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

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

[1]Zhejiang University [2]The Hong Kong Polytechnic University [3]Ant Group
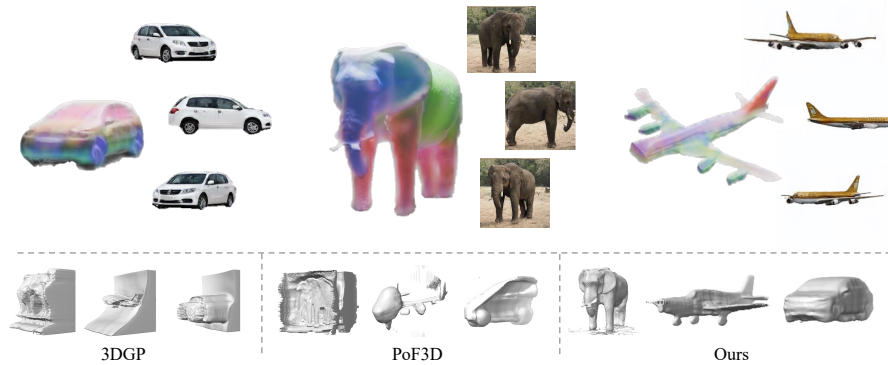
Fig. 1: We propose to perform on-the-fly pose estimation of training images with a learned template feature field. Our method enables learning complete 3D shapes from challenging datasets, including elephants, planes, and in-the-wild car datasets. While other methods have half or inaccurate geometry.

Abstract. Collecting accurate camera poses of training images has been shown to well serve the learning of 3D-aware generative adversarial networks (GANs) yet can be quite expensive in practice. This work targets learning 3D-aware GANs from unposed images, for which we propose to perform on-the-fly pose estimation of training images with a learned template feature field (`TeFF`). Concretely, in addition to a generative radiance field as in previous approaches, we ask the generator to also learn a field from 2D semantic features while sharing the density from the radiance field. Such a framework allows us to acquire a canonical 3D feature template leveraging the dataset mean discovered by the generative model, and further efficiently estimate the pose parameters on real data. Experimental results on various challenging datasets demonstrate the superiority of our approach over state-of-the-art alternatives from both the qualitative and the quantitative perspectives. Project page: https://XDimlab.github.io/TeFF.

Keywords: 3d-aware generation, semantic features, pose estimation

---

[*] Corresponding author.

1   Introduction

3D assets are an important part of popular media formats such as video games, movies, and computer graphics. Compared to designing 3D content manually which is expensive and time-consuming, learning generative 3D models from data is a promising alternative to reduce the design effort. It is particularly attractive to learn 3D generative models from easy-to-obtained 2D images that generalize to different categories.

Recent 3D-aware GANs have shown great success in this direction [4, 5, 31]. Most methods are designed for datasets with known camera pose distribution defined with respect to a canonical object pose. It has been proved that knowing the exact camera poses of real images further improves the learned geometry [3, 48]. However, estimating camera poses of real images requires domain-specific 3D knowledge about the object category, which is difficult to acquire for most in-the-wild objects. To remove the assumption of known camera poses, there are some recent attempts to use a generator to jointly learn the camera pose distribution and the 3D contents. Despite achieving promising performance on faces and synthetic objects [27, 33], these methods struggle in object categories with more complex pose distributions such as multi-peak distribution. This is mainly due to the fact that the generated camera poses and object poses are entangled in the 2D image space, leading to highly ill-posed solution space. For example, the objects may be generated facing different directions with a fixed camera so that the synthesized images can match the target distribution, as the result of 3DGP [35] and PoF3D [33] shown in Fig. 2. This leads to incomplete geometry as shown in Fig. 1 as parts of the geometry are never observed.

In this work, we tackle this challenge by disentangling the estimation of camera pose distribution from the training of the 3D-aware GANs. Our key idea is to learn a 3D semantic template feature field along with the generative model and define the pose estimation as an auxiliary task taking the template feature field as the canonical object space. This template feature field is automatically learned from self-supervised 2D features, *e.g.*, DINO features [2], which are semantically aligned across instances of different appearances, thus enabling pose estimation. Specifically, We augment the generative radiance field with a semantic feature field while sharing the density. Then we acquire a template feature field leveraging the dataset mean discovered by the generative model. With the learned 3D template feature field and a 2D semantic feature of a real image, we estimate the camera pose of the real image by solving a 3D-2D pose estimation task. For robust and efficient pose estimation, we propose to project the 3D template to the 2D image space using a discretized set of camera poses and find the one with the best matching. This allows for achieving global optimum pose estimation when an ideal template is provided. To address the slow matching process when the camera pose discretization space is of high dimensional, we propose to discretize the azimuth and elevation only, and leverage phase correlation to efficiently solve scale and in-plane rotation. We demonstrate that our method can generate full geometry even in complex pose distribution, including real-world cars, planes, and in-the-wild animals like elephants.
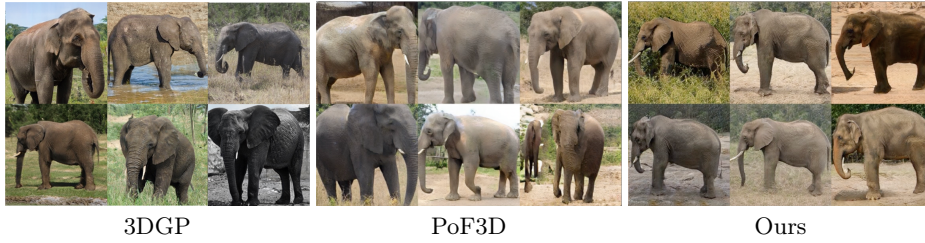
Fig. 2: Qualitative Comparison of 3DGP ($\{cols.1-3\}$), PoF3D ($\{cols.4-6\}$), Ours ($\{cols.7-9\}$) when rendering in the same camera pose. Note that objects generated by 3DGP and PoF3D face different directions with the same camera pose while ours face the same direction.

Our contributions can be summarized as follows, 1) We present a novel 3D-aware generative model that jointly learns a semantic template feature field to enable pose estimation of real-world images on the fly, thus enabling training without known pose distribution. 2) We propose to efficiently solve the camera pose estimation by incorporating phase correlation for estimation scale and in-plane rotation. 3) Our model learns 3D-aware generative models on multiple challenging datasets, including real-world cars, planes, and elephants.

2   Related Work

3D-Aware Image Synthesis: Recently, 3D-aware image synthesis has attracted growing attention by lifting the generator to the 3D space. A key to 3D-aware image synthesis is the choice of the underlying 3D representation. Early methods attempt to learn 3D voxel grid [13, 26] and mesh [21], yet these discretized representations limit the image fidelity. More recent methods exploit neural radiance fields [24] for 3D-aware image synthesis   [4, 5, 7, 8, 10–12, 15, 16, 23, 28, 29, 31, 32, 34, 39–42, 50]. Albeit achieving photorealistic 3D-aware image synthesis, these methods focus on datasets with known camera pose distribution, e.g., human and cat faces or synthetic car datasets.

3D-Aware Image Synthesis from Unposed images: There are a few attempts to address the task of 3D-aware unposed image synthesis [27, 33, 35, 46]. CAMPARI [27] designs the camera generator as a residual function to map a prior pose distribution to the target pose distribution. However, their performance is highly sensitive to the given pose prior. PoF3D [33] frees the requirements of 3D pose priors by learning a pose-free generator which map a Gaussian distribution to the target distribution and learning a pose-aware discriminator in an adversarial manner. These methods only consider camera poses of two degrees of freedom (DoF), e.g., azimuth and elevation angles, based on the assumption that all cameras lie on a sphere. Recently, 3DGP [35] extends the DoF from two to six by additionally modeling the field of view of the camera
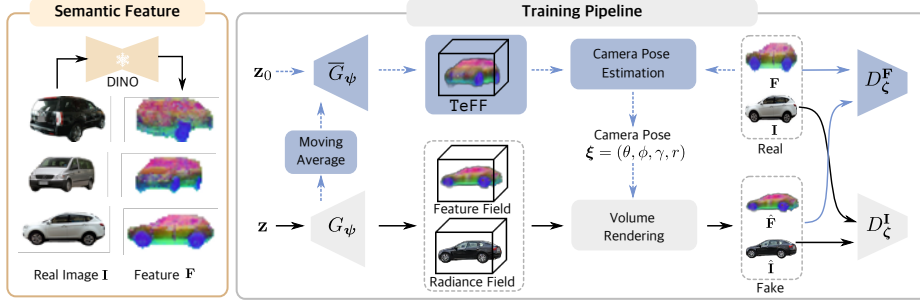
Fig. 3: Method Overview. We augment the generative radiance field with a semantic feature field, enabling estimating camera poses of real images on the fly to facilitate the 3D-aware GAN training. Specifically, we map a randomly sampled noise vector to a radiance field and a semantic feature field. By taking the mean shape of the feature field, we obtain a 3D template feature field. This allows us to perform efficient 2D-3D pose estimation to estimate camera poses of real images, which are in turn fed into the generator to perform volume rendering. The blue part is the auxiliary task of pose estimation we introduced.

and the look-at point. [46] model the translation and the rotation of the camera with 6 DoF. All these methods adopt a generator to jointly learn the camera pose distribution and the 3D contents. Despite performing well for data with a simple pose distribution, e.g., Gaussian distribution, these methods are prone to fail when the pose distribution becomes more complex. We instead learn a 3D semantic template feature field to solve the camera pose of each real image. Our method can generate full geometry even in complex pose distribution.

Neural Feature Fields: Self-supervised models, like DINO [2], can solve video instance segmentation and tracking by calculating the similarity among features in adjacent frames. [1] demonstrates that DINO features also work well on co-segmentation and point correspondence of different instances. Reconstruction methods have explored incorporating DINO features into neural fields for scene decomposition, editing, and semantic understanding [18,19,37,43]. In this work, we explore the semantic-aware characteristics of such features to enable camera pose estimation across different instances within the same category.

3   Method

Fig. 3 gives an overview of our method. We augment a 3D-aware GAN using our Template Feature Fields (`TeFF`) to enable learning from real-world images without camera pose distributions. The key to our approach is the joint learning of the generative radiance field and a semantic feature field, where a 3D template feature field can be extracted to solve the camera poses of the real images on the fly. In the following sections, we first introduce the offline semantic feature extraction in Sec. 3.1. Next, we present our generator augmented with `TeFF` in

Sec. 3.2. We further elaborate on the camera pose estimation of the real images based on the 2D feature maps and the 3D template feature field in Sec. 3.3, followed by the discriminator and implementation details in Sec. 3.4 and Sec. 4.1.

3.1   Semantic Feature Extraction

In this paper, we refer to semantic features as feature maps that are semantically aligned across instances of different appearances and poses, as illustrated in Fig. 3 (left). For example, the car wheels share the same features across different instances. Existing methods have demonstrated that such features can be leveraged to establish correspondences between different instances [1, 2], potentially enabling their relative pose estimation. In this work, we propose to extract semantic features from each real image to enable its camera pose estimation.

Specifically, we extract semantic features $\mathbf{F} \in \mathbb{R}^{H \times W \times F}$ from real RGB images $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ based on DINO [2], as we observe that DINO generalizes well and provides meaningful semantic features for various categories. In practice, we feed RGB images to the DINO model to get DINO feature maps. Then we extract foreground masks from these RGB images and apply the mask to DINO feature maps. This allows us to focus on the features of foreground instances.

3.2   Generator with `TeFF`

Generative Radiance and Feature Fields: We build our generator based on EG3D [3] but with a key difference in that the model jointly generates radiance fields and feature fields. Conditioned on a noise $\mathbf{z} \in \mathbb{R}^M$ sampled from a Gaussian distribution, the generator $G_{\boldsymbol{\psi}}$ generates a radiance field and a feature field that maps a 3D point $\mathbf{x}$ to a RGB value $\mathbf{c}$, feature value $\mathbf{f}$, and density value σ:

$$G_{\boldsymbol{\psi}} : \mathbb{R}^3 \times \mathbb{R}^M \to \mathbb{R}^3 \times \mathbb{R}^F \times \mathbb{R}^+ \quad (\mathbf{x}, \mathbf{z}) \mapsto (\mathbf{c}, \mathbf{f}, \sigma) \tag{1}$$

where F denotes the dimension of the feature vector. In practice, the generator produces two tri-planes, one for the color and density, the other for the feature.

The generative radiance field is rendered to a 2D image via the volume rendering operation. Given an estimated camera pose $\boldsymbol{\xi}$ (discussed in Sec. 3.3), the generator is queried on $H \times W$ rays with N sampling points on each ray. Let $\{\mathbf{c}_i, \mathbf{f}_i, \sigma_i\}_{i=1,\dots,N}$ denote the queried color, feature, and density values of one ray, the pixel color $\mathbf{c}_r$ and feature $\mathbf{f}_r$ at the corresponding ray is obtained via:

$$\pi : (\mathbb{R}^3 \times \mathbb{R}^+)^N \to \mathbb{R}^3 \qquad \{(\mathbf{c}_i, \mathbf{f}_i, \sigma_i)\} \mapsto \mathbf{c}_r, \mathbf{f}_r$$

$$\mathbf{c}_r = \sum_{i=1}^{N} T_i\, \alpha_i\, \mathbf{c}_i \quad \mathbf{f}_r = \sum_{i=1}^{N} T_i\, \alpha_i\, \mathbf{f}_i \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j) \quad \alpha_i = 1 - \exp(-\sigma_i \delta_i)$$

where T_i and α_i denote the transmittance and alpha value at a sample point $\mathbf{x}_i$, δ_i denotes the distance interval between two adjacent sample points. Note that the same density values are shared for rendering the color $\mathbf{c}_r$ and the feature
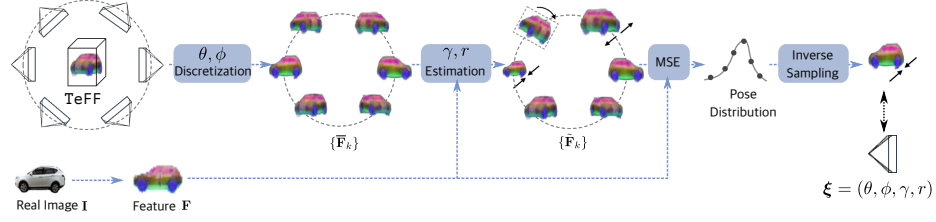
Fig. 4: Camera Pose Estimation. We leverage the template feature field to estimate camera poses of 2D real images. We discretize the azimuth θ and elevation ϕ angles and render the feature field from these discretized camera poses. Then we use phase correlation to estimate the scale and the in-plane rotation in the 2D image space and warp each of the rendered templates based on the solution. We calculate the mean square error between the warped rendering and the real feature and further obtain the probability distribution function of the camera pose. Finally, we sample the camera pose using inverse sampling.

$\mathbf{f}_r$. By compositing all ray values we obtain a color image $\hat{\mathbf{I}} \in \mathbb{R}^{H \times W \times 3}$ and a feature map $\hat{\mathbf{F}} \in \mathbb{R}^{H \times W \times F}$.

Following EG3D [3], we apply a super-resolution module to the rendered RGB image for upsampling via neural rendering. Note that the super-sampling module is not shown in Fig. 3 for brevity. We do not perform super-resolution modules to the rendered feature maps as we observe that low-resolution feature maps are sufficient for our purpose of pose estimation.

Template Feature Field: We propose to leverage the mean feature fields automatically discovered by the generator to enable the subsequent camera pose estimation. This mean feature field can be considered as a template for the full category. To obtain this template feature field, we first apply the exponential moving average to the generator G_{ψ} during training to get $\overline{G}_{\psi}$. Next, we feed the mean noise vector $\mathbf{z_0}$ to the moving averaged generator $\overline{G}_{\psi}$ to decode the feature and density for the template.

Background Generator: In order to estimate camera poses focusing on the objects, we disentangle the foreground and background generation. Following VoxGRAF [32], we generate the background of the RGB image with a 2D GAN for efficiency. Specifically, we use the StyleGAN2 generator [17] with reduced channel size as modeling the background requires less capacity than generating the full image. We use the same latent code $\mathbf{z}$ of the foreground for the background generator to allow for modeling correlations like lighting between the two. The final RGB image is obtained by alpha compositing the rendered foreground image and the 2D background.

3.3 Camera Pose Estimation

We leverage the 3D template feature field to estimate camera poses of 2D real images, which can be formulated as a 2D-3D camera pose estimation task. This

is possible since the features are semantically aligned across different instances. The task of 2D-3D matching is non-trivial. Many existing 2D-3D pose matching methods require establishing 2D-3D correspondences [30], followed by solving a Perspective-n-Point (PnP) problem for pose estimation. However, in our preliminary trials, it is non-trivial to establish correct 2D-3D correspondences (e.g., the left leg of an elephant in the 2D image might be matched to a right leg in the template), thus the estimated poses are unreliable. To tackle this challenge, we propose an efficient yet accurate way of solving the camera pose via grid search. The core idea is to discretize the camera pose space and select the one that renders a feature map that best matches the real feature. We now introduce how we define the camera model and how the grid search is efficiently implemented.

Camera Model: We assume our camera lies on a sphere and looks at the sphere center, with freedom degrees of azimuth, elevation, in-plane rotation, and sphere radius. More formally, we parameterize the camera pose $\boldsymbol{\xi} = (\theta, \phi, \gamma, r)$. Here, r denotes the radius of the sphere. $\theta \in [0, 2\pi]$ and $\phi \in [0, \pi]$ denote the azimuth and elevation angles of the camera's translation in the spherical coordinate system. By looking at the sphere center from the direction determined by θ and ϕ and performing the in-plane rotation γ, we can determine the camera's rotation. Note that, r determines the scale of objects in the image, we use scale to refer it in the following.
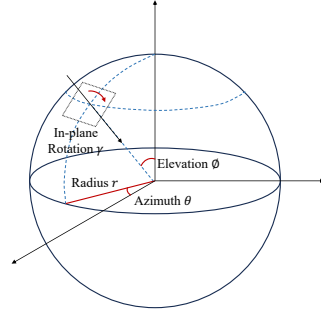

Fig. 5: Camera Model.

Azimuth and Elevation Discretization: With this 4 DoF camera model, a naïve grid search can be implemented by discretizing all four variables $(\theta, \phi, \gamma, r)$ to form a large number of camera poses, and render the feature field into a set of feature maps, followed by comparing them with a feature map of the real image and selecting the best-matching one. This exhaustive grid search allows for finding the best-matching pose among the candidates, yet it is extremely computationally expensive. Hence, we propose to combine grid search with the efficient phase correlation approach [20], allowing for efficiently estimating the camera pose with a high accuracy. Specifically, we discretize the azimuth θ and elevation ϕ angles into N_θ and N_ϕ values, respectively. Next, we render the template feature fields from these discretized azimuth θ and elevation ϕ angles, yielding a set of 2D template features $\{\overline{\mathbf{F}}_k\}_{k=1}^{N_\theta \times N_\phi}$.

Scale and In-plane Rotation Estimation: Given a semantic feature map $\mathbf{F}$ of a real image $\mathbf{I}$, we may retrieve its best-matching template feature from $\{\overline{\mathbf{F}}_k\}$ and assign the corresponding pose to $\mathbf{I}$. However, this does not model scale r and in-plane rotation γ. Therefore, we further estimate the r and γ between the real feature map $\mathbf{F}$ and *each* 2D template feature $\overline{\mathbf{F}}_k$. Considering the efficiency, we employ phase correlation (PC) [20], a classical frequency-domain-based approach that allows for estimating the scale and in-plane rotation between two images

efficiently, without relying on any explicit correspondence or initialization. This is because PC estimates scale and in-plane rotation following the same idea as grid search, yet the calculations are performed in the frequency domain for fast computation. Based on the estimated scale r and in-plane rotation γ, we warp each 2D template image $\overline{\mathbf{F}}_k$ to get $\tilde{\mathbf{F}}_k$. Note that, the idea of performing image registration in the frequency domain is also studied in other works for different purposes [9, 49].

Camera Pose Sampling: We calculate the mean square error (mse) between each $\tilde{\mathbf{F}}_k$ and the real semantic feature map $\mathbf{F}$ as:

$$e_k(\tilde{\mathbf{F}}_k, \mathbf{F}) = \|\tilde{\mathbf{F}}_k - \mathbf{F}\|_2^2 \tag{2}$$

Ideally, we can assign the camera pose of $\tilde{\mathbf{F}}_k$ corresponding to the lowest e_k to the real feature map $\mathbf{F}$. However, the feature template may not exactly match each real image, especially at the early training stage, occasionally leading to inaccurate pose estimation. Therefore, we form a probability distribution function (PDF) over all possible $N_\theta \times N_\phi$ poses and sample from it during training. Specifically, we sample a camera pose using inverse sampling according to the PDF of the camera pose. The pose probability $p(k)$ is calculated by:

$$p(k) = softmax(-e_k(\tilde{\mathbf{F}}_k, \mathbf{F}) * \tau) \tag{3}$$

where τ is the temperature, which controls the sharpness of the probability distribution function. At the early of training, we use low temperature τ and then increase the temperature linearly. Since we discretize the camera pose space, we further add a slight Gaussian noise to the sampled pose. This allows us to sample continuous camera pose locations from each discretized bin. During inference, we select the camera pose with the maximum likelihood.

3.4   Discriminator and GAN Training

Discriminator: In order to train our feature field, we add a feature discriminator $D_\zeta^{\mathbf{F}}$ on top of the original image discriminator $D_\zeta^{\mathbf{I}}$. $D_\psi^{\mathbf{I}}$ is used to discriminate real RGB images and fake RGB images, whereas $D_\zeta^{\mathbf{F}}$ learns to discriminate the semantic features. Following [3], we adopt the dual discriminator with camera pose condition for $D_\zeta^{\mathbf{I}}$, i.e., the input to $D_\zeta^{\mathbf{I}}$ is an upsampled low-resolution RGB image, a high-resolution RGB image, and the corresponding camera pose. The input to $D_\zeta^{\mathbf{F}}$ is the low-resolution RGB image and the corresponding semantic feature map, to encourage the learned feature field to match with the corresponding radiance field. We stop the gradient backpropagation from $D_\zeta^{\mathbf{F}}$ into the RGB values following [36].

GAN Training Given the latent code $\mathbf{z}$, a real image $\mathbf{I}$ and its corresponding semantic feature $\mathbf{F}$ sampled from the real-data distribution $p_\mathcal{D}$, we train our model using non-saturated GAN loss with R1 regularization [22].

$$\begin{aligned} \mathcal{L} = &\mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0,1)} \left[f(D_\zeta^{\mathbf{I}}(G_\psi(\mathbf{z}))) + f(D_\zeta^{\mathbf{F}}(G_\psi(\mathbf{z}))) \right] \\ &+ \mathbb{E}_{\mathbf{I},\mathbf{F} \sim p_\mathcal{D}} \left[f(-D_\zeta^{\mathbf{I}}(\mathbf{I})) + \lambda \|\nabla D_\zeta^{\mathbf{I}}(\mathbf{I})\|^2 + f(-D_\zeta^{\mathbf{F}}(\mathbf{F})) + \lambda \|\nabla D_\zeta^{\mathbf{F}}(\mathbf{F})\|^2 \right] \end{aligned}$$

Training Strategy: The pose estimation is done on the fly in the early stage. We observe that once the model learns a reasonable template, we can use the camera poses estimated from a fixed template for the remaining training. We update the template once every 16 iterations before 3k iterations and then update the template once epoch.

4   Experiments

Datasets: We evaluate on four datasets, including Shapenet Cars [3,6], Comp-Cars [44], SDIP Elephant [25], and LSUN Plane [47]. Shapenet Cars is a synthetic dataset that contains car images with ground truth camera poses. Note that the ground truth poses are only used for evaluation for all methods. As the camera poses of the original dataset are roughly uniformly distributed, we consider a more challenging scenario by building a sub-dataset where the pose distribution has multiple peaks, with around 103K images. CompCars contains 136k unposed images capturing the entire cars with different styles. We mask the background to black and filter the data with bad mask estimation and extreme scale, leading to 110k images. We use the SDIP Elephant dataset following [35] and filter the elephants with the half body, which contains around 20k unposed images. LSUN Plane is a dataset that contains unposed images of different planes. We mask the background to white and filter the data with extreme scale and occluded planes, leading to 130k images. In the experiments, we use the resolution of 256×256 for SDIP Elephant, CompCars and LSUN Plane, and 128×128 for Shapenet Cars.

Baselines: We consider three baselines: EG3D [3], PoF3D [33] and 3DGP [35]. EG3D is a state-of-the-art 3D-aware GAN for posed images, whereas PoF3D and 3DGP both learn from unposed images while jointly learning the camera pose distribution. For EG3D we uniformly sample pose with azimuth angle sampled from 360 degrees and elevation from 85 degrees to 95 degrees. Note that the original EG3D conditions the generator and discriminator on the camera poses. We do not use this pose condition for EG3D due to the lack of image-pose pair data. For 3DGP, we set the prior azimuth angle with a uniform distribution of 360 degrees and set the prior elevation angle with a uniform distribution from 85 degrees to 95 degrees.

Metrics: We use three metrics to evaluate the performance, including Frechet Inception Distance (FID) [14], Depth Error [3], and Kullback-Leibler (KL) Divergence. FID is measured between 50K generated images and all real images. Note that, we evaluate the FID using two sets of camera poses. As the camera pose distributions estimated by different methods are different, we first sample images using the ground truth pose distribution for evaluation as "FID_{gt}". This ensures a fair comparison of different methods. In cases where the ground truth pose distribution is not available on CompCars, SDIP Elephant and LSUN Plane, we render fake images from a uniform distribution with azimuth angle sampled from 360 degrees and evaluate their fidelity as "FID_{360}". This is rationale as those datasets contains images captured from 360 degree azimuth angles. We

further evaluate FID using the estimated camera poses, denoted as "FID_{est}" to reflect the image fidelity at the learned camera distributions. Depth Error is used to assess the quality of geometry. Following [3], we use a monocular depth estimation algorithm [45] to get the pseudo depth and use a mask estimation algorithm [38] to obtain the foreground mask. We randomly generate 10k images and only evaluate the depth error of the foreground objects. We normalize the depth by subtracting the mean depth of the current sample and dividing the standard deviation depth of the dataset. Similar to FID we also evaluate "$Depth_{gt}$"/"$Depth_{360}$" and "$Depth_{est}$". The standard deviation depth uses the one calculated in the estimated camera pose setting since the depth estimation and mask estimation may fail in the 360 camera pose setting. For pose distribution estimation, we randomly sample 10k ground truth poses and estimated poses, and then calculate their KL Divergence. We adopt this distribution loss since the baseline methods do not provide an estimated camera pose for each real image but learn a distribution.

4.1   Implementation Details

In our implementation, We extract the semantic features from DINO [2] using real images of 256×256 resolution and get the DINO features of 64×64 resolution. We apply Principal Component Analysis to the feature maps and keep the first three principal components of the original DINO features for training efficiency. This means the feature channel F is set to 3.

The standard deviation of the noise is $1/6$ of the discrete interval. We discretize the azimuth θ and elevation ϕ angles into N_θ and N_ϕ values, respectively. The azimuth range is 360 degrees and N_θ is 36 for all datasets. The elevation range is 180 degrees and N_ϕ is 18 for the ShapeNet Cars dataset. For other datasets, the elevation range is from 85 to 95 degrees and N_ϕ is 3 since the elevation variation is small.

4.2   Comparison to the Baselines

Qualitative Comparison: Fig. 6, Fig. 7, Fig. 8 shows the rendering comparison against the baselines on SDIP Elephant, CompCars, and LSUN Plane. We render the image from 360 degrees at 60-degree intervals. The bottom right of the image illustrates the corresponding geometry. EG3D [3], 3DGP [35], and PoF3D [33] perform well in some specific views but fail to perform 360-degree rendering with consistency due to the incomplete or inaccurate geometry. Instead, our method can perform 360-degree rendering with good fidelity. Besides, our method allows for learning the complete geometry with high quality, whereas the baselines fail to reconstruct the geometry correctly or only reconstruct half of the object.

Quantitative Comparison: We report the quantitative evaluation of baselines and our method in Tab. 1 and Tab. 2. The result suggests that EG3D trained with unknown camera pose distribution can achieve a low FID, yet results in
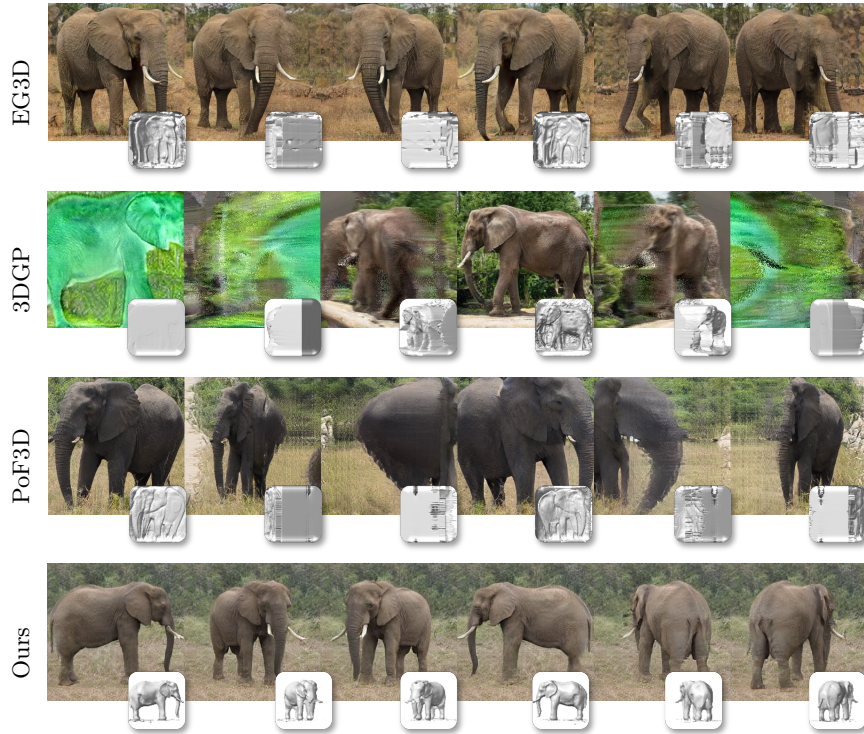
Fig. 6: Qualitative Comparison on SDIP Elephant. We show each sample from 360 viewing directions.

the worst geometry. Note that EG3D achieves the best FID_{360} since the azimuth angle is uniformly sampled from 360 degrees during training. Despite having high fidelity for all surrounding viewpoints, EG3D does not recover the correct geometry and thus yields bad $Depth_{360}$, e.g., as shown in Fig. 6. 3DGP and PoF3D achieve the best FID when evaluated on their estimated camera poses, whereas the performance drops significantly when evaluated using camera poses uniformly sampled on CompCars, SDIP Elephant and LSUN Plane. The reason is that 3DGP and PoF3D learn a collapsed camera pose distribution, i.e., it only covers a small range of the underlying true pose distribution. Our method instead retains a similar performance when evaluated using the uniform distribution, i.e., our FID_{360} and FID_{est} are close, thanks to the fact that our method learns to recover a full object in the 3D space for reasonable rendering from all surrounding viewpoints. This is also verified by the fact that our method performs the best in terms of the $Depth_{360}$. Note that 3DGP achieves the best $Depth_{est}$ since it uses the depth maps as supervision during training, and we use the same monocular depth estimation algorithm to provide depth supervision for 3DGP and for evaluating its depth performance. Despite not relying on any depth
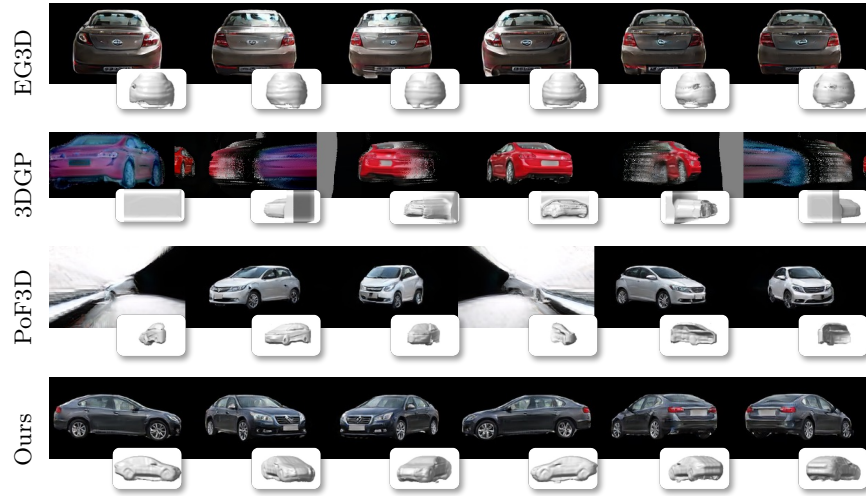
Fig. 7: Qualitative Comparison on CompCars. We show each sample from 360 viewing directions.

Method	Multiple-Peak Shapenet Cars				CompCars			
	$\text{Depth}_{gt}\downarrow$	$\text{Depth}_{est}\downarrow$	$\text{FID}_{gt}\downarrow$	$\text{FID}_{est}\downarrow$	$\text{Depth}_{360}\downarrow$	$\text{Depth}_{est}\downarrow$	$\text{FID}_{360}\downarrow$	$\text{FID}_{est}\downarrow$
EG3D	0.61	0.61	7.25	11.89	0.95	0.95	**7.06**	7.06
3DGP	4.84	**0.46**	139.48	**4.93**	4.02	0.37	187.20	8.34
PoF3D	0.65	0.70	12.72	6.45	10.31	0.70	44.52	**6.62**
Ours	**0.53**	0.52	**5.95**	6.55	**0.31**	**0.29**	27.71	20.60

Table 1: Quantitative Comparison on Shapenet Cars and CompCars.

supervision, our method even achieves comparable Depth_{est} compared to 3DGP on the car datasets in Tab. 1.

Pose Distribution Comparison: We use the ShapeNet Cars with ground truth poses to evaluate the accuracy of the estimated camera pose distribution. We discretize the azimuth angle and the elevation angle into 24 and 12 intervals, and calculate the discrete probability distributions over these intervals. Fig. 9 demonstrate the comparison between the GT and estimated pose distribution of different methods regarding azimuth and elevation angles. It is shown that the pose distribution of 3DGP is collapsed to

Method	θ KL $\downarrow$	ϕ KL $\downarrow$
3DGP	40.4571	39.3625
PoF3D	4.4829	0.5495
Ours	**0.0555**	**0.0696**

Table 3: Pose Distribution Comparison.

a small range. PoF3D covers a larger range but fails to capture the multi-modal distribution: the learned distribution remains Gaussian, similar to its initialization. In contrast, our estimated pose distribution is notably more accurate. This is also verified by the quantitative comparison of the KL divergence in Tab. 3, where our distribution is significantly better than that of 3DGP and PoF3D. This superior pose distribution contributes to our ability to better recover the full object geometry.

Fig. 8: Qualitative Comparison on LSUN Plane. We show each sample from 360 viewing directions with a 60-degree interval.

Method	SDIP Elephant				LSUN Plane			
	$\text{Depth}_{360}\downarrow$	$\text{Depth}_{est}\downarrow$	$\text{FID}_{360}\downarrow$	$\text{FID}_{est}\downarrow$	$\text{Depth}_{360}\downarrow$	$\text{Depth}_{est}\downarrow$	$\text{FID}_{360}\downarrow$	$\text{FID}_{est}\downarrow$
EG3D	1.10	1.10	6.03	6.03	1.19	1.19	**7.27**	7.27
3DGP	3.29	**0.30**	196.04	**2.18**	3.84	**0.33**	196.29	**3.79**
PoF3D	3.14	1.08	36.32	3.04	1.37	1.07	16.26	6.07
Ours	**0.60**	0.68	**5.51**	5.44	**0.78**	0.78	14.20	16.55

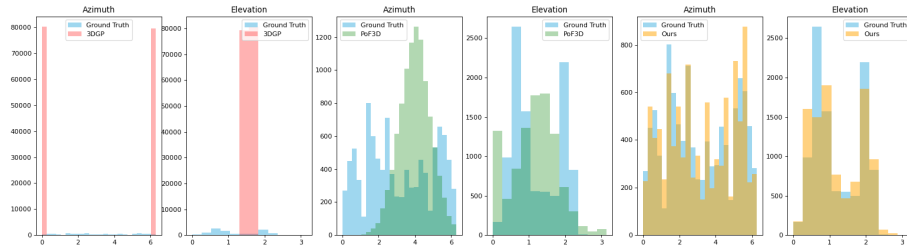**Table 2: Quantitative Comparison** on SDIP Elephant and LSUN Plane.

Fig. 9: Pose Distribution Comparison on Shapenet Cars.

4.3 Ablation Study

Template Feature Field: We now ablate the design choice of using template feature fields for pose estimation. Alternatively, one may directly obtain a *template radiance field* (TeRF) for pose estimation, without introducing the additional feature branch. This template radiance field and the real RGB images can be leveraged to solve the camera pose estimation task using the method introduced in Sec. 3.3. To further enhance the robustness against color variation, one may further convert the RGB images to gray for pose estimation. We compare our semantic feature-based method to these

Method	TeRF$_{RGB}$	TeRF$_{Gray}$	**TeFF(Ours)**
θ KL $\downarrow$	0.0663	0.0656	**0.0555**
ϕ KL $\downarrow$	0.1422	0.1490	**0.0696**

Table 4: Ablation Study on Shapenet Car.

DoF	Depth$_{360}$ $\downarrow$	Depth$_{est}$ $\downarrow$	FID$_{360}$ $\downarrow$	FID$_{est}$ $\downarrow$
θ, ϕ	4.98	1.16	39.66	**11.09**
θ, ϕ, γ, r	**0.31**	**0.29**	**27.31**	20.60

Table 5: Ablation Study on CompCars.

two alternatives on the Shapenet Cars dataset with ground truth pose distribution. As shown in Tab. 4, directly leveraging a template radiance field for pose estimation leads to worse pose estimation, while our `TeFF` is semantically aligned across instances of different appearances and achieves better pose estimation.

Freedom of Degrees: To demonstrate the effect of leveraging four degrees of freedom (DoF), we conduct experiments on the CompCars dataset since this dataset has a large 2D scale variation. We compare with an alternative that omits the phase correlation for estimating in-plane rotation γ and scale r. Although training with only two DoF (θ, ϕ) achieves lower FID$_{est}$ as shown in Tab. 5, we observe that the generated 3D samples have wrong geometry. This observation is also supported by the worse FID$_{360}$ and depth metrics, demonstrating the importance of modeling the scale and in-plane rotation in the camera model.

5   Limitation

Our method can not model images with significant perspective distortion. To fit the distribution of real images, the perspective effect is modeled by generated geometry and leads to the failure case like Fig. 10. The 2D-3D pose estimation may be influenced by the geometry of the instance since we use the mse to select the best-matching camera

Fig. 10: Failure case.

pose. In the future, we will explore disentangling the geometry information during the matching process. Besides, we do not model the articulation of the instance. The generator may model different articulations in different views and occasionally lead to instances with multiple legs.

6   Conclusion

We present `TeFF`, a novel approach to enable learning 3D-aware generative models from in-the-wild images with unknown camera pose distribution. By estimating the camera poses of real images on the fly, we demonstrate that our method is capable of recovering the full 3D geometry from datasets of challenging distributions. Despite achieving promising results on in-the-wild images, our method learns a single feature template and hence is only applicable to a specific category. Future work may explore the usage of multiple templates for learning a single 3D-aware generative model on images of multiple categories.

References

1. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep vit features as dense visual descriptors. ECCVW What is Motion For? (2022)
2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
3. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3d generative adversarial networks. In: CVPR (2022)
4. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3D generative adversarial networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
5. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: Pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
6. Chang, A.X., Funkhouser, T.A., Guibas, L.J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: Shapenet: An information-rich 3d model repository. arXiv.org **1512.03012** (2015)
7. Chen, X., Deng, Y., Wang, B.: Mimic3d: Thriving 3d-aware gans via 3d-to-2d imitation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
8. Chen, X., Huang, J., Bin, Y., Yu, L., Liao, Y.: Veri3d: Generative vertex-based radiance fields for 3d controllable human image synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8986–8997 (October 2023)
9. Chen, Z., Xu, X., Wang, Y., Xiong, R.: Deep phase correlation for end-to-end heterogeneous sensor measurements matching. In: Proceedings of the 2020 Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 155, pp. 2359–2375. PMLR (16–18 Nov 2021), https://proceedings.mlr.press/v155/chen21g.html
10. Deng, K., Liu, A., Zhu, J., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. arXiv.org **2107.02791** (2021)
11. Deng, Y., Yang, J., Xiang, J., Tong, X.: Gram: Generative radiance manifolds for 3d-aware image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10673–10683 (2022)
12. Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. Int. Conf. Learn. Represent. (2022)
13. Henzler, P., Mitra, N.J., Ritschel, T.: Escaping plato's cave: 3d shape from adversarial rendering. In: Int. Conf. Comput. Vis. (2019)
14. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Adv. Neural Inform. Process. Syst. (2017)
15. Jo, K., Jin, W., Choo, J., Lee, H., Cho, S.: 3d-aware generative model for improved side-view image synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22862–22872 (2023)
16. Jo, K., Shim, G., Jung, S., Yang, S., Choo, J.: Cg-nerf: Conditional generative neural radiance fields. arXiv.org (2021)

17. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: Proc. CVPR (2020)
18. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: International Conference on Computer Vision (ICCV) (2023)
19. Kobayashi, S., Matsumoto, E., Sitzmann, V.: Decomposing nerf for editing via feature field distillation. In: Advances in Neural Information Processing Systems. vol. 35 (2022), https://arxiv.org/pdf/2205.15585.pdf
20. Kuglin, C.D.: The phase correlation image alignment method (1975), https://api.semanticscholar.org/CorpusID:61133413
21. Liao, Y., Schwarz, K., Mescheder, L., Geiger, A.: Towards unsupervised learning of generative models for 3d controllable image synthesis. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
22. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: Int. Conf. Mach. Learn. (2018)
23. Michael, N., Andreas, G.: Giraffe: Representing scenes as compositional generative neural feature fields. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021. pp. 11453–11464. IEEE Computer Society (2021)
24. 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: Eur. Conf. Comput. Vis. (2020)
25. Mokady, R., Yarom, M., Tov, O., Lang, O., Daniel Cohen-Or, Tali Dekel, M.I., Mosseri, I.: Self-distilled stylegan: Towards generation from internet photos (2022)
26. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.: Hologan: Unsupervised learning of 3d representations from natural images. In: Int. Conf. Comput. Vis. (2019)
27. Niemeyer, M., Geiger, A.: Campari: Camera-aware decomposed generative neural radiance fields (2021)
28. Or-El, R., Luo, X., Shan, M., Shechtman, E., Park, J., Kemelmacher, I.: Stylesdf: High-resolution 3d-consistent image and geometry generation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
29. Pan, X., Xu, X., Loy, C.C., Theobalt, C., Dai, B.: A shading-guided generative implicit model for shape-accurate 3d-aware image synthesis. In: Adv. Neural Inform. Process. Syst. (2021)
30. Peng, S., Liu, Y., Huang, Q.X., Bao, H., Zhou, X.: Pvnet: Pixel-wise voting network for 6dof pose estimation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4556–4565 (2018), https://api.semanticscholar.org/CorpusID:57189382
31. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: Adv. Neural Inform. Process. Syst. (2020)
32. Schwarz, K., Sauer, A., Niemeyer, M., Liao, Y., Geiger, A.: Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. Adv. Neural Inform. Process. Syst. (2022)
33. Shi, Z., Shen, Y., Xu, Y., Peng, S., Liao, Y., Guo, S., Chen, Q., Yeung, D.Y.: Learning 3d-aware image synthesis with unknown pose distribution (2023)
34. Shin, M., Seo, Y., Bae, J., Choi, Y.S., Kim, H., Byun, H., Uh, Y.: Ballgan: 3d-aware image synthesis with a spherical background. arXiv preprint arXiv:2301.09091 (2023)
35. Skorokhodov, I., Siarohin, A., Xu, Y., Ren, J., Lee, H.Y., Wonka, P., Tulyakov, S.: 3d generation on imagenet. In: International Conference on Learning Representations (2023), https://openreview.net/forum?id=U2WjB9xxZ9q

36. Sun, J., Wang, X., Zhang, Y., Li, X., Zhang, Q., Liu, Y., Wang, J.: Fenerf: Face editing in neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7672–7682 (2022)

37. Tschernezki, V., Laina, I., Larlus, D., Vedaldi, A.: Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In: Proceedings of the International Conference on 3D Vision (3DV) (2022)

38. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. https://github.com/facebookresearch/detectron2 (2019)

39. Xiang, J., Yang, J., Deng, Y., Tong, X.: Gram-hd: 3d-consistent image generation at high resolution with generative radiance manifolds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2195–2205 (2023)

40. Xu, X., Pan, X., Lin, D., Dai, B.: Generative occupancy fields for 3d surface-aware image synthesis. In: Adv. Neural Inform. Process. Syst. (2021)

41. Xu, Y., Peng, S., Yang, C., Shen, Y., Zhou, B.: 3d-aware image synthesis via learning structural and textural representations. IEEE Conf. Comput. Vis. Pattern Recog. (2022)

42. Xue, Y., Li, Y., Singh, K.K., Lee, Y.J.: Giraffe hd: A high-resolution 3d-aware generative model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18440–18449 (2022)

43. Yang, J., Ivanovic, B., Litany, O., Weng, X., Kim, S.W., Li, B., Che, T., Xu, D., Fidler, S., Pavone, M., Wang, Y.: Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. arXiv preprint arXiv:2311.02077 (2023)

44. Yang, L., Luo, P., Loy, C.C., Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. pp. 3973–3981. IEEE Computer Society (2015). https://doi.org/10.1109/CVPR.2015.7299023, https://doi.org/10.1109/CVPR.2015.7299023

45. Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., Shen, C.: Learning to recover 3d scene shape from a single image. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR) (2021)

46. Yu, F., Wang, X., Li, Z., Cao, Y., Shan, Y., Dong, C.: GET3D-: learning GET3D from unconstrained image collections. CoRR **abs/2307.14918** (2023). https://doi.org/10.48550/ARXIV.2307.14918, https://doi.org/10.48550/arXiv.2307.14918

47. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv.org **1506.03365** (2015)

48. Zhao, X., Ma, F., Güera, D., Ren, Z., Schwing, A.G., Colburn, A.: Generative multiplane images: Making a 2d gan 3d-aware. In: Proc. ECCV (2022)

49. Zhao, X., Zhao, Z., Schwing, A.G.: Initialization and alignment for adversarial texture optimization. In: European Conference on Computer Vision (ECCV) (2022)

50. Zhou, P., Xie, L., Ni, B., Tian, Q.: CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. arXiv.org (2021)