Neural Networks for 3D Data

CS 6384 Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Neural Networks for Images and Languages

- Image recognition

- Natural Language Understanding

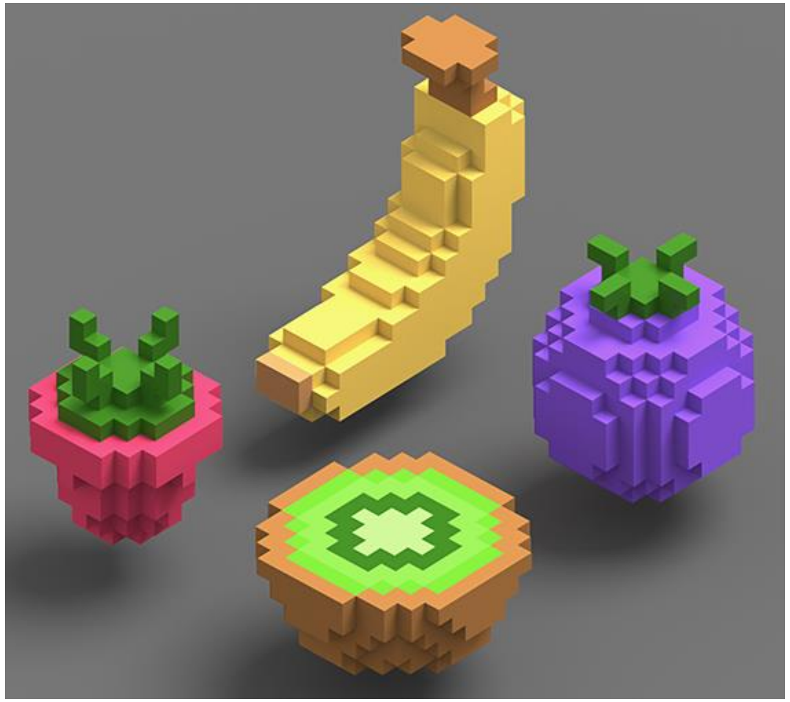
Google Translation

ImageNet classification

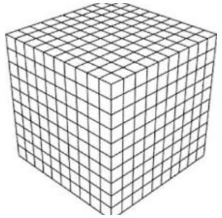
3D Data

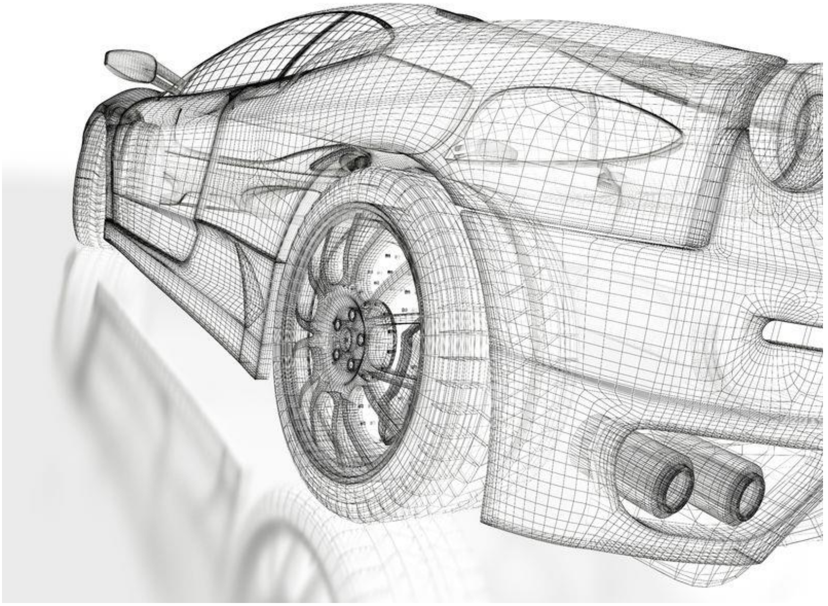
Can we use neural networks for these 3D data?

3D points



3D Voxels

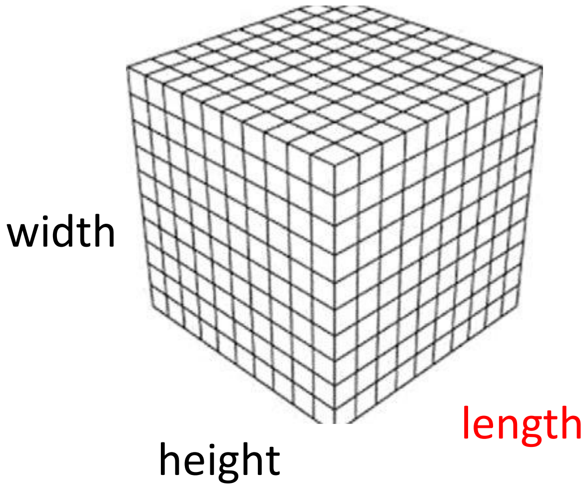


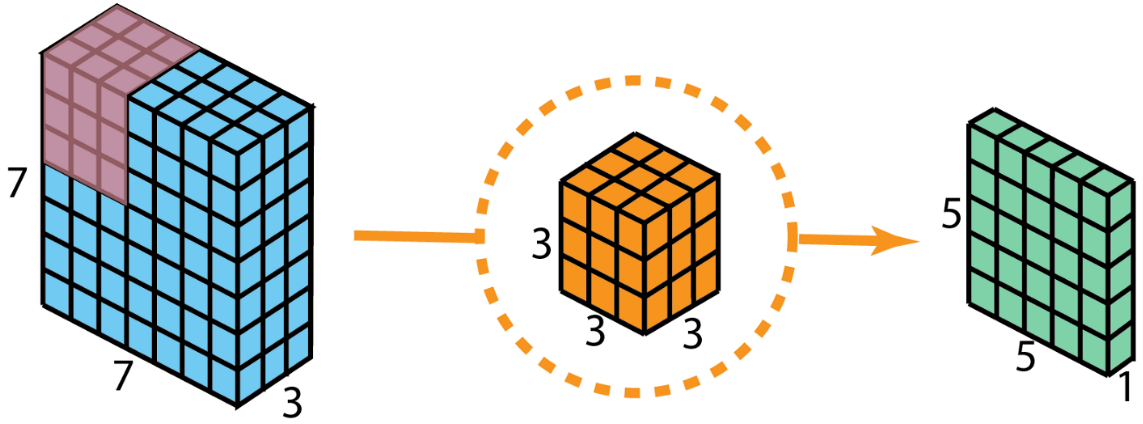


3D Meshes

3D Voxels

- Add an additional dimension to images
  - Images [height, width, 3]
  - Voxels [height, width, length, 3] (the last dimension can change depending on what data to store)
- Use 3D convolutions



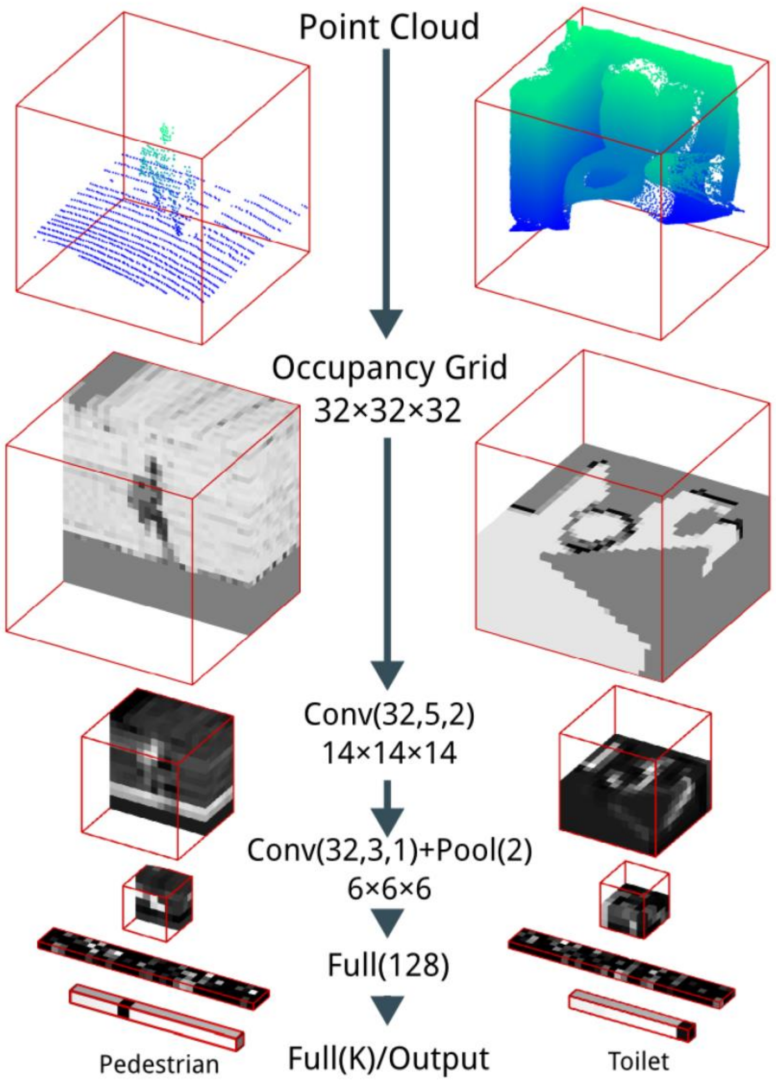


VoxNet

- Input: Volumetric occupancy grid
  - Each voxel stores the probability of that voxel is occupied
- 3D convolution layer

$$C(f, d, s)$$

filters filter size stride

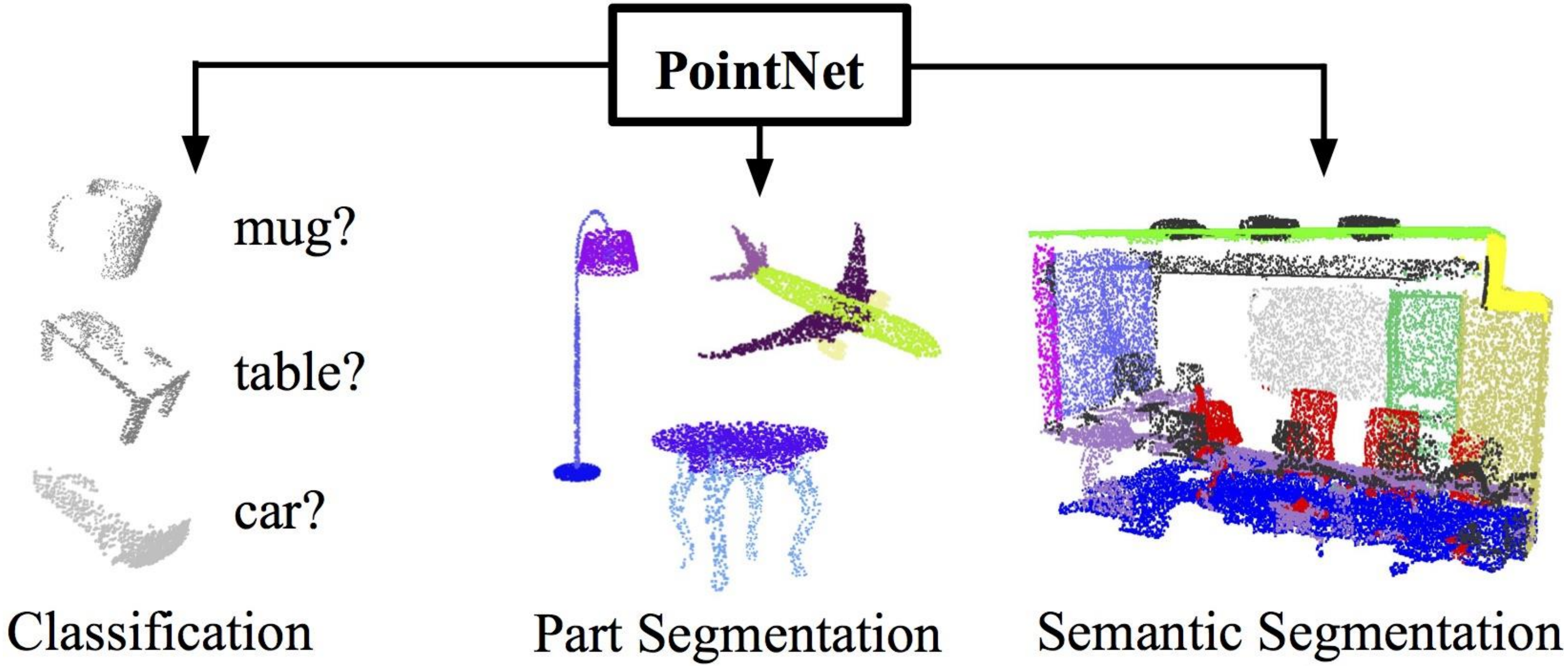


VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. Maturana & Scherer, IROS'15

3D Points

- 3D convolution is expensive
- 3D points $N \times 3$
  - A set, irregular format
  - Cannot directly apply 2D convolution or 3D convolution
  - Invariant to permutation and rigid transformation

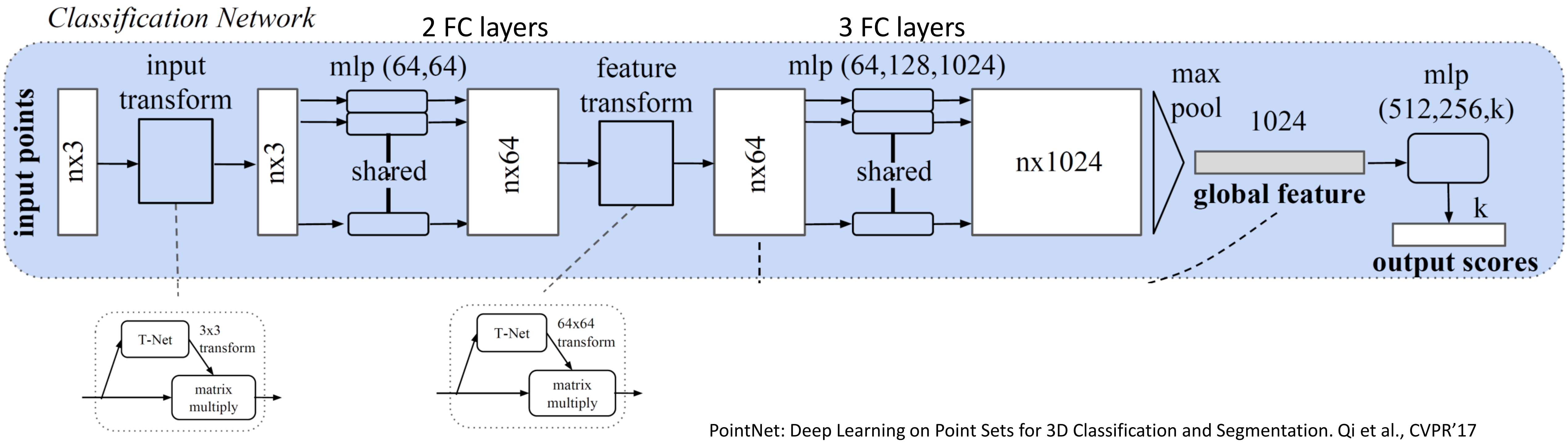
PointNet



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Qi et al., CVPR'17.

PointNet

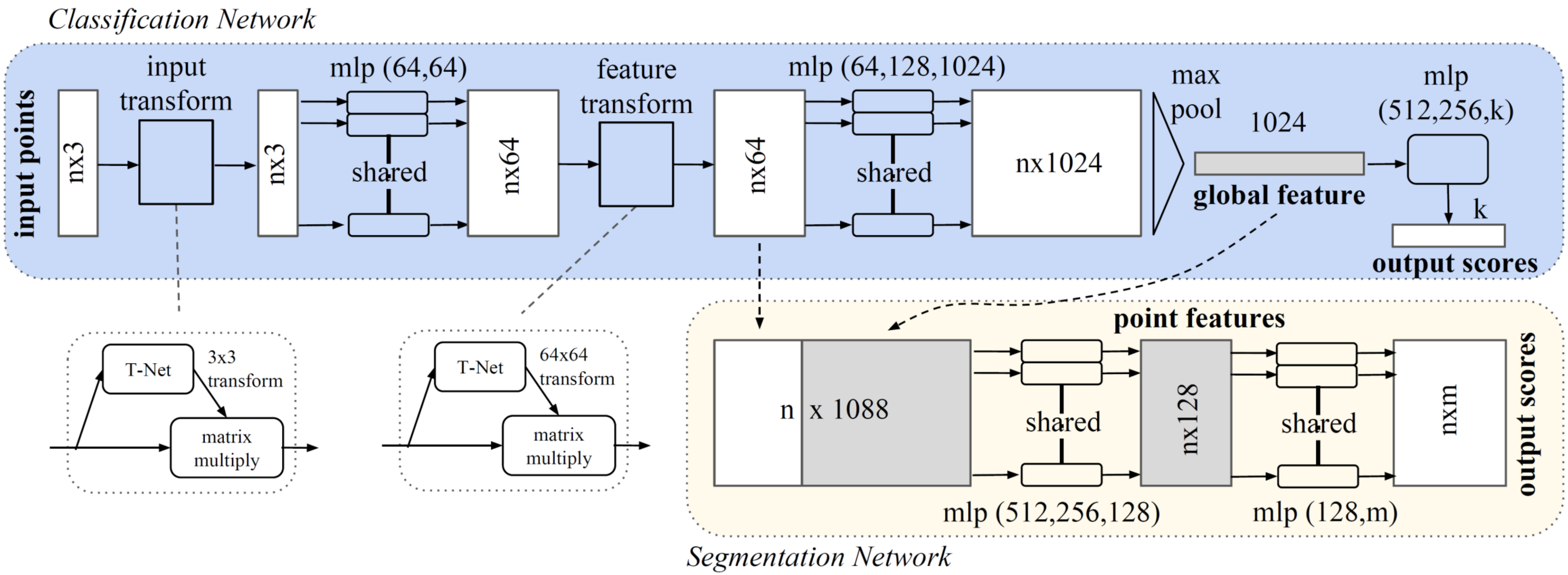
- Design principle
  - Invariant to permutation and rigid transformation
  - Per-point feature extraction and max-pooling



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Qi et al., CVPR'17

PointNet

- Point-wise labeling



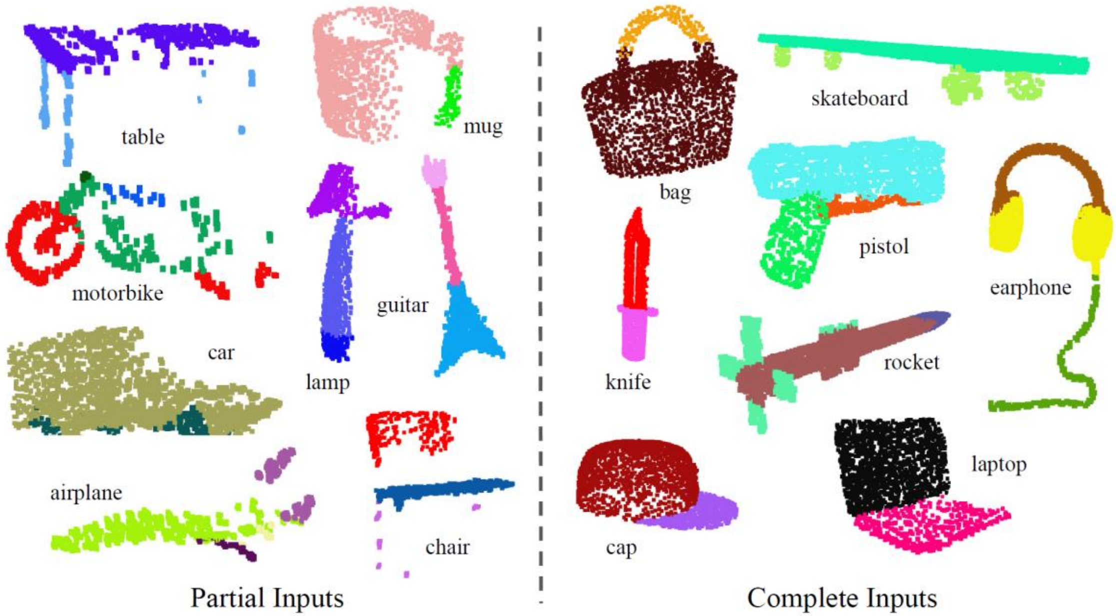
PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Qi et al., CVPR'17

PointNet

	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [28]	volume	1	77.3	84.7
VoxNet [17]	volume	12	83.0	85.9
Subvolume [18]	volume	20	86.0	**89.2**
LFD [28]	image	10	75.5	-
MVCNN [23]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

Table 1. **Classification results on ModelNet40.** Our net achieves state-of-the-art among deep nets on 3D input.

3D Shape Classification



Part segmentation

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Qi et al., CVPR'17

PointNet++

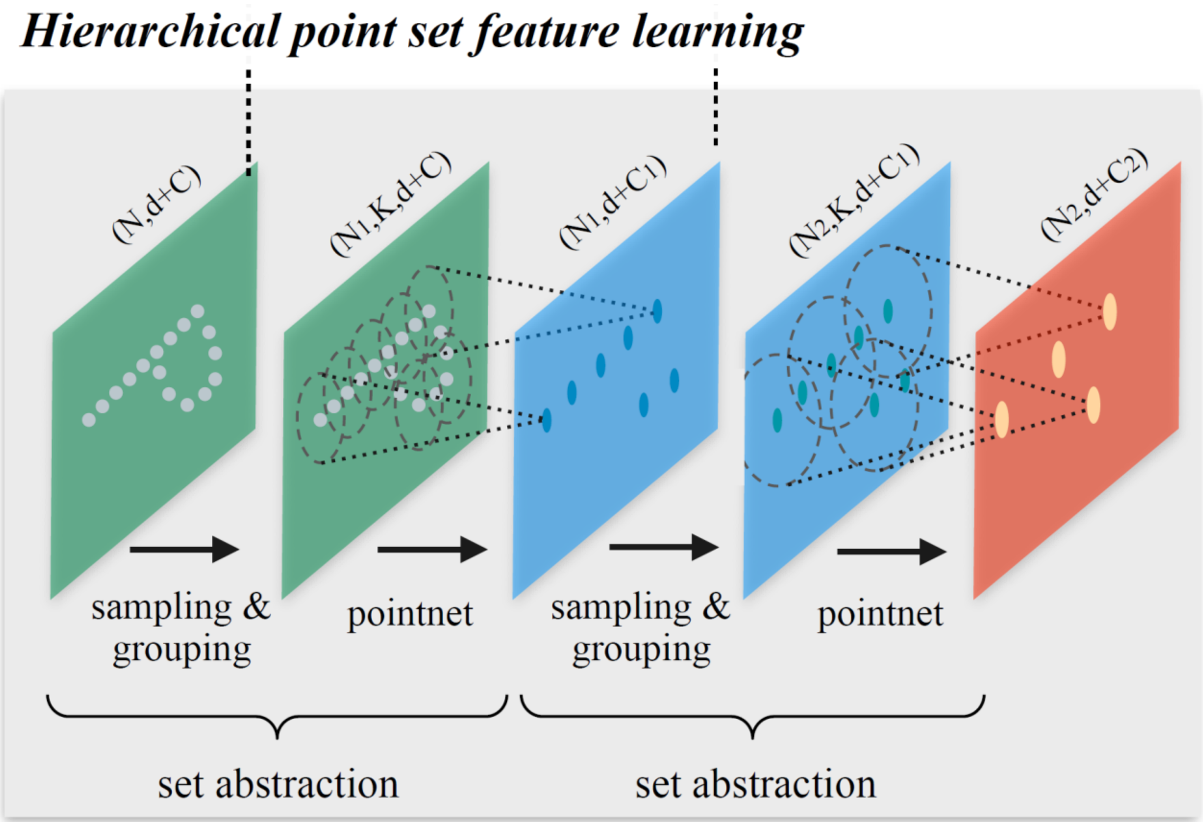
- PointNet cannot capture local structures of the point clouds
    - Per-point feature extraction and max-pooling

- PointNet++
    - A hierarchical neural network on 3D points

    - Use PointNet as a building block, extract features in a hierarchical way

PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Qi et al., NuerIPS'17

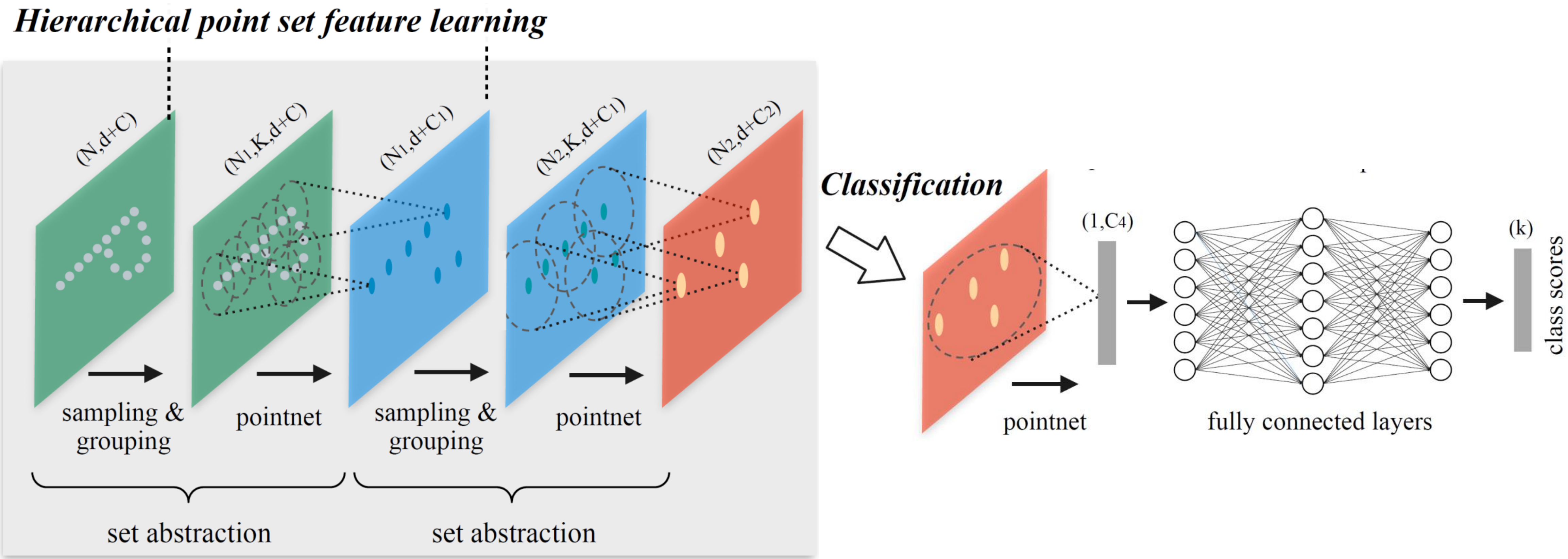
PointNet++



- Set abstraction levels (3 levels used)
  - Sampling layer (farthest point sampling), sample N' points (centroids)
  - Grouping layer, find K nearest neighbors for each centroid
    - Ball query
    - KNN
  - PointNet layer, extract a feature vector with dimension C' for each centroid and its neighbors

PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Qi et al., NuerIPS'17

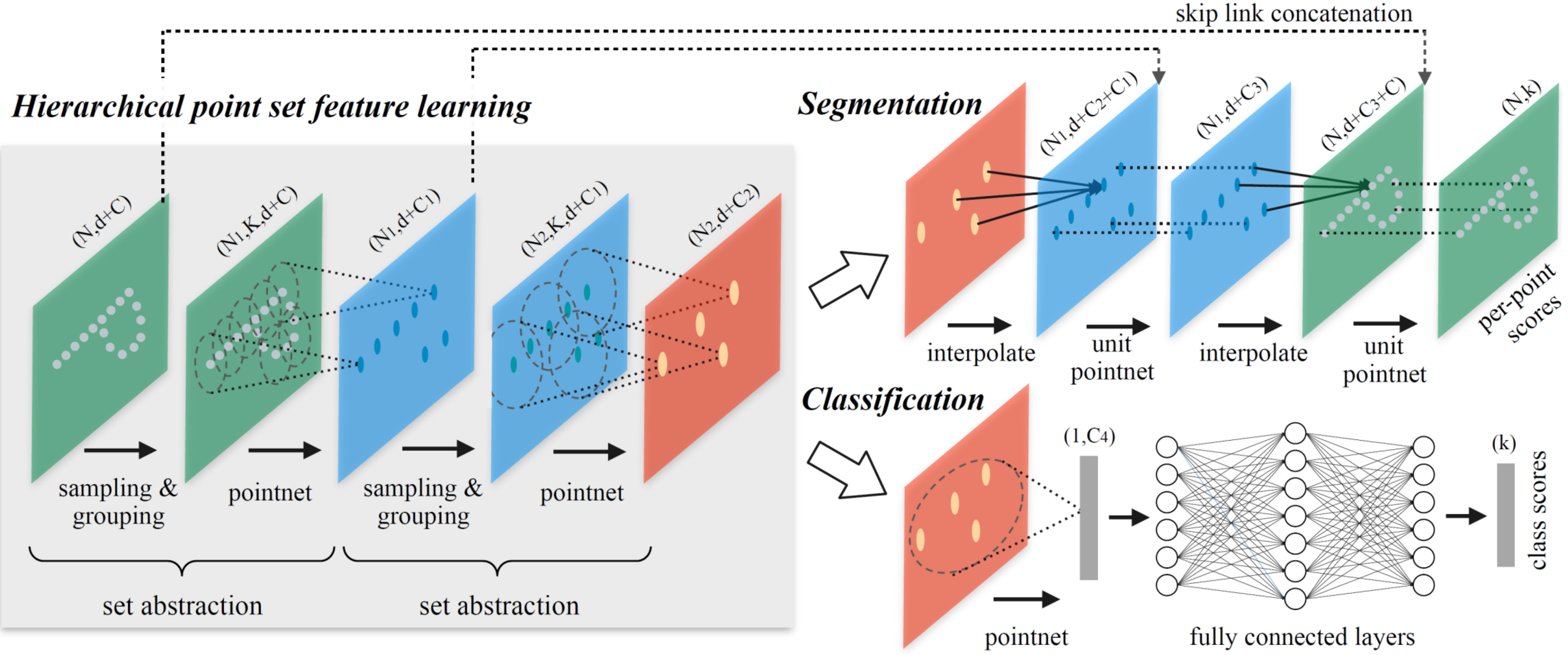
PointNet++



PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Qi et al., NuerIPS'17

PointNet++

Point Feature Propagation for Set Segmentation



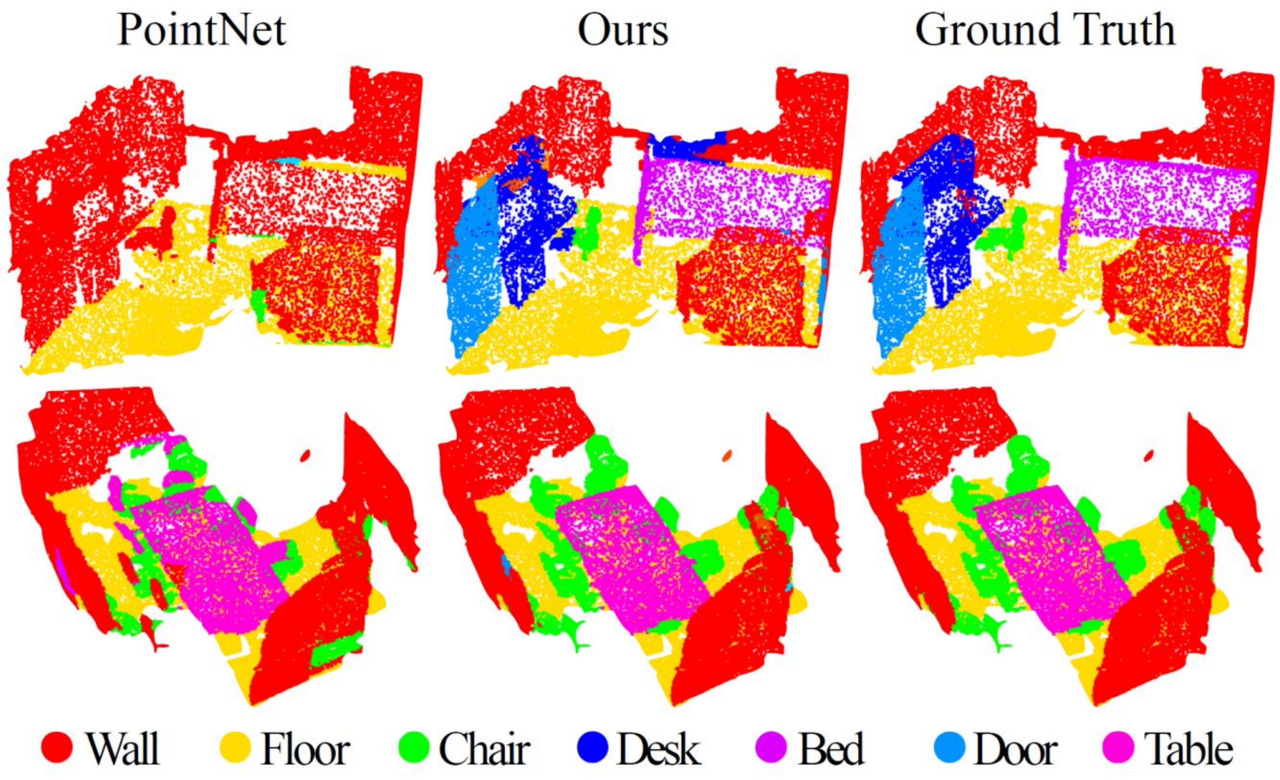
PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Qi et al., NuerIPS'17

PointNet++

Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Table 2: ModelNet40 shape classification.

3D Shape Classification



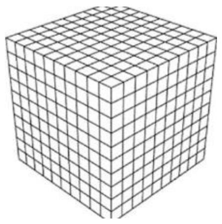
3D point segmentation

PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Qi et al., NuerIPS'17

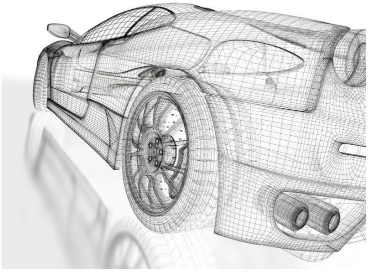
Implicit Representations of 3D Data

- Explicit shape representations

3D points

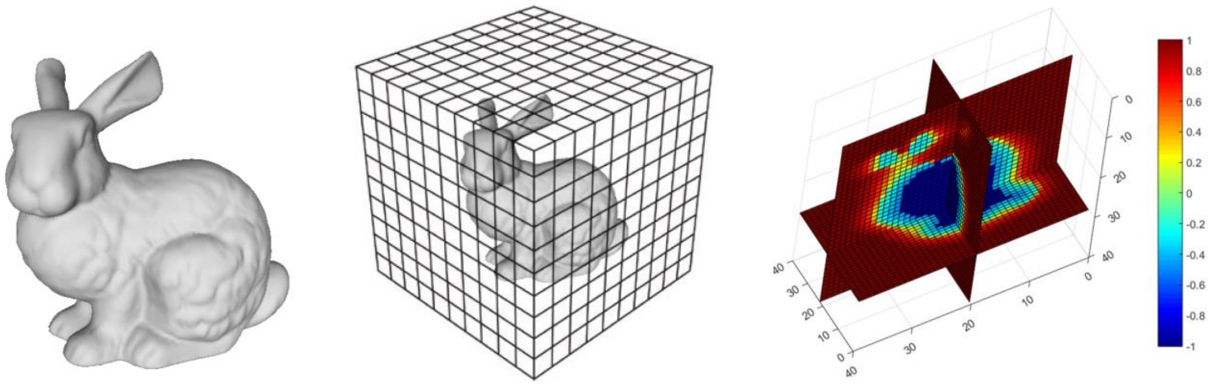


3D Voxels



3D Meshes

- Implicit shape representations
  - Use a function to encode the 3D shape
  - Example: Signed Distance Fields (SDFs)



(a) Surface view. (b) Bounding volume. (c) Generated SDF.

Signed Distance Fields for Rigid and Deformable 3D Reconstruction. Miroslava Slavcheva.

Occupancy Network for 3D Reconstruction

- Occupancy function

$$o : \mathbb{R}^3 \rightarrow \{0, 1\}$$

3D location

- Training a neural network to learn the following function

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

3D location

Input for 3D reconstruction: image, point cloud

Probabilities for occupancy

- Image: ResNet
- Points: PointNet

Occupancy Networks: Learning 3D Reconstruction in Function Space. Mescheder et al., CVPR'19

Occupancy Network for 3D Reconstruction

- Training

Cross-entropy loss function

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

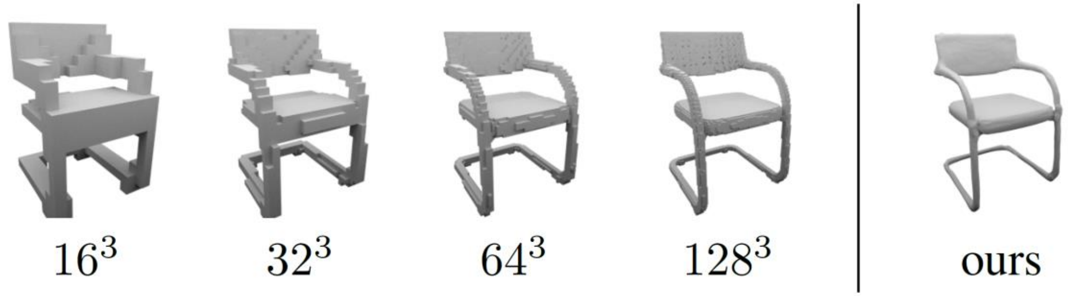
3D point j for image i

image i

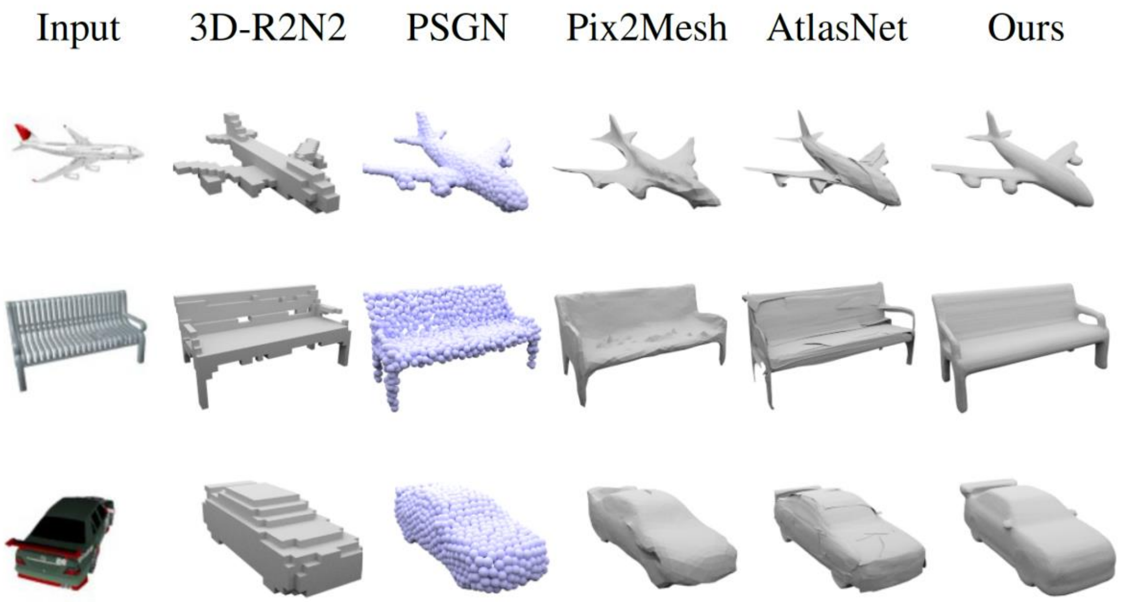
Ground truth occupancy

Occupancy Networks: Learning 3D Reconstruction in Function Space. Mescheder et al., CVPR'19

Occupancy Network for 3D Reconstruction



Continuous shape representation



Single image 3D reconstruction

Occupancy Networks: Learning 3D Reconstruction in Function Space. Mescheder et al., CVPR'19

DeepSDF

- Signed distance function

$$SDF(\boldsymbol{x}) = s : \boldsymbol{x} \in \mathbb{R}^3,\ s \in \mathbb{R}$$

- Train a neural network to predict SDFs

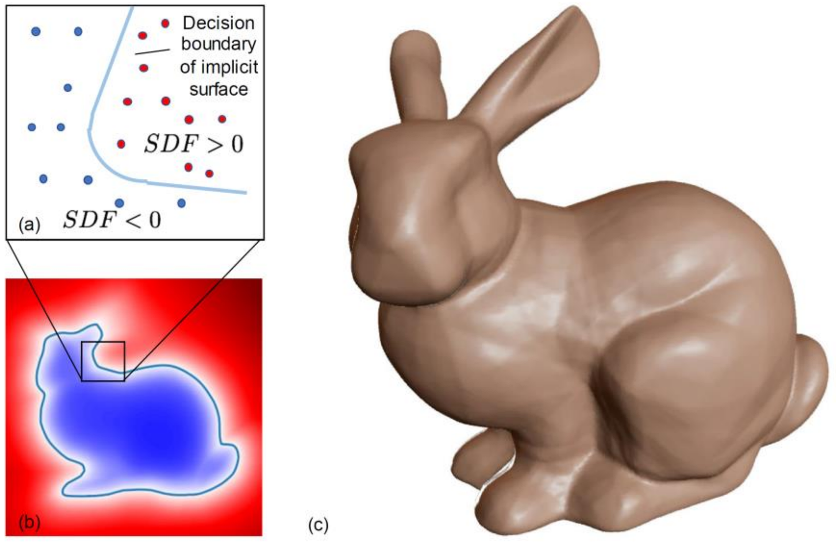
$$f_\theta(\boldsymbol{x}) \approx SDF(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \Omega$$

- Loss function

$$\mathcal{L}(f_\theta(\boldsymbol{x}), s) = |\,\text{clamp}(f_\theta(\boldsymbol{x}), \delta) - \text{clamp}(s, \delta)\,|$$

$$\text{clamp}(x, \delta) := \min(\delta, \max(-\delta, x))$$

distance from the surface over
which we expect to maintain a metric SDF

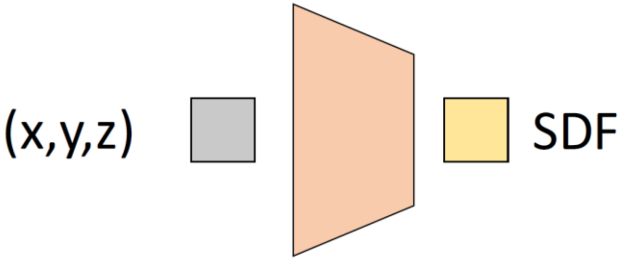


- 8 FC layers with dropout
- 512-d FC layer with ReLU
- Output with tanh

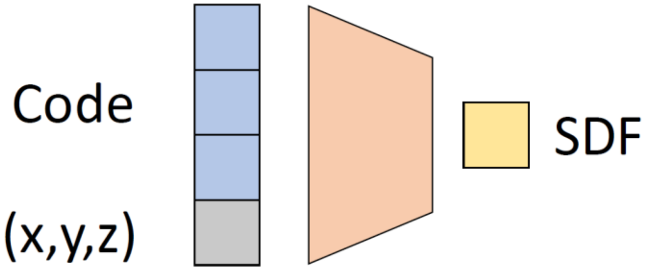
DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. Park et al., CVPR'19

DeepSDF

- Learning the latent space of shapes



(a) Single Shape DeepSDF



(b) Coded Shape DeepSDF

$$f_\theta(\boldsymbol{x}) \approx SDF(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \Omega$$

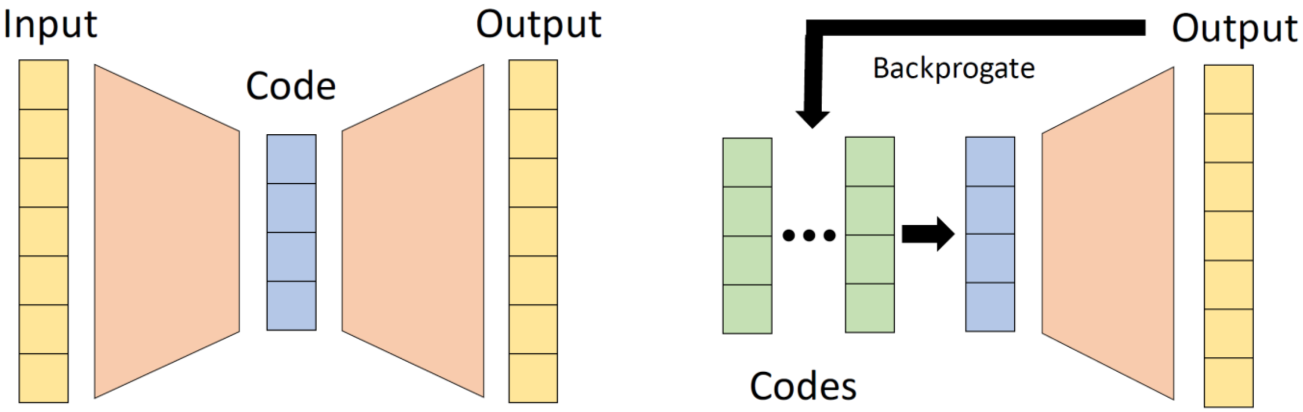
$$f_\theta(\boldsymbol{z}_i, \boldsymbol{x}) \approx SDF^i(\boldsymbol{x})$$

Code for shape i

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. Park et al., CVPR'19

DeepSDF

- Auto-decoder



(a) Auto-encoder **(b)** Auto-decoder

- Training objective

$$\underset{\theta,\{\boldsymbol{z}_i\}_{i=1}^{N}}{\arg\min} \sum_{i=1}^{N} \left(\sum_{j=1}^{K} \mathcal{L}(f_\theta(\boldsymbol{z}_i, \boldsymbol{x}_j), s_j) + \frac{1}{\sigma^2}||\boldsymbol{z}_i||_2^2 \right)$$

- Inference

$$\hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\arg\min} \sum_{(\boldsymbol{x}_j, \boldsymbol{s}_j) \in X} \mathcal{L}(f_\theta(\boldsymbol{z}, \boldsymbol{x}_j), s_j) + \frac{1}{\sigma^2}||\boldsymbol{z}||_2^2$$

Shape completion from partial point clouds

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. Park et al., CVPR’19

DeepSDF

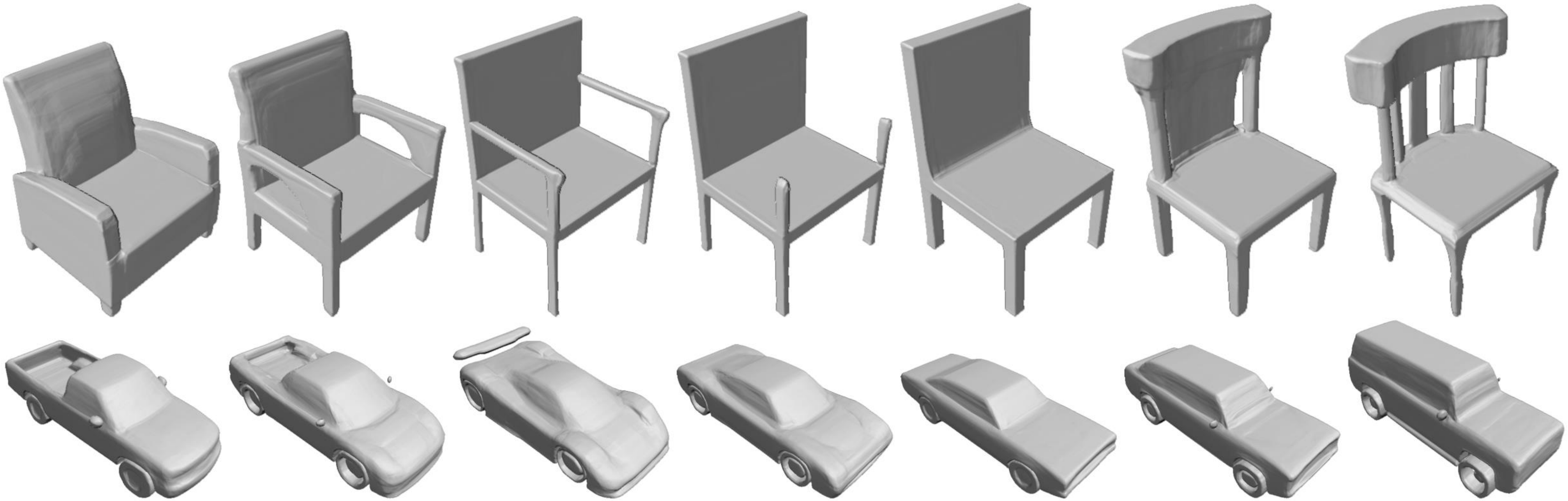
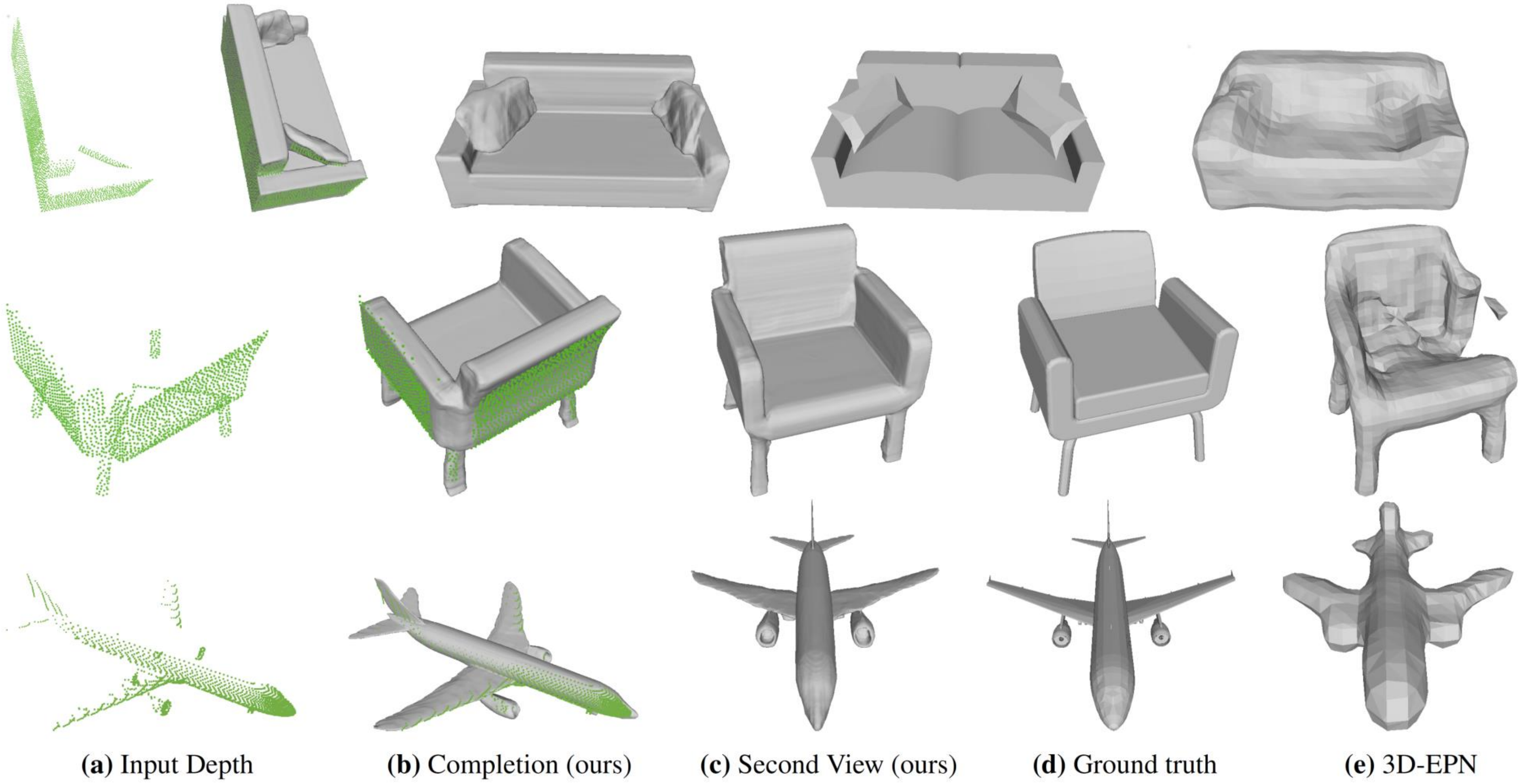


Figure 1: DeepSDF represents signed distance functions (SDFs) of shapes via latent code-conditioned feed-forward decoder networks. Above images are raycast renderings of DeepSDF interpolating between two shapes in the learned shape latent space. Best viewed digitally.

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. Park et al., CVPR'19

DeepSDF



DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. Park et al., CVPR’19

Neural Radiance Fields (NeRF)

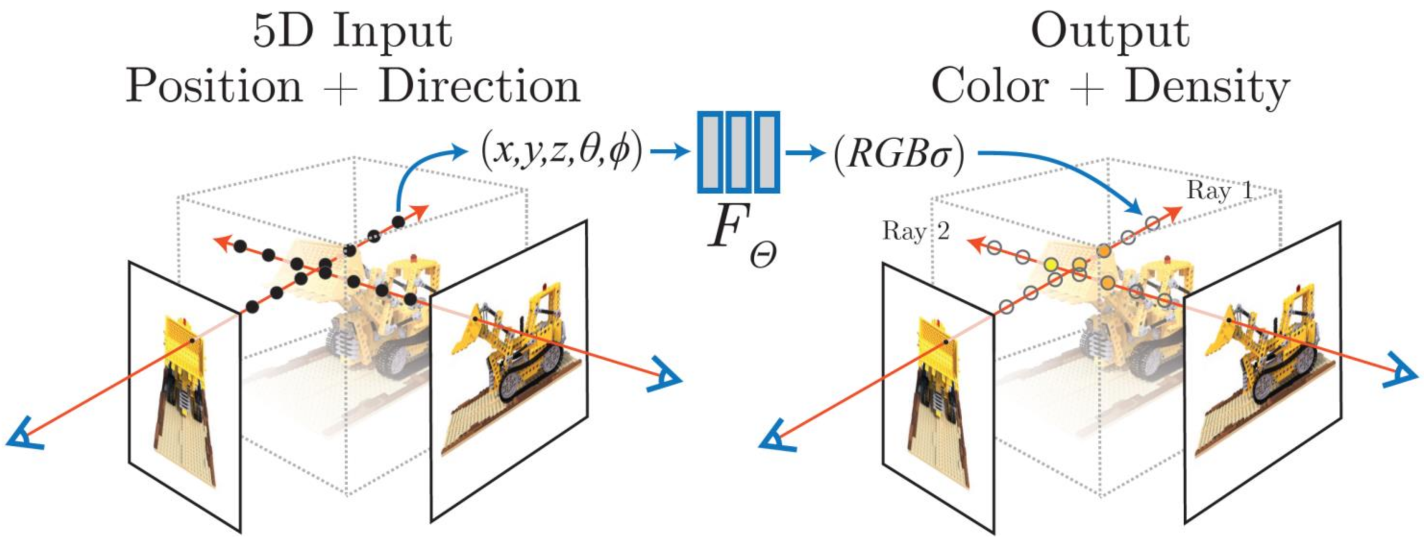
- Represent 3D scenes with color information (geometry + appearance)
- Learning a 5D vector-valued function

$$F_\Theta(x, y, z, \theta, \phi) = (r, g, b, \sigma)$$

3D location — Viewpoint: azimuth, elevation — Color (RGB) — Density

$$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$$

Unit vector for direction



NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. Mildenhall et al., ECCV'20

Neural Radiance Fields (NeRF)

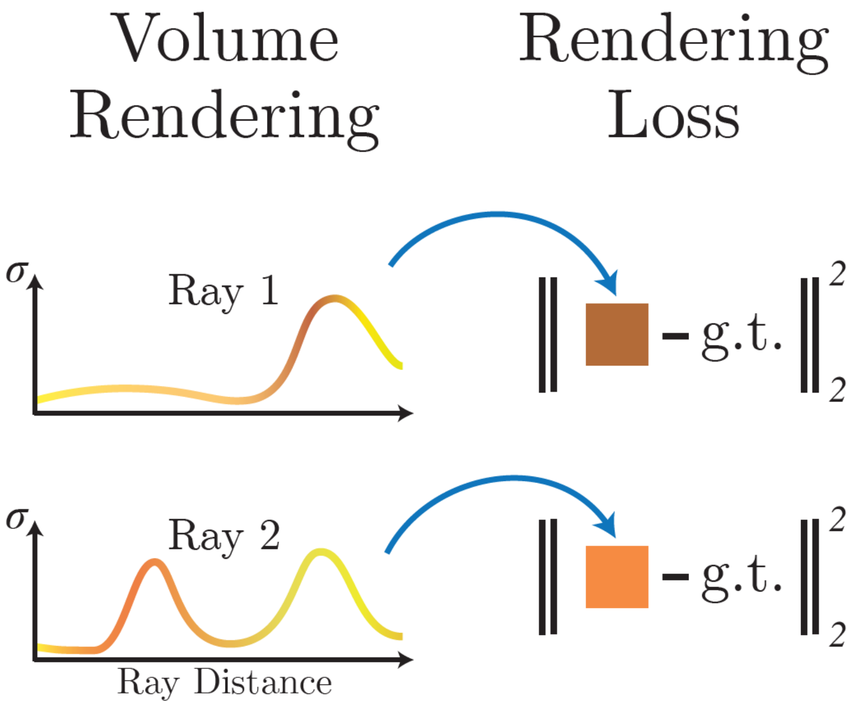
- Volumetric rendering

Ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$

Color of the ray $C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt$

Density Color

Probability that the ray travels from t_n to t without hitting any other particle $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$



Rending: find C(r) for a camera ray traced through each pixel

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. Mildenhall et al., ECCV'20

Neural Radiance Fields (NeRF)

View Synthesis

https://www.matthewtancik.com/nerf

Summary

- Neural networks can be applied to 3D data
  - Shape recognition, shape reconstruction
  - Point cloud segmentation
  - View synthesis
  - Etc.

- Explicit 3D representations
  - Voxels, points, meshes

- Implicit 3D representations
  - Learn a function to represent the 3D shape (occupancy, SDFs, radiance fields)

Further Reading

- VoxNet https://www.ri.cmu.edu/pub_files/2015/9/voxnet_maturana_scherer_iros15.pdf
- PointNet https://arxiv.org/abs/1612.00593
- PointNet++ https://arxiv.org/pdf/1706.02413.pdf
- Occupancy Network https://arxiv.org/abs/1812.03828
- DeepSDF https://arxiv.org/abs/1901.05103
- NeRF https://arxiv.org/abs/2003.08934
- NeRF Explosion 2020 https://dellaert.github.io/NeRF/