Generating 3D Objects with Limited Data

Eric Ryan Chan
Stanford University
erchan@stanford.edu

Abstract

Generative models have rapidly gained popularity for convincing synthesis of images, videos, 3D objects, and other media. However, while image-synthesizing GANs are mature, 3D-model synthesis is a much less explored field. Current 3D GANs require many thousands of examples to train effectively and are capable of operating off of only simple, low-fidelity 3D datasets like ShapeNet Cars. When trained on very small datasets, such as a small collection of 3D models, the discriminator easily memorizes the real examples and causes training to collapse. Drawing from advances in neural rendering and motivated by recent success with differentiable augmentation, we demonstrate that randomized rendering is a simple yet effective way of augmenting the training dataset. We show that training a generative model from a handful of 3D examples is possible and achieve synthesis of high-quality, complex 3D objects.

1. Introduction

The last few years have seen immense progress in Generative Adversarial Networks (GANs), with state-of-the-art models capable of generating high-resolution, photorealistic images indistinguishable from real photos [15, 17, 18]. However, while exciting, the most famous of these GANs are generally confined to two-dimensions, with no understanding of 3D structure.

Recent work has taken the ideas behind 3D Generative Adversarial Networks into three dimensions. Simply by replacing the 2D convolutional layers with 3D convolutional layers, the most basic of 3D GANs [41][5] have been successful in generating rough voxel grids corresponding to 3D shapes.

At training, these approaches generate an occupancy grid that captures the discretized shape of an object. The discriminator, in turn, sees either the generated occupancy grid or the occupancy grid of a real object. As the direct analogue of training 2D GANs, this method is simple, well proven, and effective for large datasets such as ShapeNet.

However, such an approach has been found to be sub-optimal when training from small datasets. Recent work [43][16] has shown that when attempting to train on relatively small datasets (in the thousands rather than tens of thousands), image quality plummets. Training from very small datasets (in the hundreds) leads to heavily corrupted images at best or complete training collapse at worst.

The intuition behind this instability is that when training from very small datasets, the discriminator easily memorizes the small set of real samples. Once the discriminator has overfit, it ceases to provide useful gradients to the generator. Recent work [15][43] has concurrently identified a method to alleviate discriminator overfitting: data augmentation. With heavy augmentation (e.g. crops, masking, rotations, color-shift, etc.) Karras et al. and Zhao et al. have demonstrated the capability of 2D GANs for training from datasets of as little as a hundred examples.

Inspired by the success of data augmentation in training 2D generative models from hundreds of images, we investigate training a 3D generative model from a handful of 3D objects, a problem not previously investigated in the literature. As you can imagine, the requirement of a large dataset of models constrains the possible applications for 3D GANs. Just think of the possibilities if a 3D artist were able to generate new assets from dozens rather than thousands of examples!

In this work, we demonstrate by leveraging recent advances in neural rendering and 3D-aware Image Synthesis, training a 3D generative model from a handful of 3D objects is possible. The input to our pipeline is a small collection of 3D objects. The output of our pipeline is a trained model which, when sampled, produces 3D objects that match the distribution of the input objects.

Our contributions are the following:

- We motivate and discuss the problems associated with low-shot 3D generation.
- We introduce a dataset of Greco-Roman busts as a benchmark for low-shot 3D generation.
- We propose a novel architecture that leverages a convolutional backbone and locally-conditioned neural-implicit decoder that allows for efficient training.
• We demonstrate that the proposed approach synthesizes convincing 3D models from limited examples.

• We provide qualitative and quantitative results that support our hypothesis that randomized rendering counters discriminator overfitting and improves stability and quality.

2. Related Work

Differentiable Augmentation for 2D Image GANs The performance of traditionally-trained GANs deteriorates rapidly when trained on small datasets. When trained from few examples, the discriminator easily overfits and memorizes the true images. By memorizing rather than generalizing, the discriminator fails to provide informative gradients to the generator. Unchecked, this leads to training instability, divergence, and eventual collapse. Recent work [43][16] has explained this phenomenon and offered a solution: differentiable augmentation. In differentiable augmentation, both generated and real examples are augmented with a differentiable transformation (e.g. noise injection, color shift, flipping, cropping, etc.) before being viewed by the discriminator. In essence, differentiable augmentation gives each example a distribution of appearances, making it more difficult for the discriminator to memorize. Just as dataset augmentation is often used when training supervised-learning models to increase the effective size of the training dataset, improve robustness, and guard against overfitting, augmentation in GAN training has been shown to reduce the discriminator’s susceptibility to overfitting.

Implicit neural representations and rendering. Neural implicit scene representations promise 3D-structure-aware, continuous, memory-efficient representations for shape parts [9, 8], objects [32, 25, 1, 10, 42, 6, 2], or scenes [7, 38, 13, 33, 37]. These representations can be supervised with 3D data, such as point clouds, and optimized as either signed distance functions [32, 25, 1, 10, 38, 13, 33, 36] or occupancy networks [24, 4]. Using neural rendering [40], implicit neural representations can also be trained using multiview 2D images [34, 38, 30, 29, 26, 42, 21, 14, 22]. Temporally aware extensions [28] and multimodal variants that add part-level semantic segmentation [20] have also been proposed.

Recently, local-implicit representations [21][2][39] have emerged as an alternative to fully-implicit representations. Rather than parameterizing the representation as a single fully-connected MLP, local-implicit methods couple a fully-connected ‘decoder’ with a large number of spatial embeddings. As the additional spatial embeddings increase the capacity of the model, such methods can often get away with decreasing the size of the decoder, improving runtime performance. However, local-implicit representations have generally been confined to representing single scenes, with little work examining the characteristics local-implicit representations when generalizing across multiple scenes. This work explores the promise of local implicit representations in the GAN setting, where generalization across a large number of scenes is of paramount importance.

3D GANs Several works explore generative 3D object synthesis. 3D-GAN[41] is a straightforward approach that uses 3D convolutions to generate occupancy grids representing 3D shapes. Extensions include 3D generation conditioned on images and shape generation with texture synthesis [44]. In order to get around the memory-requirements and resolution constraints of voxel-grids, more recent approaches have leveraged neural implicit representations to allow for the generation of continuous shapes[19]. Alternative approaches focus on 2D synthesis, and produce consistent 3D representations as intermediate results. Platonic GAN [11] and HoloGAN[27] learn latent voxel representations to allow for view-consistent rendering. GRAF [35] and pi-GAN [3] learn generative models for implicit radiation fields, achieving better multi-view consistency compared to earlier approaches. Still further works instead rely on inverting pre-trained 2D GANs, which have been shown to have a notion of 3D shape even without explicit 3D representations[31]. However, these ‘3D-Aware’ models have largely been focused on large 2D image datasets, where overfitting is not a problem and generalization naturally arises from many examples.

3. Dataset and Evaluation

While previous 3D GANs required vast datasets of 3D models, such as ShapeNet, we assert that our method achieves high-fidelity results even when trained on a small collection of real data. We train our model on a collection of six photoscans of Greco-Roman busts, selected from SketchFab. The size of the dataset was kept intentionally small in order to amplify effects caused by overfitting. We evaluate the trained model by computing the Frechet Inception Distance[12] of 8000 2D-renderings of generated 3D objects with 8000 2D-renderings of the ground-truth 3D ob-
Figure 2: A visualization of our differentiable rendering procedure. Given a conditioned radiance field, we cast rays from the camera origin \( o \), sample density \( \sigma \) and color \( c \) values along each ray, and calculate pixel color \( C \) using Eq. 1.

4. Approach

4.1. Randomized Neural Rendering as Data Augmentation

Recent work [15][43] has shown that 2D image GANs benefit greatly from differentiable augmentation. By augmenting images with differentiable transformations, we can make it tougher for the discriminator to simply memorize the training dataset, preventing catastrophic training collapse.

While similar transformations, e.g. rotations, noise, etc. could be applied to 3D voxel-grid-based GANs in order to improve training performance, we assert that a more elegant solution exists: randomized neural rendering. Rather than supplying the discriminator with the full shape of the object, as is done with traditional 3D GANs [41][5], we instead give the discriminator a 2D rendering from a random camera pose. Now, in order to memorize a specific object from the training set, the discriminator must memorize every possible rendering from random angles. Just as differentiable augmentation produces a distribution of appearances for each image, making it harder for the discriminator to memorize, randomized neural rendering produces a distribution of appearances for each 3D model.

In practice, we precompute a set number of renderings of true 3D objects using Blender and we use differentiable volumetric rendering[26] to produce 2D renderings from synthesized objects.

4.2. Differentiable Rendering

We render a neural radiance field from arbitrary camera poses using neural volume rendering. For this purpose, we cast rays from the camera origin \( o \) and compute the integrals along each ray through the volume. At every sample, our generator predicts the volume density \( \sigma \) and color \( c \). The pixel color \( C \) for a camera ray \( r(t) = o + td \) with near and far bounds \( t_n \) and \( t_f \) is then calculated using the volume rendering equation [23]:

\[
C(r) = \int_{t_n}^{t_f} T(t) \sigma (r(t)) c (r(t), d) \, dt,
\]

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

Our approach implements a discretized form of this equation using the stratified and hierarchical sampling approach introduced by NeRF [26] (see Fig. 2).

This neural rendering approach, which is also adopted by GRAF [35] and pi-GAN [3], is agnostic to image size and offers full control over camera pose, focal length, aspect ratio, and other parameters.

4.3. Model Architecture

Previous approaches to neural rendering in a GAN framework [35][3] rely on fully-implicit MLP backbones. While powerful, the computational complexity of these large MLP-based backbones scales linearly with the number of samples needed for rendering. Because neural volumetric rendering requires sampling a large number (batch size \( \times \) image height \( \times \) image width \( \times \) ray samples) of samples, neural-rendering GANs have traditionally been slow and memory-intensive to train. In order to reduce complexity, we rely primarily on 3D-convolutional backbone which synthesizes a coarse 3D feature grid. The 3D feature grid locally conditions a very small, inexpensive, two-layer MLP. Because much of the model capacity is contained within the efficient convolutional backbone, the proposed approach requires roughly 1/4 of the memory of previous approaches[35][3] while achieving comparable quality.

5. Experiments and Analysis

In this section, we seek to answer the question of whether randomized neural rendering is an effective form of augmentation for training 3D GANs. We hypothesize that randomized rendering may help alleviate discriminator overfitting by increasing the effective size of the training dataset.
and making it more difficult for the discriminator to memo-
rrize the true images. In order to test this hypothesis, we train
GANs to generate 3D objects from our dataset of Greco-
Roman statues under three levels of randomized rendering.

We evaluate three training settings (4-View, 64-View,
and 256-View), corresponding to the number of distinct
views supplied to the model. The 4-View setting augments
the training dataset with four random renderings of each
3D object while the 256-View setting augments the training
dataset with 256 random renderings of each 3D object. Note
that all runs include the same number (six) of ground-truth
objects. In practice, the number of views can be made arbi-
trarily large, and is bounded only by Blender pre-rendering
time of the ground-truth 3D models.

In this section, we compare these three training settings.
We provide qualitative and quantitative evaluations of the
training settings and investigate whether randomized ren-
dering helps prevent discriminator overfitting and improve
training stability.

5.1. Results

Does randomized rendering reduce discriminator over-
fitting? In order to determine whether augmenting the
dataset with multiple views of each object reduces overfit-
ting, we plot discriminator classification accuracy as a func-
tion of training iterations. In the adversarial framework, the
discriminator is a binary classifier that predicts the realness of each example. It is trained to classify generated examples as fake and real examples as real. In order to measure how well the discriminator generalizes, we plot discriminator training accuracy and discriminator validation accuracy (calculated on held-out real images) as a function of iterations.

Figure 6 shows that all three settings result in significant discriminator overfitting. In all three cases, training performance quickly approaches perfect performance, while validation performance deteriorates to only slightly better than random guessing. This suggests that the discriminator is memorizing the training dataset rather than generalizing.

However, if we compare training and validation performance between our three settings, we see that while randomized rendering doesn’t solve discriminator overfitting, it is a significant improvement over the settings with fewer images. Figure 6 shows that having additional views in the dataset does improve validation accuracy, at a slight cost to training accuracy.

Does randomized rendering improve training stability?
We train each of the three training settings to 30,000 iterations. Figure 5 shows random generated examples at each run’s best checkpoint. 4-View was unable to obtain stability at any point in training and quickly collapsed before it produced successful generations. 64-View succeeded in generating rough shapes but diverged after 16k iterations, leading to training collapse. 256-View maintained training stability for at least 30k iterations without suffering a noticeable collapse. Empirically, we have significantly stabilized training by augmenting with randomized rendering.

Does randomized rendering improve result quality?
Figure 5 gives a qualitative comparison of renderings from the three separate training runs. It is clear that randomized rendering leads to better qualitative results—while 4-View was unable to achieve any stable results and 64-View was only moderately successful in generating coarse statue shapes, 256-View generated examples that are recognizable as statues. Figure 4 shows samples generated with 256-View from multiple camera angles.

We calculate Frechet Inception Distance [12] between renderings of the true objects and renderings of generated objects in order to get a quantitative comparison of visual quality. Table 1 provides quantitative results for the three training settings. The numbers back up the qualitative results; randomized rendering seems to significantly improve generated example quality.

6. Discussion
The ability to train Generative Adversarial Networks from limited data is prerequisite for many interesting applications, where we might have limited training data that fits the task at hand. It is of even greater importance for 3D models, since high-quality 3D models are far more difficult to acquire than 2D images. To our knowledge, even the smallest datasets used to train prior 3D GANs consisted of several thousand unique examples. In this work, we demonstrate that randomized neural rendering is an effective form of data augmentation that allows for stable training even from only a handful of 3D models. We motivate randomized rendering as a method to reduce discriminator overfitting and provide qualitative and quantitative results that support the efficacy of the proposed approach.

References


[34] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned


