Learning implicit fields for generative shape modeling[1]

July 1, 2020

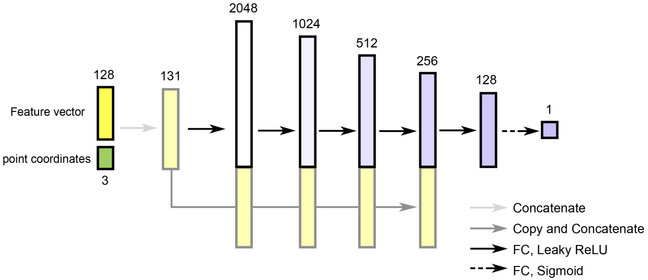
[1] "Learning implicit fields for generative shape modeling" by Chen and Zhang, 2019

Overview

- There are different ways to represent 3D shapes: voxel grids, octrees, point clouds, etc.
- It is quite expensive (in terms of GPU memory) to train a decoder for such kind of outputs
- So authors employed an INR-like decoder to output 3D shapes
- They test the approach on several tasks:
  - Auto-encoding
  - 3D-shape generation
  - 2D-shape generation (generating MNIST digits)
- They obtain strong results, but the model is slower to train and run

Implicit Field Decoder

Authors propose *IM-NET*: an INR-like decoder

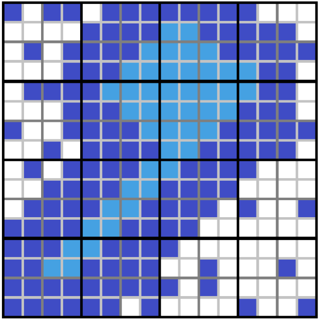


- This decoder:
  - takes a coordinate (x, y, z) and an image embedding
  - outputs 1 or 0, denoting if a point is a part of the shape or not
- They train it with the MSE loss

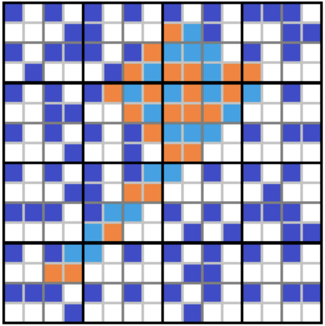
Data preparation

Authors perform the following data preparation steps:

- Downsample training examples to different resolutions: $16^3, 32^3, 64^3, 128^3$.
- Sample points near closer to a surface with higher probability



(a) Main strategy: sample points 2 voxels away from the surface

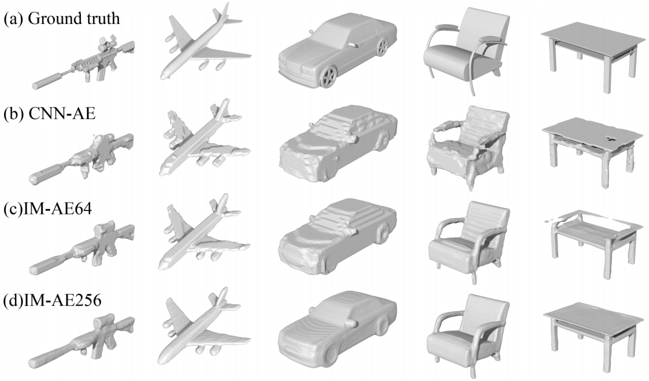


(b) Auxiliary strategy: sample points evenly with a stride

3D-shape auto-encoding

- Use a CNN-encoder (with conv3d layers) as Encoder
- Use a IM-NET as Decoder
- They use a CNN-decoder as a baseline decoder

3D-shape auto-encoding samples



3D-shape auto-encoding results

	Plane	Car	Chair	Rifle	Table
CNN64-MSE	**1.47**	**4.37**	**7.76**	**1.62**	**5.80**
IM64-MSE	2.14	4.99	11.43	1.91	10.67
CNN64-IoU	**86.07**	**90.73**	**74.22**	**78.37**	**84.67**
IM64-IoU	78.77	89.26	65.65	72.88	71.44
CNN64-CD	**3.51**	5.31	**7.34**	**3.48**	**7.45**
IM64-CD	4.22	**5.28**	8.96	3.78	12.05
IM256-CD	4.23	5.44	9.05	3.77	11.54
CNN64-LFD	3,375	1,323	2,555	**3,515**	**1,824**
IM64-LFD	3,371	1,190	2,515	3,714	2,370
IM256-LFD	**3,236**	**1,147**	**2,453**	3,602	2,201

- IM-NET works worse on all the metrics except LFD
- But visually its samples are better
- Authors claim that all the metrics except LFD are bad (and provide some argumentation for this)
- LFD (Light Field Descriptor) is computed by taking several 2D renderings of two shapes from different angles and comparing the results

3D-shape generation

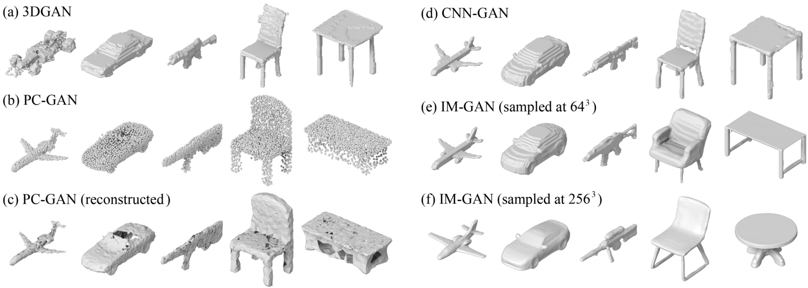
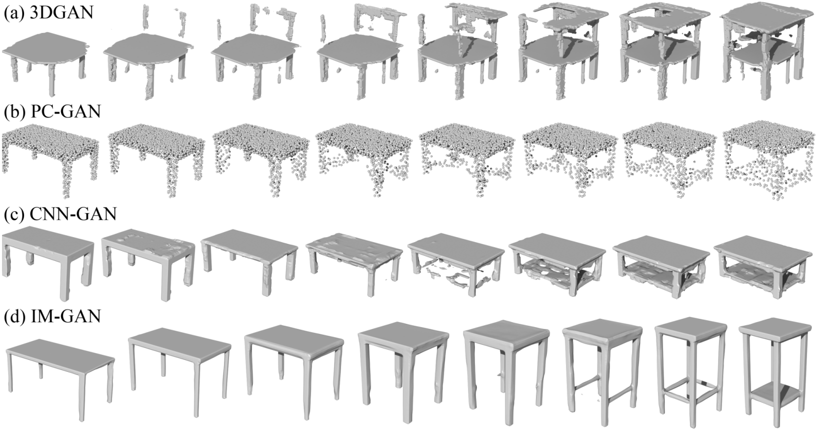


Figure: Authors trained a GAN model on latent codes of the autoencoder and used IM-NET or CNN Decoder to decode the generated codes

3D-shape interpolation

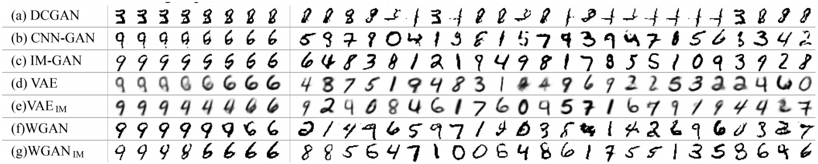


2D-shape generation

2D shape generation models:

- Train a GAN model on latent codes of the autoencoder and use IM-NET or CNN decoder
- Train VAE/WGAN/DCGAN
- Train VAE/WGAN but use IM-NET Decoder instead of decoder/generator part

For 2D-shape generation, they train all the models on only 5000 images.



Conclusion

Pros

- Smoother surfaces compared to voxel grids
- An ability to learn a complete shape (no need in parts annotation)
- Super-resolution and progressive growing without architectural changes
- Traditional CNN decoders are limited by GPU memory size (we cannot output high resolutions) during training, while INRs are not (since we can compute values only in points of interest)

Cons

- Much longer training time (up to $30\times$)
- Authors didn't test the approach on more complex datasets
- We need to train 1 model per category (likely because of weights sharing in Decoder)