt the provided [4] evaluation metrics to evaluate mAP of all classes. We follow a similar setting in [41]. We use a 10% subset of real data from the nuScenes training set. The subset is sampled scene-wise to mimic label-efficient scenarios.

Asset Bank Creation   For StyleGAN2 [21], we adopt the official model[1] trained on the LSUN Car dataset [47] to synthesize the multi-view images in an offline manner. We synthesize images with pose label $\mathbf{P}$, which is sampled from the surface of a unit sphere that looks at the origin, with $0 - 360°$ in azimuth and $0 - 20°$ in elevation. We use segmentation network PointRend [22] to extract binary mask of objects. During inference, images with a confidence lower than 0.5 will be regarded as noisy data to filter out. We created a large object bank of 1000 different vehicles, and each vehicle is assigned 200 multi-view images (note that nothing prevents us from generating a larger amount of data). Then, we conduct the lifting process in the whole asset bank.
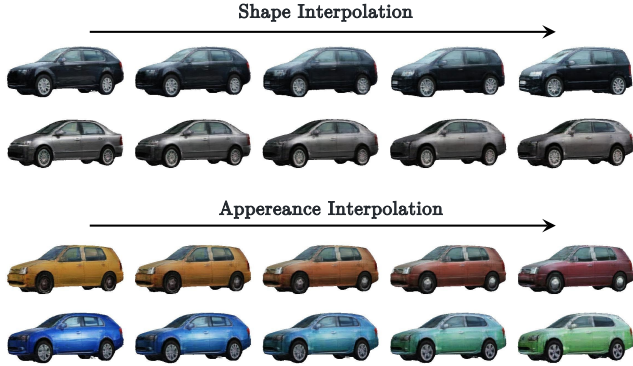
---

[1] https://github.com/NVlabs/stylegan2

Figure 6. Visualization of interpolation. We linearly interpolate the latent codes of NeRF from left to right. Our method faithfully generates a smooth and meaningful transition from one to another, in both shape and appearance.

4.2. Comparisons

Qualitative Results   We compare the visual quality with the existing methods in Fig. 5. We utilize the foreground mask in the generative model to filter out the background pixels. Without any deformation of objects or broken parts, our method is significantly more realistic than other methods. Fig. 8 further shows the composition result of generated objects and the original datasets, where our method achieves harmonious visual quality without any post-processing.

Quantitative Results   In Tab. 1, we provide the experimental results of leveraging the augmented datasets to train 3D object detectors. Following existing settings [24, 41], we use CenterNet [53] (KITTI) and FCOS3D [40] (nuScenes) trained on original data as baseline detectors. We compare our method with several methods: **ShapeNet model** denotes that we randomly select 20 CAD models from ShapeNet car, rendering the models under different positions and rotations to generate an augmented dataset. **GIRAFFE** and **GIRAFFE HD**: Since they don't contain real-world scale and can not generalize to new camera parameters, we only paste their generated objects in the center of images and leverage object height to recover the ground truth depth. We lighten their color for such issues. **SimpleCopyPaste** is

Method	Easy	Mod	Hard
Original data	21.57	15.51	13.58
ShapeNet model	19.64	15.75	13.80
GIRAFFE [28]	20.79	15.10	12.50
GIRAFFE HD [45]	19.30	14.44	12.39
SimpleCopyPaste [24]	22.23	15.47	13.24
Ours	**24.07**	**18.09**	**15.06**

Table 1. Quantitative Evaluation on the KITTI validation dataset. $AP|_{40}$ of 3D bounding box on the Car category are reported.
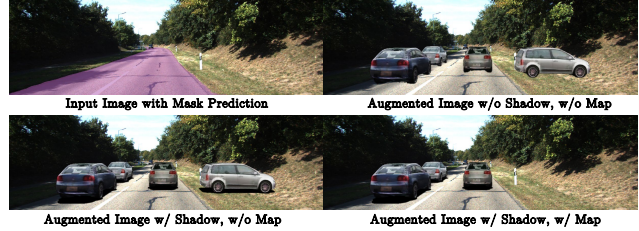

Figure 7. Visualization of different composition strategies. "Map" denotes only sample objects in drivable areas. "Shadow" denotes whether to add a shadow at the bottom of objects.

SOTA copy-paste augmentation method in monocular 3D detection. As discussed in [24], it is restricted to using existing objects with fixed viewpoints to do augmentation, while our method can generate novel objects in arbitrary positions and rotations. Consequently, our method can effectively enhance the diversity of training data and achieve better results as shown in Tab. 1.

Except for the KITTI dataset, we also validate the effectiveness of our method on the nuScenes dataset. We augment the full-surround view images of dataset with our method. We use the same training strategy and model hyperparameters as [40] but do not supervise attributes and velocity as they are not included in the augmented data. It is worth mentioning that, by only augmenting the car objects, we can achieve consistent gain among the majority of classes even though we do not directly augment these classes.

Method	mAP	Car	Bus	Pedestrian	Motorcycle
Original data	0.136	0.231	0.072	0.277	0.142
Ours	**0.146**	**0.254**	**0.086**	**0.297**	**0.154**

Table 2. Experimental results of FCOS3D [40] on the nuScenes validation set. mAP represents the mean average precision of 10 object categories. We additionally present the AP of other classes that are not directly augmented (*e.g.*, bus, pedestrian, and motorcycle).

Unsupervised 3D Object Detection   Besides augmenting existing datasets, our framework can create datasets from scratch, which means unsupervisedly train a 3D detector. Specifically, we select the images with more than 90% of background pixels as empty images by using a pretrained segmentation network [37]. Then we populate novel objects into these nearly-empty background images. During the whole

Method	Easy	Mod	Hard
SDF-Label [48]	1.23	0.54	N/A
Our unsup. data	**5.15**	**3.80**	3.48

Table 3. Comparison with unsupervised 3D detection methods. Note that SDF-Label [48] additionally uses LiDAR point clouds for accurate depth retrieval.

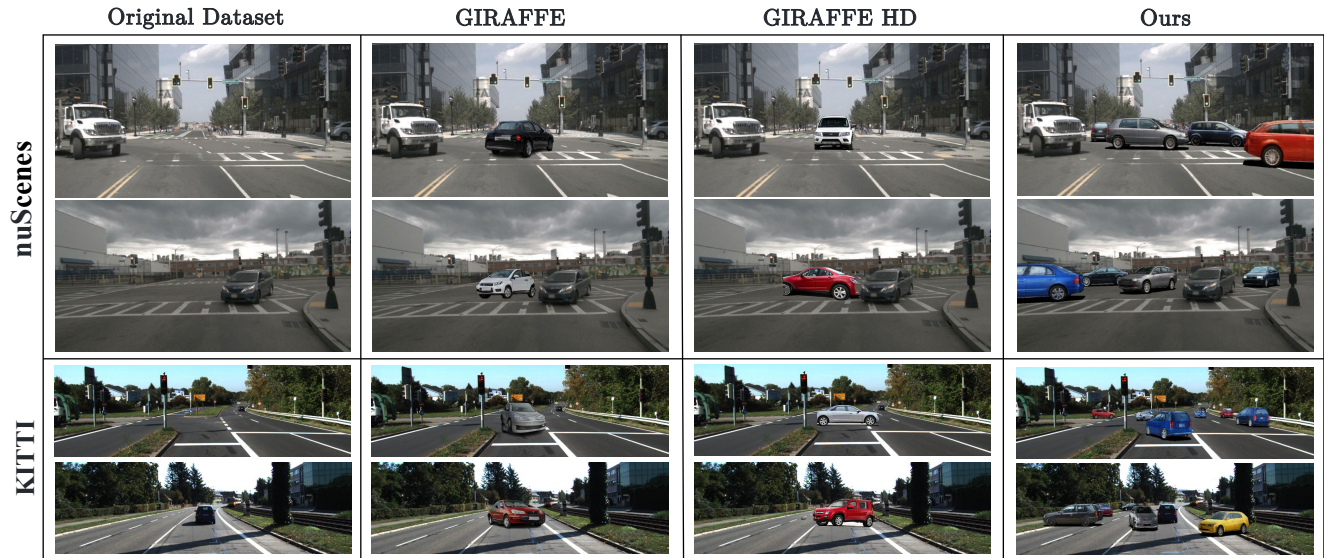| | Original Dataset | GIRAFFE | GIRAFFE HD | Ours |

Figure 8. Qualitative comparison of 3D generative synthesis serves as data augmentation on nuScenes and KITTI datasets. Our method is able to generate more realistic shapes and textures, and the resulting objects blend more naturally with the background image.

process, the construction of the training data does not require any human annotation for 3D bounding boxes. As shown in Tab. 3, our trained model outperforms existing unsupervised training solution by a large margin.

4.3. Ablation Study

We ablate different composition strategies that affect performance in the downstream task in Tab. 4. We fix the sampling pose of all synthesized objects and train the same model [53] with different composition strategies. As shown in Fig. 7 and Tab. 4, the improvement of downstream task performance is consistent with the enhanced visual quality. These indicate the importance of realistic simulation in downstream tasks.

Map	Shadow	Easy	Mod	Hard
		22.35	17.36	14.53
✓		22.85	16.49	14.08
	✓	23.08	17.71	15.02
✓	✓	**24.07**	**18.09**	**15.06**

Table 4. Ablation of different composition strategies on the KITTI validation dataset. "Map" denotes road layout aware sampling. "Shadow" denotes whether to cast a shadow at the bottom of objects.

5. Limitation and Future Work

Relation Reasoning　The proposed method augments the existing dataset by sampling novel objects uniformly on the ground plane. If objects are not properly sampled in reasonable positions, they can cause unharmonious in the augmented images. For example, a vehicle may not follow the lane lines or even rush to a building. Also, relations between traffic participants are not modeled realistically. In the future, these issues could be addressed by traffic flow simulation [38, 39], or geometry-aware composition [10].

Photorealistic Appearance　Since our generated objects have a different data source from the original datasets. There exist an illumination gap between generated object and the existing environment. For example, the rendered object in driving scene should be darker in the afternoon, compared with midnoon. Lighting estimation [41, 46] can help with this problem. In the meantime, high-quality material properties should be estimated in detail. Techniques proposed in GET3D [15] could potentially be used to generate objects with high-fidelity textures.

6. Conclusion

We propose Lift3D, a 3D generation framework that provides high-resolution synthesis and tight 3D annotation, with the goal of training data generation. Compared with other 3D generative models relying on a 2D upsampler, our lifting process gives explicit 3D information of generated objects and hence the objects align well with given 3D labels. Also, our approach provides more realistic appearance generation by distilling the knowledge from pretrained StyleGAN2. Based on the proposed framework, we can enhance original datasets both in quantity and diversity. Experimental results show significant improvements over baseline in monocular 3D object detection tasks. We hope this work can take us a step further in 3D asset generation for driving scene simulation.

References

[1] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *IJCV*, 2018. 2

[2] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017. 5

[3] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 1, 2

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 2, 6

[5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 3, 5

[6] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *CVPR*, 2021. 2, 3

[7] 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. 4

[8] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 6

[9] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In *NeurIPS*, 2015. 6

[10] Yun Chen, Frieda Rong, Shivam Duggal, Shenlong Wang, Xinchen Yan, Sivabalan Manivasagam, Shangjie Xue, Ersin Yumer, and Raquel Urtasun. Geosim: Realistic video simulation via geometry-aware composition for self-driving. In *CVPR*, 2021. 3, 5, 8

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[12] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *CVPR*, 2022. 3

[13] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *ICCV*, 2019. 5

[14] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 1, 2

[15] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *NeurIPS*, 2022. 8

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 2

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 6

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 2

[19] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *ICLR*, 2022. 3

[20] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *NeurIPS*, 2020. 3, 4

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 3, 5, 6

[22] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 6

[23] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *CVPR*, 2022. 1, 2

[24] Qing Lian, Botao Ye, Ruijia Xu, Weilong Yao, and Tong Zhang. Exploring geometric consistency for monocular 3d object detection. In *CVPR*, 2022. 2, 3, 7

[25] Feng Liu and Xiaoming Liu. 2d gans meet unsupervised single-view 3d reconstruction. In *ECCV*, 2022. 3

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3

[27] Siva Karthik Mustikovela, Shalini De Mello, Aayush Prakash, Umar Iqbal, Sifei Liu, Thu Nguyen-Phuoc, Carsten Rother, and Jan Kautz. Self-supervised object detection via generative image synthesis. In *ICCV*, 2021. 2

[28] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 1, 2, 3, 6, 7

[29] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-Resolution 3D-Consistent Image and Geometry Generation. In *CVPR*, 2022. 3

[30] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *ICCV*, 2017. 1

[31] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 1, 2

[32] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *CVPR*, 2021. 3

[33] Yichun Shi, Divyansh Aggarwal, and Anil K Jain. Lifting 2D stylegan for 3D-aware face generation. In *CVPR*, 2021. 3

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6

[35] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 5

[36] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *arXiv preprint arXiv:2206.10535*, 2022. 3

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 5, 7

[38] Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *CVPR*, 2021. 8

[39] Shuhan Tan, Kelvin Wong, Shenlong Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Scenegen: Learning to generate realistic traffic scenes. In *CVPR*, 2021. 8

[40] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: Fully convolutional one-stage monocular 3d object detection. In *ICCVW*, 2021. 7

[41] Zian Wang, Wenzheng Chen, David Acuna, Jan Kautz, and Sanja Fidler. Neural light field estimation for street scenes with differentiable virtual object insertion. In *ECCV*, 2022. 5, 6, 7, 8

[42] Xinshuo Weng, Yunze Man, Jinhyung Park, Ye Yuan, Dazhi Cheng, Matthew O'Toole, and Kris Kitani. All-In-One Drive: A Large-Scale Comprehensive Perception Dataset with High-Density Long-Range Point Clouds. *arXiv*, 2021. 1

[43] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *TPAMI*, 2022. 4

[44] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, 2022. 3

[45] Yang Xue, Yuheng Li, Krishna Kumar Singh, and Yong Jae Lee. Giraffe hd: A high-resolution 3d-aware generative model. In *CVPR*, 2022. 1, 2, 3, 6, 7

[46] Yao Yao, Jingyang Zhang, Jingbo Liu, Yihang Qu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf: Neural incident light field for physically-based material estimation. *arXiv preprint arXiv:2203.07182*, 2022. 8

[47] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[48] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3d objects with differentiable rendering of sdf shape priors. In *CVPR*, 2020. 7

[49] Wenwei Zhang, Zhe Wang, and Chen Change Loy. Multi-modality cut and paste for 3d object detection. *arXiv preprint arXiv:2012.12741*, 2020. 3

[50] Yuxuan Zhang, Wenzheng Chen, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. Image gans meet differentiable rendering for inverse graphics and interpretable 3d neural rendering. In *ICLR*, 2021. 3

[51] Xiaoming Zhao, Fangchang Ma, David Güera, Zhile Ren, Alexander G. Schwing, and Alex Colburn. Generative multiplane images: Making a 2d gan 3d-aware. In *Proc. ECCV*, 2022. 3

[52] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. *arXiv preprint arXiv:2110.09788*, 2021. 3

[53] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 7, 8