# GVP: Generative Volumetric Primitives

Mallikarjun B R [1,2], Xingang Pan [1], Mohamed Elgharib [1], Christian Theobalt [1,2],
[1] Max Planck Institute for Informatics [2] Saarland University

## Abstract

*Advances in 3D-aware generative models have pushed the boundary of image synthesis with explicit camera control. To achieve high-resolution image synthesis, several attempts have been made to design efficient generators, such as hybrid architectures with both 3D and 2D components. However, such a design compromises multiview consistency, and the design of a pure 3D generator with high resolution is still an open problem. In this work, we present Generative Volumetric Primitives (GVP), the first pure 3D volumetric generative model that can sample and render 512-resolution images in real-time. GVP jointly models a number of volumetric primitives and their spatial information, both of which can be efficiently generated via a 2D convolutional network. The mixture of these primitives naturally captures the sparsity in the 3D volume. The training of such a generator with a high degree of freedom is made possible through a combination of adversarial and knowledge distillation training. The learned model exhibits dense 3D correspondences between samples. We provide exhaustive qualitative and qualitative evaluations for dense correspondences. Experiments on several datasets demonstrate superior efficiency, 3D consistency, and the emergence of dense correspondences of GVP over the state-of-the-art.*

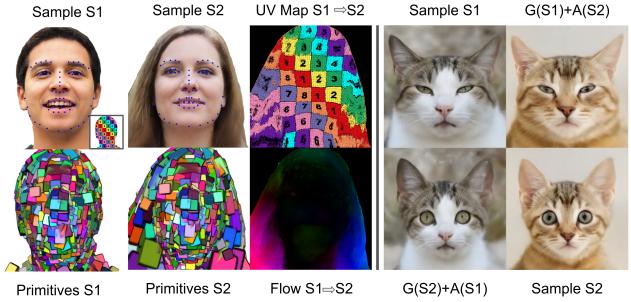


Figure 1. We propose the first 3D-aware generative model that relies on pure 3D volumetric representation and can render images at a resolution of $512 \times 512$ in real-time. Our method provides dense 3D correspondences between samples. Here we show 2 random samples (S1, and S2) and their primitives visualization. We visualize the learned dense correspondences in 3 ways. We track the predicted key points on Sample S1 to Sample S2 using primitives geometric properties. Similarly, we show the tracked UV map (Sample S1-inset) from Sample S1 to S2 (top-right). Finally, we visualize the magnitude and direction (bottom-right) of tracked pixels. We also provide appearance and geometry transfer between 2 samples on the right side of the vertical line. S1 and S2 are 2 random samples. We take the color and density output (denoted A) corresponding to a sample and primitives' spatial information (denoted G) corresponding to the other to render the final result.

## 1. Introduction

Synthesizing photorealistic objects with high-resolution and multi-view consistency is a long-term goal in computer vision and graphics. Recently, advances in 3D-aware generative adversarial networks (GANs) have made inspiring progress towards this goal [5, 6, 14, 22, 27]. By learning from unstructured 2D images, 3D-aware GANs can synthesize different views of generated objects with explicit control over camera pose. This is made possible through 3D generator architectures that naturally possess a 3D inductive bias. The key challenge towards a high-quality 3D-aware GAN lies in the design of an efficient 3D generator. For example, voxel-based 3D representation [15] suffers from cubic memory growth, and thus can only afford a limited resolution. Recently, a more widely used representation for 3D-aware GAN is neural radiance fields (NeRF) [5, 21, 27], which integrates a coordinate-based MLP and volume rendering. As NeRF requires querying the MLP densely across 3D space, it is prohibitively computationally expensive for high-resolution image rendering. Several works propose to improve efficiency via a compromised hybrid generator that models a 3D feature space via NeRF followed by a 2D CNN-based renderer that upsamples the feature to a high-resolution 2D image [6, 14, 22, 23]. However, the use of a 2D CNN compromises 3D consistency and is vulnerable to artifacts for camera views unseen during training. Although a few recent works attempt to improve the efficiency of a pure 3D generative model [28, 29], the synthesis of high-resolution images in real-time remains an open problem.

In this work, we present Generative Volumetric Primitives (GVP), which, to the best of our knowledge, is the

first 3D-aware generative model that is based on a pure 3D volumetric representation and can sample and render 512-resolution images in real-time. To achieve high rendering efficiency, we take the first attempt to build a 3D generative model based on the mixture of volumetric primitives (MVP) representation [19]. Each primitive model the color and density of a local volume and a number of primitives are composed together to form the entire 3D volume, as shown in Fig. 2. Unlike the original MVP that heavily relies on mesh tracking to determine the spatial information (*i.e.*, position, orientation, and scale) of the primitives, our generator learns to model the volumetric primitives and their spatial information jointly. Such a representation can be efficiently modeled via a 2D convolutional network architecture and naturally captures spatial sparsity, thus is extremely efficient to sample and render compared to previous 3D-aware GANs.

The use of multiple movable primitives also introduces a high degree of freedom to our generator, making it unstable to train via only a conventional adversarial loss on the raw training images. To address this issue, we propose to regularize adversarial training with knowledge distillation of a pretrained 3D-aware GAN, EG3D [6]. Such knowledge distillation significantly stabilizes training and inherits the generative capability of EG3D. Although EG3D has inconsistent rendering for out-of-distribution camera views, we found the in-distribution views sufficient to train our model with high quality, which would then be able to perform consistent rendering for any camera views.

We conduct an extensive evaluation of GVP on several datasets including human face, cat face, and cars. Thanks to the highly efficient design, GVP achieves much faster rendering than previous pure 3D GANs and preserves better multiview consistency compared with those with hybrid architectures. Visualization of the learned primitives demonstrates that GVP effectively captures the sparsity of the 3D volume, avoiding redundant computation and memory costs. The predicted positions of the volumetric primitives also adapt to different samples, providing dense correspondence between samples. Fig **??** show the quality of correspondences obtained between 2 samples using tracked landmarks, UV map and magnitude and orientation of tracked pixels. We also provide quantitative analysis to evaluate learned correspondences.

## 2. Related Works

Building generative models is a long-standing computer vision problem. Early efforts include works such as learning 3D Morphable models of the human face [2, 25]. These approaches, however, lack photorealism and are trained using controlled data captured with multi-view camera rigs. With the advances in deep learning, several generative models have been proposed [16, 17]. These models are learned purely from in-the-wild 2D images and produce highly photorealistic results. However, they are 2D-based and hence produce results that are not multi-view consistent. The past couple of years witnessed a rapidly growing interest in 3D generative models. This is largely fueled with the recent advances in implicit scene representations and Neural Radiance Fields (NeRF) [21, 24] which quickly revolutionized 3D scene understanding and rendering in an unprecedented manner. The methods that utilizes surface based representation [13], although have the advantage of efficiency, they lack in photorealistic rendering. This is because surface based representation inherently can not handle translucent and thin structures. Although GET3D [13] can be sampled and rendered in realtime, it lacks in photorealism.

The works of Chan *et al*. [5] and Schwarz *et al*. [27] were one of the first to show how to marry volumetric [21] representation with adversarial training. D3D [30] extended similar approach that could disentangle shape from appearance. But these methods use co-ordinate based MLP to parameterize their generator and train in an adversarial manner. As they need to densely sample space to render the scene, it is quite slow. VoxGRAF [28] acknowledges the fact that volumetric rendering is usually very slow due to querying multiple points along rays. To address this limitation, VoxGRAF utilizes sparse voxel grid representations. This improves computational efficiency, but still, their method is not real-time even at $256 \times 256$ resolution. They [28] can render in real-time only after they have the scene generated. On the other hand, our method can both synthesize the scene and render faster than real-time rate. Deng *et al*. [12] proposed an alternative solution to handle the large computational requirements of volumetric rendering. Their approach uses 2D manifolds to guide point sampling and radiance field learning. These manifolds are embedded in the 3D volume and help in reducing the computational complexity while still producing multi-view consistent results. However, such manifolds can lead to clear rendering artifacts, especially in side-view angles, where the manifolds are almost parallel to camera rays. Rebain *et al*. [26] showed it is possible to learn a generative model using single-view in-the-wild 2D images without adversarial training. They follow an autodecoder architecture that learns a shared latent representation. The method is trained in a self-supervised manner, where an off-the-shelf 2D landmark detector is used to determine the camera poses. While most of these methods [5, 28] and more [8, 29, 32] produce photorealistic renderings that are 3D-consistent, they suffer from expensive computational requirements that prohibit them from rendering in real-time.

To address the expensive computational requirement of the previously discussed generative models, another class of methods performs volumetric rendering in a low-resolution instead [1, 6, 14, 22, 23, 31, 34]. This is then followed by a super-resolution network that upscales the output to the full

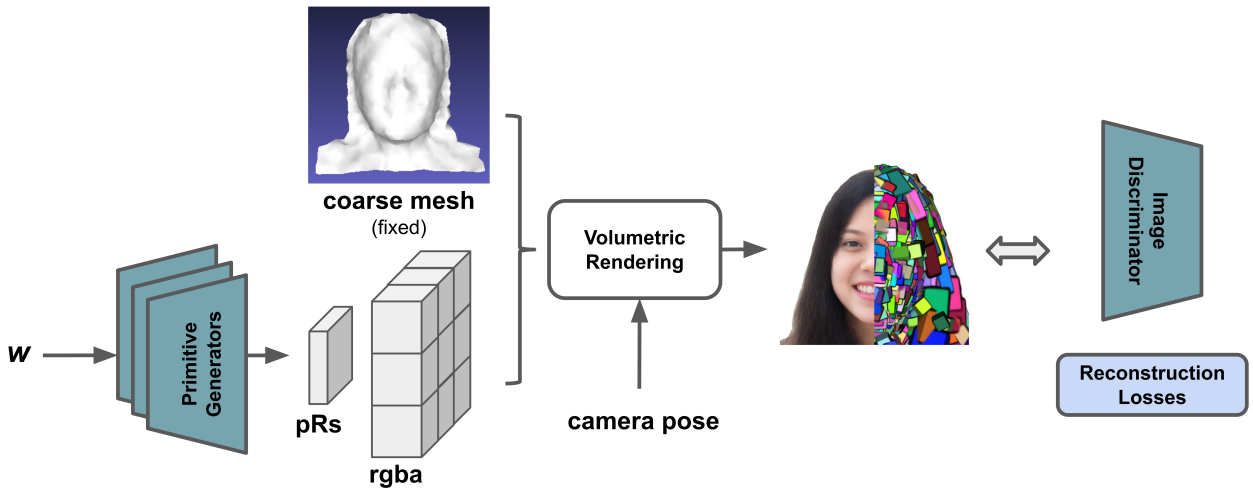


Figure 2. **Method overview:** We propose a generative model that represents each sample using a number of volumetric primitives (voxel grid) that can be rendered very efficiently in real time. Our method consists of 3 2D-CNN-based generators. The geometric generator takes latent vector $w$ and outputs the position, rotation, and scales (pRs) of primitives The second generator takes the latent vector $w$ as input and outputs the density values of all the primitives at once. And the last generator takes latent vector $w$ and viewing direction as input and outputs color content of the primitives. The scene can then be volumetrically rendered efficiently to synthesize samples under any pose.

resolution. An example is the method EG3D by Chan *et al*. [6]. This method extracts 2D StyleGAN2 features [17] and utilizes a 3D tri-planar representation together with a lightweight feature decoder to extract 3D features. A volumetric renderer operating in a low resolution produces an image output, which is then passed through the super-resolution network. The method is trained in an adversarial manner using in-the-wild 2D images. EG3D [6] produces impressive multi-view consistent photorealistic results and is significantly more computationally efficient than previously discussed methods [5, 28, 29, 32]. This super-resolution-based framework has also been utilized in several other works [1, 14, 23, 34], including methods that handle articulated objects [1] and others that use an SDF representation [23]. However, they suffer from one main limitation, the fact that it is not real-3D. While the super-resolution network helps in reducing computation significantly, it doesn't guarantee multi-view consistent results. Hence, the framework fails when rendering from viewpoints that are out of the training camera pose distribution.

## 3. Method

The goal of our method is to build efficient generative volumetric models. We take inspiration from the recent method, MVP [19] to represent the 3D scene. In specific, MVP [19] proposes an efficient 3D scene representation with a mixture of volumetric primitives. But to train MVP, they need registered meshes of an identity performing diverse expressions. Our goal is to build and generalize the training across different categories, where sophisticated 3D morphable models don't exist. Here we show, it's possible to build a generative model without the need for sophisticated morphable models for different categories. One way to train this framework is by learning the whole model end-to-end in an adversarial manner using only loss from an image-based discriminator similar to existing methods [5]. Our initial experiments in that direction proved to be unstable because of multiple moving components. The overview of our method can be found in Fig. 2 To address this issue, we regularize adversarial training with knowledge-distillation technique using pre-trained EG3D [6] models. In the following, we describe the representation of our model in detail in Sec. 3.1 and also provide information about training in Sec. 3.2.

### 3.1. Representation

**Volumetric Primitives:** As mentioned before, we take inspiration from MVP [19] to model our scene efficiently. The efficiency arises because of mainly 2 reasons. One, by having only sparsely distributed volumetric primitives in the scene, they can avoid sampling unnecessary regions that do not contribute to volumetric rendering. Second, the content of the primitives can be generated at once by an efficient CNN network instead of multiple queries required by a coordinate based MLP methods [5]. As we want to learn a generative model for a category, let $\mathbf{w} \in \mathbb{R}^{512}$ be the latent vector that defines a sample. Let $N_{prim}$ be the number of primitives in the scene and each of the primitives is a 3D voxel grid representing only a small region in the 3D space. The position, orientation, and scale of $k^{th}$ primitive in 3D space are defined by $\mathbf{t}_k(\mathbf{w}) \in \mathbb{R}^3$, $\mathbf{R}_k(\mathbf{w}) \in \text{SO}(3)$, and $\mathbf{s}_k(\mathbf{w}) \in \mathbb{R}^3$ respectively. The content of each primitive

is defined by a dense voxel grid $V_k(\mathbf{w}) \in \mathbb{R}^{4\times M_x \times M_y \times M_z}$ that contains color and opacity information. In all our experiments, we use $M_x = M_y = M_z = 32$. The payload of the volumetric primitive contains color and opacity and is defined as a function of sample $\mathbf{w}$.

**Guide Mesh:** MVP [19] utilizes tracked mesh to guide the position and orientation of primitives. Although it is possible to obtain tracked mesh for faces, obtaining the same for other categories like cats and cars becomes challenging. Our surprising finding is that, using our training strategy, we do not need registered meshes across different samples of the category to train the primitives. We make use of a fixed mesh as the guide mesh, $\mathcal{M}$ for all samples, and learn the delta position, orientation, and scales for all the primitives. In order to get the fixed guide mesh $\mathcal{M}$, we extract a mesh from a manually chosen EG3D [6] generated sample and build its UV map using Blender [10]. We observe that the performance of our model does not depend much on the type of sample chosen. We obtain the guide mesh's contribution of primitive position and orientation similar to MVP and let it be $\hat{\mathbf{t}}_k(\mathcal{M})$ and $\hat{\mathbf{R}}_k(\mathcal{M})$. We refer readers to MVP [19] for the exact details. And we also learn delta position, orientation, and scales as a function of latent vector $\mathbf{w}$ to generalize to different samples. We set $\hat{\mathbf{s}}_k$ to a fixed value for all primitives (empirically found ). The final position, orientation, and scale are defined by:

$$\mathbf{t}_k(\mathbf{w}) = \hat{\mathbf{t}}_k(\mathcal{M}) + \delta_{\mathbf{t}_k}(\mathbf{w})$$

$$\mathbf{R}_k(\mathbf{w}) = \hat{\mathbf{R}}_k(\mathcal{M}) \cdot \delta_{\mathbf{R}_k}(\mathbf{w})$$

$$\mathbf{s}_k(\mathbf{w}) = \hat{\mathbf{s}}_k + \delta_{\mathbf{s}_k}(\mathbf{w})$$

**Network Architecture:** Our model mainly consists of 3 generators. The 3 generators are geometric generator $\mathcal{G}_{geo}$ and two payload generators $\mathcal{G}_a, \mathcal{G}_{rgb}$. The geometric generator $\mathcal{G}_{geo} : \mathbb{R}^{512} \rightarrow \mathbb{R}^{9\times N_{prim}}$ is for obtaining the delta position, orientation, and scale of the primitives. And the payload generators $\mathcal{G}_a : \mathbb{R}^{512} \rightarrow \mathbb{R}^{1\times M^3 \times N_{prim}}$ and $\mathcal{G}_{rgb} : \mathbb{R}^{512+3} \rightarrow \mathbb{R}^{3\times M^3 \times N_{prim}}$ are used to obtain the opacity and color values.

**Efficient Differentiable Rendering** Once we have the primitives and their attributes, we make use of efficient differentiable rendering to render our scene. For a given ray $r_{\mathbf{p}}(t) = \mathbf{o_p} + t\mathbf{d_p}$ with starting position $\mathbf{o_p}$ and ray direction $\mathbf{d_p}$,

$$\mathcal{I}_p = \int_{t_{\text{min}}}^{t_{\text{max}}} \mathbf{V}_{\text{col}}(r_{\mathbf{p}}(t)) \cdot \frac{dT(t)}{dt} \cdot dt \ .$$

$$T(t) = \min\Big(\int_{t_{\text{min}}}^{t} \mathbf{V}_{\alpha}(r_{\mathbf{p}}(t)) \cdot dt, 1\Big) \ .$$

Here, $\mathbf{V}_{\text{col}}$ and $\mathbf{V}_{\alpha}$ are the color and opacity values at a given point on the primitive.

### 3.2. Training

One way to train this framework is by learning the whole model end-to-end in an adversarial manner using only loss from an image-based discriminator similar to existing methods [5]. Our initial experiments in that direction proved to be unstable because of multiple moving components. To stabilize training, we propose to regularize adversarial training with knowledge-distillation technique using pretrained EG3D [6] models. The generators are trained with a combination of adversarial and supervised training. As EG3D can synthesize fairly consistent multi-view data for in-distribution camera views, we make use of it as the supervision. Since the latent space of EG3D is known to be well structured, we mimic its properties by using their latent space $\mathbf{w}$ as $\mathbf{w}$ in our model. Using EG3D intermediate latent space is critical in building a high-quality model as we will show later. Given multi-view renderings $\mathbf{X}$ and their corresponding latent space $\mathbf{w}$, we employ reconstruction loss between the prediction and $\mathbf{X}$. This help in structure the learning process. As we do not want to be limited by the knowledge of EG3D only, we also randomnly sample $\mathbf{w}$ for certain percentage of iterations in the training stage and employ discriminator loss.

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{disc}} + \lambda_{perc}\mathcal{L}_{\text{perc}} \tag{1}$$

where $\mathcal{L}_{\text{rec}}$ is an $L_1$ reconstruction loss between rendered images from our model and $\mathbf{X}$. The discriminator loss is defined as follows,

$$\mathcal{L}_{\text{disc}} = f(D(\mathcal{I}(\mathbf{w}))) + f(-D(\mathbf{X})) + \lambda_{reg}\|\nabla D(\mathbf{X})\|^2. \tag{2}$$

where $D$ is an image-based discriminator, $f(u) = -\log(1+\exp(-u))$, and $\lambda_{reg}$ is the coefficient for $R_1$ regularization. The discriminator is trained with a combination of FFHQ dataset [17] and images synthesized from EG3D and jointly represented as $\mathbf{X}$ for ease of reading. Please note for iterations, where we randomly sample $\mathbf{w}$, we do not enforce $\mathcal{L}_{\text{rec}}, \mathcal{L}_{\text{perc}}$ losses. Although discriminator loss increases the sharpness of the results, it also introduces grid-like artifacts. We found having an additional perceptual loss helps in mitigating such effects. The perceptual loss $\mathcal{L}_{\text{perc}}$ is an LPIPS-based [33] loss between rendered images and $\mathbf{X}$.

### 3.3. Correspondences

Once trained our model can be used to obtain dense correspondences between any 2 scenes generated by the model using geometric properties of the primitives. Since we represent the scene in volumetric sense, we make use of expected depth for each pixel and its corresponding point in


Figure 3. Here we show samples of our model trained on FFHQ dataset. We also provide renderings of primitives. The color coding of primitives is consistent across samples. The semantically similar parts are occupied by the same colored primitives across different samples, which provides correspondence information.

3D to show correspondence results. In order to obtain correspondence, for a given pixel and the point at its expected depth, we first store the set of points on the ray and its relative position on the overlaped primitives and weightage of primitives contribution to the final color for a given sample. Since we observe that same primitives occupy similar semantic part of the scene across different samples of the model, to obtain correspondence in another sample of the model, we simply do weighted average of new 3D points corresponding to pre-stored primitives.

## 4. Results

**Datasets** We demonstrate the results of our method on three datasets: FFHQ [17], AFHQv2-Cats [9], and Shapenet Cars [7]. FFHQ [17] is a large-scale portrait dataset of human faces under diverse camera poses. AFHQ-Cats [9] is a similar dataset for cat portraits. Shapenet Cars [7] is a synthetically rendered image dataset of cars.

**Baselines** We compare our model with 4 baselines. The first one is EG3D [6] as this is the state-of-the-art method that can synthesize high-resolution multi-view images in real-time. We also train another model (MVPA) with the same architecture as our model, but with a learnable latent code for each sample. This is similar to auto-decoder training followed in LolNeRF [26] but trained with the dataset described above. This experiment shows the importance of leveraging the $\mathbf{w}$ latent space in EG3D to train our models in a stable manner. Another naive way to train MVP is to employ adversarial loss only. We call this model MVPG, which has same model architecture as our method. This baseline shows that scene representation with high degrees of freedom is hard to train using naive adversarial training only. Finally, we qualitatively compare to Authentic Volumetric Avatars [4]. Although it [4] is not a generative model, since they also use MVP [19] representation, we provide a qualitative comparison. Please note editing expressions is not a goal of our method, unlike [4].

**Training Details** In addition to the datasets mentioned above, we also created a synthetic dataset of around 600k multi-view images and their latent spaces by sampling the pre-trained EG3D model on the FFHQ dataset. Similarly, we sample 300k images for the training model on AFHQ cats and Shapenet cars datasets. Similar to MVP [19], we also use opacity fade factor and volume minimization prior. The opacity fade factor encourages primitives to explain the scene content by the movement of primitives rather than to have primitives just modify their payload content. The volume minimization prior makes the maximum usage of voxels to explain the opaque part of the scene rather than also including transparent parts. We refer readers to MVP [19] for more details. We set $N_{prim} = 1024$, $\lambda_{perc} = 20$ and $\lambda_{reg} = 0.0001$ and use mesh $\mathcal{M}$ with 1024 vertices in all our experiments. The discriminator used for the discriminator loss is similar to the one used in $\pi$-GAN [5] and is learned with a learning rate of 1e-5. We train our models at $512 \times 512$ resolution with a batch size of 32 images for around 5 days on 4 Quadro RTX 8000 GPUs. We use Adam optimizer [18] with a learning rate of $0.001$ for the parameters of the generator. Please refer to the supplemental document for more details.

**Qualitative Results** We present the qualitative results of our method in Fig. 3 and Fig. 4 with models trained on FFHQ, AFHQ, and Shapenet car datasets respectively. Our method can synthesize multi-view consistent results under


Figure 4. Here we show samples of our model trained on AFHQ and Cars datasets. We also provide renderings of primitives. The color coding of primitives is consistent across samples. Please note that our method is robust to different car shapes, even though we use a coarse fixed mesh as an initialization.

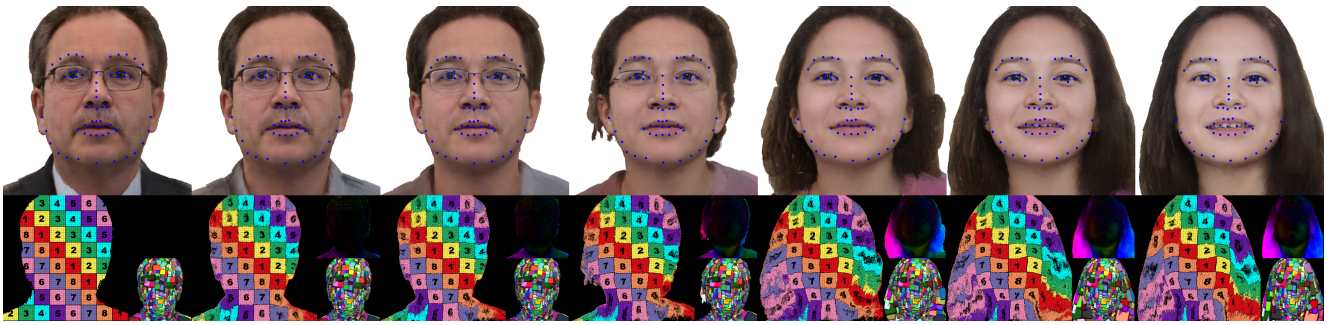


Figure 5. Here we show latent interpolation results of 2 samples. We also visualize the primitives, tracked landmarks, UV map, and magnitude and orientation of tracked correspondences from Sample 1. This shows the quality of dense correspondences learned.

| FFHQ | MVPA | MVPG | EG3D | Ours– | Ours |
|---|---|---|---|---|---|
| FID ↓ | 90.97 | 180.28 | 6.14 | 31.95 | 31.23 |
| Speed ↓ | 22.1 | 22.62 | 46.1 | 22.6 | 22.6 |

Table 1. Quantitative comparisons using the FID score metric (a lower value is better) on the FFHQ dataset and computation time (in milliseconds). Here we see that ours performs better than MVPA, MVPG, while EG3D performs the best in FID score. Our method is considerably faster than EG3D. Our method has other advantages of having pure 3D representation, rendering speed, and can provide correspondences across different samples.

different camera positions. It can also be observed that our method can generalize well to different categories of objects. We also provide visualization of primitives rendering with consistent color coding across various samples. One can observe the correspondence between semantically similar parts of samples. In Fig. 5, we provide interpolation results between 2 randomly sampled faces. We track the

| AFHQ | EG3D | Ours– | Ours |
|---|---|---|---|
| FID ↓ | 3.63 | 19.10 | 14.95 |

Table 2. Quantitative comparisons using the FID score metric (a lower value is better) on the AFHQ dataset. Our method has other advantages of having pure 3D representation, rendering speed, and can provide correspondences across different samples.

landmarks and UV map through interpolation. We also visualize the magnitude and orientation of tracked pixels. We can observe how correspondences follow the semantics of the samples. To evaluate the correspondences between samples in another way, we provide appearance and geometry transfer between 2 samples in Fig. 8. In specific, let $\mathbf{w}_1$ and $\mathbf{w}_2$ be 2 latent vectors of 2 different samples. We take the color and density output corresponding to $\mathbf{w}_1$ and primitives spatial information (i.e., the output of geometry generator) corresponding to $\mathbf{w}_2$ to render the final result. The results of interpolation and correspondences are better appreciated

| | F1 (Out-Out) ↑ | F1 (Out-In) ↑ |
|---|---|---|
| EG3D | 0.2448 | 0.1832 |
| Ours | 0.3579 | 0.2721 |

Table 3. Quantitative comparisons using the F1 score of detected Facial Action Units (a higher value is better). Out-Out denotes the metric computed using renderings of the same sample under 2 views, which are both outside training-camera-pose-distribution. Out-In denotes the metric computed using renderings of the same sample under 2 views, with 1 view in training-camera-pose-distribution and the other being outside. Here we can see that our method clearly outperforms EG3D in both metrics.

in the video. Please refer to the supplemental video.

Next, we provide a qualitative comparison of our method with 4 baselines. In Fig. 6, we compare our method with EG3D [6], MVPA and MVPG. MPVA has the same architecture and supervision as our method and the only difference is that we use the output of the mapping network of EG3D [6] as our latent space and MVPA learns the latent vectors while training similar to LoLNeRF [26]. One can observe the random samples of MVPA clearly suffer from severe artifacts. MVPG trained only with adversarial loss converges to bad optima because of lot of freedom in moving primitives, resulting in non-realistic looking samples.

We also provide a qualitative comparison with EG3D [6] in Fig. 6 7. The samples from EG3D have slightly better perceptual quality than that of our method in capturing high-frequency details. This quality comes because of their 2D super-resolution module, which is known to capture high-frequency details better. In contrast, our representation is purely 3D without any 2D components. This ensures multi-view consistency even for extreme out-of-distribution camera views. In Fig. 7, we provide renderings of both ours and EG3D in training camera distribution and a sample of rendering with a camera placed far from training distribution. As can be observed, EG3D suffers from degraded quality because of inconsistency in the 2D super-resolution module. In contrast, our method retains the quality in out-of-training camera distribution.

In Fig. 9, we provide the monocular fitting results to a given input image. Given an input image, we use our trained model to fit to the input image and show reconstruction and novel view renderings. We obtain the camera pose for the input image using off-the-shelf face pose estimator [11]. Then we optimize the latent vector $w$ for about 1200 iterations using reconstruction loss and simultaneously optimize the generator parameter and latent vector for another 800 iterations to get the better fitting. In the same Fig. 9, we provide a comparison with Authentic Volumetric Avatars(AVA) [4]. While AVA is not a generative model, it can generate a volumetric avatar using a phone


Figure 6. Here, we compare with 3 baselines. We provide 2 novel view renderings of a random sample for all methods. MVPA and MVPG have very poor random samples in its latent space. EG3D can synthesize high-quality samples, but with the help of a 2D-based super-resolution module. In contrast, our method is synthesized with pure 3D representation and is faster than EG3D.

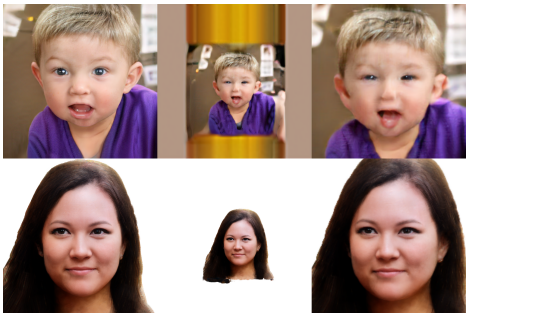


Figure 7. Comparison in out-of-training-distribution of camera pose. The 1st column shows samples for our and EG3D in training distribution and the 2nd column shows rendering with the camera placed a bit far and the 3rd column shows the zoomed-in version of the 2nd column. Note EG3D quality is not 3D-consistent.

scan, we thus qualitatively compare with this method as well. Please note AVA is not a main comparison, as this method targets expression retargeting, while we don't. [4] heavily relies on the tracked mesh to place the primitives and suffers in the case of long hair, which results in truncated results. In contrast, our method is robust to long hair, illumination, and spectacles.

**Quantitative Results** *Correspondences:* We quantitatively evaluate learned correspondences using landmark error. We randomly sample 2 sets of 1000 samples from our model. Then we obtain 2D landmarks [3] for the first set of images and track its correspondence in the second set. We then measure average landmark deviation in the second set using estimated landmarks from an off-the-shelf detector [3]. We found this error to be 7.52 pixels.

We provide commonly used FID scores to evaluate gen-

erative models in Tabs. 1, 2. We sample 20k images (FFHQ) and 10k images(AFHQ) for all the methods at $512\times512$ resolution. We can observe that EG3D [6] performs better than our model. Although perceptually samples from our model look good, we believe it suffers a higher score because our model has discontinuous representation in 3D space. This sometimes leads to piecewise linear regular patterns.

But our model is robust to out-of-training-distribution camera renderings. To evaluate this scenario, we sample 1024 identities for both our model and EG3D and render them under 2 camera distributions. One distribution is similar to training data, as we can easily obtain that by using off-the-shelf face model estimators [11]. And the other is out of this distribution, by placing the camera a bit far from the face. To evaluate the multi-view consistency of these renderings, we obtain the active Facial Active Units using state-of-the-art detectors [20] and compute the mean F1 score of all pairs of renderings for both methods. Our method clearly outperforms EG3D in this evaluation, proving the advantage of having pure 3D representation over a hybrid generator that relies on 2D super-resolution blocks. In spite of having pure 3D representation, our method can render images faster than EG3D. We report the average time taken to render $512 \times 512$ images (Tab. 1). Please note that the evaluation was done on an NVIDIA A40 GPU. The computation time for the EG3D model is calculated using their version which volumetrically renders at $64\times64$ and upsamples to $512 \times 512$. The version with $128 \times 128$ volumetric rendering is twice as slow as the one with $64 \times 64$.

*Discussion:* Although our FID scores are comparatively higher, it's essential to recognize that our method offers unique advantages that comparable methods cannot deliver. We emphasize that constructing a 3D-aware generative model is an important problem with implications for various tasks. Developing a model with a pure volumetric representation that is highly efficient is paramount, and currently, no method meets these criteria. Our baselines MVPG and MVPA show the complexity of building a 3D-aware generator with an efficient scene representation using a naive training paradigm. Furthermore, our approach stands out by providing 3D dense correspondences, a capability unmatched by any other method at this resolution. We hope that our innovative approach to training the model serves as inspiration for future endeavors in constructing efficient pure volumetric 3D-aware GANs. In light of these advantages, we believe that a subpar FID score alone does not define the extent of our contribution, as other crucial aspects enhance the value we bring to the community.

*Ablative:* In Tabs. 1, 2, we provide FID of our model trained without adversarial training (Ours–). As can be observed, adversarial training (Ours) helps in improving the quality of our model.


Figure 8. Here we provide appearance and geometry transfer between 2 samples. S1 and S2 are 2 random samples. We take the color and density output (denoted A) corresponding to a sample and primitives' spatial information (i.e., the output of the geometry generator) corresponding to the other to render the final result.

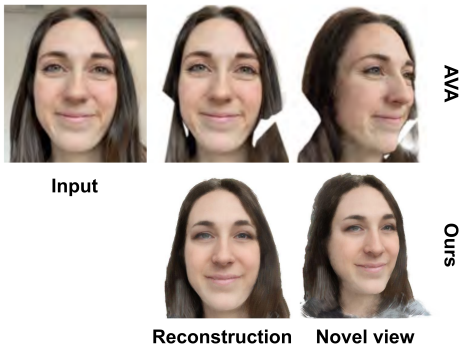


Figure 9. Here we show a comparison with Authentic Volumetric Avatar (AVA) [4]. The first column shows the input image and the second column shows the reconstruction using AVA and ours. The third column shows the novel view rendering of both methods. As can be seen, AVA suffers from truncated hair reconstruction when reconstructing long hair, while ours faithfully reconstructs hair.

## 5. Conclusion

We presented the first 3D-aware generative model with pure 3D volumetric scene representation that can be rendered at more than 43 FPS at $512 \times 512$ resolution. Moreover, our method provides dense correspondence between samples by tracking the primitive position, orientation, and scale. Although our FID score is slightly worse than that of methods that rely on 2D super-resolution modules, our method has the advantage of robustness to camera views because of pure 3D representation, and computation speed and can provide correspondences. We believe our method takes a solid step further in building high-quality, pure 3D, efficient 3D-aware generative models that can provide dense correspondences and can inspire future work in this direction.

## References


[1] Alexander W. Bergman, Petr Kellnhofer, Wang Yifan, Eric R. Chan, David B. Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. In *NeurIPS*, 2022.

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, page 187–194, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[3] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[4] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, Yaser Sheikh, and Jason Saragih. Authentic volumetric avatars from a phone scan. *ACM Trans. Graph.*, 41(4), 2022.

[5] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proc. CVPR*, 2021.

[6] 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.

[7] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[8] Xingyu Chen, Yu Deng, and Baoyuan Wang. Mimic3d: Thriving 3d-aware gans via 3d-to-2d imitation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[10] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[11] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2019.

[12] Yu Deng, Jiaolong Yang, Jianfeng Xiang, and Xin Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[13] 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 *Advances In Neural Information Processing Systems*, 2022.

[14] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *International Conference on Learning Representations*, 2022.

[15] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019.

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

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021.

[20] Cheng Luo, Siyang Song, Weicheng Xie, Linlin Shen, and Hatice Gunes. Learning multi-dimensional edge feature-based au relation graph for facial action unit recognition. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1239–1246, 2022.

[21] 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.

[22] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13503–13513, 2022.

[24] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, page 296–301, USA, 2009. IEEE Computer Society.

[26] Daniel Rebain, Mark J. Matthews, Kwang Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *Computer Vision Pattern Recognition (CVPR)*, 2022.

[27] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[28] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[29] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. EpiGRAF: Rethinking training of 3d GANs. In *Advances in Neural Information Processing Systems*, 2022.

[30] Ayush Tewari, Mallikarjun B R, Xingang Pan, Ohad Fried, Maneesh Agrawala, and Christian Theobalt. Disentangled3d: Learning a 3d generative model with disentangled geometry and appearance from monocular images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022.

[31] Jianfeng Xiang, Jiaolong Yang, Yu Deng, and Xin Tong. Gram-hd: 3d-consistent image generation at high resolution with generative radiance manifolds, 2022.

[32] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. *arXiv preprint arXiv:2112.10759*, 2021.

[33] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[34] Xuanmeng Zhang, Zhedong Zheng, Daiheng Gao, Bang Zhang, Pan Pan, and Yi Yang. Multi-view consistent generative adversarial networks for 3d-aware image synthesis. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18429–18438, 2022.