Occupancy Networks

Learning 3D Reconstruction in Function Space

Thomas Wimmer, Lucas McIntyre
Analysis and Deep Learning on Geometric Data

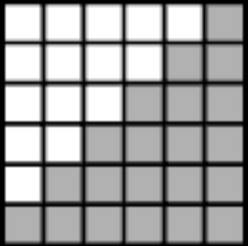
Mescheder, Lars, et al. "Occupancy networks: Learning 3d reconstruction in function space." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

What will this presentation cover?

- Motivation
- The model
- Experiments:
    1) 3D reconstruction from embedded representations
    2) 3D reconstruction from single observation
    3) 3D object generation
- Follow-up work

What are problems with current methods?

Voxels

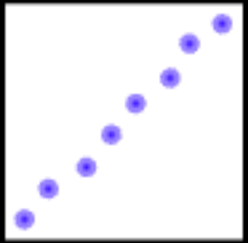


- Discretization of the 3D space into a grid of voxels
- Large Memory Footprint ($\mathcal{O}(n^3)$)

Meshes

- Discretization of the surface into vertices and faces
- Meshes are hard to predict for NNs (complex structure)

Point clouds



- Discretization of the surface into 3D points
- Lacking connectivity / topology

What else could we do?

What is the key idea?

No explicit representation (≈discretization)

→ Model surface implicitly as decision boundary of a non-linear classifier

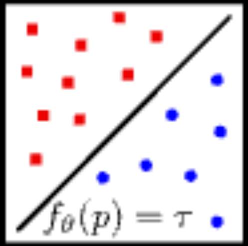
Occupancy function: $o : \mathbb{R}^3 \to \{0,1\}$

3D Reconstruction: $f_\theta : \mathbb{R}^3 \times \mathcal{X} \to [0,1]$

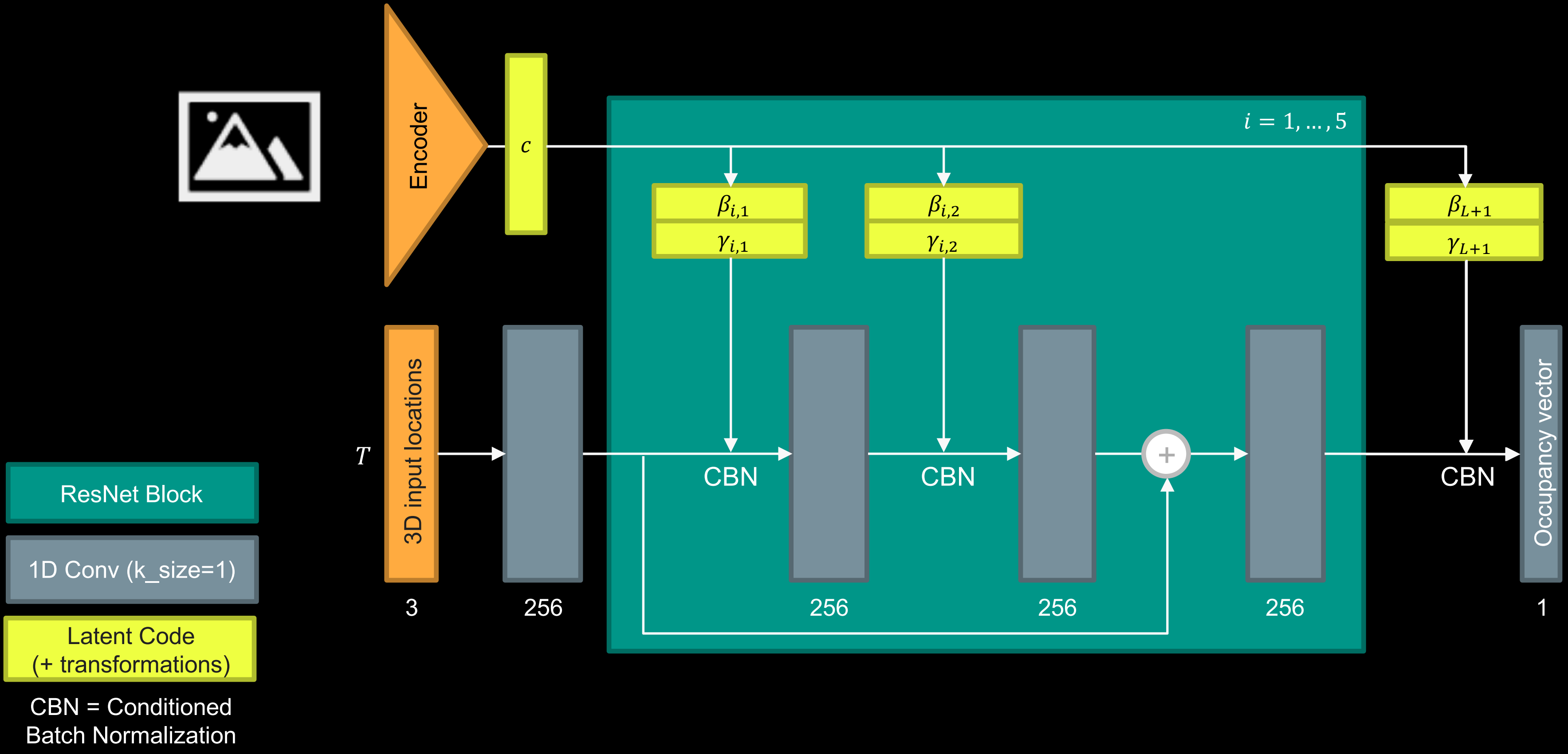
3D Location

Condition
(e.g. 2D image)

Occupancy
Probability



Okay, but how does this look like in practice?



Okay, but how does this look in practice?

Supervised Learning

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{K} BCE(f_\theta(p_{ij}, z_i), o_{ij})$$

- BCE: Binary Cross-Entropy
- K randomly sampled points p_{ij} for each training sample i (usually $K = 2048$)
- f_θ: Occupancy Network
- z_i: Condition for training sample i
- o_{ij}: Ground-truth occupancy

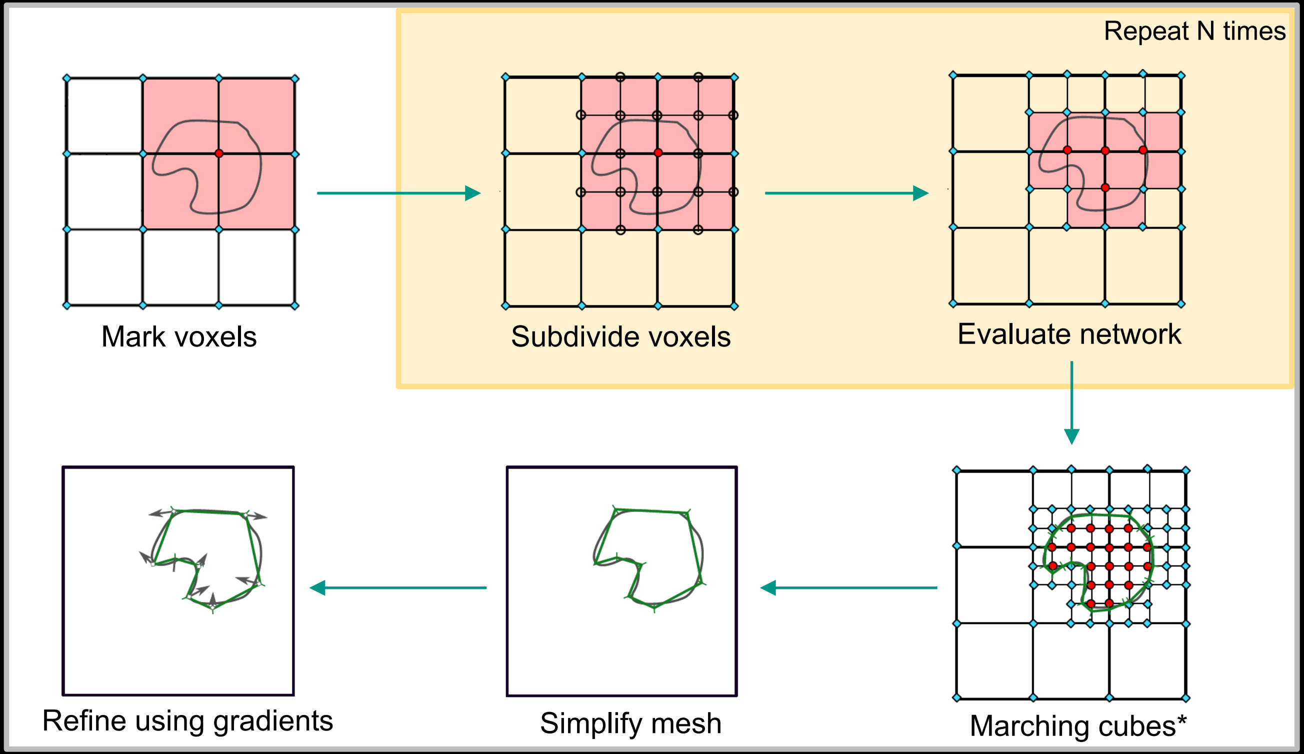
Okay, but how does this look in practice?

Unsupervised Learning

$$\mathcal{L}(\theta, \psi) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{K} BCE\left(f_\theta\left(p_{ij}, z_i\right), o_{ij}\right) + KL\left[q_\psi\left(z | \left(p_{ij}, o_{ij}\right)_{j=1:K}\right) \parallel p_0(z)\right]$$

- BCE: Binary Cross-Entropy
- K randomly sampled points p_{ij} for each training sample i (usually $K = 2048$)
- f_θ: Occupancy Network
- z_i: Condition for training sample i
- o_{ij}: Ground-truth occupancy
- KL: Kullback-Leibler divergence
- q_ψ: Encoder (cf. Variational Autoencoder)

Can we convert into explicit representations?



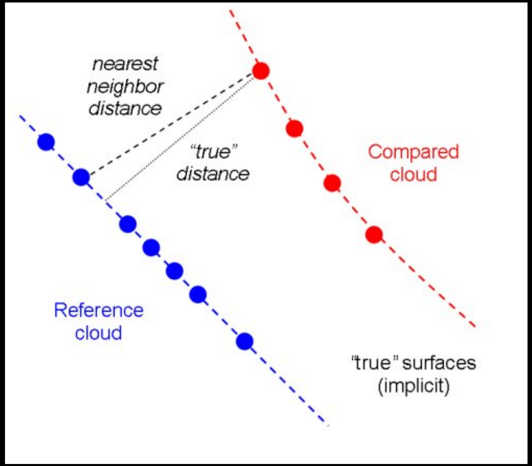
* Lorensen, William E., and Harvey E. Cline. "Marching cubes: A high resolution 3D surface construction algorithm." *ACM siggraph computer graphics* 21.4 (1987): 163-169.

How do we quantify the results?

↑ **Volumetric IoU** (estimated using 100k randomly sampled points from the bounding volume)

↓ **Chamfer-L_1 distance** (estimated by randomly sampling 100k points from both meshes)

- Accuracy Metric: Mean distance of points on output mesh to closest point in ground-truth mesh
- Completeness Metric: Same as the accuracy metric, but reversed



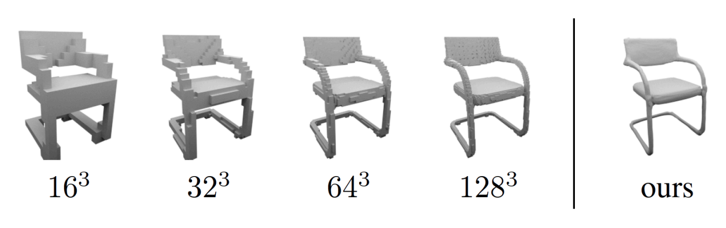
↑ **Normal consistency score** (mean absolute dot product of the normal in one mesh and the corresponding nearest neighbour in the other mesh)

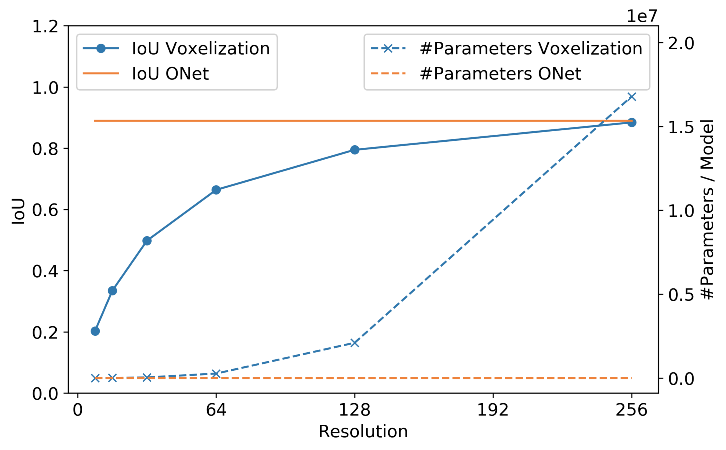
Which experiments can we conduct?

1) 3D Reconstruction from embedded representation

- "chair" category of the ShapeNet Dataset

→ Embed each training sample in a 512 dimensional latent space and train the neural network to reconstruct the 3D shape





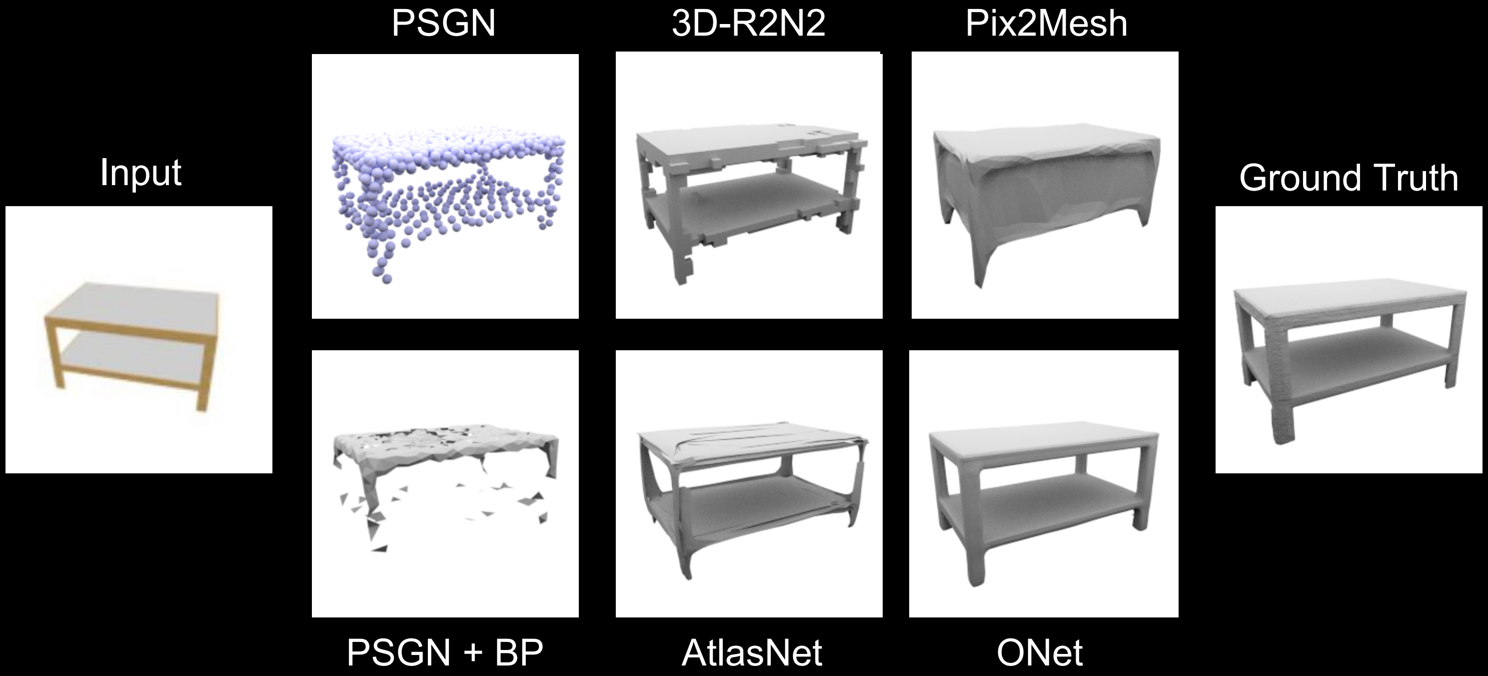
Which experiments can we conduct?

2) Single Image 3D Reconstruction – Setup

- ShapeNet Dataset
- Training carried out only on synthetic data
- Comparison against SOTA (2019) models generating different 3D data representations
- Tests on realistic data (KITTI, Online Products)

Which experiments can we conduct?

2) Single Image 3D Reconstruction – Qualitative Results



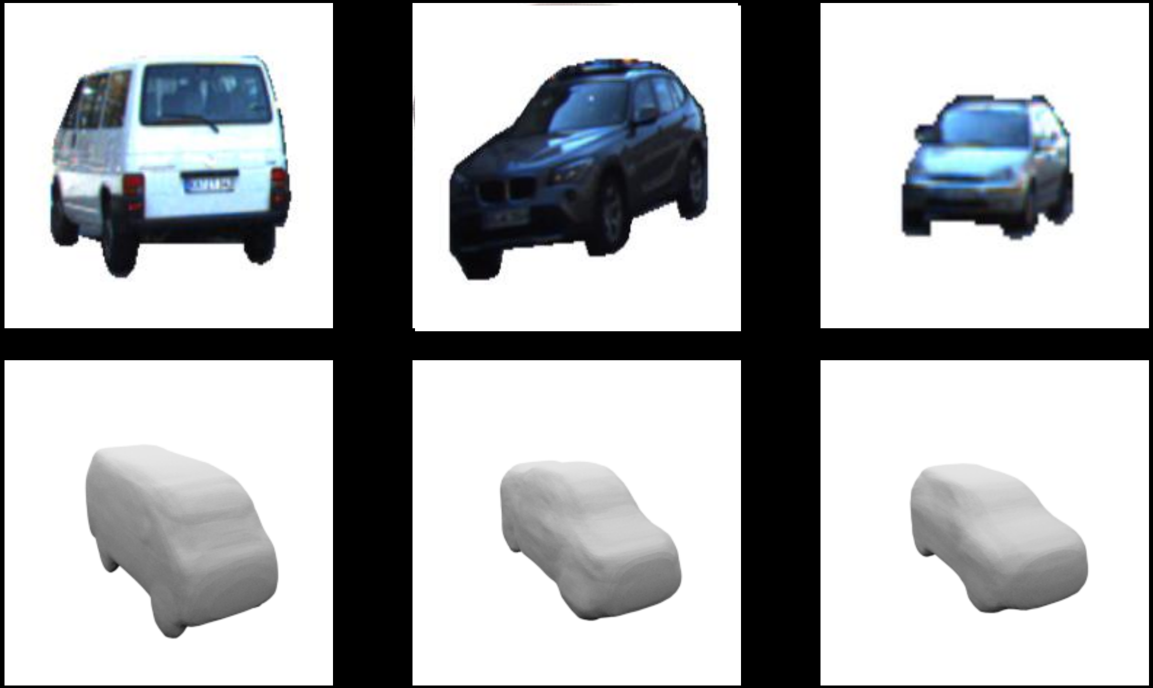
Which experiments can we conduct?

2) Single Image 3D Reconstruction – Quantitative Results

	IoU					Chamfer-L_1					Normal Consistency				
category	3D-R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	3D-R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	3D-R2N2	PSGN	Pix2Mesh	AtlasNet	ONet
airplane	0.426	-	0.420	-	**0.571**	0.227	0.137	0.187	**0.104**	0.147	0.629	-	0.759	0.836	**0.840**
bench	0.373	-	0.323	-	**0.485**	0.194	0.181	0.201	**0.138**	0.155	0.678	-	0.732	0.779	**0.813**
cabinet	0.667	-	0.664	-	**0.733**	0.217	0.215	0.196	0.175	**0.167**	0.782	-	0.834	0.850	**0.879**
car	0.661	-	0.552	-	**0.737**	0.213	0.169	0.180	**0.141**	0.159	0.714	-	0.756	0.836	**0.852**
chair	0.439	-	0.396	-	**0.501**	0.270	0.247	0.265	**0.209**	0.228	0.663	-	0.746	0.791	**0.823**
display	0.440	-	**0.490**	-	0.471	0.314	0.284	0.239	**0.198**	0.278	0.720	-	0.830	**0.858**	0.854
lamp	0.281	-	0.323	-	**0.371**	0.778	0.314	0.308	**0.305**	0.479	0.560	-	0.666	0.694	**0.731**
loudspeaker	0.611	-	0.599	-	**0.647**	0.318	0.316	0.285	**0.245**	0.300	0.711	-	0.782	0.825	**0.832**
rifle	0.375	-	0.402	-	**0.474**	0.183	0.134	0.164	**0.115**	0.141	0.670	-	0.718	0.725	**0.766**
sofa	0.626	-	0.613	-	**0.680**	0.229	0.224	0.212	**0.177**	0.194	0.731	-	0.820	0.840	**0.863**
table	0.420	-	0.395	-	**0.506**	0.239	0.222	0.218	0.190	**0.189**	0.732	-	0.784	0.832	**0.858**
telephone	0.611	-	0.661	-	**0.720**	0.195	0.161	0.149	**0.128**	0.140	0.817	-	0.907	0.923	**0.935**
vessel	0.482	-	0.397	-	**0.530**	0.238	0.188	0.212	**0.151**	0.218	0.629	-	0.699	0.756	**0.794**
mean	0.493	-	0.480	-	**0.571**	0.278	0.215	0.216	**0.175**	0.215	0.695	-	0.772	0.811	**0.834**

Which experiments can we conduct?

2) Single Image 3D Reconstruction – Real Data



Which experiments can we conduct?

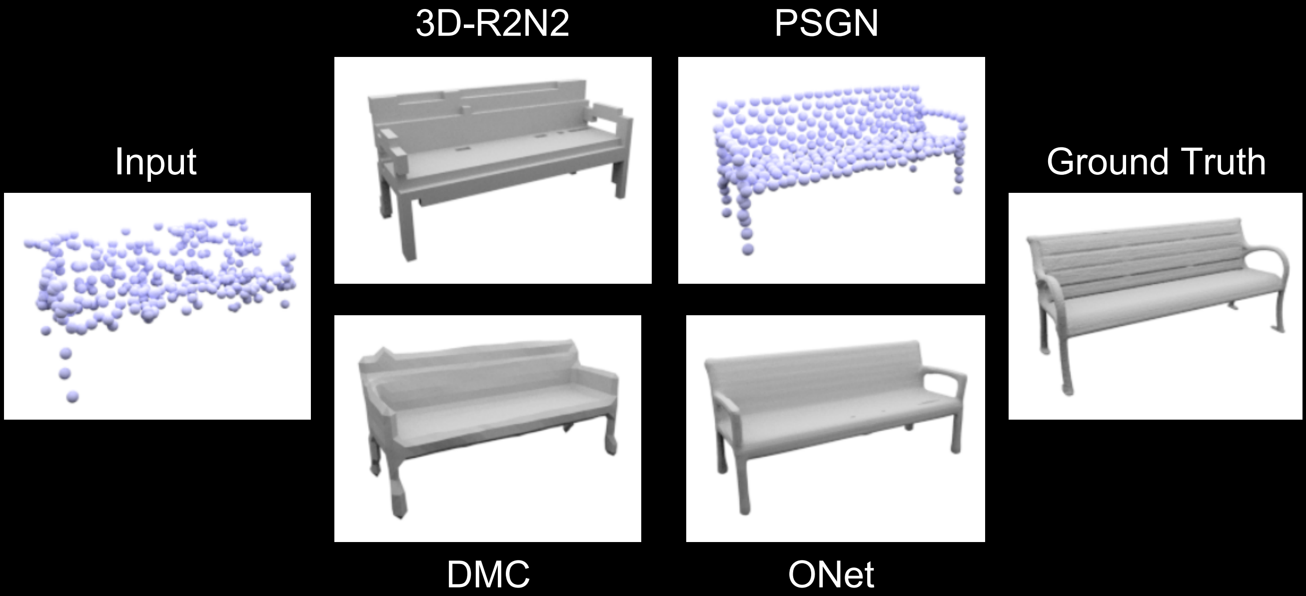
3) Point Cloud Completion

- Reconstruction of meshes from noisy point clouds
- Subsampling of 300 points from the surfaces of ShapeNet models and adding Gaussian noise

	IoU	Chamfer-L_1 †	Normal Consistency
3D-R2N2	0.565	0.169	0.719
PSGN	-	0.144	-
DMC	0.674	0.117	0.848
ONet	**0.778**	**0.079**	**0.895**

Which experiments can we conduct?

3) Point Cloud Completion



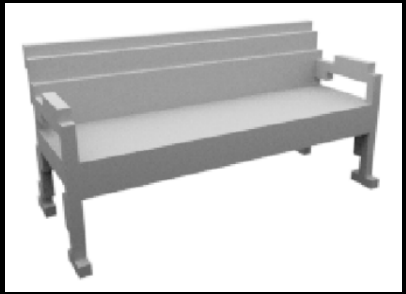
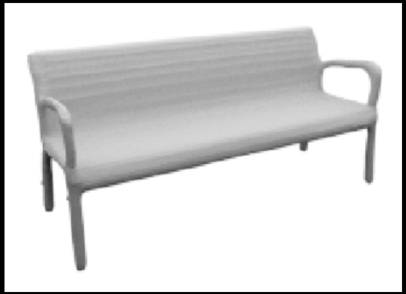
Which experiments can we conduct?

4) Voxel Super-Resolution

Input: coarse 32^3 voxelizations of a ShapeNet mesh

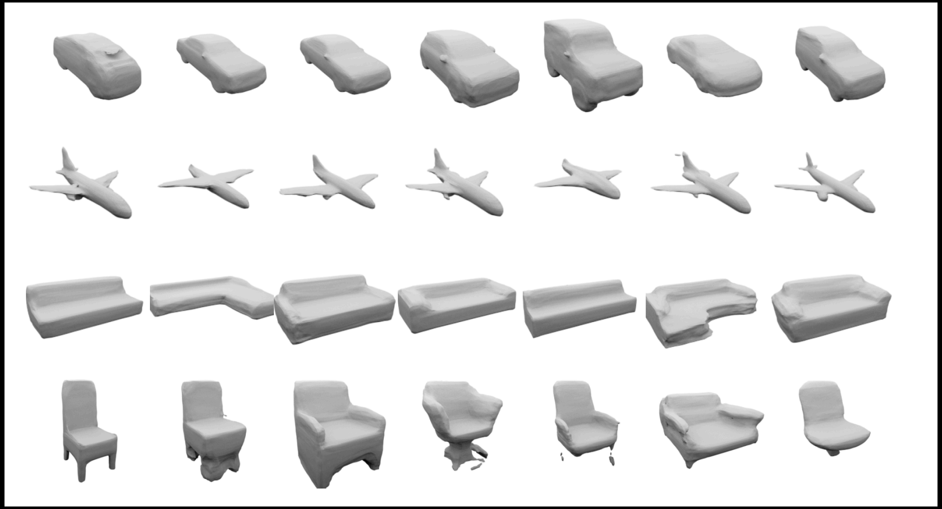
Task: Reconstruction of a high-resolution mesh

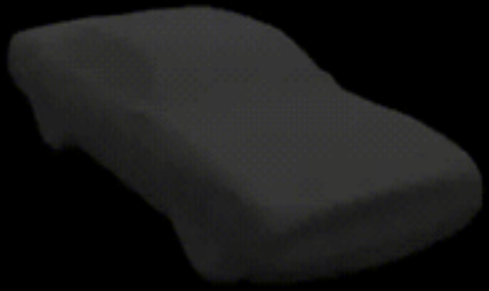
	IoU	Chamfer-L_1	Normal Consistency
Input	0.631	0.136	0.810
ONet	**0.703**	**0.109**	**0.879**


Are there any other cool properties?

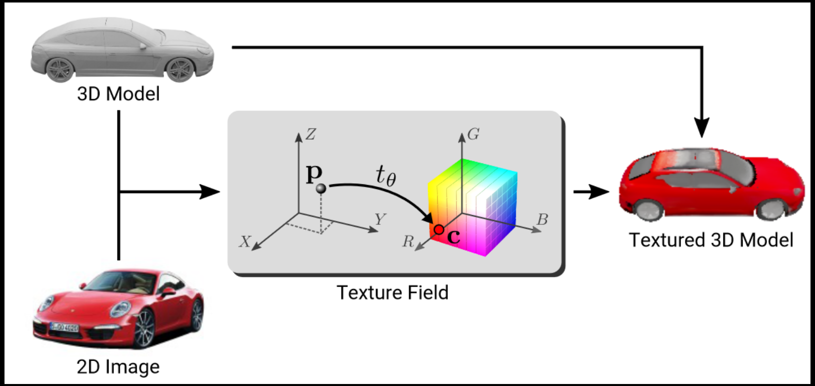
5) Shape generation and Latent Space Interpolations



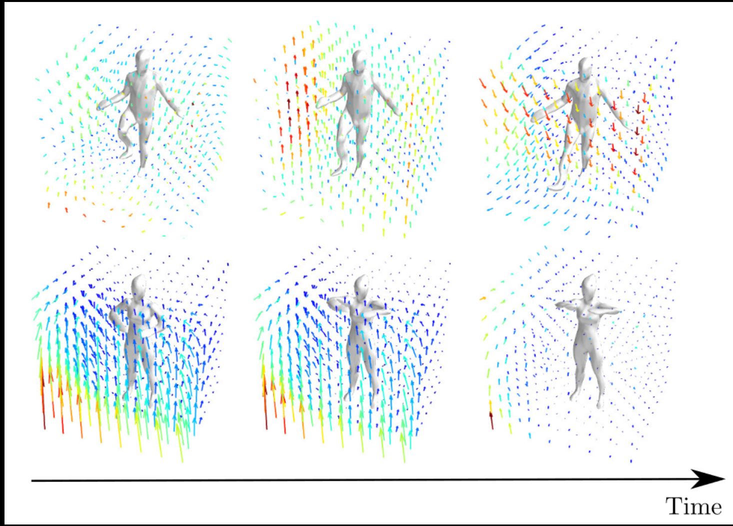


Any follow-up works?

Texture Fields*



Occupancy Flow°



* Oechsle, Michael, et al. "Texture fields: Learning texture representations in function space." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.
° Niemeyer, Michael, et al. "Occupancy flow: 4d reconstruction by learning particle dynamics." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

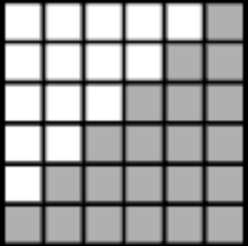
Any follow-up works?

NeRF*

* Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis." *Communications of the ACM* 65.1 (2021): 99-106.

What have we achieved?

Voxels

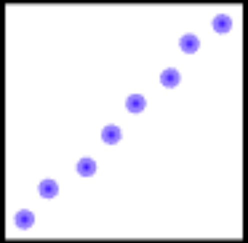


- Discretization of the 3D space into a grid of voxels
- Large Memory Footprint ($\mathcal{O}(n^3)$)

Meshes

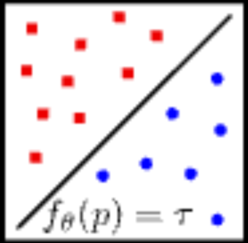
- Discretization of the surface into vertices and faces
- Meshes are hard to predict for NNs (complex structure)

Point clouds



- Discretization of the surface into 3D points
- Lacking connectivity / topology

Occupancy Networks



- Implicit representation of the shapes
- Arbitrary topology and resolution
- Low memory footprint

Occupancy Networks

Learning 3D Reconstruction in Function Space

Thomas Wimmer, Lucas McIntyre

Analysis and Deep Learning on Geometric Data

Refining the output mesh

1. Simplification using the Fast-Quadric-Mesh-Simplification*
2. Refining the output mesh using first and second order information:

$$\sum_{k=1}^{K}(f_\theta(p_k,x)-\tau)^2+\lambda\left\|\frac{\nabla f_\theta(p_k,x)}{\|\nabla f_\theta(p_k,x)\|}-n(p_k)\right\|^2$$

- Sample random points p_k from each face of the mesh
- $n(p_k)$: Normal vector of mesh at p_k

→ Can efficiently normalized using double backpropagation°

* Garland, Michael, and Paul S. Heckbert. "Simplifying surfaces with color and texture using quadric error metrics." *Proceedings Visualization'98 (Cat. No. 98CB36276)*. IEEE, 1998.
° Drucker, Harris, and Yann Le Cun. "Improving generalization performance using double backpropagation." IEEE transactions on neural networks 3.6 (1992): 991-997.

Ablation Study

Sampling Strategy

	IoU	Chamfer-L_1	Normal Consistency
Uniform	**0.571**	**0.215**	0.834
Uniform (64)	0.554	0.256	0.829
Equal	0.475	0.291	**0.835**
Surface	0.536	0.254	0.822

(a) Influence of Sampling Strategy

Effect of architecture

	IoU	Chamfer-L_1	Normal Consistency
Full model	**0.571**	**0.215**	**0.834**
No ResNet	0.559	0.243	0.831
No CBN	0.522	0.301	0.806

(b) Influence of Occupancy Network Architecture

Limits of the proposed method

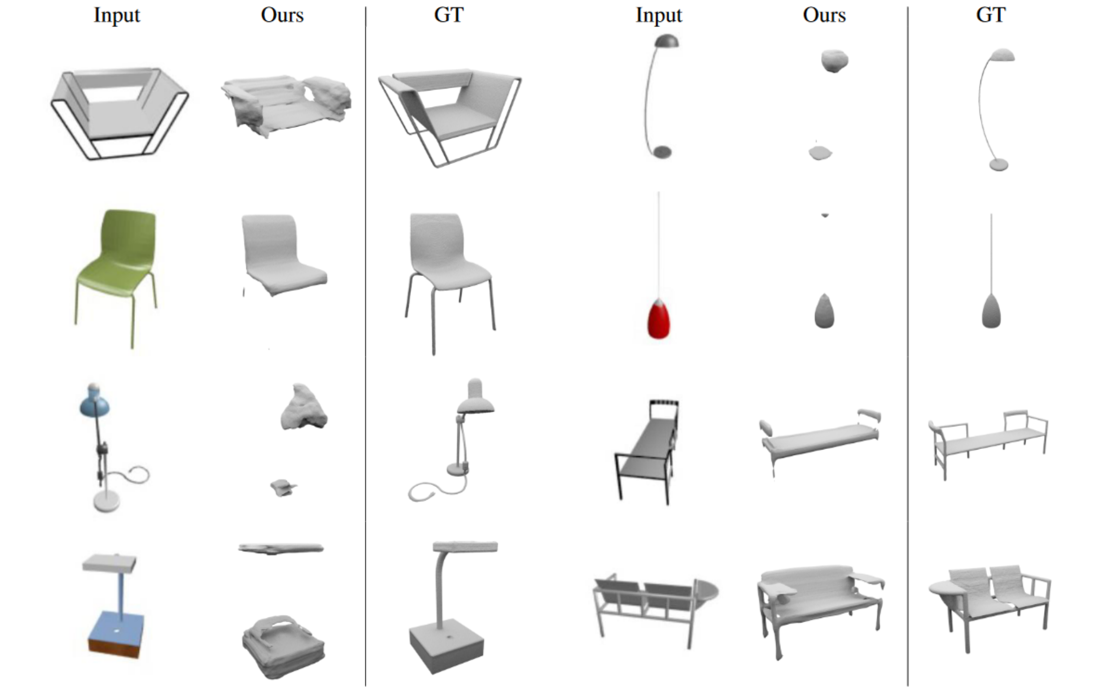


Figure 11: **Failure Cases.** While our method generally performs well, it struggles with extremely thin object parts and objects that are very different from the objects seen during training. These kinds of objects are especially frequent for the "lamp' category of the ShapeNet dataset. The input is shown in the first column, the other columns show the results for our method compared to the ground truth.