Adversarial generation of continuous implicit shape representations

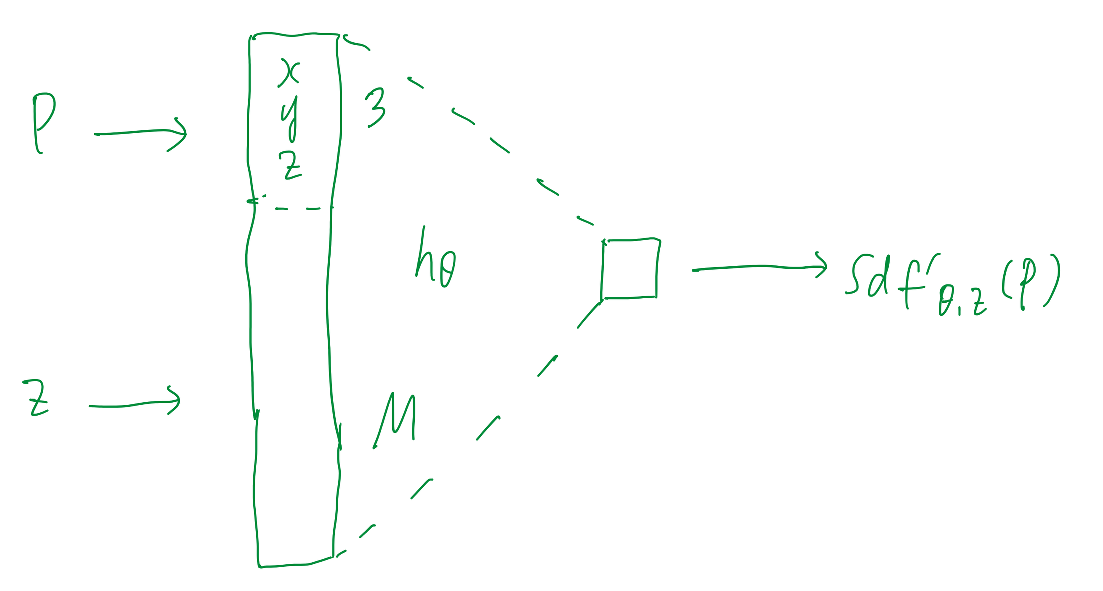
Saturday, December 26, 2020 8:52 AM

In this paper, we propose a GAN that generates 3D shapes, the GAN uses a DeepSDF network as a generator, and either a 3D CNN or a Pointnet as the discriminator.

DeepSDF.
The standard way of representing 3D shapes in deep learning modules uses voxel volumes, they are a generalization of images to 3D space and use voxels instead of pixels. With the 3D CNN approach, concepts from deep learning for images can be applied to 3D shapes.

A voxel volume contains a rasterized representation of the signed distance field of the shape. The signed distance function is a function that maps a point in 3D space to a scalar signed distance value.

The idea behind the DeepSDF network, is to train a neural network to predict the value of the signed distance directly for an arbitrary point in space. Thus, the network learns a continuous representation instead of a rasterized one.



To learn a representation of multiple shapes, the DeepSDF receives a latent code as an additional input.

The decision boundary of the network is the surface of the learned shape. For a given latent code, a mesh can be created using Marching Cubes by evaluating the network for a raster of points. The resolution of that raster can be selected arbitrarily after the network was trained.

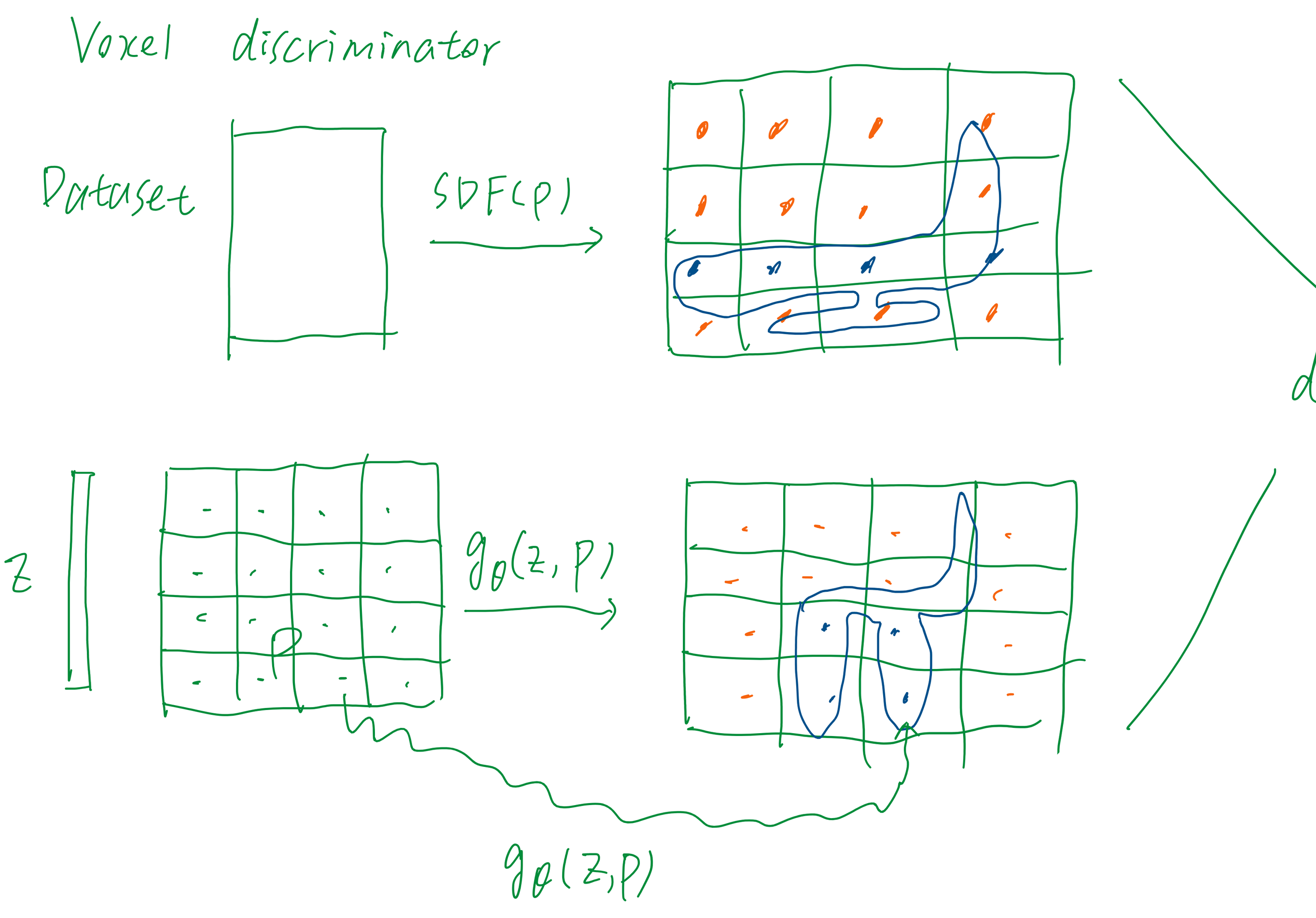
The DeepSDF network is trained on a dataset of 3D points with corresponding SDF values. These points are in part uniformly sampled and in part normally distributed around the shape surface, resulting in a high density of training data near the surface.

The DeepSDF autodecoder works like an autoencoder, but without the encoder. The latent codes are assigned randomly and then optimized during training using SGD.

GAN.
If the GAN training is successful, the GAN reaches an equilibrium, where the generator has learned a mapping from the latent distribution to the underlying distribution of the dataset, and the discriminator assesses the generated and real samples with the same score.

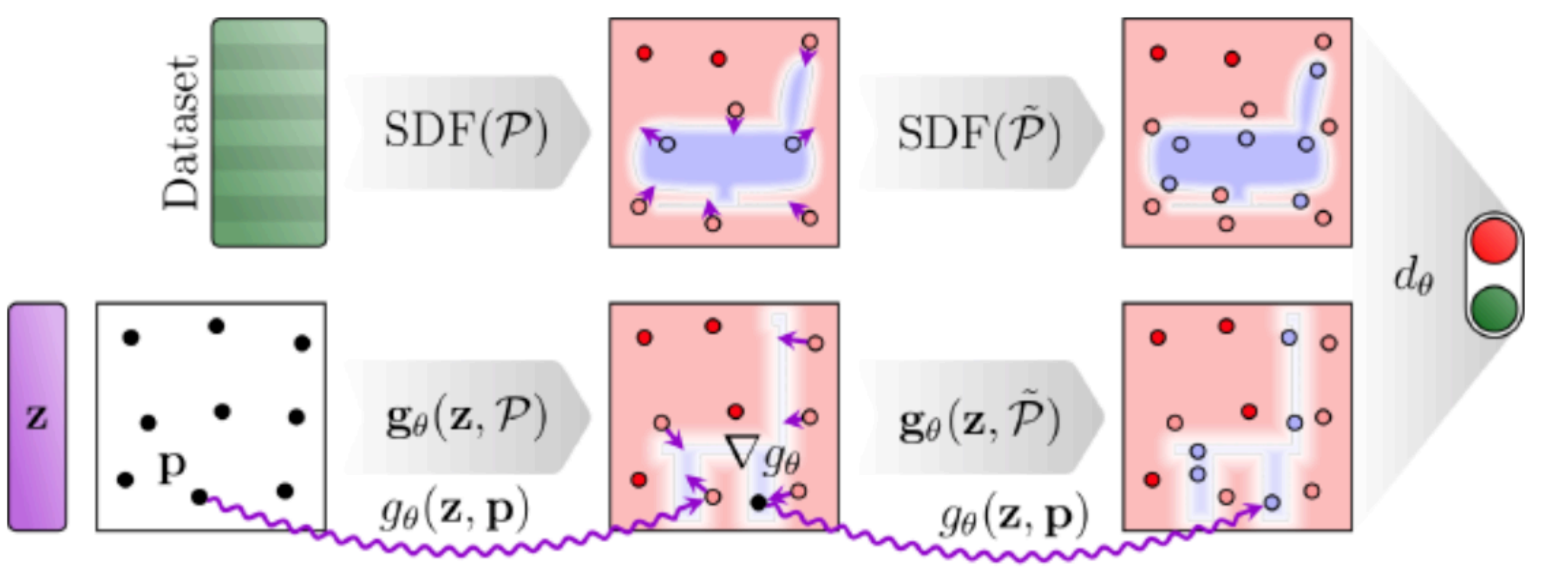
Usually, the discriminator is a mirred version of the generator. In the case of the DeepSDF, this is not feasible, because a single sample of the DeepSDF network provides only the SDF for one point. From one point alone, a discriminator could not access if the sample is realistic. Instead, the discreminator needs multiple points to judge the output value in context.

Voxel discriminator



One solution of the problem of missing context, is to use a 3D CNN as the discriminator. In this case, the generator is evaluated for a batch of raster points, and the resulting SDF values are rearranged into a voxel volume. The training data for the voxel discriminator are voxel volumes, e.g. $8^3, 16^3, 32^3, 64^3$ etc. The generator is able to generalize from the raster points to intermediate points that it was not trained on.

Pointnet discriminator



Since the voxel based discriminator keeps some of the disadvantages of fully voxel based GANs, we propose another approach that doesn't use voxels at all.

The pointnet is a neural network that can operate on point clouds. In our case, we sample a point cloud of uniform points, and use the generator to predict the SDFs for the points. The Pointnet then receives the position of the points, and the signed distance values as a "feature vector" with one element. This way, we avoid the fixed raster points, and use always changing query points. A pointnet typically infers information from the spatial structure of the points, which in our case is random, Regardless, we found that the Pointnet can be used as the descriminator in our case.

To train the GAN with the Pointnet discriminator, we use a ground truth dataset of uniformly sampled points with their corresponding SDFs. We "grow" the point clouds during training simply by increasing the number of points.

When the surface of the generated SDF is reconstructed using Marching Cubes, only values close to sign changes matter. We would like the network to spend more model capacity on values close to the surface, as they influence the result the most. We achieved that by refining the network with additional sample points close to the surface.

The gradient of the SDF gives us the direction towards the shape's surface, and for a neural network, the gradient is easily computed. Using that, we can move randomly sampled points closer to the surface of the generated shape. The non-uniform point cloud can then be evaluated by the descriminator. Since the Pointnet takes the positions of the points into account, it could discern a uniformly sampled point cloud from a surface point cloud. Thus, we add surface points to the ground truth data as well.

We chose an isosurface level of 0.04 to reduce the missing geometry, at the cost of slightly rounded corners. Since we clip the ground truth SDF at −0.1 and 0.1, the isosurfaces of generated SDF outside of that range are not usable.