GaussiGAN: Controllable Image Synthesis with 3D Gaussians from Unposed Silhouettes

Youssef A. Mejjati[1]
Isa Milefchik[2]
Aaron Gokaslan[3]
Oliver Wang[4]
Kwang In Kim[5]
James Tompkin[2]

[1] University of Bath
[2] Brown University
[3] Cornell University
[4] Adobe
[5] UNIST

Abstract

We present an algorithm that learns a coarse 3D representation of objects from unposed multi-view 2D mask supervision, then uses it to generate detailed mask and image texture. In contrast to existing voxel-based methods for unposed object reconstruction, this approach learns to represent the generated shape and pose with a set of self-supervised canonical 3D anisotropic Gaussians via a perspective camera and a set of per-image transforms. This allows robust estimates of a 3D space for the camera and object, while baselines sometimes struggle to reconstruct coherent 3D spaces in this setting. We show results on synthetic datasets with realistic lighting, and demonstrate object insertion with interactive posing. With our work, we help move towards structured representations that handle more real-world variation in learning-based object reconstruction. https://visual.cs.brown.edu/gaussigan

1 Introduction

Inferring poseable 3D object representations from image data for tasks like controllable generation is complex: objects are visible under unknown perspective 3D cameras, have different forms and shapes, have moving parts in different poses, and vary appearance e.g, due to lighting. Oftentimes, we have little supervision for learning parametrized models for these properties. Yet, we would like a flexible structure that does not presume an underlying template and can be shared between images. This task is similar to discovering an 'artist's mannequin' for an object.

We consider a setting with only mask supervision. This problem is related to shape from silhouette, which assumes known camera poses and rigid objects and recovers shape via intermediate voxels. However, in our setting, we do not know the camera poses and the articulated object in the dataset has unknown pose per image. Prior approaches to related problems rely on voxel prediction [7], which can conflate 3D spaces across images in the unposed camera setting (Fig. 1), and deep voxel-based representations [32] that provide too much freedom to recover coherent 3D spaces (Fig. 1, right).

Instead of voxels, we aim to estimate object 'parts' in a robust way; this lets us abstract the structure from the detail in this challenging inference case. For this, we introduce a mixture of anisotropic 3D Gaussians as a coarse implicit geometry proxy. These are low dimensional to infer, have an analytically-differentiable projection model under perspective cameras, are composable for parts, can represent position, scale, and rotation to model part pose transforms, and are simple to self-supervise. Even then, inferring a coherent 3D space across images is tricky. For this, we employ a canonical 3D

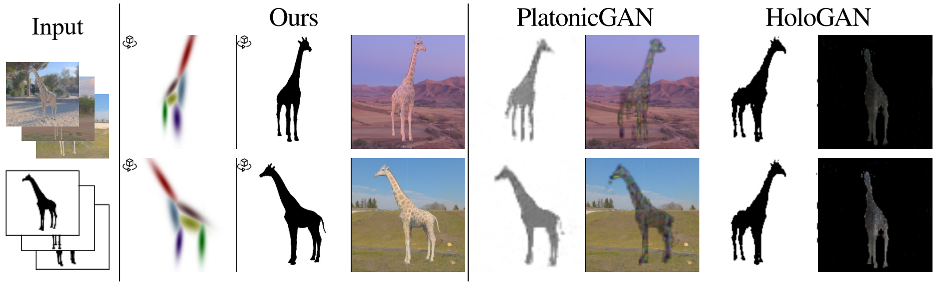


Figure 1: From object masks with unknown camera and object pose, we infer a set of 3D Gaussians to represent coarse shape and pose. This allows 'rigging' detailed 2D mask and texture generation, disentangling of texture variation (here illumination), and interactive camera and pose control. PlatonicGAN [7] generates inconsistent shapes from silhouettes, and texturing is a challenge. HoloGAN [32] on silhouettes has an inconsistent 3D space and, without explicit shape, rotation and lighting are entangled on RGB foregrounds. *Please view in Adobe Acrobat to see animations.*

Gaussian set plus per-image camera and per-Gaussian transformation parameters that describe camera and object pose for each image. Through training via 2D silhouette reconstruction, our representation and losses associate object parts with Gaussians, despite not having any part-level supervision.

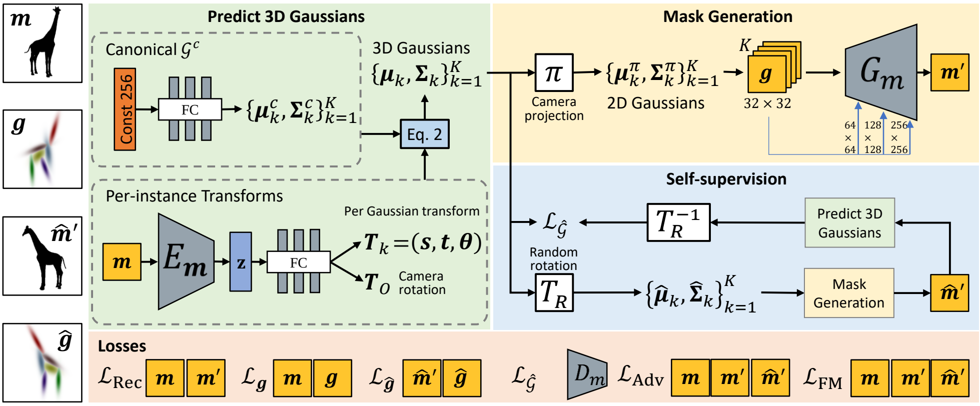
For evaluation, we control input variation using synthetic data with varying camera pose, object pose, and illumination, from which to show recovery of this low-dimensional structure. Then, using the learned Gaussians within 2D RGB generation, we disentangle pose, view-dependent texture, and shading variation caused by lighting differences. This lets us insert objects at arbitrary viewing angles into backgrounds with matched appearance, and interactively adjust object pose by directly manipulating the Gaussians of our 'artist's mannequin'. Looking forward, our work implies a structure to robustly handle pose and shape to better cope with the increased variation in 'in the wild' datasets.

2 Related work

Image and object generation and insertion. GANs have improved learning-based whole image generation [6, 56, 58], including disentangling latent features [14, 22, 38]. Research has also investigated how to learn to generate and add 2D objects to a given background [52], including 2D object shape generation [20] also via bounding boxes [54], completing bounding boxes with texture [8], learning to warp 2D foregrounds [25], insert 2D objects [37], or animate in 2D [45]. We learn an explicit 3D representation that allows controllable image generation.

Unsupervised keypoint and part detection. Gaussians are related to keypoints and parts. Learning these is possible with supervision [26, 31, 39] and without. Here, Thewlis *et al.* [49, 50] use equivariance under 2D image transformations like warping to predict object keypoints; however, this requires the transformations to be known. To address this, Jakab *et al.* [9] learn keypoints in a self supervised way by reconstructing an object's appearance and geometry from different viewpoints. To make these intuitive, Jakab *et al.* later use a skeleton prior (*e.g.*, face, eyes, nose) to guide a discriminator [10]. This has been extended to video prediction with realistic motion [16]. Some methods use Gaussians within their pipelines. Lorenz et al. [27] predict unconstrained 2D activation maps per part for unsupervised part discovery, then estimate 2D Gaussian parameters from these to mark keypoints. Instead, we directly learn a set of 3D Gaussians to describe the shape and pose of an object.

3D object representations. Learned representations exist for taking 3D input data like point clouds [1], volumes [44], meshes [2, 12, 53], or textured meshes [3], for generating 3D output data. These include techniques to fit sets of Gaussians to 3D shapes using 3D supervision [5], and by



m
g
$\hat{m}'$
$\hat{g}$
Predict 3D Gaussians
Canonical $\mathcal{G}^c$
Const 256
FC
$\{\mu_k^c, \Sigma_k^c\}_{k=1}^K$
3D Gaussians
$\{\mu_k, \Sigma_k\}_{k=1}^K$
Eq. 2
Per-instance Transforms
Per Gaussian transform
m
E_m
z
FC
$T_k = (s, t, \theta)$
T_O Camera rotation
Mask Generation
π
Camera projection
$\{\mu_k^\pi, \Sigma_k^\pi\}_{k=1}^K$
2D Gaussians
K
g
32 × 32
G_m
m'
64 × 64
128 × 128
256 × 256
Self-supervision
$\mathcal{L}_{\hat{\mathcal{G}}}$
Random rotation
T_R^{-1}
Predict 3D Gaussians
T_R
$\{\hat{\mu}_k, \hat{\Sigma}_k\}_{k=1}^K$
Mask Generation
$\hat{m}'$
Losses
$\mathcal{L}_{\text{Rec}}$ m m'
$\mathcal{L}_g$ m g
$\mathcal{L}_{\hat{g}}$ $\hat{m}'$ $\hat{g}$
$\mathcal{L}_{\hat{\mathcal{G}}}$
D_m
$\mathcal{L}_{\text{Adv}}$ m m' $\hat{m}'$
$\mathcal{L}_{\text{FM}}$ m m' $\hat{m}'$

Figure 2: Learning a K-part 3D Gaussian representation with only mask supervision $\boldsymbol{m}$. *Green:* For each instance, we predict 3D anisotropic Gaussians by combining a canonical representation with scale, rotation, and translation transforms. *Yellow:* We project these to 2D Gaussians in an analytically-differentiable way, sampled into K maps. $\boldsymbol{g}$ conditions network $G_{\boldsymbol{m}}$ to generate a detailed mask $\boldsymbol{m}'$ as a reconstruction of $\boldsymbol{m}$. *Blue:* To learn a meaningful and smooth 3D space, we self supervise by forcing a random rotation of our estimated 3D Gaussians to also produce a plausible mask $\hat{\boldsymbol{m}}'$ *and* for its 3D Gaussian prediction to be consistent after the inverse rotation. *Orange:* We penalize reconstruction losses on masks and promote realism via adversarial discrimination.

combining 3D supervision with multi-view silhouette losses [55]. Some works use pre-defined detailed canonical 3D meshes for 2D images [59], e.g., to learn surface parameterizations [19, 30]; others attempt to learn shape templates via structured implicit functions [4]. Some works learn representations from 2D input data via 3D representations [15, 35] and flow [40], and often need multi-view camera information given at training time [28, 36]. For instance, DeepVoxels [46] projects RGB values on known camera rays to learn a deep voxel space that reproduces 2D inputs when projected and decoded. Other works require object-specific pose information, such as human skeletal data [18]. Without camera poses, Lei et al. build surface parametrizations for rigid 3D objects [21].

For image generation, few works take only 2D input and *no* camera or object pose information for supervision—this is hard as there is no explicit constraint on the 3D space. Schwarz et al. and Niemeyer and Geiger generate fields for 3D objects [34, 44]. Liao et al. use cube and sphere mesh proxies to represent multiple simple scene objects [23], as is similar to BlockGAN [33], but neither handle posable objects. HoloGAN uses deep voxels within an implicit rotation space [32], and PlatonicGAN uses discrimination on random rotations to learn a generative voxel space [7]. Different geometry and appearance proxies have different trade-offs, e.g., voxels can capture shape detail but are a high dimensional space to predict; our 3D Gaussian proxy is coarse but low dimensional and can capture transformable parts.

3 Learning Gaussian proxies for shape & pose

We wish to reconstruct a set of Gaussian proxies for an object using supervision only via performing the task of mask reconstruction. We train a network to predict a set of 3D anisotropic Gaussians as coarse proxies for the objects' shape and pose, where each Gaussian emerges to loosely represent one part of the object; the mean defines the position and its covariance defines the rotation and scale of the part. Prediction is trained by projecting Gaussians into a perspective camera and transforming them into a detailed mask via a GAN. In this process, we recover a canonical Gaussian representation for the object, from which specific pose and shape transforms are estimated per image.

Input masks and anisotropic 3D Gaussians. We start with a dataset of 256×256 binary segmentation masks $\boldsymbol{m} \in \mathcal{M}$ of an object under varying unknown camera parameters and object poses. We also require a given number K of unnormalized anisotropic 3D Gaussians $\{\mathcal{G}_k\}_{k=1}^{K}$ (Fig. 2). Each Gaussian $\mathcal{G}_k$ has mean vector $\boldsymbol{\mu}_k \in \mathbb{R}^3$ and covariance matrix $\boldsymbol{\Sigma}_k \in \mathbb{R}^{3\times3}$ with its density declared as:

$$\mathcal{G}_k(\mathbf{x}) = \exp\Big(-(\mathbf{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\Big). \tag{1}$$

Camera. We declare a general perspective pinhole camera with intrinsic matrix $\mathbf{K}$, rotation $\mathbf{R}$, and translation $\mathbf{t}$ such that camera matrix $\mathbf{P}$ is represented as $\mathbf{K}[\mathbf{R},\mathbf{t}]$. To project a 3D anisotropic Gaussian into our camera's image plane to produce a 2D anisotropic Gaussian, we use analytically-differentiable projection function π [47]. This is valid for perspective cameras, unlike orthographic [7] or para-perspective [55] projection models that are less applicable to real-world cameras. In our experiments, $\mathbf{K}$ is fixed across images and approximately matches that in the data.

Canonical Gaussians. From a 256-dimensional constant [13], we use a fully connected network $E_{\mathcal{G}^c}$ to predict the canonical 3D Gaussians $\mathcal{G}_k^c$ each of mean and covariance $(\boldsymbol{\mu}^c, \boldsymbol{\Sigma}^c)$ (Fig. 2, green).

Per-image Gaussian transforms. Given an input mask $\boldsymbol{m}$, we extract a latent pose vector $\mathbf{z} \in \mathbb{R}^8$ via a convolutional encoder network $E_{\boldsymbol{m}}$. Then, from $\mathbf{z}$, we use a fully connected network to predict two transformations: 1) A camera transformation $\mathbf{T}_O$ that moves the camera with respect to the canonical model; in our experiments, we consider a yaw rotation $\mathbf{R}_\phi$. 2) K Gaussian local transformations $\mathbf{T}_k$ consisting of scale, translation, and rotation $(\mathbf{s}_k, \mathbf{t}_k, \boldsymbol{\theta}_k)$ with each in $\mathbb{R}^3$ (Fig. 2, green, bottom). Given the canonical $(\boldsymbol{\mu}_k^c, \boldsymbol{\Sigma}_k^c)$, we obtain the per-image Gaussians $\mathcal{G}_k$ with $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ via:

$$\boldsymbol{\mu}_k = \mathbf{R}_\phi(\boldsymbol{\mu}_k^c + \mathbf{t}_k) \quad \text{and} \quad \boldsymbol{\Sigma}_k = (\mathbf{R}_\phi \mathbf{R}_{\theta_k} \mathbf{U}_k \mathbf{s}_k \mathbf{S}_k)(\mathbf{R}_\phi \mathbf{R}_{\theta_k} \mathbf{U}_k \mathbf{s}_k \mathbf{S}_k)^\top, \tag{2}$$

where $\mathbf{R}_{\theta_k}$ is the rotation matrix form of $\boldsymbol{\theta}_k$, and $\mathbf{S}_k$ and $\mathbf{U}_k$ are obtained via eigenvalue decomposition of $\boldsymbol{\Sigma}_k^c$: $\boldsymbol{\Sigma}_k^c = (\mathbf{U}_k\mathbf{S}_k)(\mathbf{U}_k\mathbf{S}_k)^\top$. $\mathbf{S}_k$ is a diagonal matrix. The square of its (j,j)-th entry represents the j-th eigenvalue of $\boldsymbol{\Sigma}_k$. This allows us to control the scale and rotation of each individual Gaussian via the matrices $\mathbf{U}_k$ and $\mathbf{S}_k$. Training covariance Σ to be positive definite can be tricky; we describe and compare our eigendecomposition approach in supplemental material.

Conditional mask synthesis. Even a large number of Gaussian proxies will not reconstruct fine mask detail. As such, we use a conditional mask generator $G_{\boldsymbol{m}}$ to add back the detail using up-sampling transposed convolutions (Fig. 2, yellow). Given the 3D Gaussians for an image, we project them to 2D Gaussians on the image plane of our camera: $\pi(\mathcal{G}_k) = (\boldsymbol{\mu}_k^\pi, \boldsymbol{\Sigma}_k^\pi)$. Then, using the 2D version of Eq. 1, we sample the density of each projected Gaussian on a raster grid to create K Gaussian maps $\{\boldsymbol{g}_k\}_{k=1}^{K}$. These are input to $G_{\boldsymbol{m}}$ to condition the synthesis of predicted mask $\boldsymbol{m}'$, which is the learned reconstruction of $\boldsymbol{m}$. We enforce a stronger effect in $G_{\boldsymbol{m}}$ by using layer-wise conditioning via Gaussian maps at $32^2, 64^2, 128^2$, and 256^2 resolutions.

3.1 Losses

We encourage our network to reconstruct an object via an energy to minimize given by:

$$\mathcal{L}(E_{\mathcal{G}^c}, E_{\boldsymbol{m}}, G_{\boldsymbol{m}}, D_{\boldsymbol{m}}) = \lambda_1\mathcal{L}_{\text{Rec}} + \lambda_2\mathcal{L}_{\boldsymbol{g}} + \lambda_3\mathcal{L}_{\widehat{\mathcal{G}}} + \lambda_4\mathcal{L}_{\widehat{\boldsymbol{g}}} + \lambda_5\mathcal{L}_{\text{Adv}} + \lambda_6\mathcal{L}_{\text{FM}}. \tag{3}$$

Reconstruction loss. We encourage synthesized mask $\boldsymbol{m}'$ to reconstruct input mask $\boldsymbol{m}$ with an L_1 loss: $\mathcal{L}_{\text{Rec}}(\boldsymbol{m}, \boldsymbol{m}') = \|\boldsymbol{m} - \boldsymbol{m}'\|_1$.

Density loss. Even though they cannot represent fine detail in $\boldsymbol{m}$, we still wish for all projected Gaussians to 1) cover regions of the mask without overlap, and 2) cover as much of the mask as possible. We encourage this via: $\mathcal{L}_{\boldsymbol{g}}(\boldsymbol{m}, \boldsymbol{g}) = \left\|\boldsymbol{m} - \Sigma_{k=1}^{K}\boldsymbol{g}_k\right\|_1$. The sum over sampled 2D Gaussians is equivalent to a grayscale version of the colored parts visualization in Figure 3. Here, both inputs are in the range $[0,1]$, and we take $\boldsymbol{g}$ at our mask resolution of 256×256.

Self-supervised transform mask loss. We wish for the 3D space expressed through our recovered object Gaussians and camera transform parameters in $\mathbf{T}_O$ to be consistent across varying camera views even though we only have mask supervision. Thus, we randomly sample a 3D transformation $\mathbf{T}_R$, again mainly as a yaw rotation, and apply it via Eq. 2 to produce rotated 3D Gaussians $\widehat{\mathcal{G}} = (\widehat{\mu}, \widehat{\Sigma})$. As before, these are then projected via π to 2D parameters $(\widehat{\mu}^{\pi}, \widehat{\Sigma}^{\pi})$, then sampled into 2D maps $\widehat{\boldsymbol{g}}$, and finally via $G_{\boldsymbol{m}}$ to generate a mask $\widehat{\boldsymbol{m}}'$ (Fig. 2, blue).

As $\widehat{\boldsymbol{m}}'$ does not correspond to a known input image, we cannot directly enforce $\mathcal{L}_{\text{Rec}}$. Instead, we encourages the projected novel view Gaussians $\widehat{\boldsymbol{g}}$ to be consistent with the synthesized novel view $\widehat{\boldsymbol{m}}'$ via a second density loss: $\mathcal{L}_{\widehat{\boldsymbol{g}}}(\widehat{\boldsymbol{m}}', \widehat{\boldsymbol{g}}) = ||\widehat{\boldsymbol{m}}' - \sum_{k=1}^{K} \widehat{\boldsymbol{g}}_k||_1$. Without this loss, $\boldsymbol{g}$ can describe well the input mask $\boldsymbol{m}$, but the rotated $\widehat{\boldsymbol{g}}$ may not describe well the generated mask $\widehat{\boldsymbol{m}}'$.

Self-supervised transform inverse 3D Gaussian loss. We can also pass $\boldsymbol{m}'$ back through our 3D Gaussian prediction stages (Fig. 2, green) to recover an estimate of the proxies under random transform $\mathbf{T}_R$. Then, we can invert this transform and penalize a loss against our initial estimate of the 3D Gaussians. With slight notation abuse: $\mathcal{L}_{\widehat{\mathcal{G}}}(\mathcal{G}, \widehat{\mathcal{G}}') = ||\mathcal{G} - \mathbf{T}_R^{-1}(\eta(\widehat{\boldsymbol{m}}'))||_1$, where η predicts 3D Gaussians for a mask.

Adversarial loss. Training using only reconstruction losses tends to produce blurry images, so we adopt an adversarial training strategy. $G_{\boldsymbol{m}}$ attempts to generate realistic masks to fool a discriminator $D_{\boldsymbol{m}}$, while $D_{\boldsymbol{m}}$ attempts to classify generated masks separately from real training masks. Within this, we also discriminate against our self-supervised transform masks $\widehat{\boldsymbol{m}}'$: these should also fool $D_{\boldsymbol{m}}$. We use a hinge-GAN loss $\mathcal{L}_{\text{Adv}}$ for better training stability [24, 29, 51]:

$$\begin{aligned}\mathcal{L}_{\text{Adv}}(G_{\boldsymbol{m}}, D_{\boldsymbol{m}}) = \mathbb{E}_{\widehat{\boldsymbol{m}}'}[\min(0, -G_{\boldsymbol{m}}(\widehat{\boldsymbol{m}}') - 1)] + \\ 2\mathbb{E}_{\boldsymbol{m}}[\min(0, G_{\boldsymbol{m}}(\boldsymbol{m}) - 1)] + \mathbb{E}_{\boldsymbol{m}'}[\min(0, -D_{\boldsymbol{m}}(\boldsymbol{m}') - 1)].\end{aligned} \tag{4}$$

To reconstruct the 3D shape within a consistent world space, along with $\boldsymbol{m}$ and $\boldsymbol{m}'$, we find that it is sufficient to give the discriminator a mask $\widehat{\boldsymbol{m}}'$ generated from only one random rotation per image (as similarly found by Henzler et al. [7]), rather than multiple random rotations.

Feature match loss. We improve sharpness by enforcing that real and generated images elicit similar deep feature responses in each layer l of the discriminator $D_M^{(l)}$ [42, 57]:

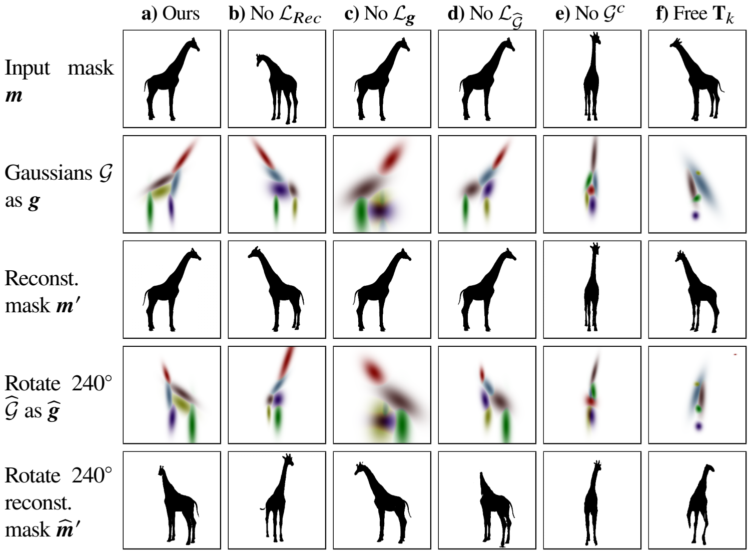
$$\mathcal{L}_{\text{FM}}(D_{\boldsymbol{m}}) = \mathbb{E}_{\boldsymbol{m}, \boldsymbol{m}', \widehat{\boldsymbol{m}}'}\left[\Sigma_{l=1}^{L} \left\| D_{\boldsymbol{m}}^{(l)}(\widehat{\boldsymbol{m}}') - \bar{D}_{\boldsymbol{m}}^{(l)}(\boldsymbol{m}) \right\|_2^2 + \left\| D_{\boldsymbol{m}}^{(l)}(\boldsymbol{m}') - \bar{D}_{\boldsymbol{m}}^{(l)}(\boldsymbol{m}) \right\|_2^2 \right], \tag{5}$$

where $\bar{D}_{\boldsymbol{m}}^{(l)}$ is the moving average of feature activations in layer l, and L is the number of layers.

Constraining pose and shape. We bound μ_k to $[-1, 1]$ and the diagonal values of Σ_k^c to $[0.01, 0.51]$. We prevent any Gaussian from being too small/too large; this encourages learning to use all Gaussians. To remove implausible canonical $\mathcal{G}^c$, we constrain $\mathbf{T}_k = (\mathbf{s}_k, \mathbf{t}_k, \theta_k)$ to produce per-image $\mathcal{G}$ that remain somewhat close to $\mathcal{G}^c$ while still giving freedom to accommodate shape and pose changes (Figure 3). Discriminating masks generated from $\mathcal{G}^c$ is also possible, with self-supervision via random transforms, and may help relax per-image transform constraints.

Importance of losses and components (Figure 3). Removing the reconstruction losses on $\boldsymbol{m}'$ allows a mask to only approximately correspond to the Gaussians as long as it satisfies the discriminator and $\mathcal{L}_{\boldsymbol{g}}$. Removing the density loss on $\boldsymbol{g}$ causes less meaningful Gaussians: they are not forced to represent the generated mask, yet $G_{\boldsymbol{m}}$ can still produces a high detail mask from these Gaussians. Finally, the transform inverse loss 'closes the loop' for the self supervision and helps maintain 3D space consistency and mask quality, especially under cases when penalizing the 2D maps $\widehat{\boldsymbol{g}}'$ alone cannot accurately predict 3D, such as when objects have strong rotation-dependent self occlusion.

Canonical $\mathcal{G}^c$ encourages a meaningful 3D space as each image should be consistent with other images. Directly estimating per-image Gaussians fails for thin front/back views as the


a) Ours
b) No $\mathcal{L}_{Rec}$
c) No $\mathcal{L}_{\boldsymbol{g}}$
d) No $\mathcal{L}_{\widehat{\mathcal{G}}}$
e) No $\mathcal{G}^c$
f) Free $\mathbf{T}_k$
Input mask $\boldsymbol{m}$
Gaussians $\mathcal{G}$ as $\boldsymbol{g}$
Reconst. mask $\boldsymbol{m}'$
Rotate 240° $\widehat{\mathcal{G}}$ as $\widehat{\boldsymbol{g}}$
Rotate 240° reconst. mask $\widehat{\boldsymbol{m}}'$

	a) Ours	b) No $\mathcal{L}_{Rec}$	c) No $\mathcal{L}_{\boldsymbol{g}}$	d) No $\mathcal{L}_{\widehat{\mathcal{G}}}$	e) No $\mathcal{G}^c$	f) Free $\mathbf{T}_k$
IoU ▲	83.96	65.09	82.98	84.75	86.62	73.16
DSSIM ▼	6.22	14.20	6.83	6.16	5.32	10.94

Figure 3: Ablations for *Giraffe*. **(a)** Our full loss model. **(b)** Without a reconstruction loss on $\boldsymbol{m}'$, the Gaussians only approximately correspond to the input mask. **(c)** Without a density loss on $\boldsymbol{g}$, the Gaussians do not well represent the input mask, yet $G_{\boldsymbol{m}}$ still produces the correct mask from these less 'coherent' Gaussians. **(d)** Not 'closing the loop' in the self-supervised loss hurts self occlusion cases or when the 2D Gaussian layouts are insufficient to recover 3D information. **(e)** Not using a canonical representation fails to rotate Gaussians for thin front/back views. **(f)** Not bounding the per-instance transforms allows nonsense canonicals.

Note: We vary input masks per column as certain effects are only visible at particular angles; please see Figure 1 and supplemental video for rotations. Gaussian colors vary across columns.

Table: Over the test set, mean IoU×100 and DSSIM×100 of reconstructed vs. ground truth masks at specific camera angles. Our qualitative results show these metrics do not tell the whole story.

self-supervised rotation must only be consistent with $\mathcal{L}_{\boldsymbol{g}}$ and discrimination $\mathcal{L}_{\text{Adv,FM}}$ (Fig. 3e), instead learning a non-linear space that only rotates between front/back views. Further, estimating μ,Σ values without the const+FC layers [13] led to worse performance.

We compute IoU and DSSIM between ground truth and generated masks (Fig. 3, bottom). Removing $\mathcal{L}_{\widehat{\mathcal{G}}}$ or $\mathcal{G}^c$ improve the metrics slightly, but qualitatively our model is more meaningful: parts can flicker in and out without $\mathcal{L}_{\widehat{\mathcal{G}}}$, and without $\mathcal{G}^c$ the mask does not rotate correctly.

4 Mask texturing

We apply our 3D Gaussian proxies as a conditioning rig to an image generation task [48]. We demonstrate object posing and inserting into an existing image (Fig. 4). For this, we condition a second separately-trained GAN on the mask and on a background image to let us approximately match scene lighting. Given limited space, please see the supplemental document for details of the architecture and combination of reconstruction and adversarial losses used.

Given a database of RGB images $\boldsymbol{i} \in \mathcal{I}$ and corresponding binary masks $\boldsymbol{m} \in \mathcal{M}$, we wish to learn a generative model of texture inside the mask conditioned on the background. First, we

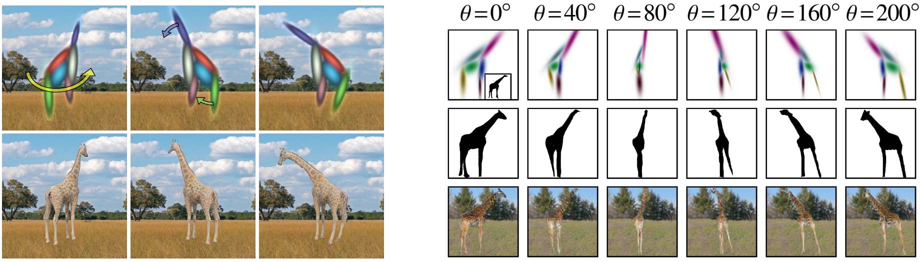


Figure 4: *Left:* Explicitly recovering camera, shape, and pose allows interactive 3D Gaussian manipulation to generate novel instances. *Right:* Our method can produce plausible 3D Gaussians from low-quality and highly-varied masks from MS COCO.

compute the background image $\boldsymbol{i}_b = \boldsymbol{i} \odot (\mathbf{1} - \boldsymbol{m})$ and the foreground image $\boldsymbol{i}_f = \boldsymbol{i} \odot \boldsymbol{m}$, where $\odot$ is the element-wise product. Next, we use an appearance encoder $E_{\boldsymbol{i}}$ to extract a latent representation $\mathbf{z}_{\boldsymbol{i}} \in \mathbb{R}^8$ for the foreground texture: $\mathbf{z}_{\boldsymbol{i}} = E_{\boldsymbol{i}}(\boldsymbol{i}_f)$; this lets us sample foregrounds at test time. We tile $\mathbf{z}_{\boldsymbol{i}}$ and concatenate it with the background image $\boldsymbol{i}_b$, and pass it into a U-Net-like network $G_{\boldsymbol{i}}$ to generate texture. In $G_{\boldsymbol{i}}$'s encoding phase, we layer-wise condition via $\mathbf{z}_{\boldsymbol{i}}$. In $G_{\boldsymbol{i}}$'s decoding phase, we concatenate the Gaussian maps $\boldsymbol{g}$ obtained from $\boldsymbol{m}$ and apply layer-wise conditioning as per $G_{\boldsymbol{m}}$. The final image $\boldsymbol{i}'$ is created from the output of $G_{\boldsymbol{i}}$ with the original background: $\boldsymbol{i}' = G_{\boldsymbol{i}}(\boldsymbol{i}_b, \mathbf{z}_{\boldsymbol{i}}, \boldsymbol{g}) \odot \boldsymbol{m} + \boldsymbol{i}_b$.

5 Experiments

Datasets. We render RGB images and masks using path tracing with ten real-world 360° HDR lighting maps of outdoor natural environments for realistic lighting. For each image, we randomly rotate the camera around the up vector at a fixed distance from the object, to match settings in the literature [32]. We use four datasets without pose variation and of increasing shape complexity (*Maple*, *Airplane*, *Carla*, and *Pegasus*), and four animated datasets with pose variation (*Bee*, *Giraffe*, *Manuel*, *Old Robot*). These include hovering and flapping wings, walking, neck bending, and dancing (each with 110-400 frames; see video). We randomly sample animation frames: poses are not matched across views or in any temporal or rotation order, and we discard object and camera poses during training. We use 1,000/2,000 images for static/animated datasets, with a random 90/10% training/test split.

Training and hyperparameters. We train mask and texture generators for 200 epochs on 2 RTX 2080 TI GPUs. We use the ADAM optimizer with a learning rate of $1e{-}4$, and $\beta = 0.5$. For static datasets, we predict the yaw rotation $\boldsymbol{R}_\phi$ per image on the canonical Gaussians. We set $\lambda_1 = 100$, $\lambda_2 = 100$, $\lambda_3 = 100$, $\lambda_4 = 100$, $\lambda_5 = 1$, $\lambda_6 = 10$. We chose λ_1, λ_2, λ_3. λ_4 over the interval $[0,10,50,100]$. For *Giraffe* with slower animation, $\lambda_3 = 10$ and $\lambda_4 = 50$ led to a slightly better Gaussians.

Baselines. To show the value of model components, we compare to HoloGAN [32], PlatonicGAN [7], and Liao et al. [23], and provide methods with *just masks* and *just RGB foregrounds*. As geometry proxies, HoloGAN and PlatonicGAN use voxels: the HoloGAN bottleneck has 64-dim. deep appearance vectors in 16^3 voxels that are projected to 2D and decoded, while PlatonicGAN directly predicts a 64^3 RGBA voxel space *per image* to handle variation. As a constraint, HoloGAN employs a weaker (but flexible) *latent* reconstruction loss vs. our projection and pixel-wise reconstruction loss. PlatonicGAN also uses a pixel-wise reconstruction loss and, via projection, this can be related to unposed voxel carving when given masks as inputs. Liao et al. estimate multiple 3D primitives (cubes or spheres) as proxies for objects with simple geometry. Please see our supplemental material for comparisons to Liao et al. and to unsupervised 2D part maps of Lorenz et al. [27].

Animated Results—Figure 1. For this qualitative overview, the 'Input' column is the training set. For the 'Our' column, we train our model and then infer Gaussians from two different test-set input masks as prompts. From each set of Gaussians, we generate a mask and then an RGB image given a background. For PlatonicGAN, we train two models: one on just masks (left), and one on masked RGB foregrounds (right). Then, given the same prompt test-set images, we show the output voxel space as occupancy (left) or as RGBA composited onto the same background image as used in the 'Our' column (right). Both 'Ours' and 'PlatonicGAN' use the same camera poses. HoloGAN has no explicit camera pose, only a latent non-metric 'rotation' variable. We train two models on just masks (left) and on RGB foregrounds (right). We see that the latent produces inconsistent rotation between the models. As rotation is latent, matching to a specific camera pose is tricky—for RGB, we see that a global rotation offset is insufficient. For compositing, HoloGAN does not model occupancy, and thresholding the black background often removes the object itself, so we leave it unedited.

Interpretation—mask only. This setting compares reconstruction of a 3D camera and object space. For the static datasets (Fig. 6), HoloGAN's deep voxels reconstruct the input masks, but its latent rotation space can be incoherent with masks at incorrect angles. PlatonicGAN's voxel spaces are naturally 3D and with shape detail, but suffer some incorrect rotations and include spurious or missing geometry. Our approach infers plausible coarse 3D structure that controls 2D mask generation.

For the animated datasets, methods must accommodate image pose variation, and all baselines perform worse. HoloGAN has both part errors (incorrect leg placement) and a low-coherence 3D space (rotation is not smooth; Fig. 1). PlatonicGAN's shapes are incorrectly reconstructed, with missing or misplaced legs and spurious content: even though the method estimates per-image shape, without a canonical model these are incorrectly corresponded in the 3D estimation task, combining parts of objects from across poses. Our method by construction has a canonical 3D $\mathcal{G}^c$ and transformable parts, producing a coherent 3D camera and coarse posable object space.

Interpretation—foreground only. Here, the task is more complex with added texture and lighting variation. Even for static scenes, HoloGAN struggles to generate high-quality appearance, and the resulting 3D spaces for dynamic scenes mix all input variations (Fig. 1) or fail to correctly rotate the image (Fig. 6, *Maple*). PlatonicGAN successfully generates detail (Fig. 1) but again these have object geometry errors and the predicted voxel coloring only approximates the intended output (Fig. 6, *all*). Liao et al. generated images are of broadly good quality, though the pose is entangled with the camera rotation and texture is low resolution and less consistent. As might be expected, our approach demonstrates that using additional mask information to separate shape and appearance allows conditioning higher-fidelity 2D texture generation with disentangled 3D camera, pose, and lighting consistency.

Quantitative results. PlatonicGAN and our method infer explicit 3D spaces. As such, we compute IoU and DSSIM on masks at a known camera angle and compare to test-set ground truth masks: if a method forms a coherent 3D camera and object space, then masks will match (Tab. 1). For methods that infer implicit 3D spaces (without meaningful angles), we compute KID and FID on generated RGB foregrounds (KID/FID are pre-trained for RGB via ImageNet).

On masks	IoU×100 ▲	DSSIM×100 ▼
Ours	**81.97**	**9.35**
PlatonicGAN [7]	77.29	21.29
On RGB	KID×100 ▼	FID×100 ▼
Ours (via masks)	**9.16 ± 0.60**	**117.81**
PlatonicGAN [7]	49.7 ± 0.89	375.26
HoloGAN [32]	32.72 ± 0.87	298.35
Liao et al. [23]	34.2 ± 0.84	292.89

Table 1: Metrics are computed per dataset and then mean averaged (supplemental has full results).

Comparison to Structure from Motion. We might consider how SfM fares given that we recover each of a camera pose, a coarse shape, and a detailed 2D mask. From our giraffe training dataset, we sample 225 images randomly, with both lighting and pose variation. Then, we perform both sparse and dense 3D reconstruction using COLMAP [43]; note that COLMAP is not designed for data like ours. On masks only, COLMAP cannot extract meaningful 2D feature points and fails. Our method works with masks because it models 3D density via Gaussians, and so fares better in this 'shape from silhouette' task. On RGB foregrounds, COLMAP can extract 2D features, but the significant lighting and pose variation cause some cameras to have no solution, only half the space to be reconstructed, and camera poses and 3D points to be noisy. Key object details are missing, such as the giraffe head, because COLMAP cannot cope with object pose variation (Fig. 5).

Real-world data. We show the benefits of a mid-level 3D structure via the managed control of variation available in synthetic data. Many other variations exist in real-world datasets. To show this gap, we demonstrate our method on highly-varied MS COCO data (Fig. 4). Our 3D space and Gaussians are plausible, even though there is significant quality variation in the hand-drawn input masks especially for front/back views; better masks would improve this [17].

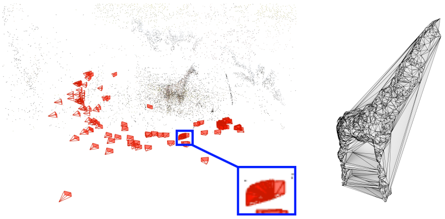


Figure 5: As a static-scene SfM method, COLMAP struggles to accurately reconstruct the object under pose variation from RGB foregrounds: note the noisy half-space camera poses with errors (inset), and missing giraffe head (right).

6 Discussion

3D/2D Bridge and Objectness. Our approach infers a coarse 3D shape and pose as an intermediate structure, rather than deep or explicit detailed 3D appearance. This is simple, robust, gives interactive control over camera and pose for 'rigging' generation, and can apply to many objects as it assumes no explicit connectivity like a skeleton. Integrating hierarchies [11] or kinematic chains could constrain the per-Gaussian transforms, though this is challenging without a template. Further, work in Gaussian transmittance functions [41] could help resolve front/back ambiguity.

Limitations—Shape and Texture. While our method can handle intra-class pose and some appearance variation (lighting), future work should explore additional structure to handle variation of objects within the same class. Our approach is limited to modest shape variation via the Gaussian covariance matrices, and large variations in shape or topology (such as a human in loose clothing) would lead to less useful Gaussians. For texture, the capability to generate variation is limited by 1) the capacity of generator $\mathcal{G}_i$ and latent control z_i, 2) as inputs to $\mathcal{G}_i$, the ability of the silhouette and Gaussian mid-level structure to determine appearance, 3) the variation within the training data. Significant extrapolation away from the training data, such as in backgrounds in supplemental Figure 6, may lead to some less plausible results. For lighting, our method does not perform an intrinsic decomposition into albedo and shading; merely shape and texture disentangling.

Illusions. Finally, humans can be fooled by silhouette illusions like the Spinning Dancer that appears to rotate either way. For certain objects and a sufficiently-small K, our output 3D space can also end up 'rotated' even though it is smooth, i.e., an input mask representing 45° is reconstructed at 315°. This can be fixed by reflecting the space.

Conclusion. As we move toward 'in the wild' settings, we need intermediate structures for varied objects and training losses that can produce meaningful 3D spaces. Given unposed silhouettes, we reconstruct a coarse Gaussian representation of 3D camera and object space along with variations in object pose. We discuss trade-offs with two voxel-based baselines, and display a potential use of our approach to condition 2D texture generators. Future work could explore deep appearance Gaussians and additional ways to self supervise reconstruction under variation to move closer still to handling complex objects in natural images.

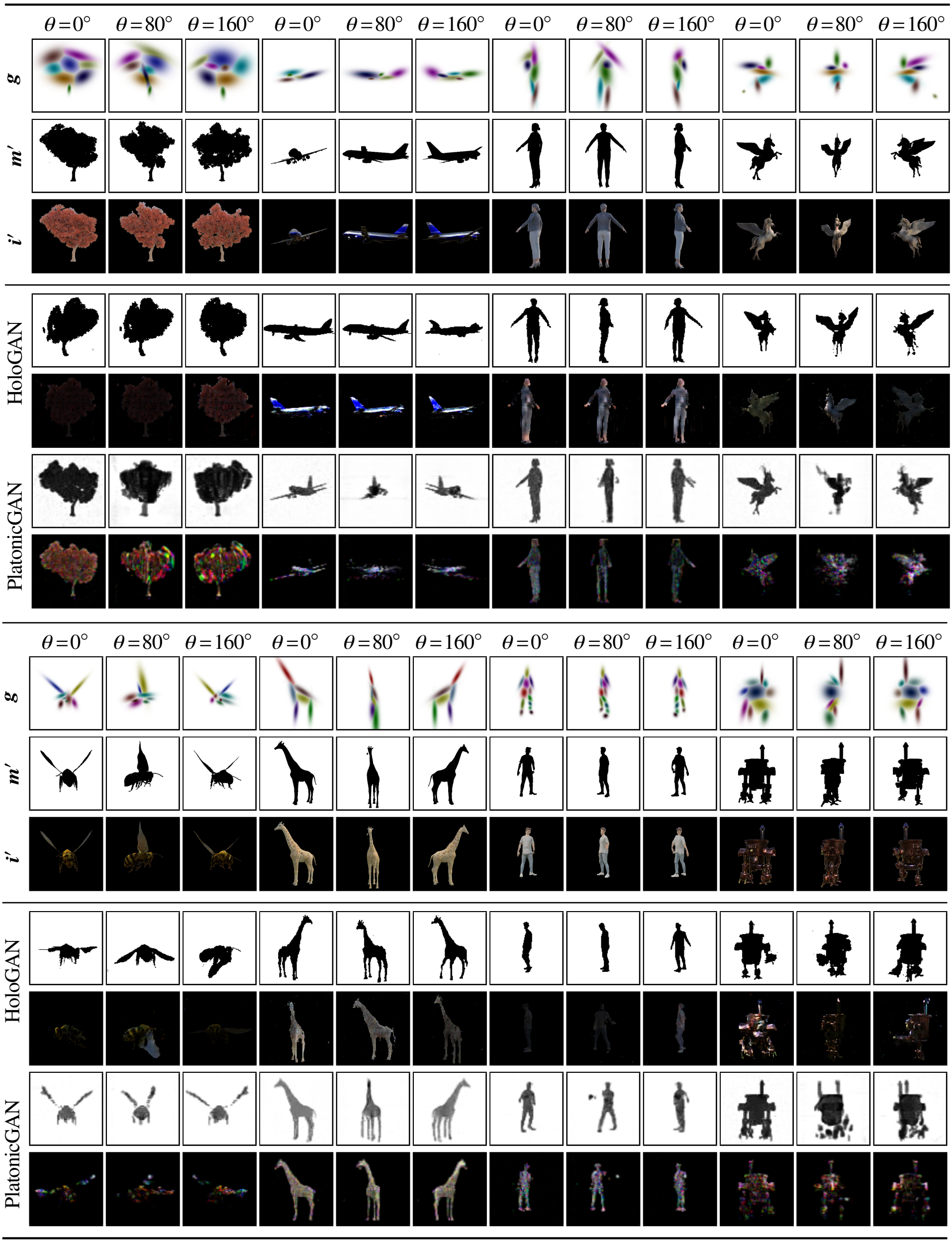


Figure 6: Please zoom in to see detail. *Rows in each block:* Reconstructed Gaussians, masks, and RGB images, across three output angles and with any texture-specific latent variables fixed, with comparisons to HoloGAN [32] and PlatonicGAN [7] run on *just masks* and *just RGB foregrounds*. Note that HoloGAN only infers a latent 'angle', making mapping to an explicit 3D space not possible. *Top block of five rows:* Datasets of objects of fixed pose showing increasing shape complexity: *Maple, Airplane, Carla, Pegasus*. *Bottom block of five rows:* Datasets of animated objects with varying pose showing increasing shape complexity: *Bee, Giraffe, Manuel, Old Robot*.

Acknowledgements

Thank you to Numair Khan for creating the Maya synthetic dataset generator, and for fruitful discussions with Helge Rhodin, Srinath Sridhar, Yuanhao Wang, Michael Snower, Stephen H. Bach, and Daniel Ritchie. Thanks also to Insoo Kim and Neil Sehgal. JT and KIK acknlowedge gifts from Adobe. KIK was supported by the National Research Foundation of Korea (NRF) grant (No. 2021R1A2C2012195). YAM thanks the Marie Sklodowska-Curie grant No 665992, and the Centre for Doctoral Training in Digital Entertainment (CDE), EP/L016540/1. This material is based on research sponsored by Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) under agreement number FA8750-19-2-1006. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) or the U.S. Government.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *ICML*, pages 40–49, 2018.

[2] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *CVPR*, 2020.

[3] Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. TM-NET: Deep generative networks for textured meshes. *arXiv preprint arXiv:2010.06217*, 2020.

[4] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *ICCV*, pages 7154–7164, 2019.

[5] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *CVPR*, pages 4857–4866, 2020.

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

[7] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping Plato's cave: 3D shape from adversarial rendering. In *ICCV*, pages 9984–9993, 2019.

[8] Seunghoon Hong, Xinchen Yan, Thomas Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. In *NeurIPS*, 2018.

[9] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, 2018.

[10] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *CVPR*, 2020.

[11] Farnoosh Javadi Fishani. Hierarchical part-based disentanglement of pose and appearance. Master's thesis, University of British Columbia, 2020. URL `https://open.library.ubc.ca/collections/ubctheses/24/items/1.0395356`.

[12] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, pages 371–386, 2018.

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CVPR*, 2019.

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020.

[15] Hiroharu Kato and Tatsuya Harada. Self-supervised learning of 3d objects from natural images. *arXiv preprint arXiv:1911.08850*, 2019.

[16] Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised keypoint learning for guiding class-conditional video prediction. In *NeurIPS*, 2019.

[17] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, pages 9799–9808, 2020.

[18] Markus Knoche, István Sárándi, and Bastian Leibe. Reposing humans by warping 3D features. In *CVPR Workshop on Towards Human-Centric Image/Video Synthesis*, 2020.

[19] Nilesh Kulkarni, Abhinav Gupta, David F Fouhey, and Shubham Tulsiani. Articulation-aware canonical surface mapping. In *CVPR*, pages 452–461, 2020.

[20] Donghoon Lee, Sifei Liu, Jinwei Gu, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Context-aware synthesis and placement of object instances. In *NIPS*, pages 10393–10403, 2018.

[21] Jiahui Lei, Srinath Sridhar, Paul Guerrero, Minhyuk Sung, Niloy Mitra, and Leonidas J. Guibas. Pix2Surf: Learning parametric 3D surface models of objects from images. In *ECCV*, 2020. URL `https://geometry.stanford.edu/projects/pix2surf`.

[22] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. MixNMatch: Multifactor disentanglement and encoding for conditional image generation. In *CVPR*, 2020.

[23] Yiyi Liao, Katja Schwarz, Lars Mescheder, and Andreas Geiger. Towards unsupervised learning of generative models for 3D controllable image synthesis. In *CVPR*, 2020.

[24] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017.

[25] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. ST-GAN: Spatial transformer generative adversarial networks for image compositing. In *CVPR*, 2018.

[26] Tony Lindeberg. Image matching using generalized scale-space interest points. *Journal of Mathematical Imaging and Vision*, 52(1):3–36, 2015.

[27] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *CVPR*, pages 10955–10964, 2019.

[28] 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.

[29] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[30] Natalia Neverova, Artsiom Sanakoyeu, Patrick Labatut, David Novotny, and Andrea Vedaldi. Discovering relationships between object categories via universal canonical maps. In *CVPR*, 2021.

[31] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[32] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. In *ICCV*, pages 7588–7597, 2019.

[33] Thu Nguyen Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy J. Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *34th Conference on Neural Information Processing Systems*, volume 2020. Neural Information Processing Systems Foundation, Inc., December 2020. NeurIPS 2020: Conference on Neural Information Processing Systems ; Conference date: 06-12-2020 Through 12-12-2020.

[34] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, pages 11453–11464, 2021.

[35] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *CVPR*, pages 4521–4529, 2018.

[36] Kyle Olszewski, Sergey Tulyakov, Oliver Woodford, Hao Li, and Linjie Luo. Transformable bottleneck networks. In *ICCV*, pages 7648–7657, 2019.

[37] Pavel Ostyakov, Roman Suvorov, Elizaveta Logacheva, Oleg Khomenko, and Sergey I. Nikolenko. SEIGAN: Towards compositional image generation by simultaneously learning to segment, enhance, and inpaint. *arXiv preprint arXiv:1811.07630*, 2018.

[38] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

[39] Varun Ramakrishna, Daniel Munoz, Martial Hebert, James Andrew Bagnell, and Yaser Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014.

[40] Jian Ren, Menglei Chai, Oliver J Woodford, Kyle Olszewski, and Sergey Tulyakov. Flow guided transformable bottleneck networks for motion retargeting. In *CVPR*, pages 10795–10805, 2021.

[41] Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. A versatile scene model with differentiable visibility applied to generative pose estimation. In *ICCV*, pages 765–773, 2015.

[42] Tim Saliman, Ian Goodfellow, Wojciech Zaremba, and Vicki Cheung. Improved techniques for training GANs. In *NeurIPS*, 2016.

[43] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[44] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In *NeurIPS*, 2020.

[45] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *CVPR*, June 2019.

[46] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3D feature embeddings. In *CVPR*, pages 2437–2446, 2019.

[47] Srinath Sridhar, Helge Rhodin, Hans-Peter Seidel, Antti Oulasvirta, and Christian Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *3DV*, pages 319–326, 2014.

[48] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging StyleGAN for 3D control over portrait images. In *CVPR*, pages 6142–6151, 2020.

[49] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks by factorized spatial embeddings. In *ICCV*, 2017.

[50] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NIPS*, 2017.

[51] Dustin Tran, Rajesh Ranganath, and David M. Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 7, 2017.

[52] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *CVPR*, 2017.

[53] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, 2017.

[54] Mehmet Ozgur Turkoglu, William Thong, Luuk Spreeuwers, and Berkay Kicanaoglu. A layer-based sequential framework for scene generation with GANs. In *AAAI*, pages 8901–8908, 2019.

[55] Kohei Yamashita, Shohei Nobuhara, and Ko Nishino. 3D-GMNet: Single-view 3D shape recovery as a gaussian mixture. *BMVC*, 2020.

[56] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. LR-GAN: Layered recursive generative adversarial networks for image generation. In *ICLR*, 2017.

[57] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

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

[59] Silvia Zuffi, Angjoo Kanazawa, Tanya Berger-Wolf, and Michael J. Black. Three-D safari: Learning to estimate zebra pose, shape, and texture from images “in the wild”. In *ICCV*, pages 5358–5367, 2019. doi: 10.1109/ICCV.2019.00546.