NYU TANDON SCHOOL OF ENGINEERING

AI4CE

Robot Vision

Segmentation & 3D Deep Learning

Dr. Chen Feng

cfeng@nyu.edu

ROB-UY 3203, Spring 2024

Overview

- Segmentation
  - Semantic Segmentation
  - Instance Segmentation
  - Panoptic Segmentation
- 3D Deep Learning
  - MVCNN
  - VoxNet
  - PointNet

  - FoldingNet
  - DeepMapping

References

- Szeliski 2022
  - Section 6.4

- Forsyth & Ponce 2011
  - Chapter 9

- Su, H., Maji, S., Kalogerakis, E. and Learned-Miller, E., 2015. Multi-view convolutional neural networks for 3d shape recognition. In Proceedings of the IEEE international conference on computer vision (pp. 945-953).
- Maturana, D. and Scherer, S., 2015, September. Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 922-928). IEEE.
- Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652-660).

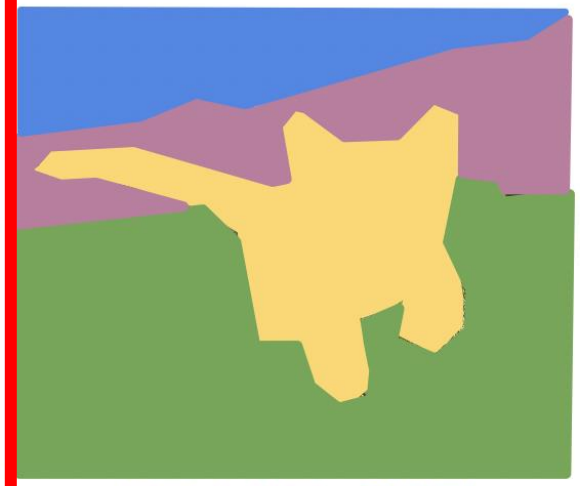
Semantic Segmentation: The Problem

Classification

CAT

No spatial extent

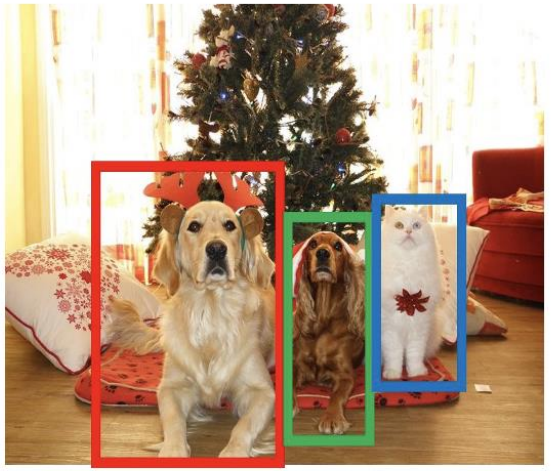
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection



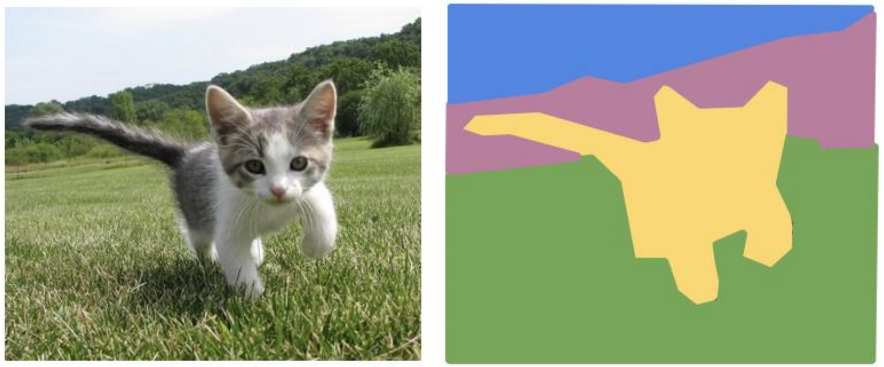
DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

Multiple Object

Semantic Segmentation: The Problem



GRASS, CAT,
TREE, SKY, ...

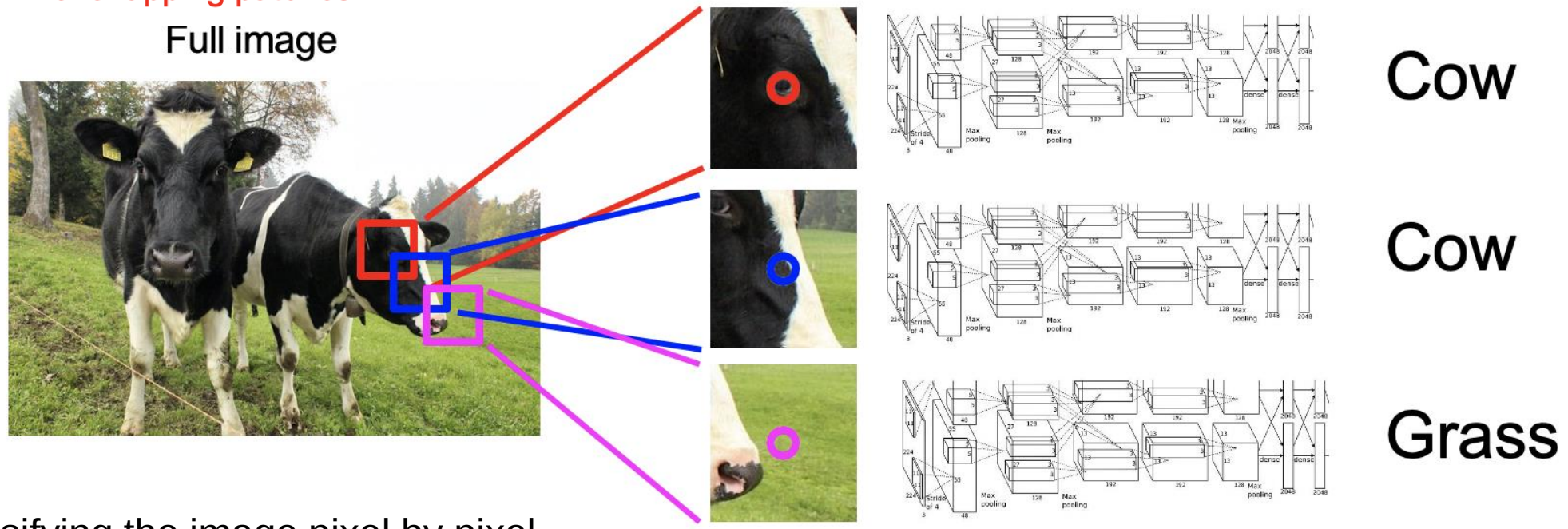
Paired training data: for each training image, each pixel is labeled with a semantic category.

?

At test time, classify each pixel of a new image.

Semantic Segmentation Idea: Sliding Window

Problem: Very inefficient! Not reusing shared features between overlapping patches

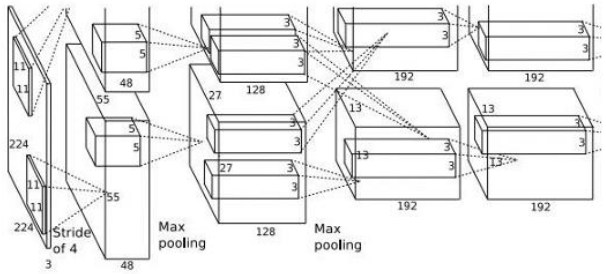


Classifying the image pixel by pixel is impossible without context. Processing small patches give some context for classifying

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

Semantic Segmentation Idea: Convolution

Full image

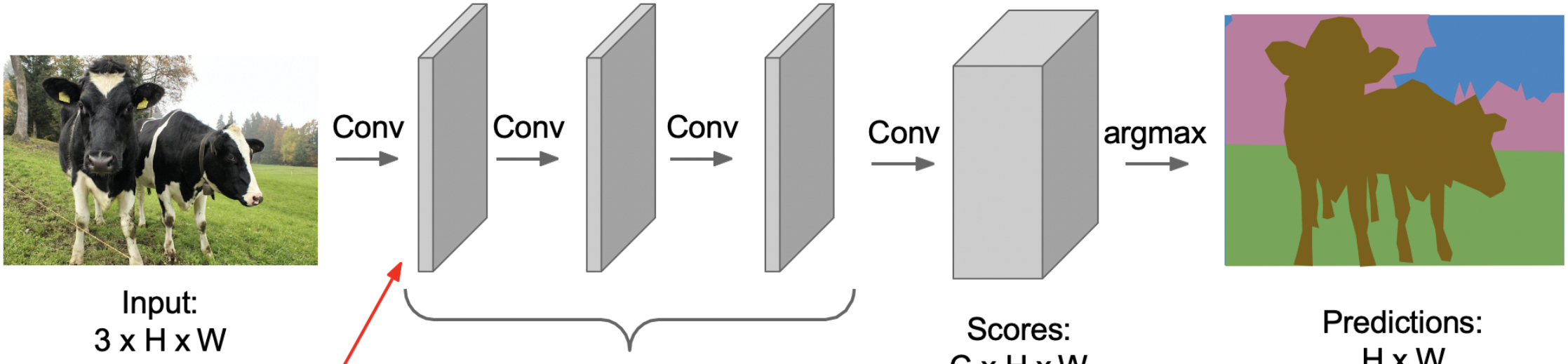


An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

Problem: classification architectures often reduce feature spatial sizes to go deeper, but semantic segmentation requires the output size to be the same as input size.

Semantic Segmentation Idea: Convolution

Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!



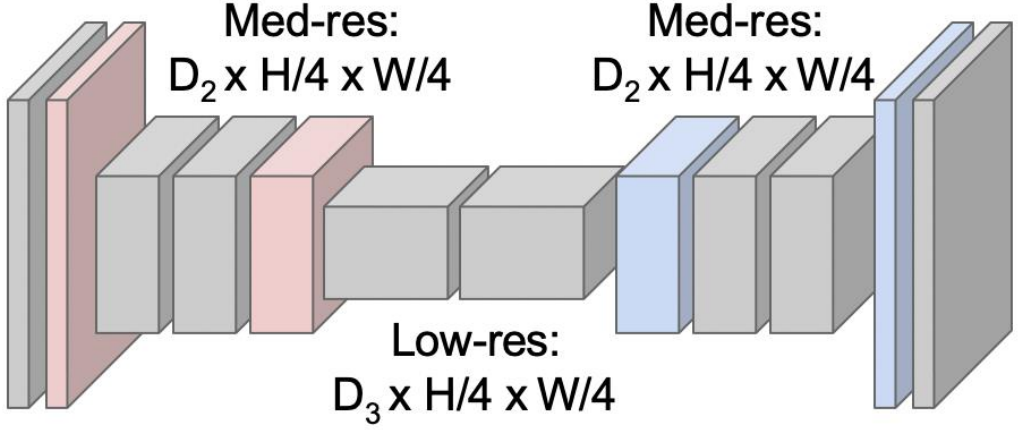
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
???

Input:
3 x H x W



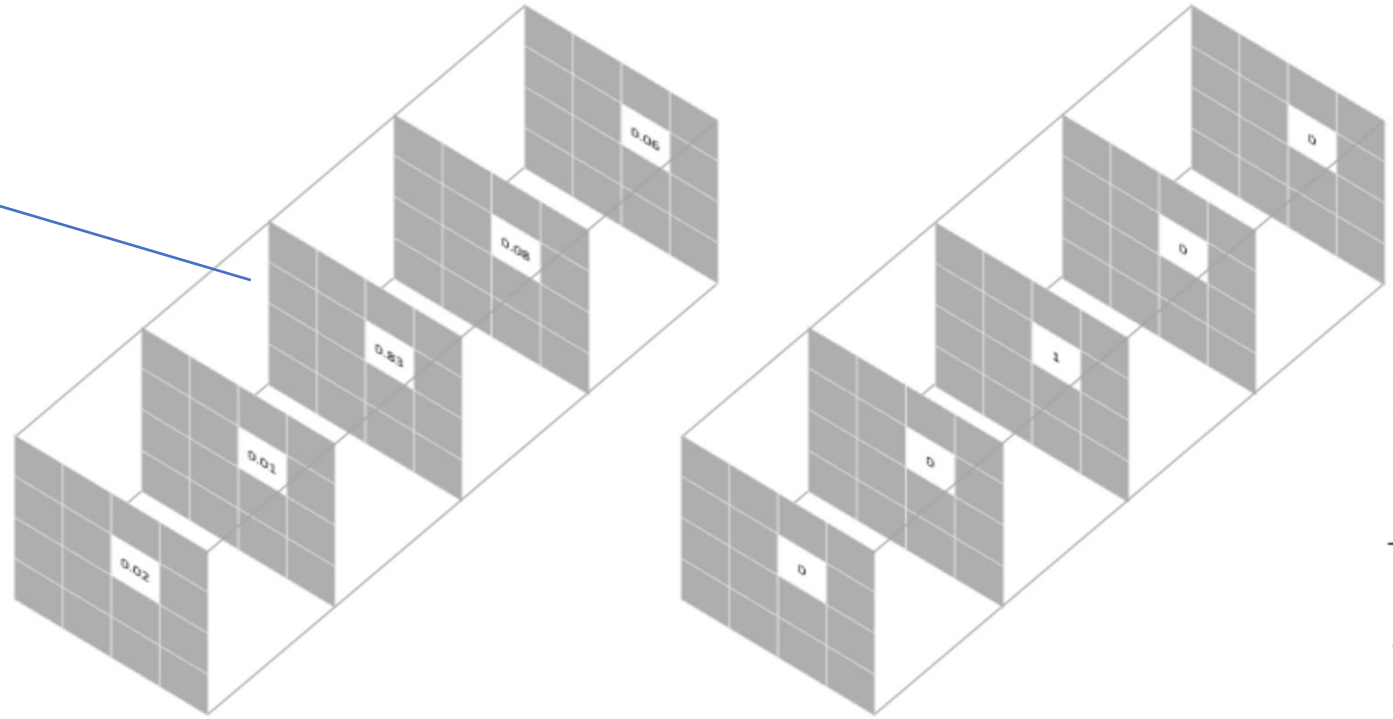
Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Semantic Segmentation Idea: Loss

- The loss function is the sum of pixel-wise multinomial logistic loss or cross entropy value over all the pixels.

C+1 channels for each pixel (including background)



Prediction for a selected pixel

Target for the corresponding pixel

Pixel-wise loss is calculated as the log loss, summed over all possible classes

$$-\sum_{classes} y_{true} \log\left(y_{pred}\right)$$

This scoring is repeated over all **pixels** and averaged

Upsampling – Max Unpooling

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

→ ... → Rest of the network

Max Unpooling
Use positions from pooling layer

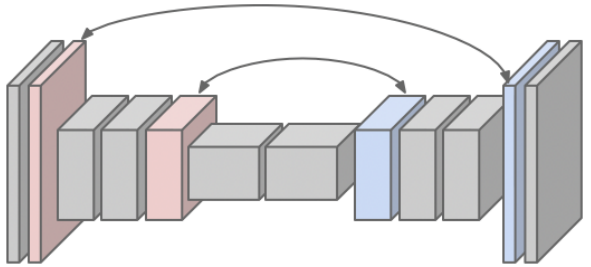
1	2
3	4

Input: 2 x 2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers

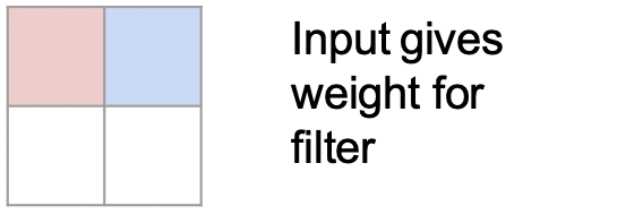


Upsamping - Transposed Convolution

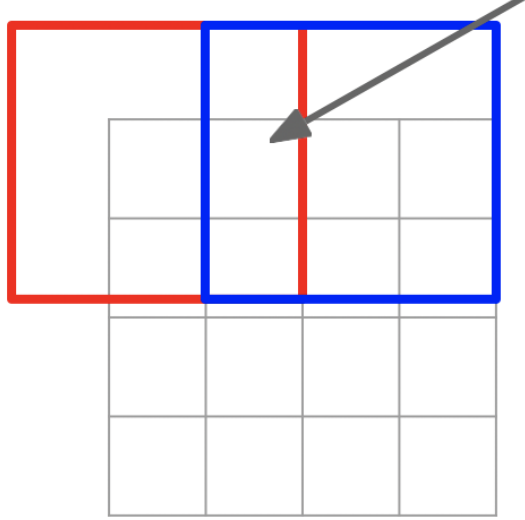
Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

Q: Why is it called transpose convolution?

3 x 3 **transpose** convolution, stride 2 pad 1



Input gives weight for filter



Sum where output overlaps

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Input: 2 x 2

Output: 4 x 4

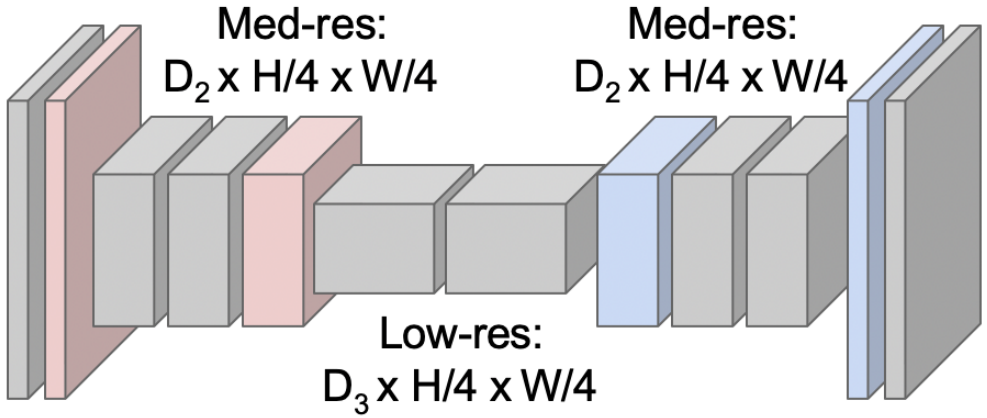
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
Unpooling or strided transpose convolution

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

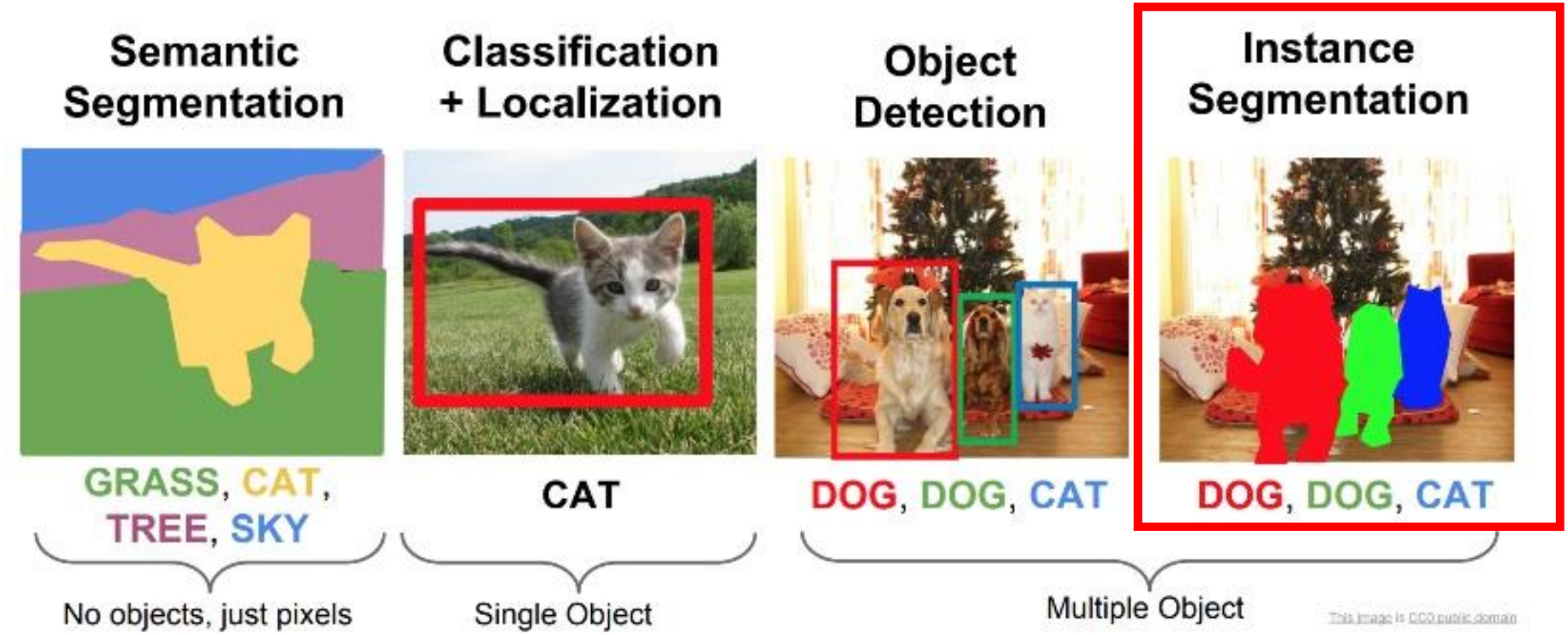
Semantic Segmentation : Evaluation Metrics

Metrics We report four metrics from common semantic segmentation and scene parsing evaluations that are variations on pixel accuracy and region intersection over union (IU). Let n_{ij} be the number of pixels of class i predicted to belong to class j, where there are n_{cl} different classes, and let $t_i = \sum_j n_{ij}$ be the total number of pixels of class i. We compute:

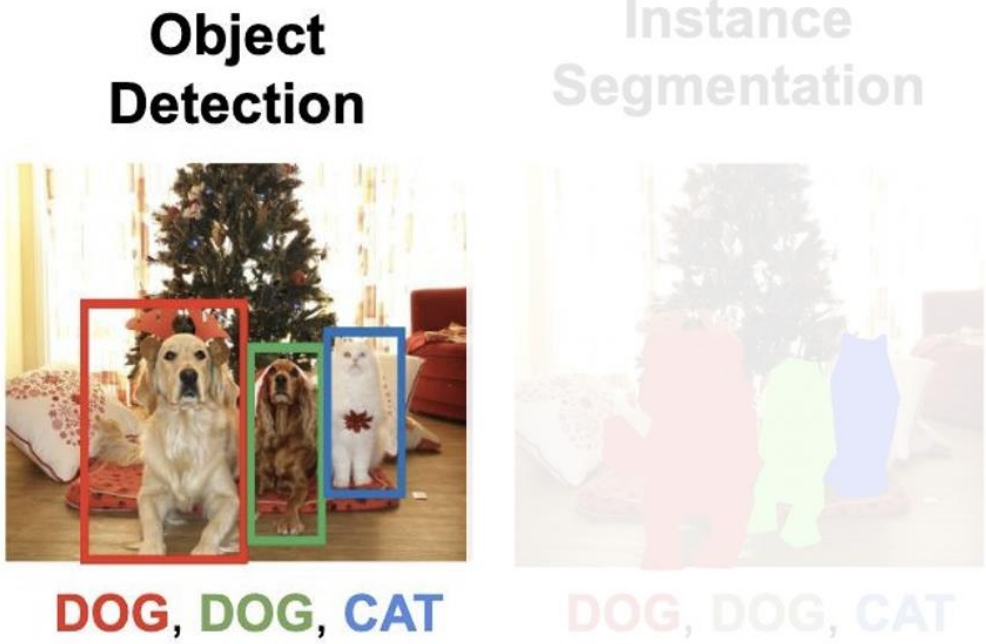
- pixel accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean accuraccy: $(1/n_{\text{cl}}) \sum_i n_{ii}/t_i$
- mean IU: $(1/n_{\text{cl}}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$
- frequency weighted IU: $\left(\sum_k t_k\right)^{-1} \sum_i t_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$

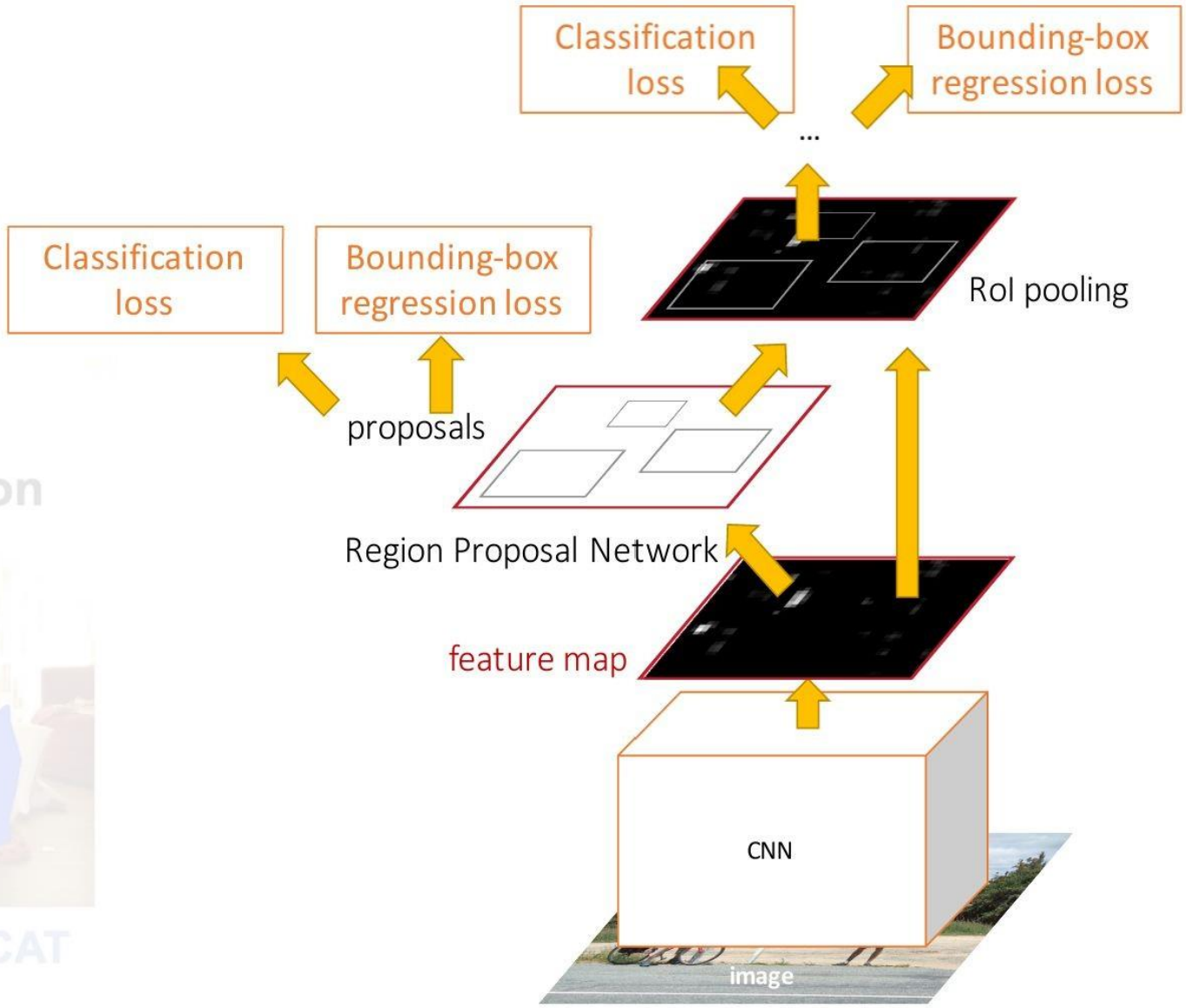
https://arxiv.org/pdf/1411.4038.pdf

Instance Segmentation

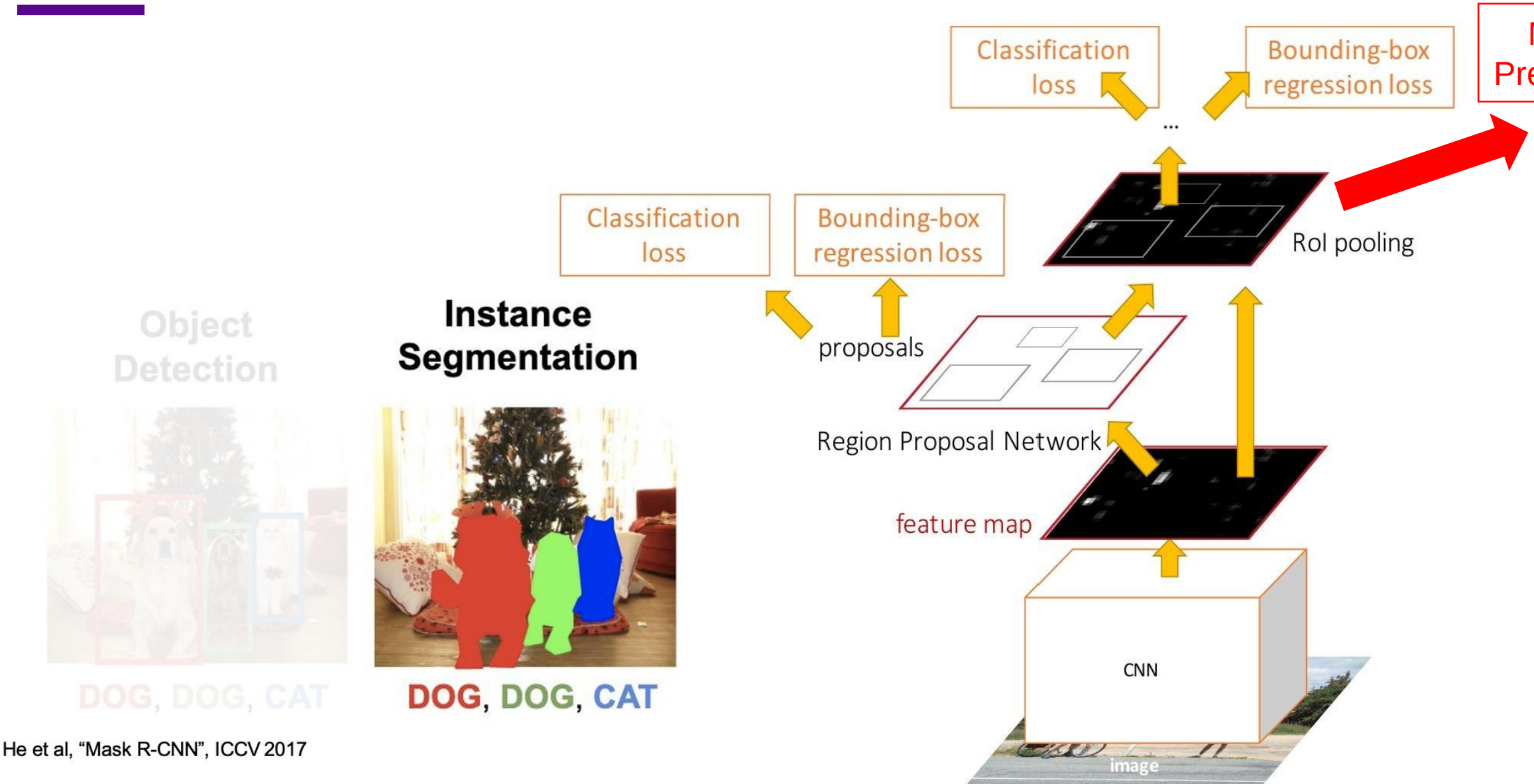


Faster R-CNN- Recall





Instance Segmentation – Mask R-CNN

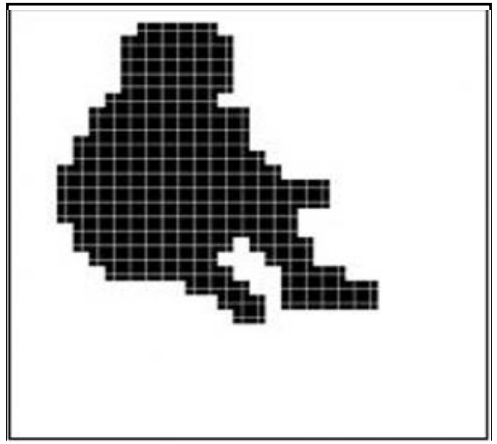


He et al, "Mask R-CNN", ICCV 2017

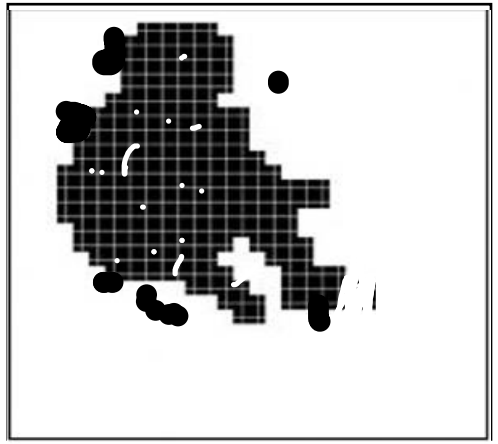
Instance Segmentation – Mask R-CNN Loss

The loss is basically the average of pixel-wise binary cross entropy

Ground truth mask



Predicted mask

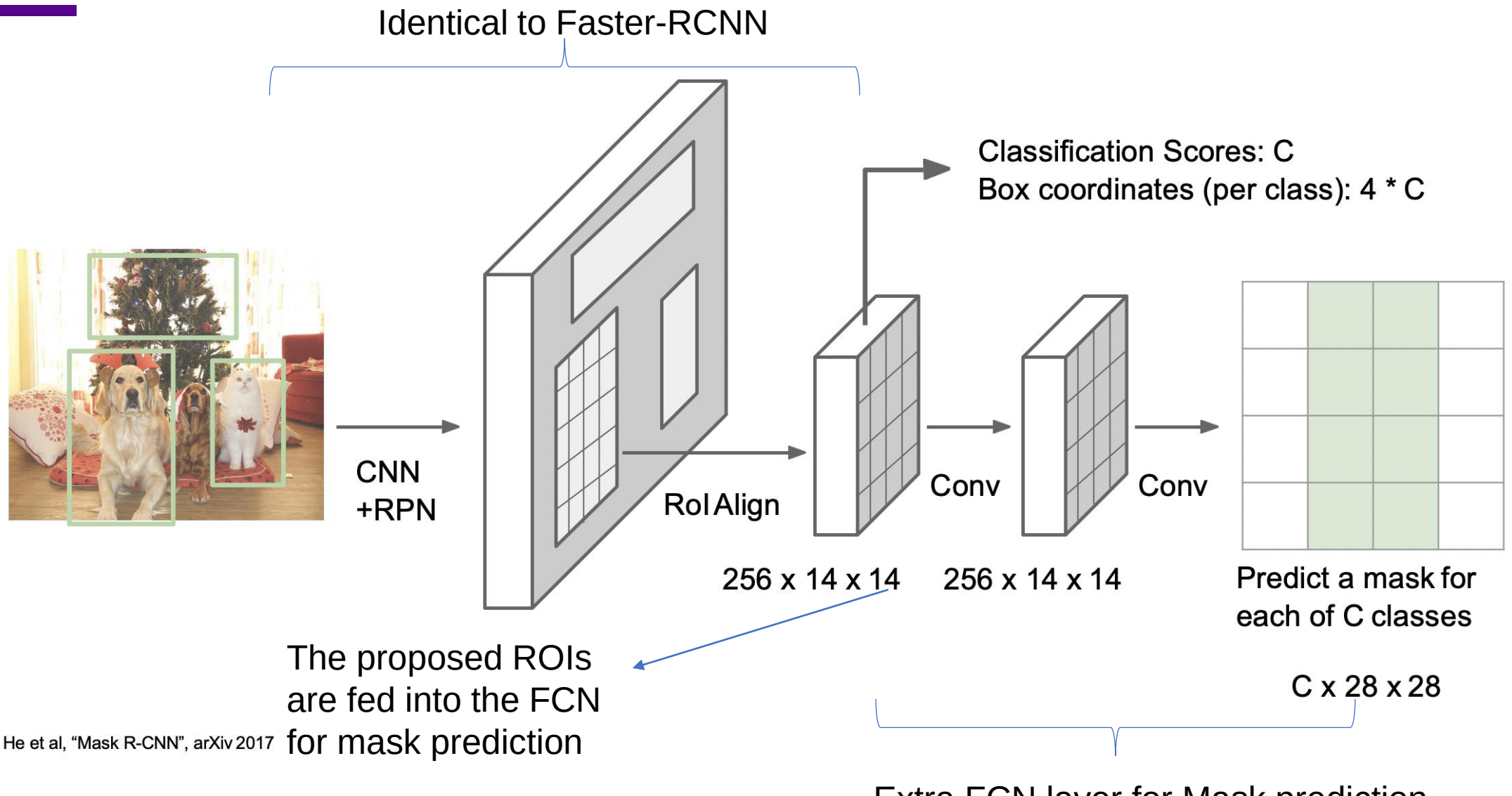


$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} \left[y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k) \right]$$

where y_{ij} is the label of a cell (i, j) in the true mask for the region of size m x m; $\hat{y}_{ij}^k$ is the predicted value of the same cell in the mask learned for the ground-truth class k.

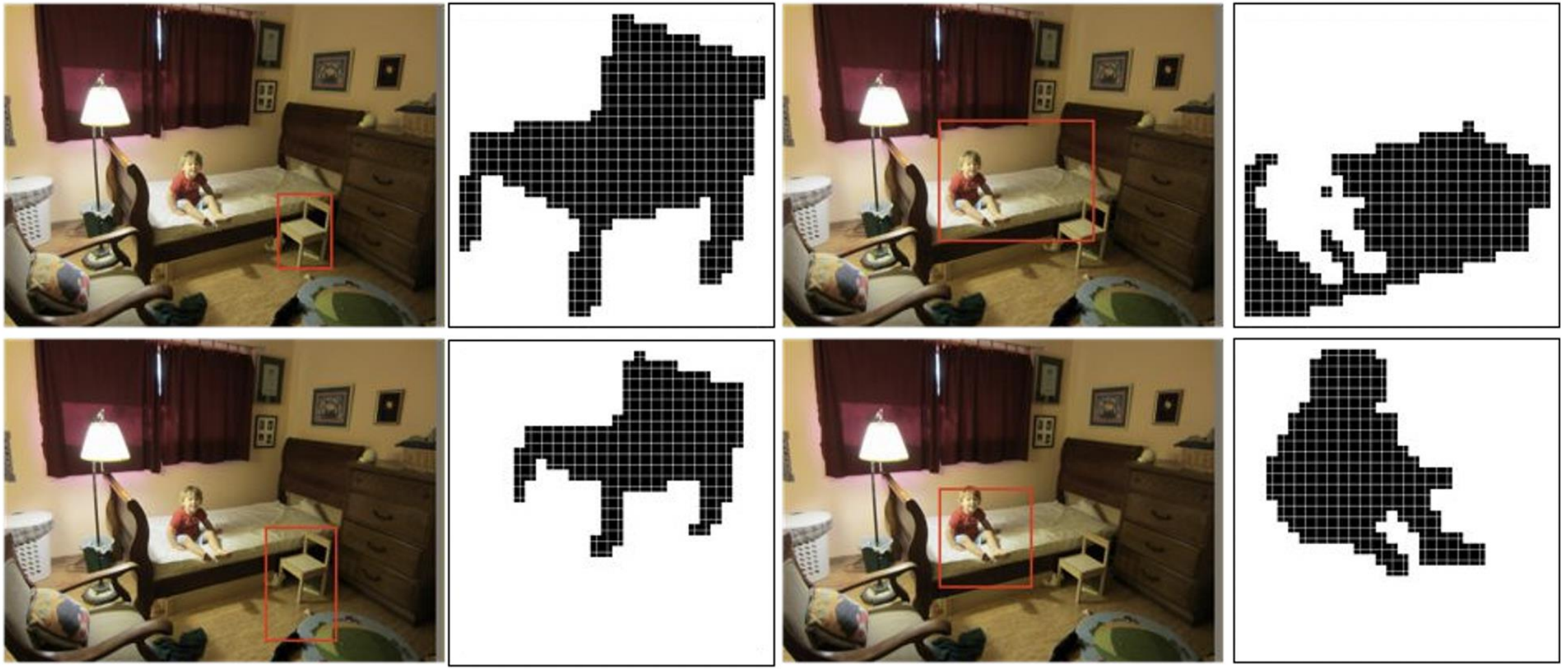
The remaining loss terms are exactly from Faster R-CNN

Mask R-CNN : Architecture



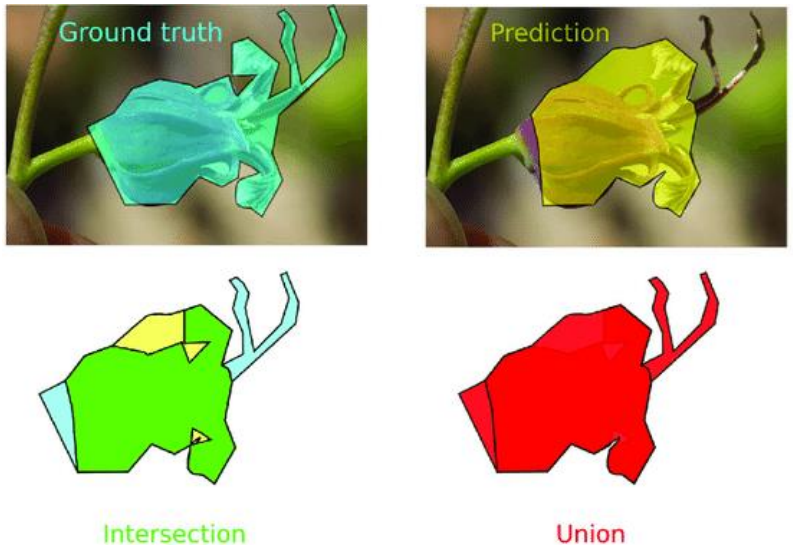
He et al, "Mask R-CNN", arXiv 2017

Mask R-CNN: Example Mask Training Targets



Mask R-CNN: Evaluation

- Mask IoU (Intersection over Union)

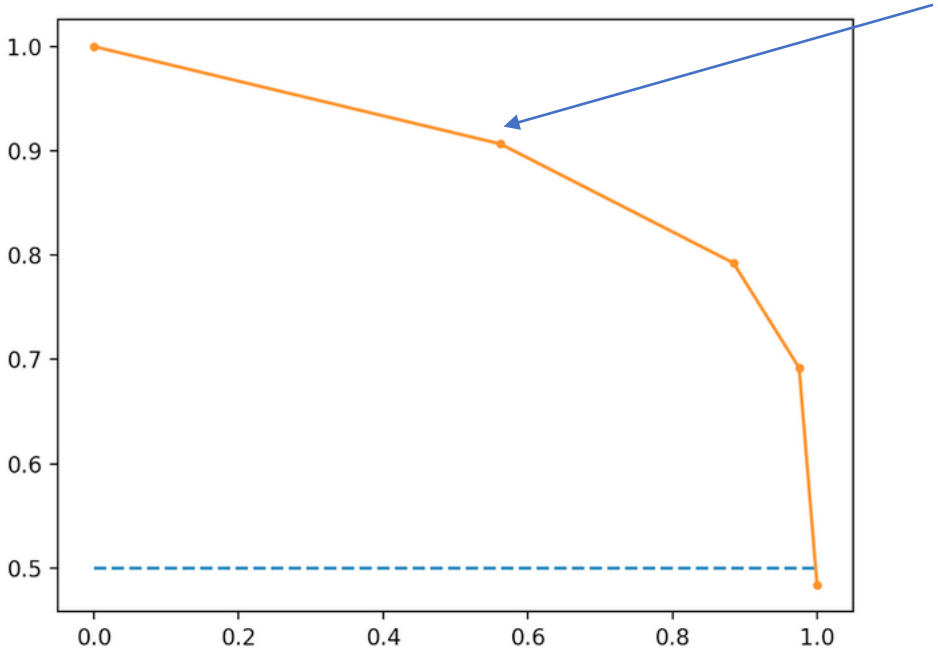


$$IoU = \frac{\text{Intersection}}{\text{Union}}$$

- mAP (Mean average precision)
  - averaged over categories and IoU thresholds

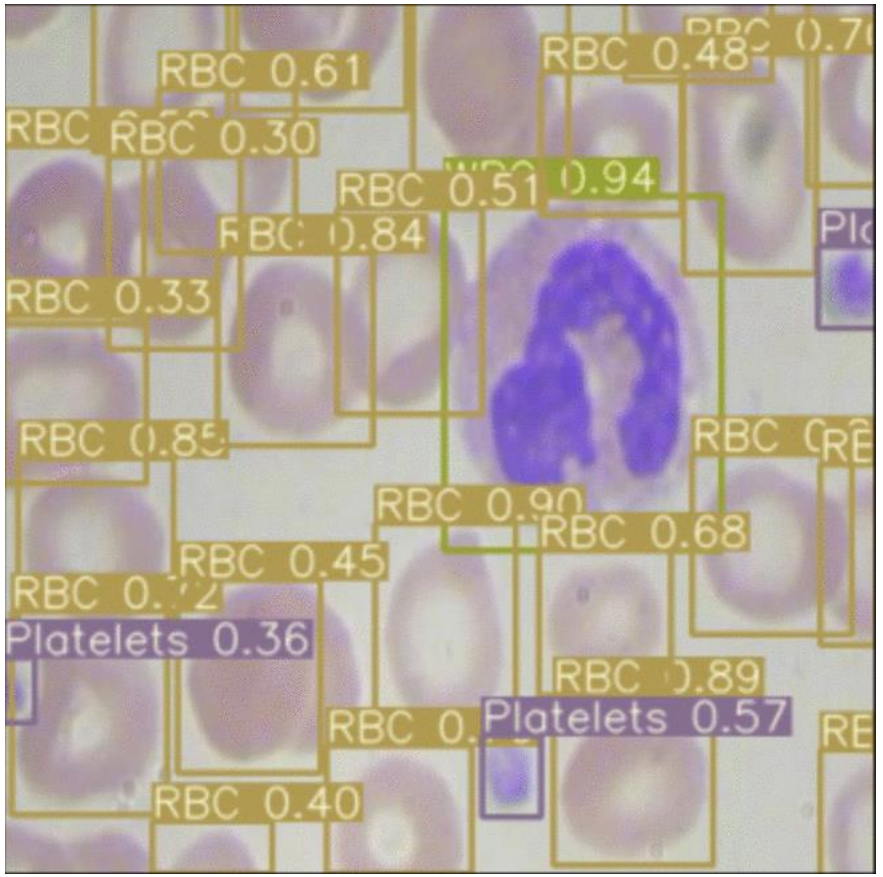
Mean Average Precision

- First, draw precision-recall curve by varying confidence threshold



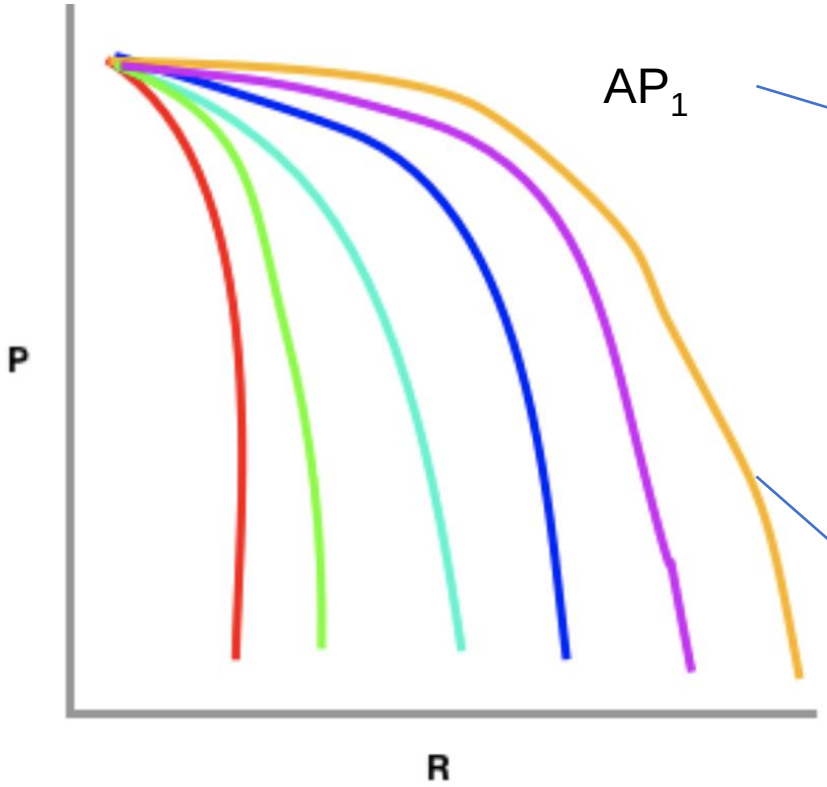
Precision and recall is plotted for each confidence threshold

Average Precision can be computed by weighted (weights = recall value) average of precision values at each confidence threshold



Mean Average Precision

- The precision and recall values depend on the IoU threshold
- Secondly, for each class, draw multiple precision-recall curves, one for each IoU

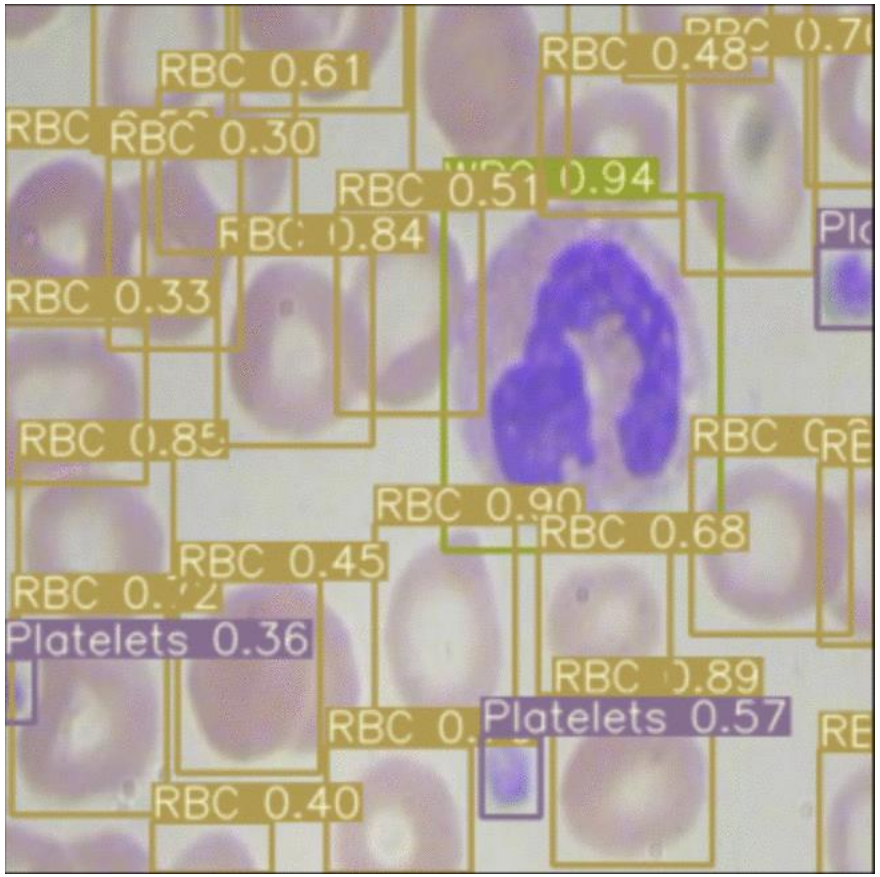


Class - A

AP_1

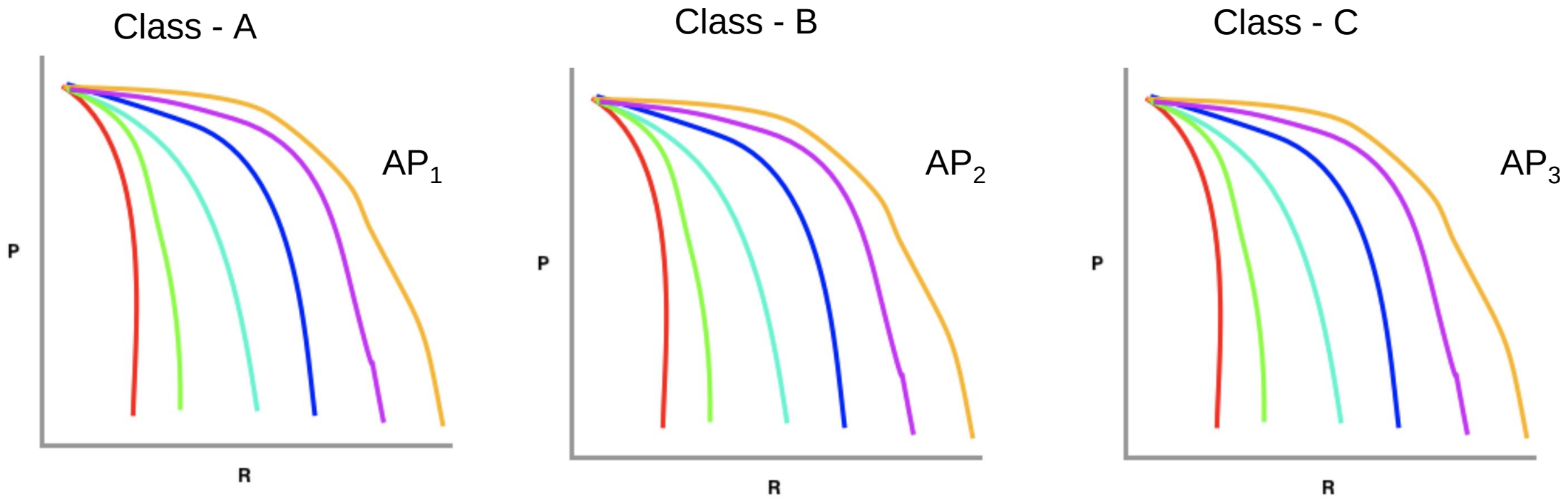
We can compute the AP for this class by averaging the Average precisions for each IoU

Each curve corresponds to a IoU threshold



https://blog.roboflow.com/mean-average-precision/

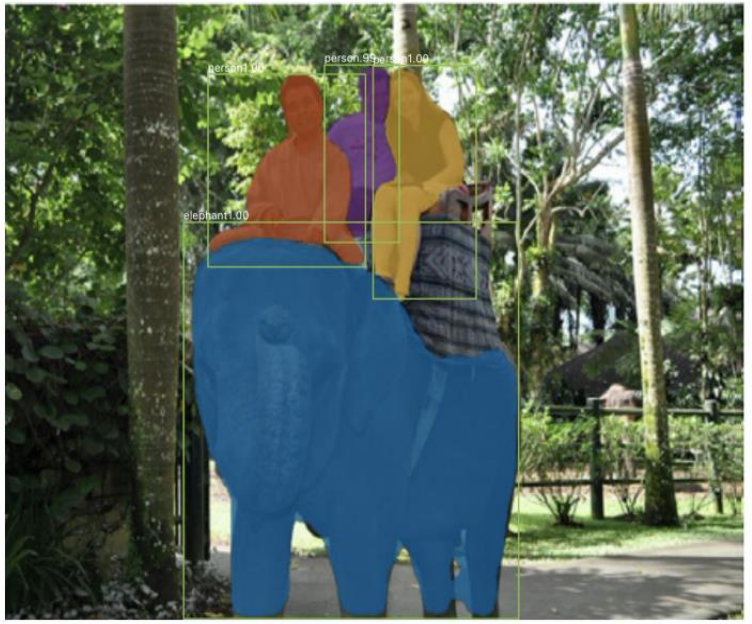
Mean Average Precision

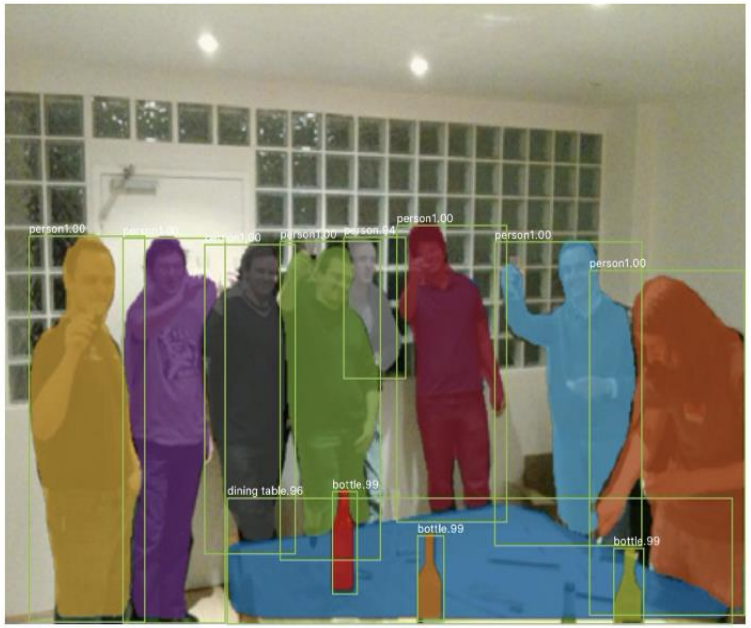


Draw multiple Precision recall curves for each class

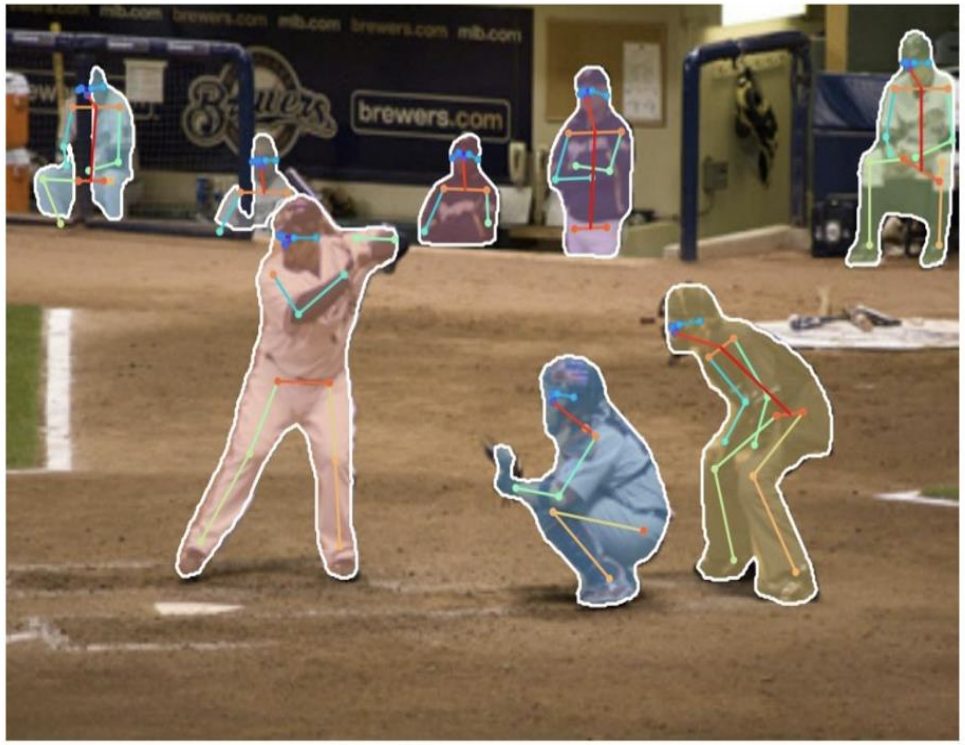
Mean Average Precision = Mean of Average Precisions of each class = 1/3 (AP_1 + AP_2 + AP_3)

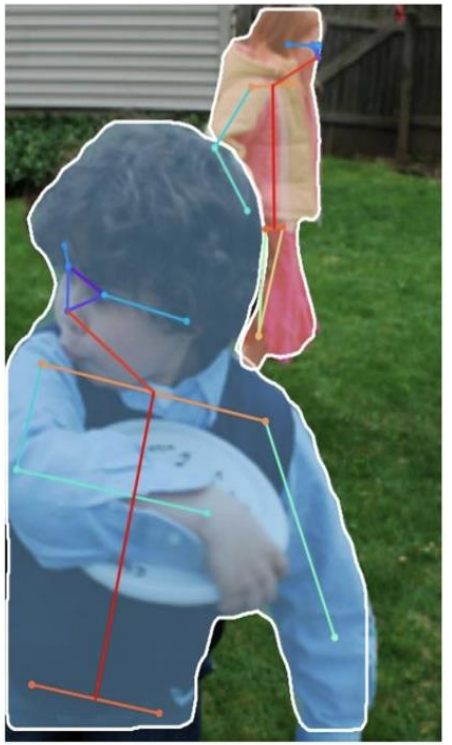
Mask R-CNN : Results





Mask R-CNN : Human Pose Estimation





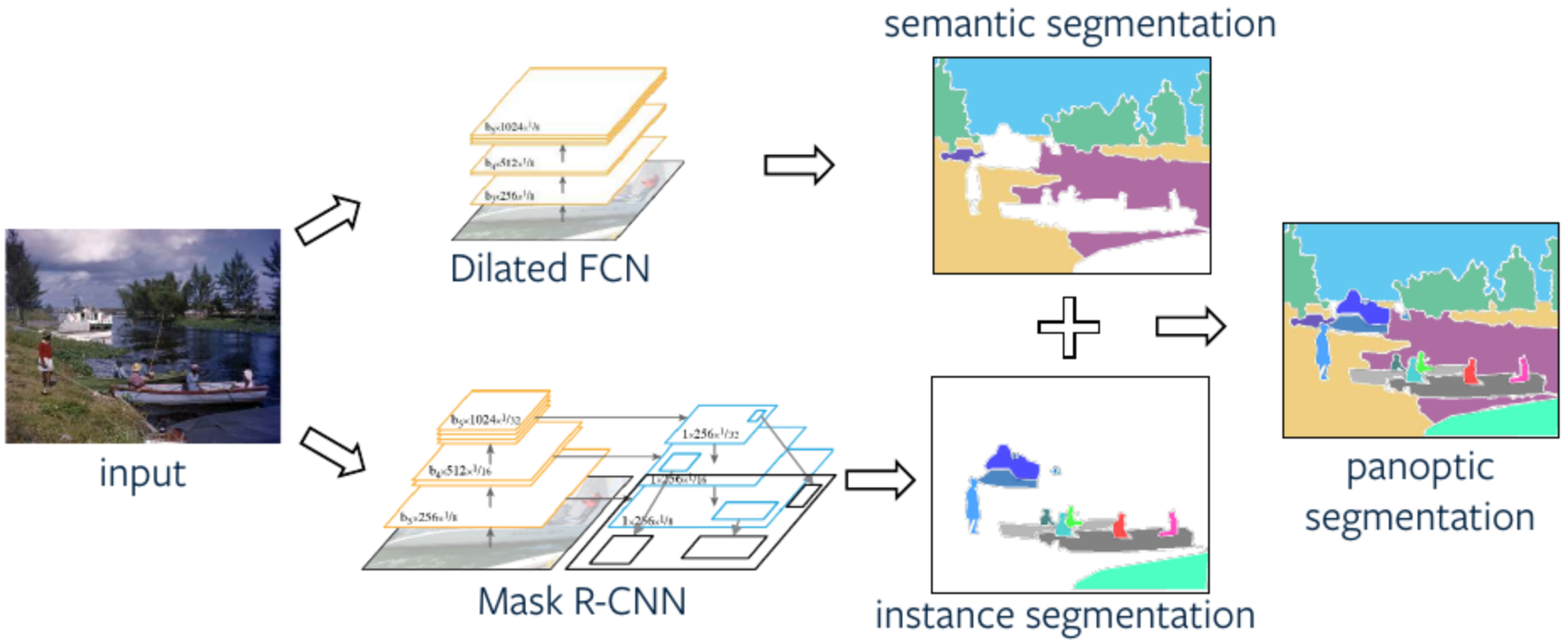
Panoptic Segmentation: Task

- In semantic segmentation, the goal is to classify each pixel into the given classes.
- In instance segmentation, we care about segmentation of the instances of objects separately.
- The panoptic segmentation combines semantic and instance segmentation such that all **pixels are assigned a class label and all object instances are uniquely segmented**.

Left: semantic segmentation, middle: instance segmentation, right: panoptic segmentation

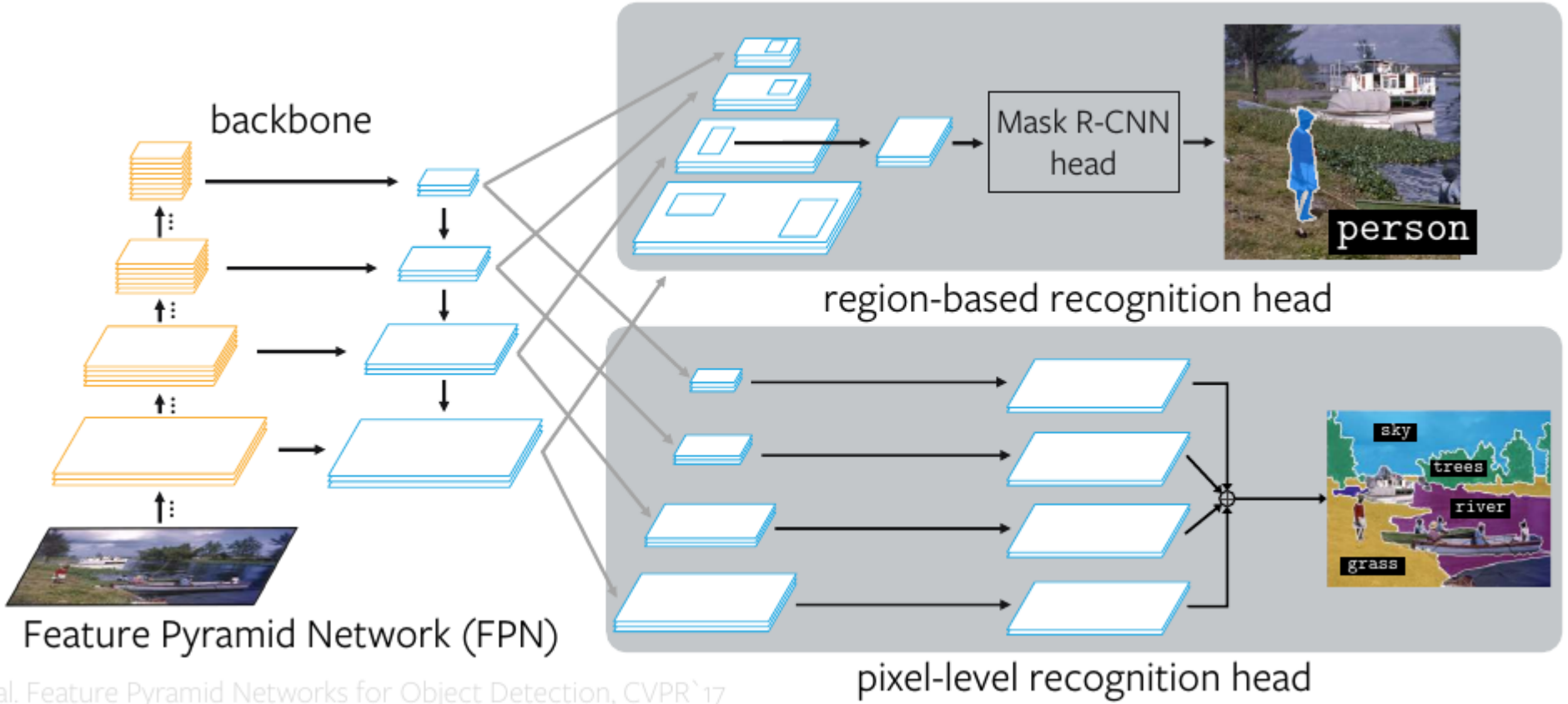
Panoptic Segmentation : Naïve Model

- One of the ways to solve the problem of panoptic segmentation is to **combine the predictions from semantic and instance segmentation models**, e.g. FCN and Mask R-CNN, to get panoptic predictions.



Panoptic Segmentation : Panoptic FPN (Feature Pyramid Network)

- The idea is to use FPN for multi-level feature extraction as backbone, which is to be used for region-based instance segmentation as in case of Mask R-CNN, and add a parallel dense-prediction branch on top of same FPN features to perform semantic segmentation.



al. Feature Pyramid Networks for Object Detection, CVPR`17

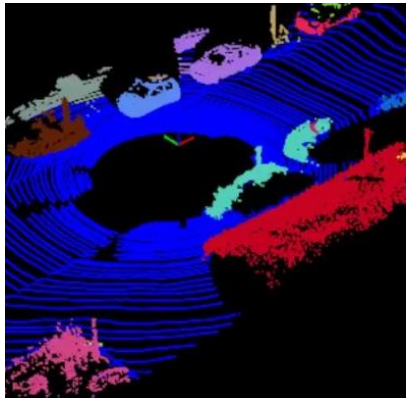
3D Deep Learning

- **Multi-view Convolutional Neural Networks for 3D Shape Recognition**
- **VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition**
- **PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation**
- **FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation**
- **DeepMapping: Unsupervised Map Estimation from Multiple Point Clouds**

Why Deep Learning on 3D Data?

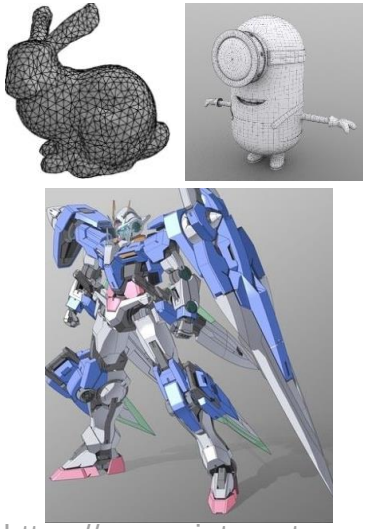
- How to enable robots to create maps and understand scenes in 3D?
- An important form of data – various application domains
- Intrinsically different than images – challenges for existing deep networks

Robotics



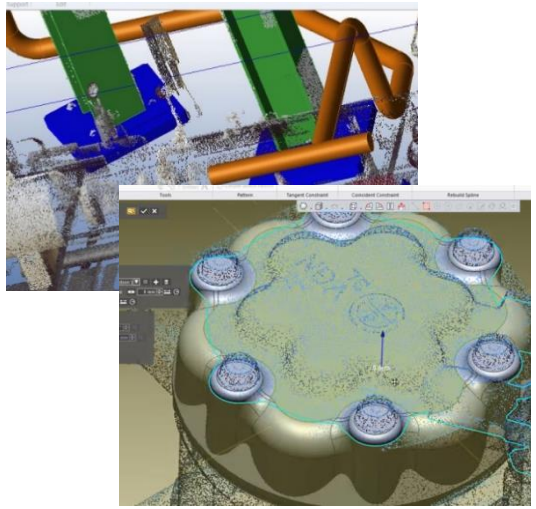
https://www.youtube.com/watch?v=7NNpvtdrHkU

Graphics/3DP



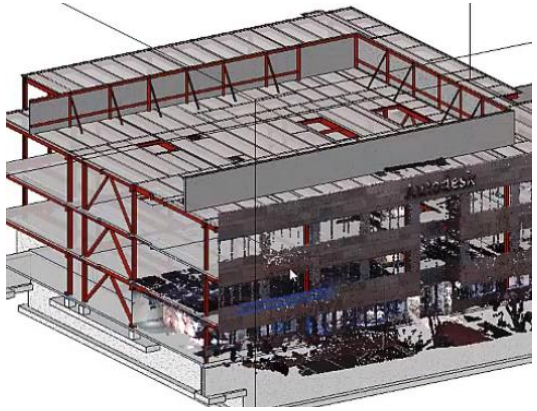
https://www.pinterest.com/pin/134756213823244639/

Mechanical Engineering



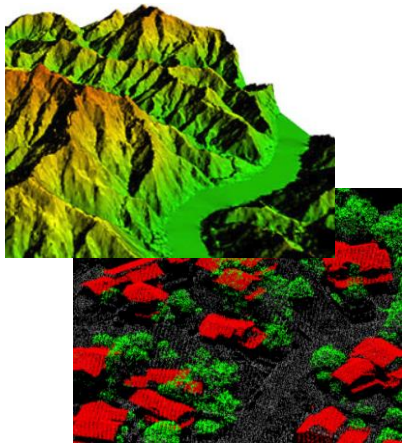
https://www.youtube.com/watch?v=UD4asn3gkNI

Civil Engineering



https://www.youtube.com/watch?v=HhV6LAZ3DN0

Geospatial Science



http://www.aamgroup.com/services-and-technology/aerial-survey

3D Input Representation

Voxel

- ✓ 3D CNN
- • Implicit representation
- × Resolution/Scalability

Multi-view

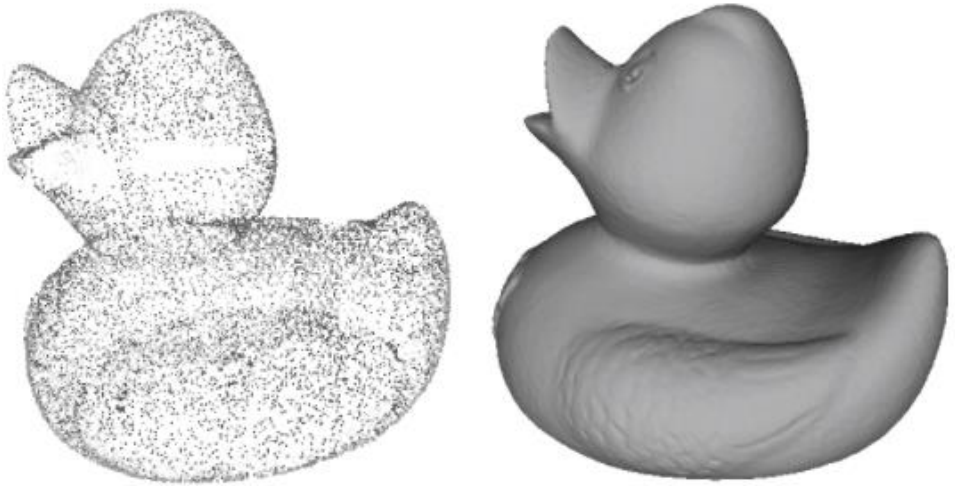
- ✓ 2D CNN
- • Generalize to points?
- × Large networks

Point Cloud/Mesh

- ✓ Raw format/Efficiency
- • Explicit representation
- × Unorganized/Unordered

https://www.planetminecraft.com/project/giant-snowman-1638162/

http://photoboothexpo.com/bullet-time-photo-booths/



https://elmoatazbill.users.greyc.fr/point_cloud/reconstruction.png

Multi-view Convolutional Neural Networks for 3D Shape Recognition

- This paper presents a novel CNN architecture that combines information from multiple views of a 3D shape into a single and compact shape descriptor offering even better recognition performance.
- The network operates on 2D renderings of a 3D object and still performs better than those that directly build on the 3D representations

2D renderings from different view

Multi-view Convolutional Neural Networks – Why ?

- One main reason for working with images is the relative efficiency of the 2D versus the 3D representations.

- For example, 3D ShapeNets use a coarse representation of shape, a **30×30×30 grid of binary voxels**. In contrast a single projection of the 3D model of the same input size corresponds to an image of **164×164 pixels.**

- Another advantage of using 2D representations is that we can leverage

  (i) Advances in image descriptors

  (ii) massive image databases (ImageNet) to pre-train our CNN architectures

Multi-view Convolutional Neural Networks - Approach

- Approach is to learn to combine information from multiple views using a unified CNN architecture that includes a view-pooling layer. All the parameters of the CNN architecture are learned discriminatively to produce a single compact descriptor for the 3D shape.

Capture images of the 3D shape from different views

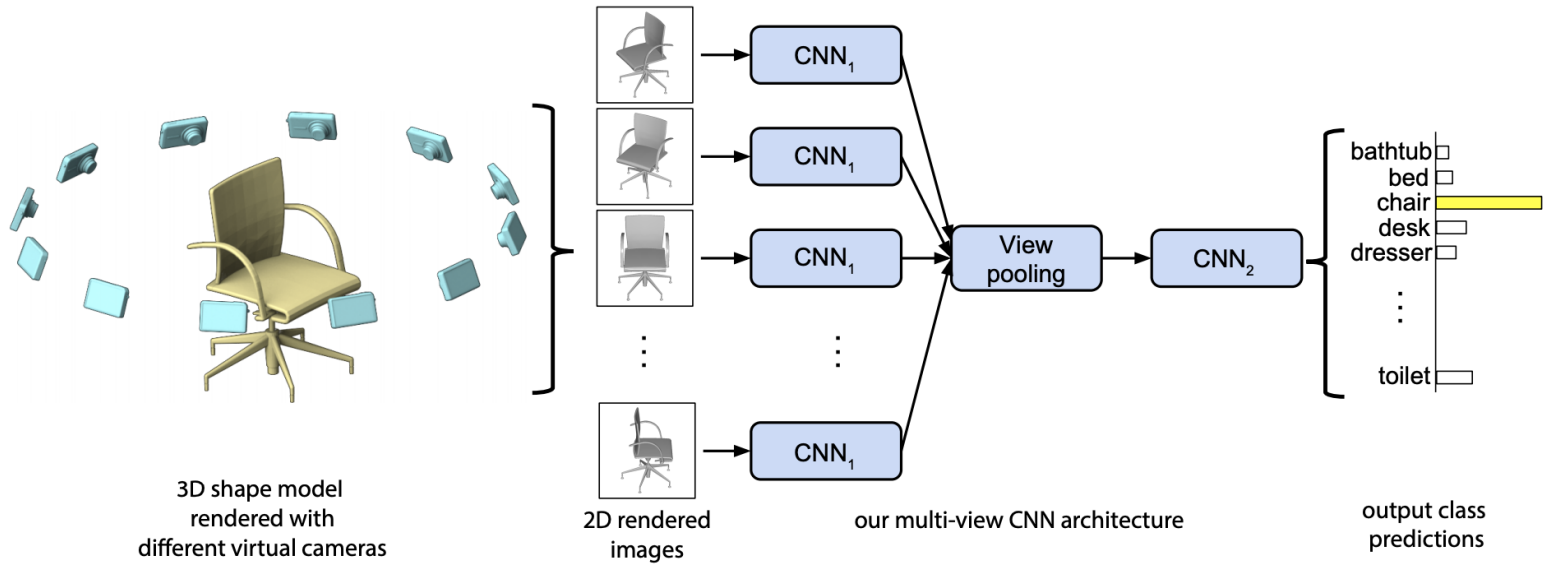


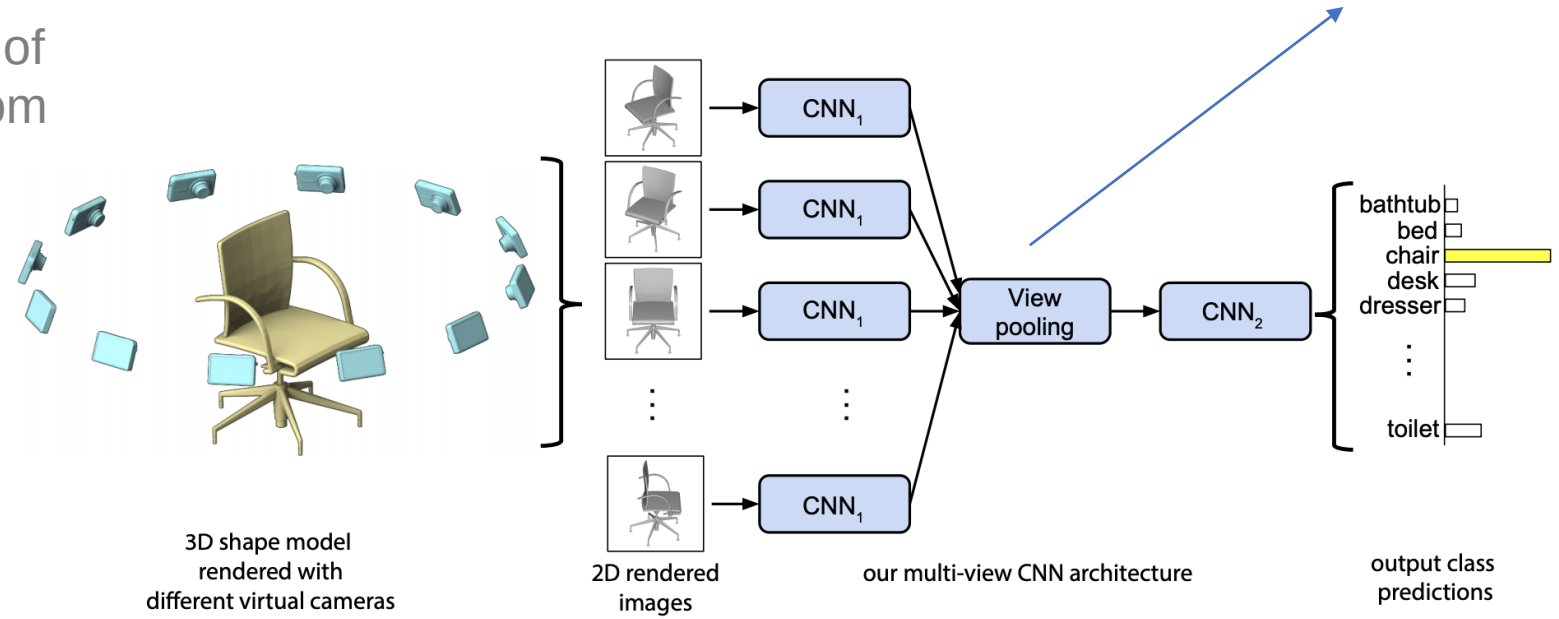
Figure 1. Multi-view CNN for 3D shape recognition (illustrated using the 1st camera setup). At test time a 3D shape is rendered from 12 different views and are passed thorough CNN_1 to extract view based features. These are then pooled across views and passed through CNN_2 to obtain a compact shape descriptor.

Multi-view Convolutional Neural Networks - Approach

Capture images of the 3D shape from different views

Pass the rendered images through common CNN

Element wise pooling across feature maps corresponding to all renderings



Output descriptor is used to classify the shape using SVM

Figure 1. Multi-view CNN for 3D shape recognition (illustrated using the 1st camera setup). At test time a 3D shape is rendered from 12 different views and are passed thorough CNN_1 to extract view based features. These are then pooled across views and passed through CNN_2 to obtain a compact shape descriptor.

Multi-view Convolutional Neural Networks - Results

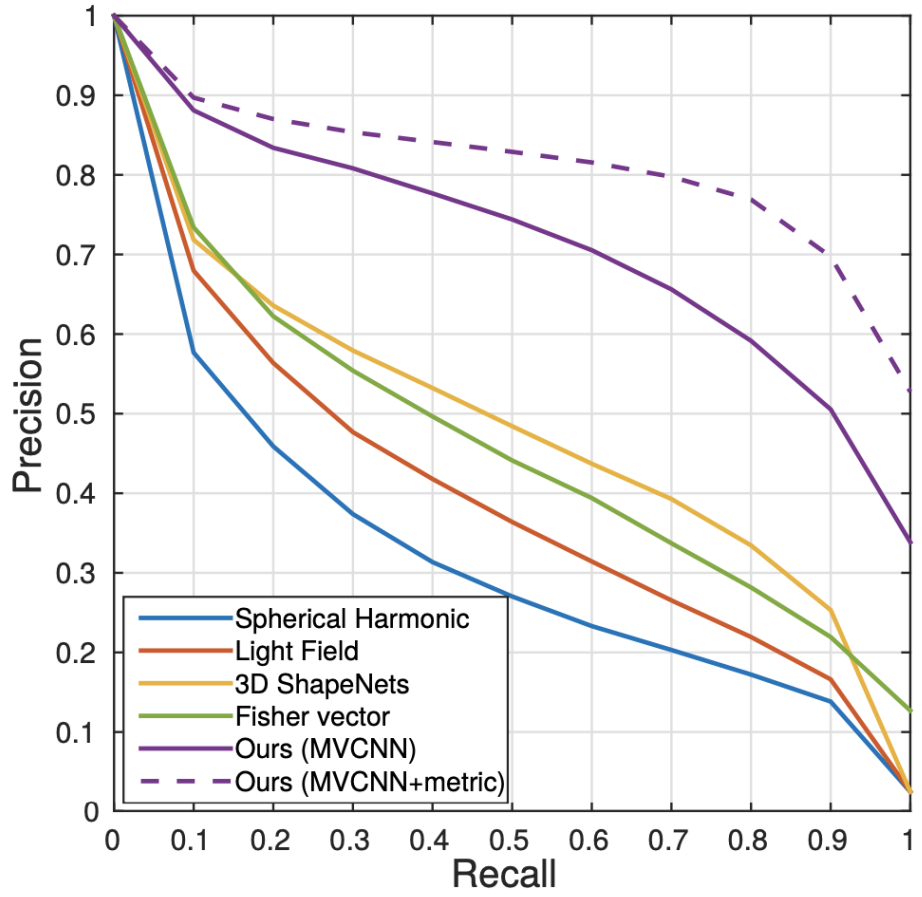


Figure 2. Precision-recall curves for various methods for 3D shape retrieval on the ModelNet40 dataset. Our method significantly outperforms the state-of-the-art on this task achieving 80.2% mAP.

Method	Aug.	Accuracy
(1) FV [30]	-	79.0%
(2) CNN M	-	77.3%
(3) CNN M, fine-tuned	-	84.0%
(4) CNN M, fine-tuned	6×	85.5%
(5) MVCNN M, fine-tuned	6×	86.3%
(6) CNN VD	-	69.3%
(7) CNN VD, fine-tuned	-	86.3%
(8) CNN VD, fine-tuned	6×	86.0%
(9) MVCNN VD, fine-tuned	6×	**87.2%**
(10) Human performance	n/a	93.0%

Table 2. Classification results on SketchClean. Fine-tuned CNN models significantly outperform Fisher vectors [30] by a significant margin. MVCNNs are better than CNN trained with data jittering. The results are shown with two different CNN architectures – VGG-M (row 2-5) and VGG-VD (row 6-9).

Multi-view Convolutional Neural Networks – Gradient Visualization

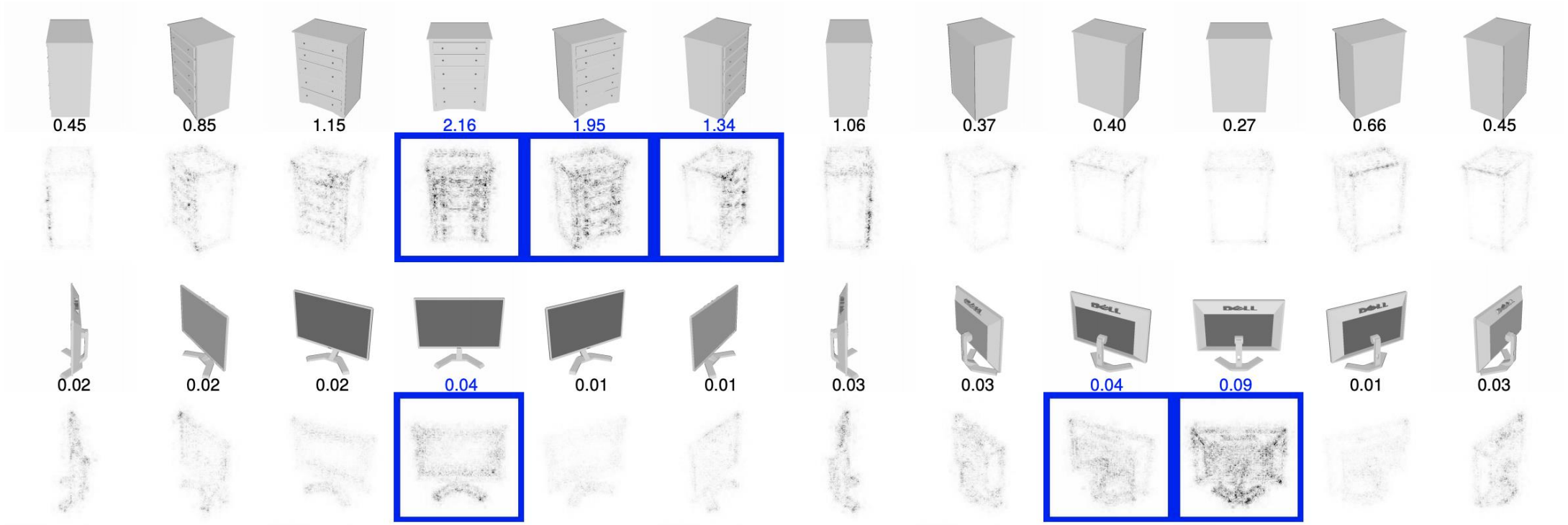


Figure 3. Top three views with the highest saliency are highlighted in blue and the relative magnitude of gradient energy for each view is shown on top. The saliency maps are computed by back-propagating the gradients of the class score onto the images via the view-pooling layer. Notice that the handles of the dresser are the most discriminative features. (Figures are enhanced for visibility.)

VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition

- Range sensors such as LiDAR and RGBD cameras are increasingly found in modern robotic systems, providing a rich source of 3D information that can aid in this task.

- However, many current systems do not fully utilize this information and have trouble efficiently dealing with large amounts of point cloud data.

- In this paper, VoxNet is introduced as an architecture to tackle this problem by integrating a Volumetric Occupancy Grid representation with a supervised 3D Convolutional Neural Network (3D CNN).

Input Representation : Volumetric Occupancy Grid

- Occupancy grids represent the state of the environment as a **3D lattice of random variables** (each corresponding to a voxel) and maintain a probabilistic estimate of their occupancy as a function of incoming sensor data and prior knowledge.

- Reason for Using Occupancy Grid as Input Representation

  (i) They allow us to **efficiently estimate** free, occupied and unknown space from range measurements.

  (ii) They can be stored and manipulated with simple and **efficient data structures.**

3D Convolutional Network Layers

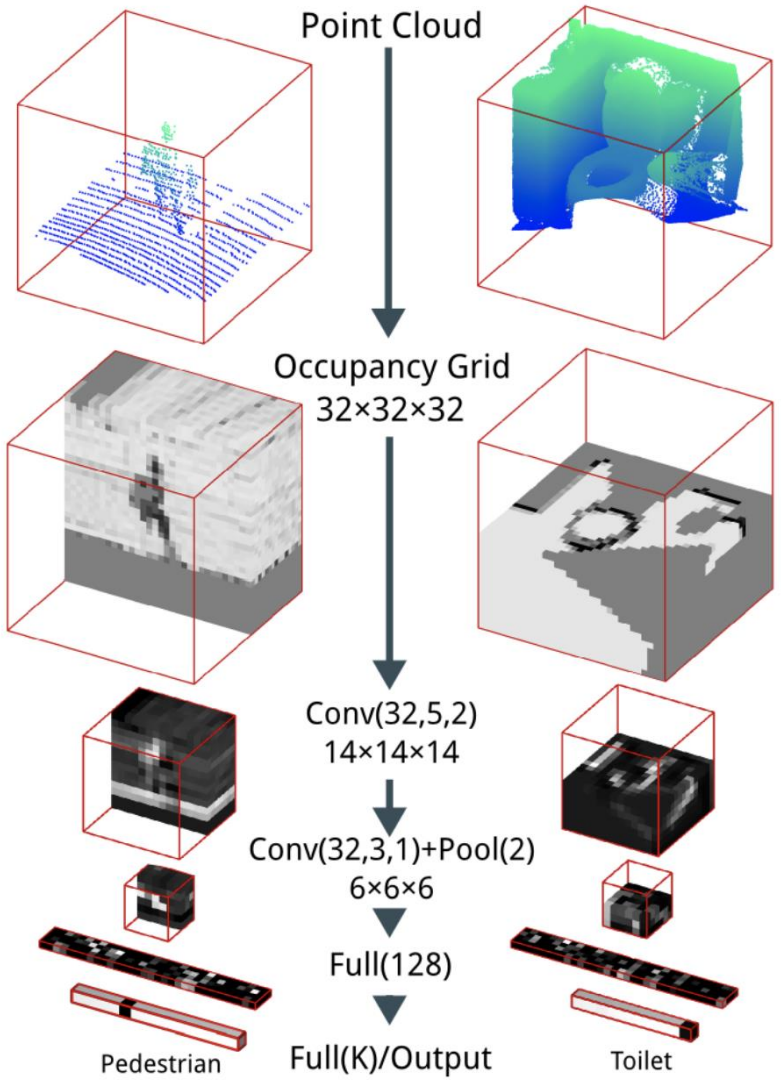
- Input Layer : This layer accepts a fixed-size grid of I×J×K voxels. In this work, we use I = J = K = 32

- Convolutional Layers : These layers accept 4-dimensional input volumes in which three of the dimensions are spatial, and the fourth contains the feature maps. The layer creates f feature maps by convolving the input with f learned filters of shape d × d × d × f ’ , where d are the spatial dimensions and f’ is the number of input feature maps.

- Pooling Layers : These layers downsample the input volume by a factor of m along the spatial dimensions

- Fully Connected Layer : Fully connected layers have n output neurons. The output of each neuron is a learned linear combination of all the outputs from the previous layer, passed through a nonlinearity.

Network Architecture



Probabilistic estimate of occupancy

Sequence of range measurements that either hit or passes through a given voxel with coordinates (i,j,k)

$$l_{ijk}^{t} = l_{ijk}^{t-1} + z^{t} l_{\text{occ}} + (1 - z^{t}) l_{\text{free}} \qquad (1)$$

where l_{occ} and l_{free} are the log odds of the cell being occupied or free given that the measurement hit or missed the cell,

Filters at the input level encode spatial structures such as planes and corners at different orientations.

The output of the network is probabilities for each class. **Loss is Multinomial negative log-likelihood and Evaluation metric is Accuracy**

Visualizations and Results

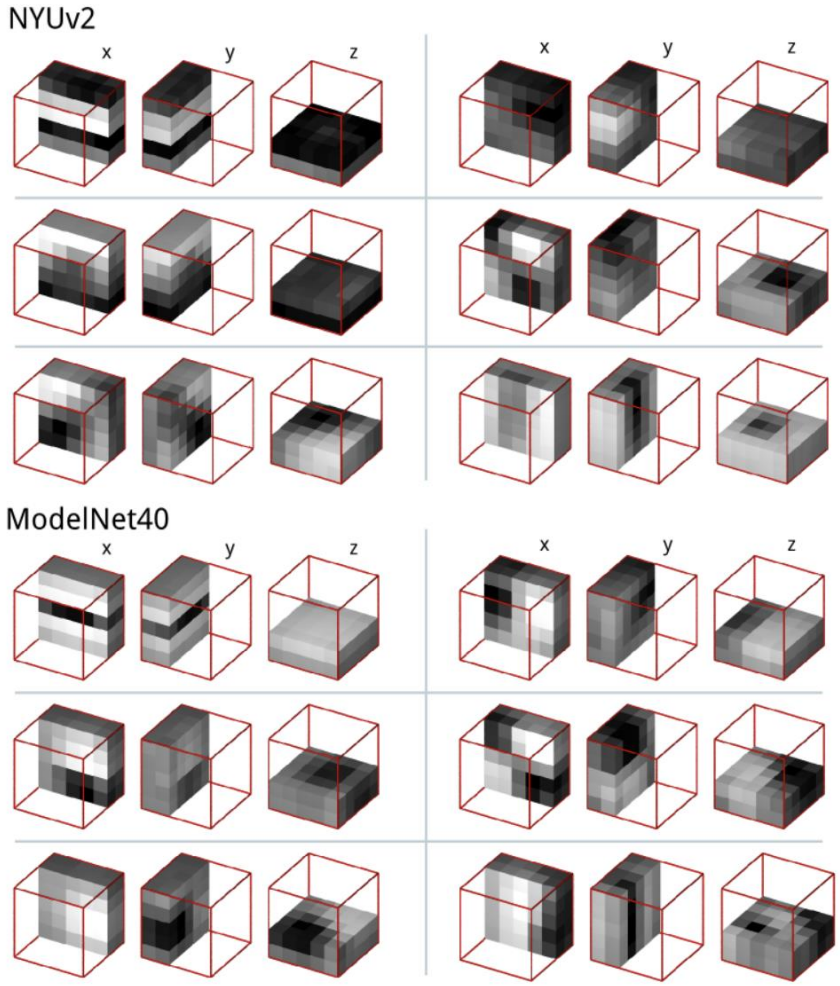


Fig. 4. Cross sections along the x, y and z axes of selected first layer filters learned in the ModelNet40 and NYUv2 datasets.

COMPARISON WITH OTHER METHODS IN SYDNEY OBJECT DATASET

Method	Avg F1
UFL+SVM[21]	0.67
GFH+SVM[37]	0.71
Multi Resolution VoxNet	**0.73**

TABLE IV

COMPARISONS WITH SHAPENET IN MODELNET (AVG ACC)

Dataset	ShapeNet	VoxNet
ModelNet10	0.84	**0.92**
ModelNet40	0.77	**0.83**

TABLE V

COMPARISON WITH SHAPENET IN NYUV2 (AVG ACC)

Dataset	ShapeNet	VoxNet	VoxNet Hit
NYU	0.58	**0.71**	0.70
ModelNet10→NYU	**0.44**	0.34	0.25

PointNet - What is a Point Cloud?

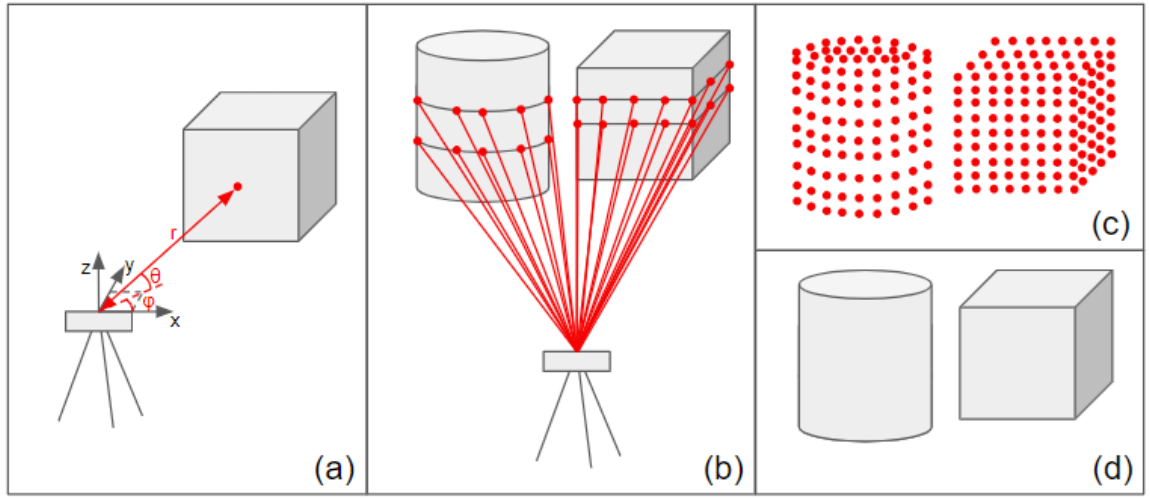


Figure 1: A point cloud from a laser scanner. See body text for more explanation.

A Trimble 3D scanning unit sweeps an area to obtain point cloud data.

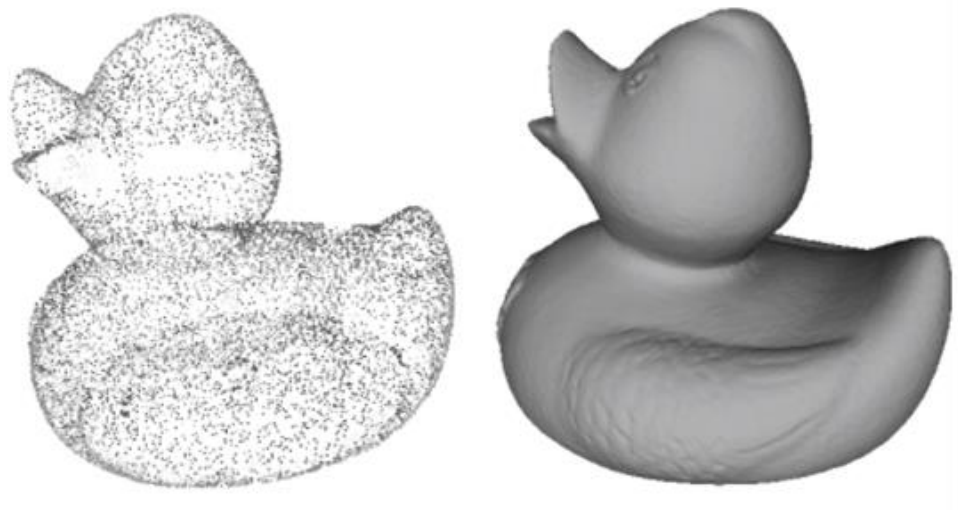
- A point cloud is nothing more than a collection of millions (sometimes billions) of points coming from a scanner. (Unorganized)
- The points only represent the surfaces of scanned objects
- Typical Scanners: 2D or 3D Lidar (Laser Scanner), Structure Sensor (1st Gen Kinect)

https://blog.bricsys.com/point-clouds-whats-the-point/

PointNet - What is Point Cloud?

Point Cloud/Mesh

- ✓ Raw format/Efficiency
- • Explicit representation
- × Unorganized/Unordered



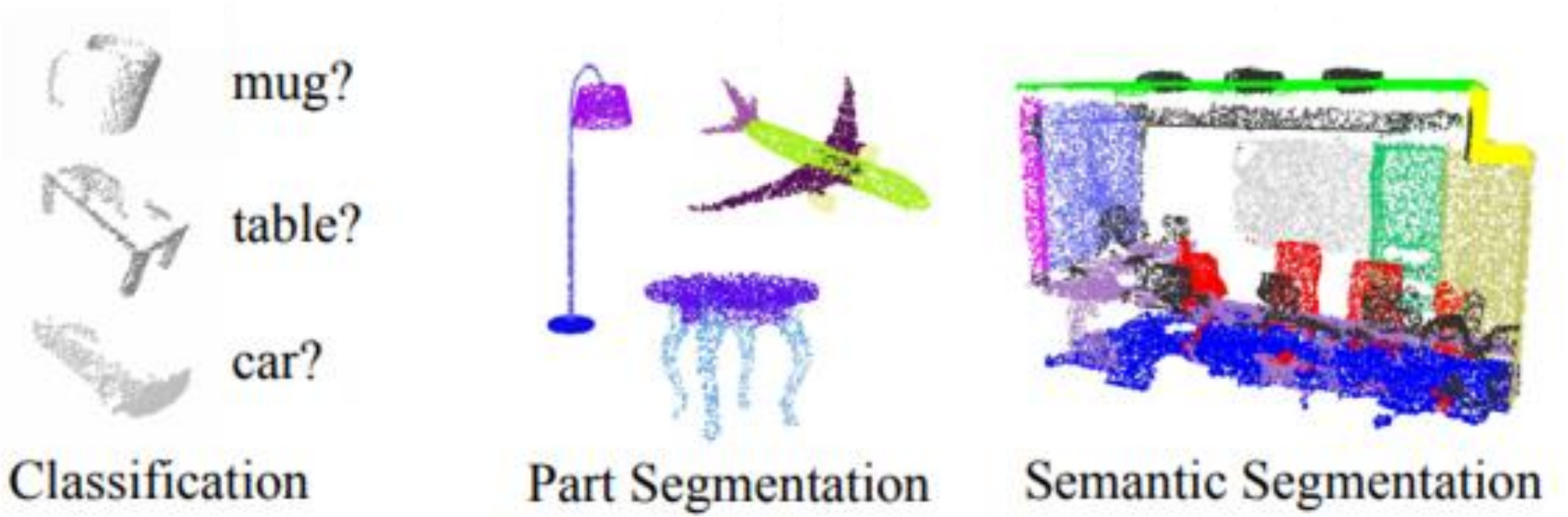
https://elmoatazbill.users.greyc.fr/point_cloud/reconstruction.png

```
ply
format ascii 1.0
element vertex 75463
property float x
property float y
property float z
property float nx
property float ny
property float nz
end_header
-472.25 -377.93 579.102 922.061 74.0133 -379.902
-471.68 -377.93 580.412 906.271 50.2959 -419.695
-468.945 -383.789 584.961 918.904 65.9779 -388.924
-466.14 -377.93 590.82 878.145 37.8206 -476.897
-469.165 -377.93 584.961 881.953 13.8531 -471.134
```

Point cloud format example:

- coordinate of each point (x, y, z)
- normal coordinate of each point (nx, ny, nz) (if have)

Machine Learning Tasks on Point Cloud Data



https://arxiv.org/pdf/1612.00593.pdf

PointNet: A Framework to Process Point Cloud with MLP

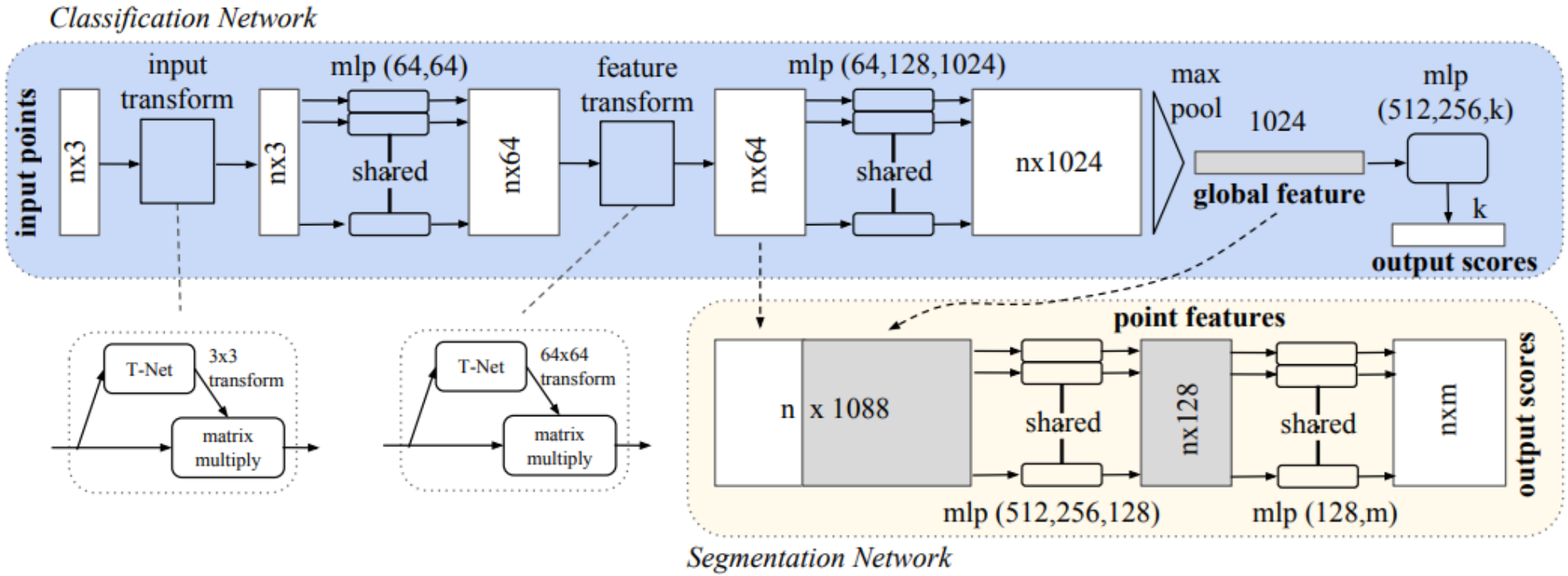


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

https://arxiv.org/pdf/1612.00593.pdf

Results of 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.

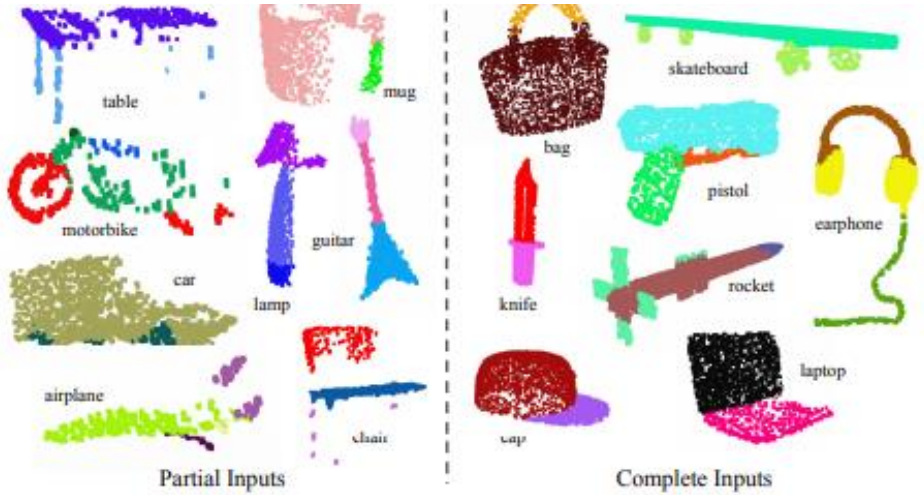


Figure 3. **Qualitative results for part segmentation.** We visualize the CAD part segmentation results across all 16 object categories. We show both results for partial simulated Kinect scans (left block) and complete ShapeNet CAD models (right block).

Results of PointNet

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [27]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [29]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9	**92.0**	85.4	**82.5**	**95.7**	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	**83.7**	**83.4**	**78.7**	**82.5**	74.9	**89.6**	**73.0**	91.5	**85.9**	80.8	95.3	65.2	**93.0**	81.2	**57.9**	**72.8**	**80.6**

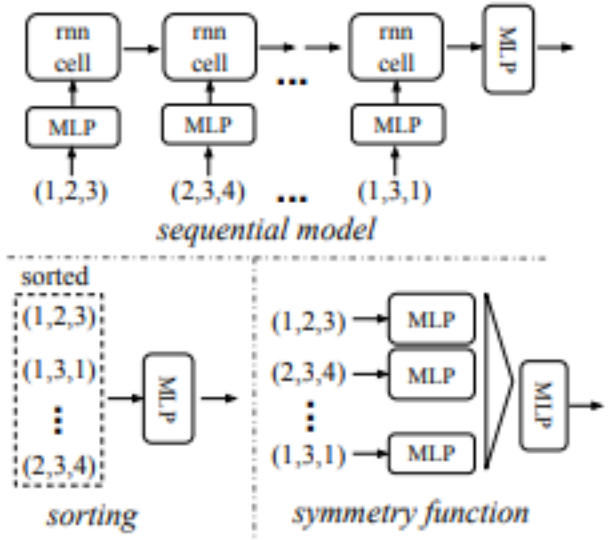
Table 2. **Segmentation results on ShapeNet part dataset.** Metric is mIoU(%) on points. We compare with two traditional methods [27] and [29] and a 3D fully convolutional network baseline proposed by us. Our PointNet method achieved the state-of-the-art in mIoU.

Figure 4. **Qualitative results for semantic segmentation.** Top row is input point cloud with color. Bottom row is output semantic segmentation result (on points) displayed in the same camera viewpoint as input.

	mean IoU	overall accuracy
Ours baseline	20.12	53.19
Ours PointNet	**47.71**	**78.62**

Table 3. **Results on semantic segmentation in scenes.** Metric is average IoU over 13 classes (structural and furniture elements plus clutter) and classification accuracy calculated on points.

How Does PointNet Achieve Point Order Invariance?



	accuracy
MLP (unsorted input)	24.2
MLP (sorted input)	45.0
LSTM	78.5
Attention sum	83.0
Average pooling	83.8
Max pooling	**87.1**

Figure 5. **Three approaches to achieve order invariance.** Multi-layer perceptron (MLP) applied on points consists of 5 hidden layers with neuron sizes 64,64,64,128,1024, all points share a single copy of MLP. The MLP close to the output consists of two layers with sizes 512,256.

Visualizing PointNet

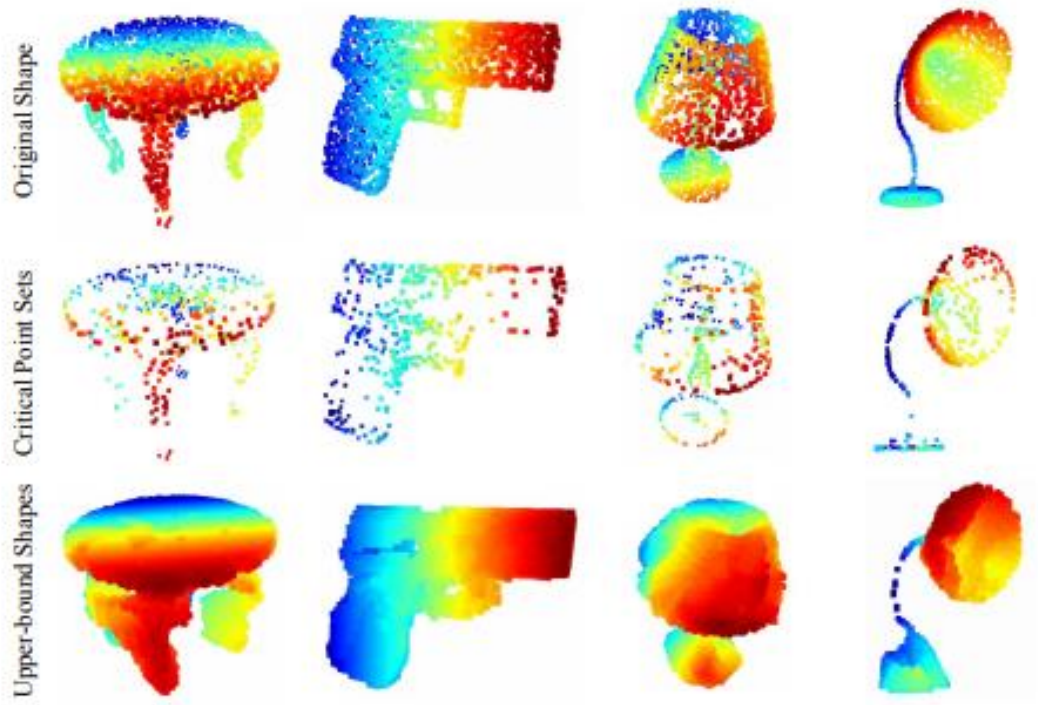
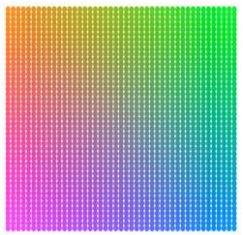
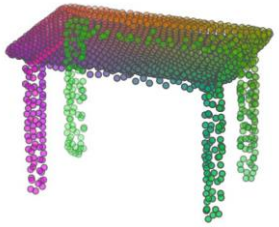
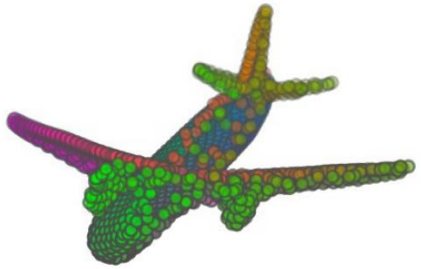


Figure 7. **Critical points and upper bound shape.** While critical points jointly determine the global shape feature for a given shape, any point cloud that falls between the critical points set and the upper bound shape gives exactly the same feature. We color-code all figures to show the depth information.

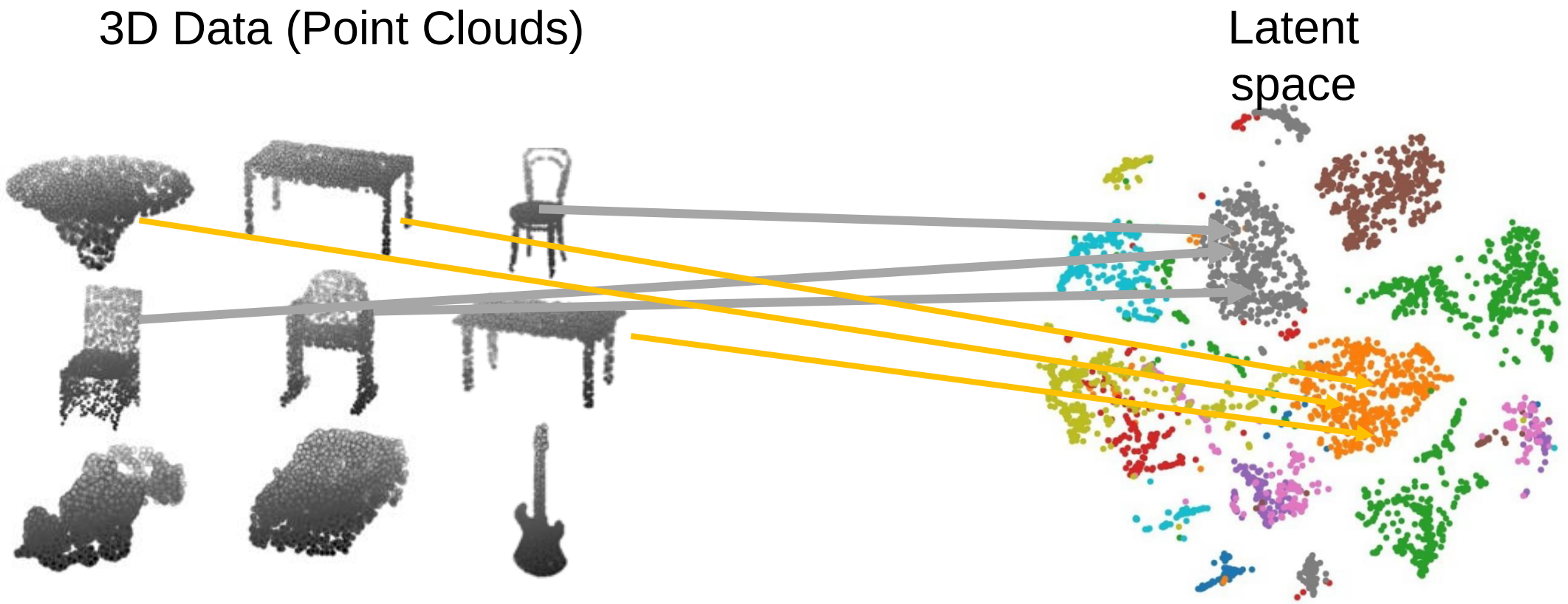



Can Neural Networks Learn Paper Folding?

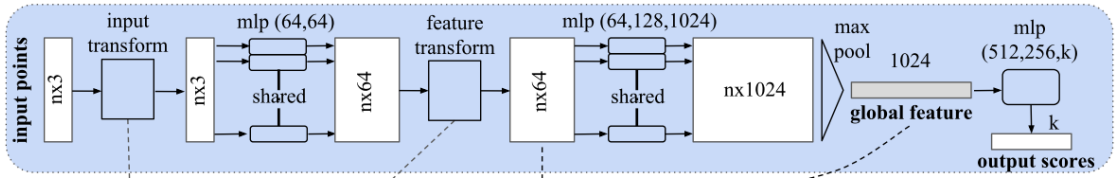


Representation Learning for Point Clouds



$\boldsymbol{F}(\boldsymbol{P})$: point cloud $\rightarrow$ codeword

Baseline Point Cloud Auto-encoder



[Qi et al., CVPR'17]

Input

Encoder

1xK

codeword

Decoder (MLP)

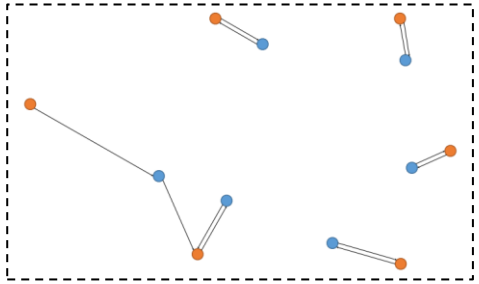
[Achlioptas et al., arxiv'17]

3Nx1

Reshape

Nx3

Output



Chamfer Distance, Earth-Mover's Distance, etc.

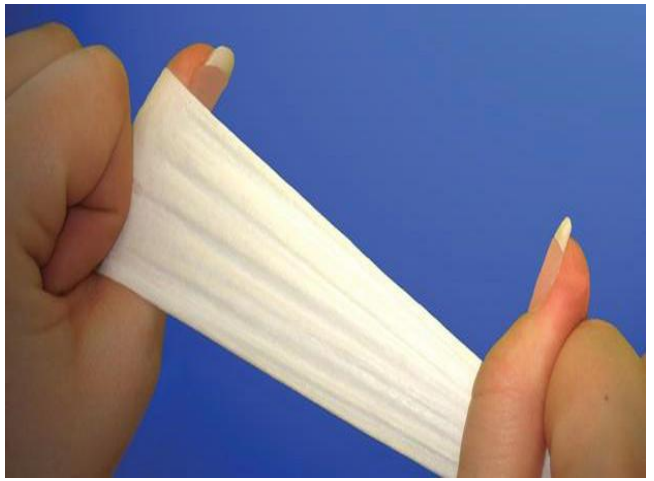
$$\max\left\{\frac{1}{|S|}\sum_{\mathbf{x}\in S}\min_{\hat{\mathbf{x}}\in\hat{S}}\|\mathbf{x}-\hat{\mathbf{x}}\|_2,\ \frac{1}{|\hat{S}|}\sum_{\hat{\mathbf{x}}\in\hat{S}}\min_{\mathbf{x}\in S}\|\hat{\mathbf{x}}-\mathbf{x}\|_2\right\}$$

Intuition of FoldingNet: Elastic Paper Folding

- 3D point clouds are often obtained from object surfaces
  - Discretized from CAD models
  - Sampled from line-of-sight sensors
- 3D object surfaces are intrinsically 2D-manifolds
  - Can be transformed from a 2D plane, through the Origami operations
  - This 2D-3D mapping is known as parameterization/cross-parameterization


Fold


Tear

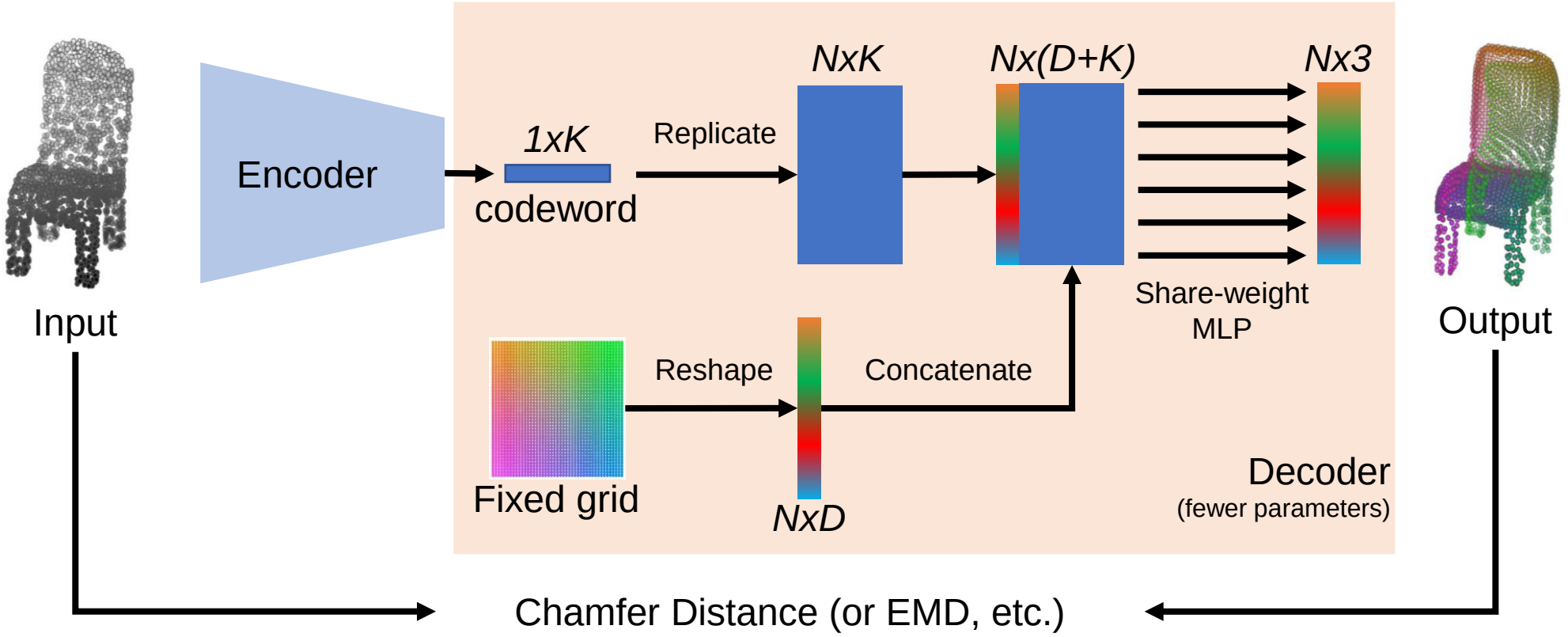

Stretch


Glue

FoldingNet Auto-encoder Framework

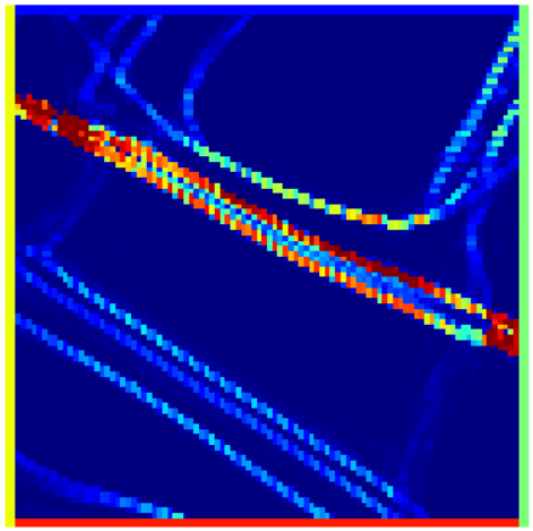
FoldingNet decoder is a *Deep Parametric Surface*:

$$x_i = f_\theta(g_i, c) \in R^3, \forall g_i \in \text{the fixed/random grid}$$

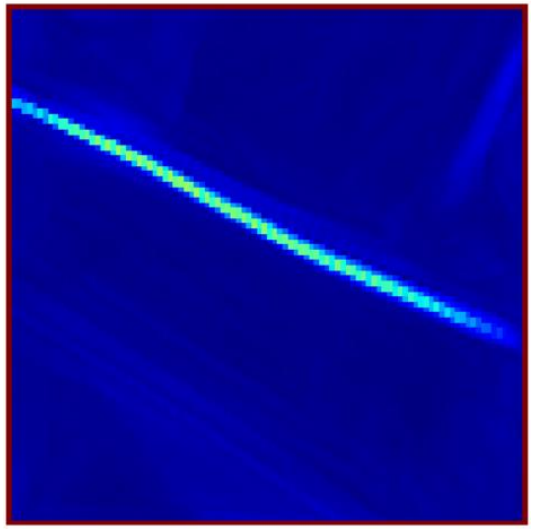


Learned Folding Profile - Sofa

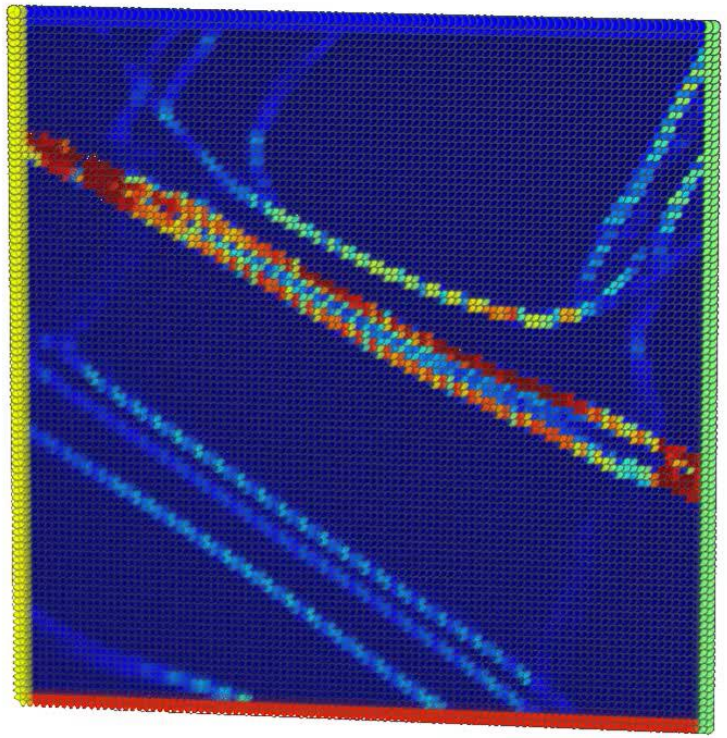
Folding Creases
(Curvature)



Tear/Stretch
(Neighbor Distance)



Folding Animation: Sofa
(colored by curvature)



Training Process Visualization

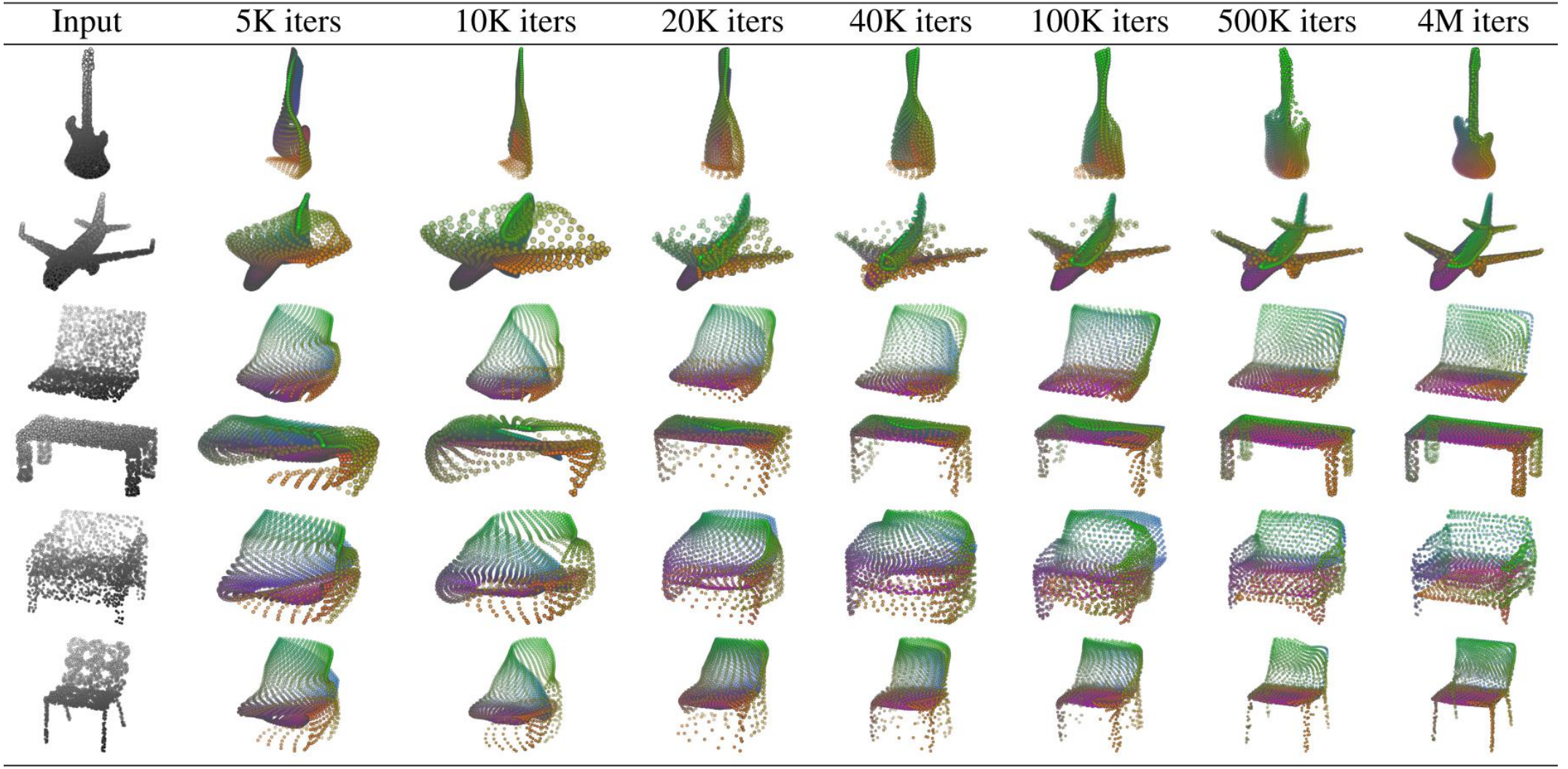


Table 2. Illustration of the training process. Random 2D manifolds gradually transform into the surfaces of point clouds.

Codeword Space Visualization

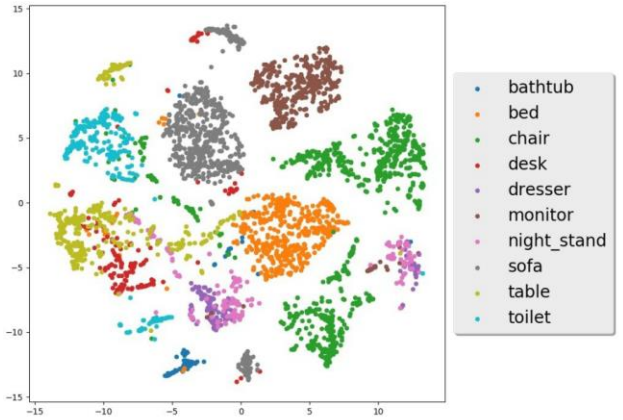
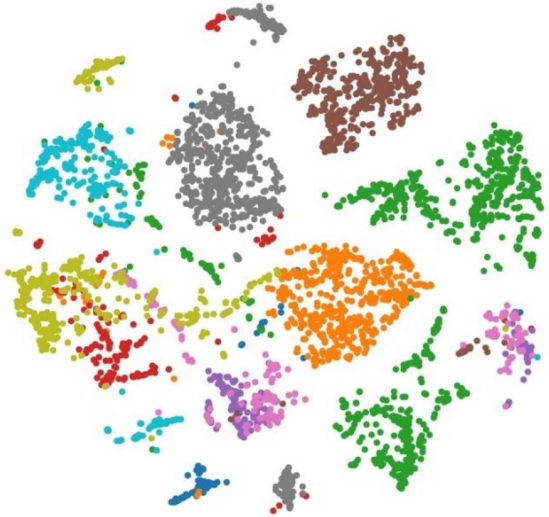


Figure 2. The T-SNE clustering visualization of the codewords obtained from FoldingNet auto-encoder.

Shape Interpolation



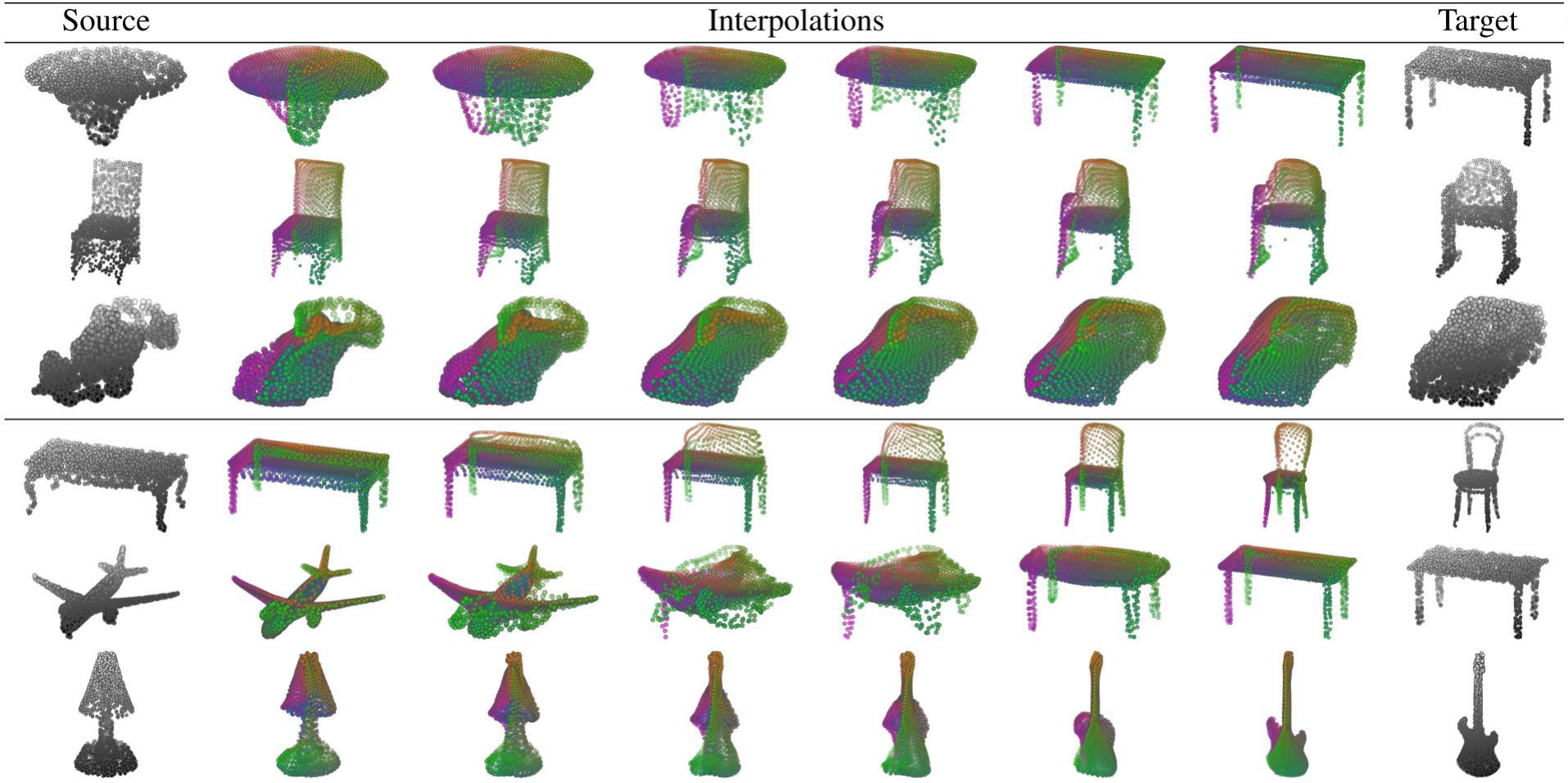


Table 3. Illustration of point cloud interpolation. The first 3 rows: intra-class interpolations. The last 3 rows: inter-class interpolations.

Quantitative Evaluation of the Learned Features

- Transfer Classification

Method	MN40	MN10
SPH [26]	68.2%	79.8%
LFD [8]	75.5%	79.9%
T-L Network [19]	74.4%	-
VConv-DAE [45]	75.5%	80.5%
3D-GAN [56]	83.3%	91.0%
Latent-GAN [1]	85.7%	**95.3%**
FoldingNet (ours)	**88.4%**	94.4%

Table 5. The comparison on classification accuracy between FoldingNet and other unsupervised methods. All the methods train a linear SVM on the high-dimensional representations obtained from unsupervised training.

- Semi-supervised Learning

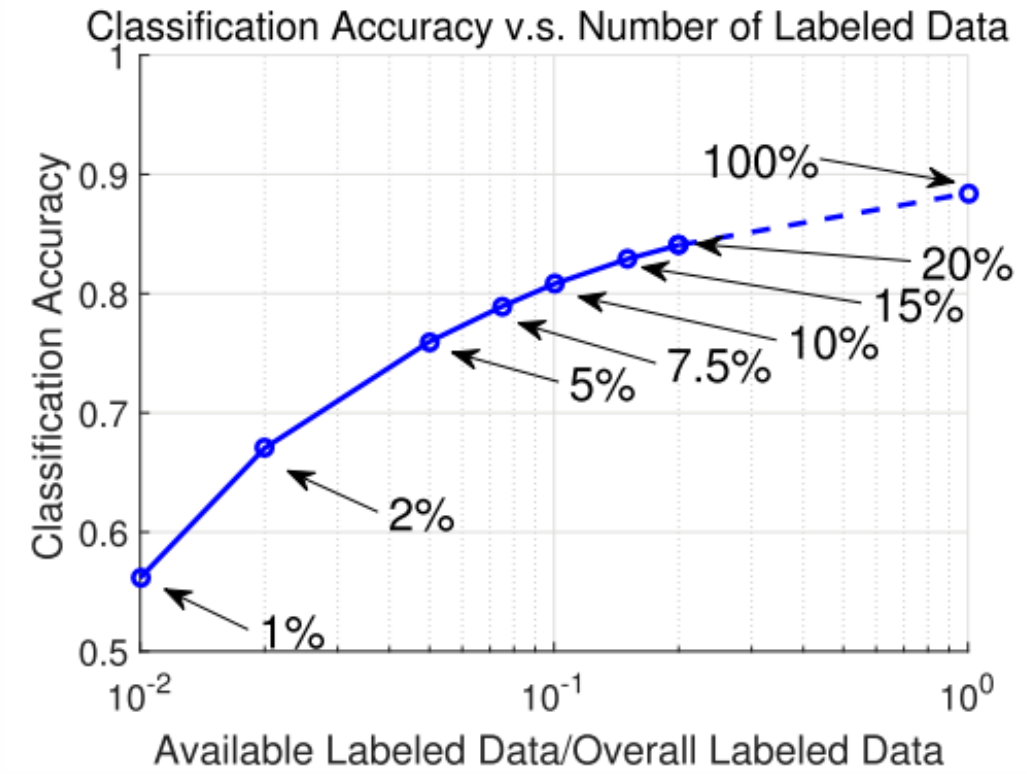


Figure 4. Linear SVM classification accuracy v.s. percentage of available labeled training data in ModelNet40 dataset.

Ablation: Decoder Variations

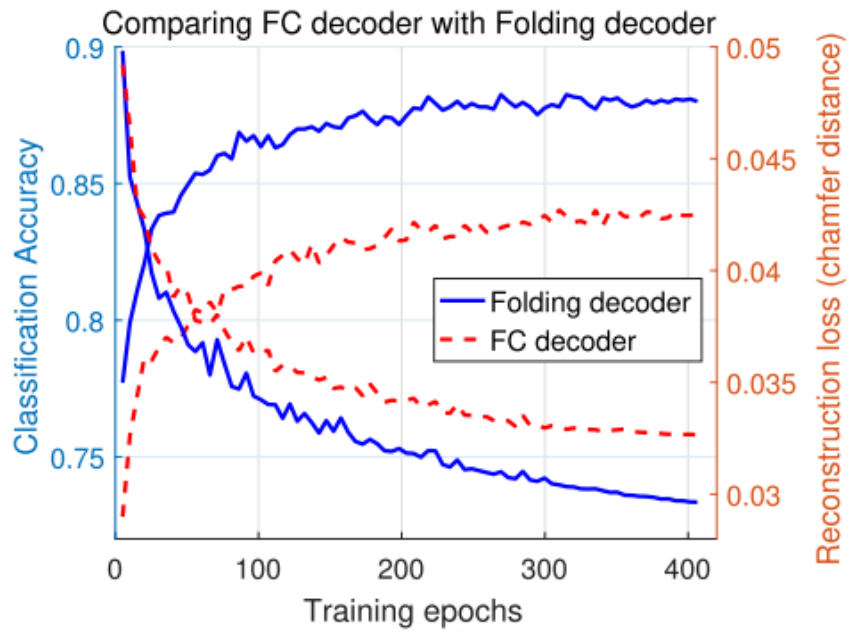


Figure 5. Comparison between the fully-connected (FC) decoder in [1] and the folding decoder on ModelNet40.

Grid Setting	#Folds	Test Cls. Acc.	Test Loss
regular 2D	2	88.25%	0.0296
regular 2D	3	88.41%	0.0290
regular 1D	2	86.71%	0.0355
regular 3D	2	88.41%	0.0284
uniform 2D	2	87.12%	0.0321

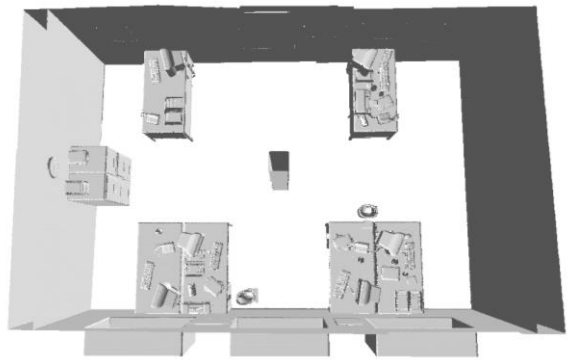
Table 6. Comparison between different FoldingNet decoders. "Uniform": the grid is uniformly random sampled. "Regular": the grid is regularly sampled with fixed spacings.

	Cl. Acc.	Tst. Loss	# Params.
FoldingNet	88.41%	0.0296	1.0×10^6
Deconv	88.86%	0.0319	1.7×10^6

Table 7. Comparison of two different implementations of the folding operation.

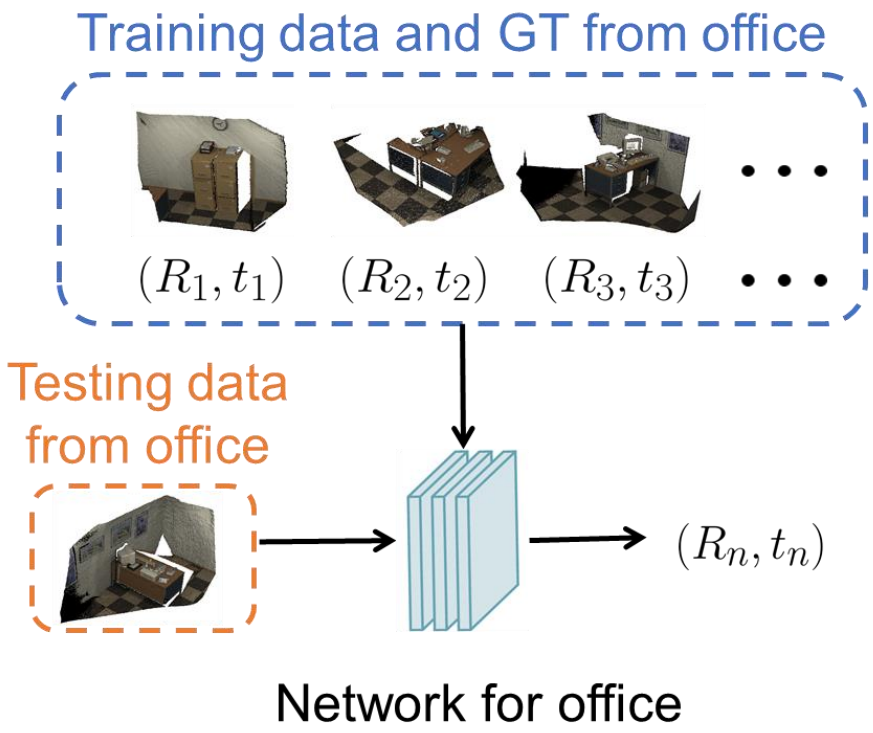



Can We Black-Box Point Cloud Mapping?



Motivation: Deep Learning for Geometric Vision Problems

- Common supervised deep learning pipeline for mapping/registration
  - Collect training dataset and ground truth poses
  - Training and testing on the same scene [PoseNet ICCV'15, MapNet CVPR'18, etc.]



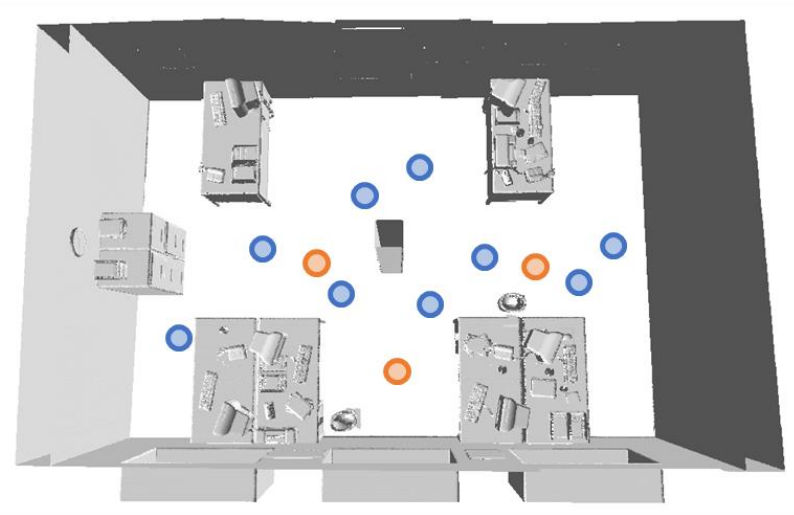
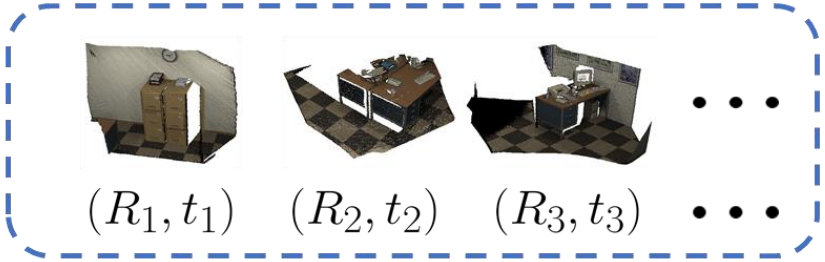

Office

Image source: S. Choi, et. al. CVPR, 2015.

Motivation: Deep Learning for Geometric Vision Problems

- Issues of these supervised approaches
  - Collect training dataset and ground truth poses
    - Ground truth is not always easy to obtain: SLAM, surveying, …

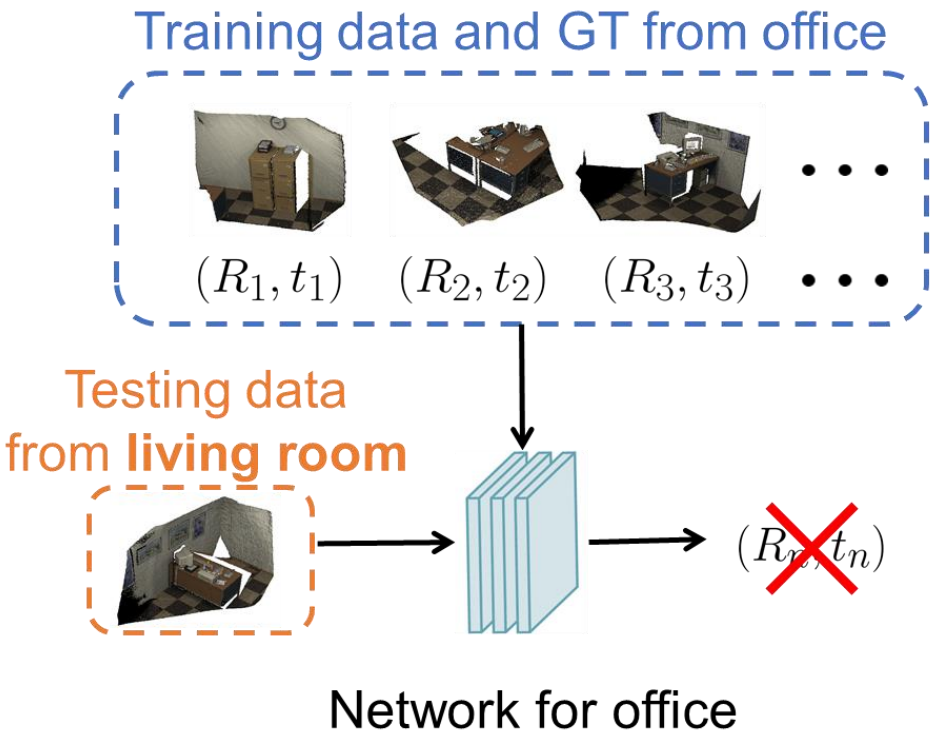
Training data and GT from office



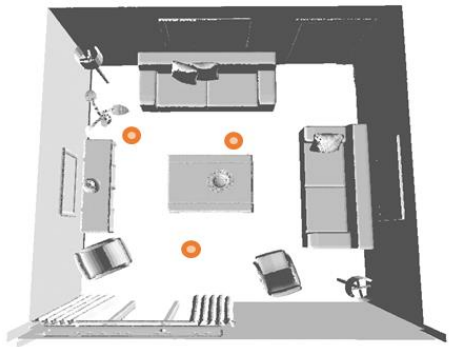
Office

Motivation: Deep Learning for Geometric Vision Problems

- Issues of these supervised approaches
  - Training and testing on the same scene
    - Network trained on “office” might not generalize to “living room”

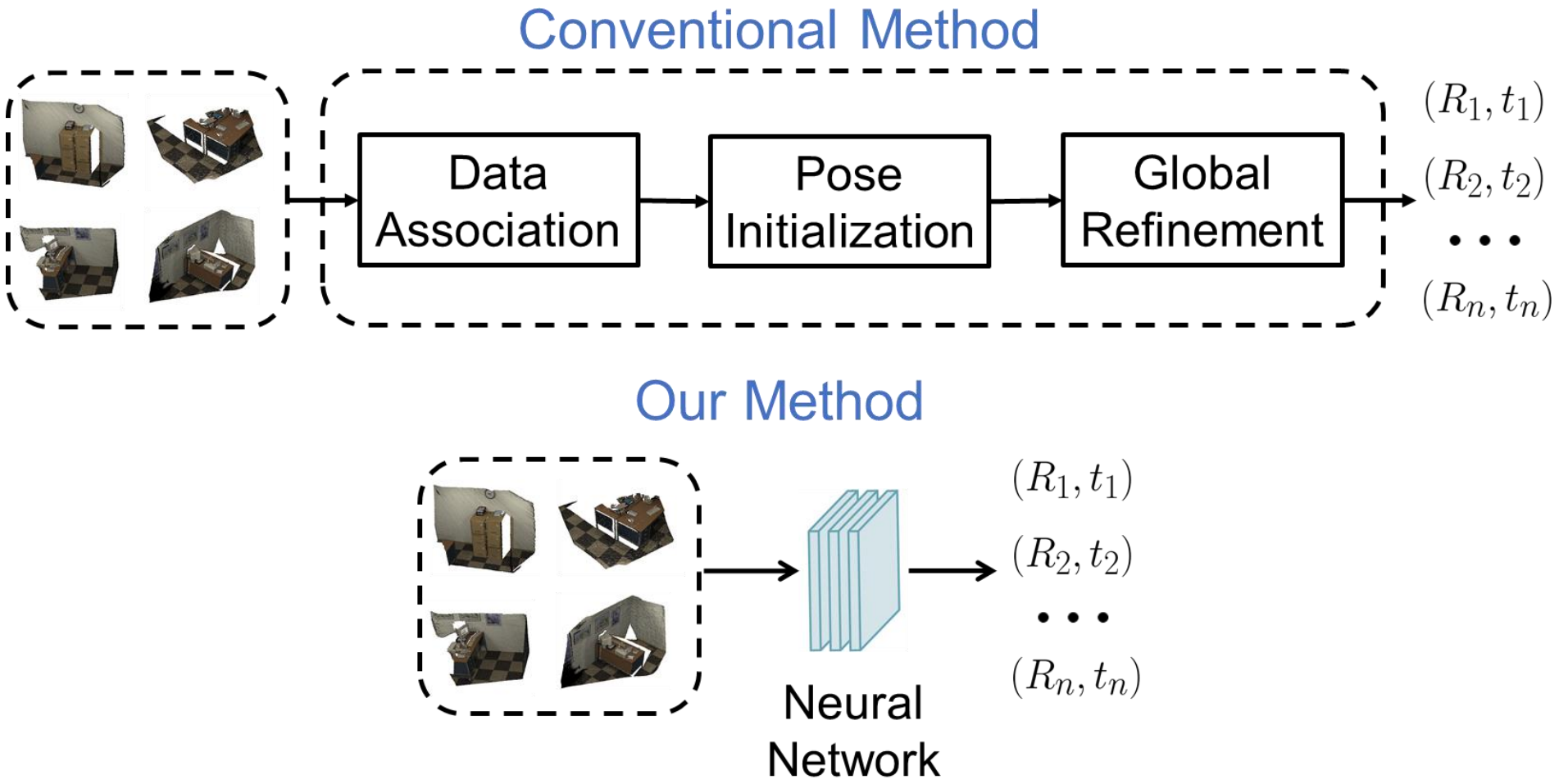



Office


Living room

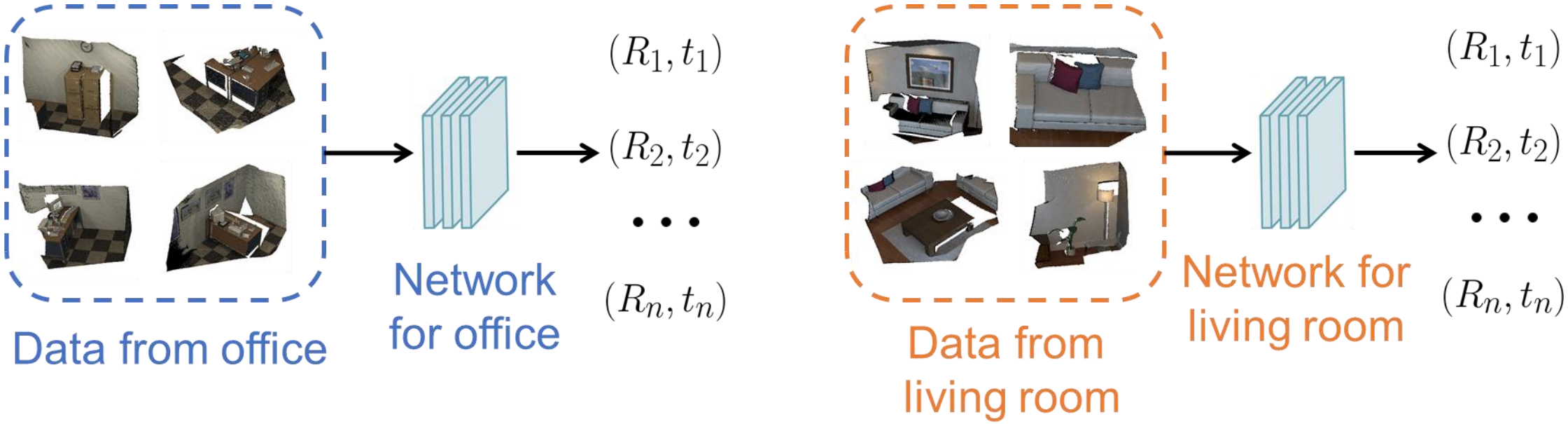
Our Problem Formulation

- Use neural network to model traditional registration process



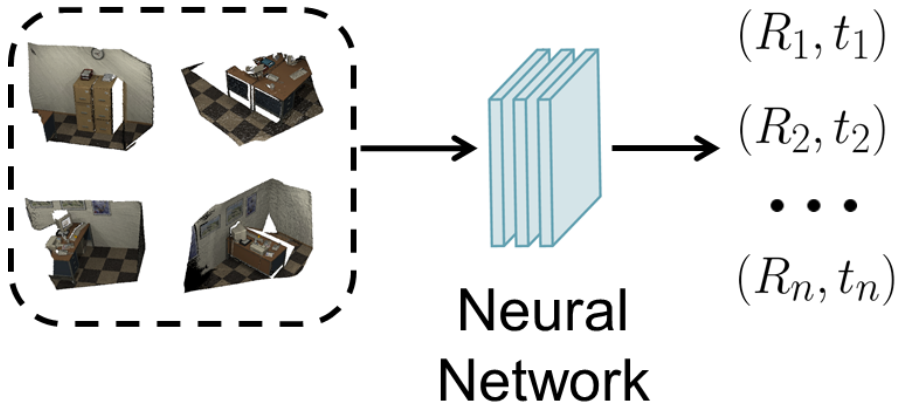
Our Problem Formulation

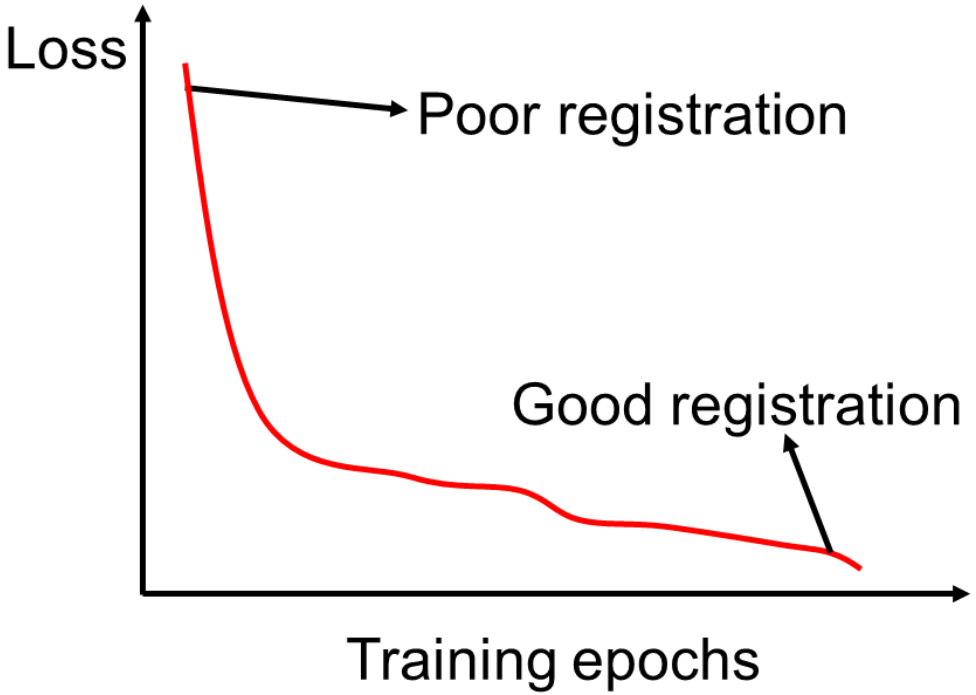
- Use neural network to model traditional registration process
- Train a neural network for each registration problem



Our Problem Formulation

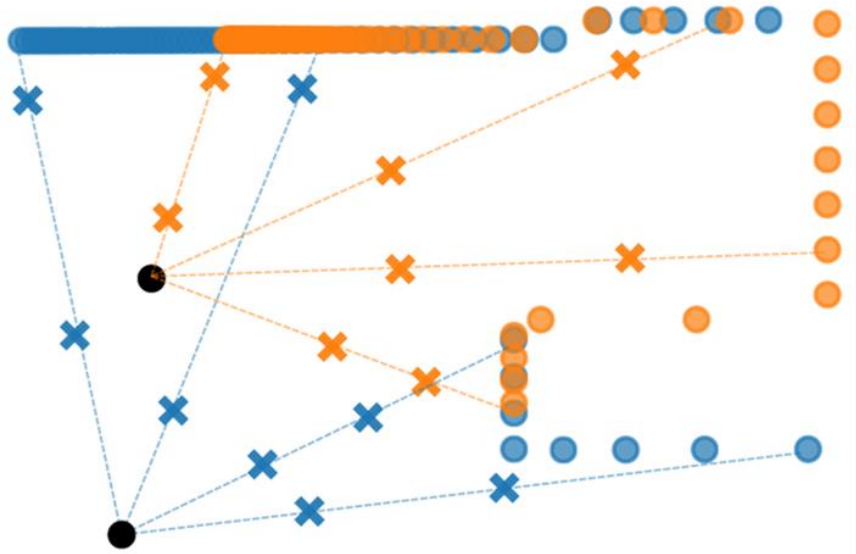
- Solve registration as an unsupervised “training” of neural network
  - Do not rely on ground truth data
  - Unsupervised loss measures registration quality
    - Minimize loss = encourage alignment



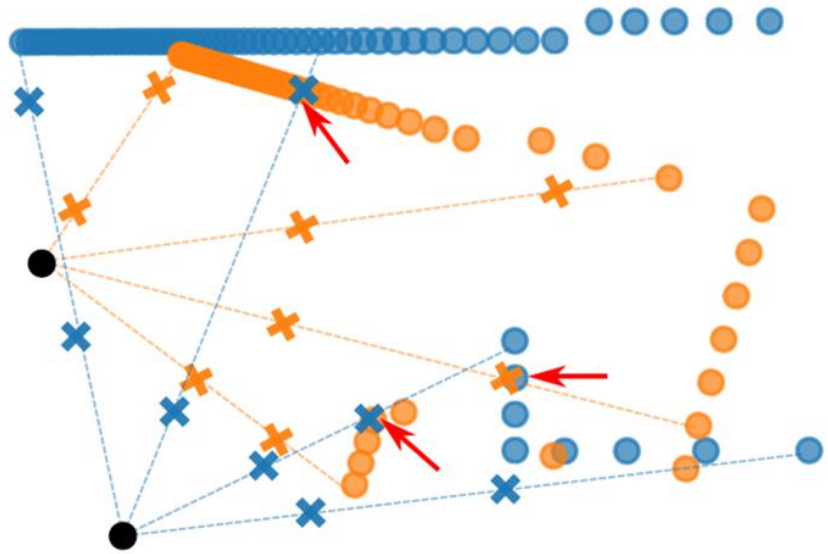


DeepMapping Loss

- How do we design unsupervised loss to reflect the registration quality?
  - Utilize both points in observed point clouds and those in free-space
- Free-space points ✖ in one point cloud not conflict with observed points ● in another point cloud

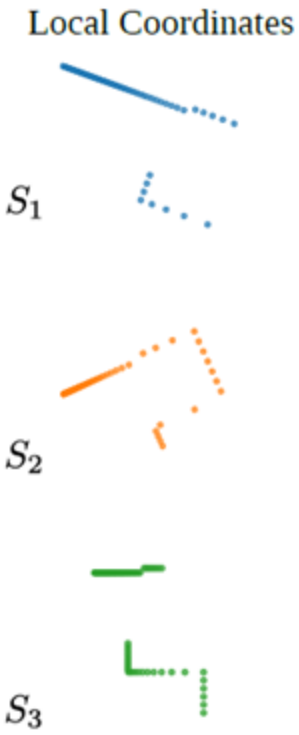


Correctly aligned
No free-space inconsistency

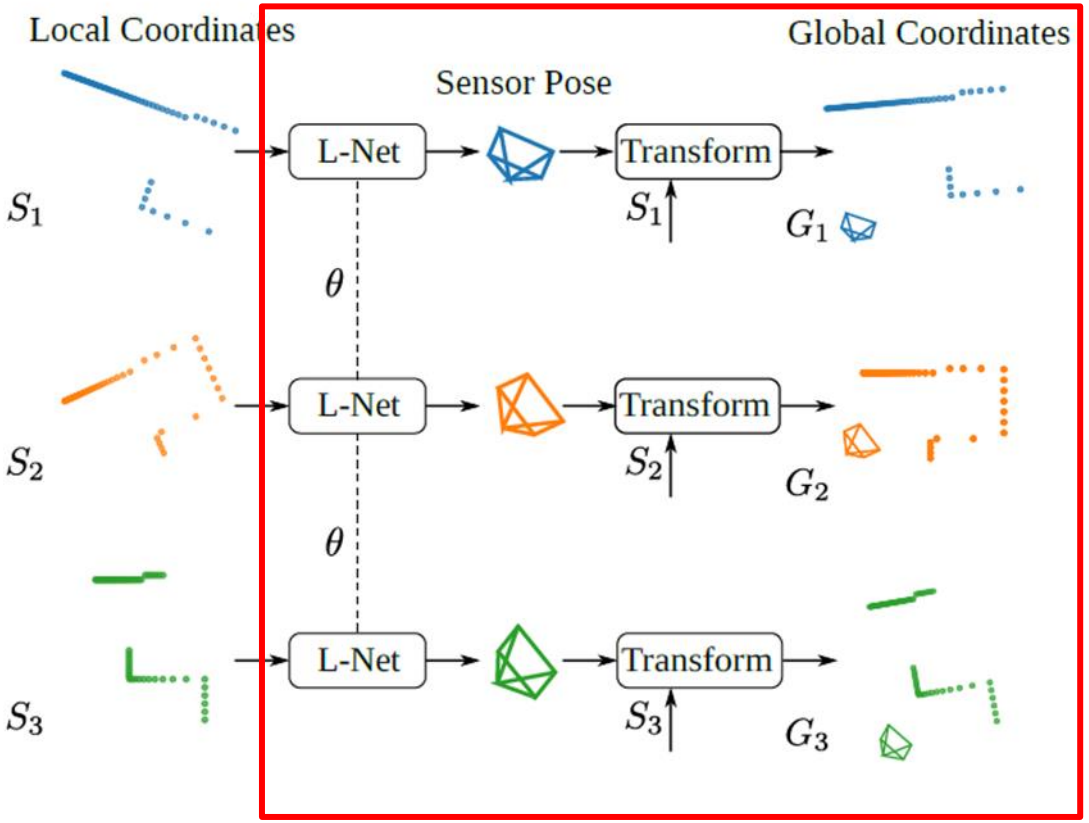


Mis-aligned
Free-space inconsistency

DeepMapping Pipeline



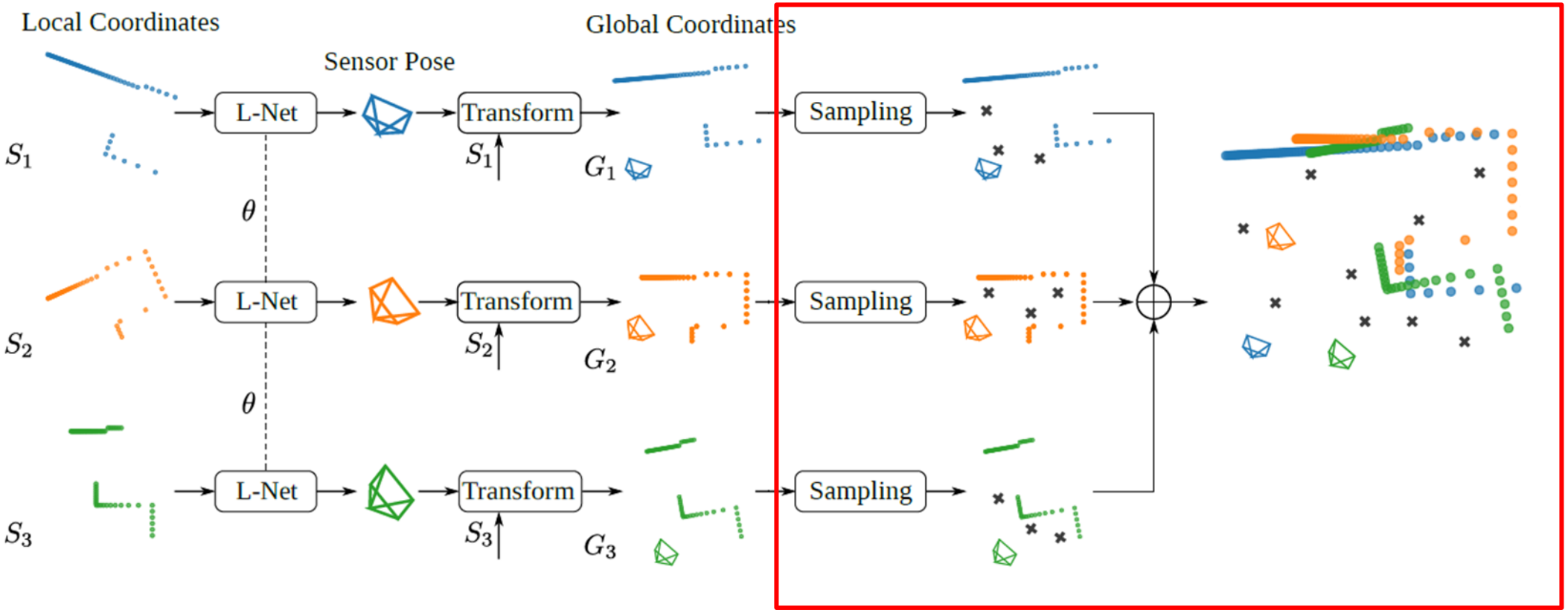
DeepMapping Pipeline: Localization Network



Localization Network (L-Net)

$$f_\theta : S_i \mapsto T_i$$

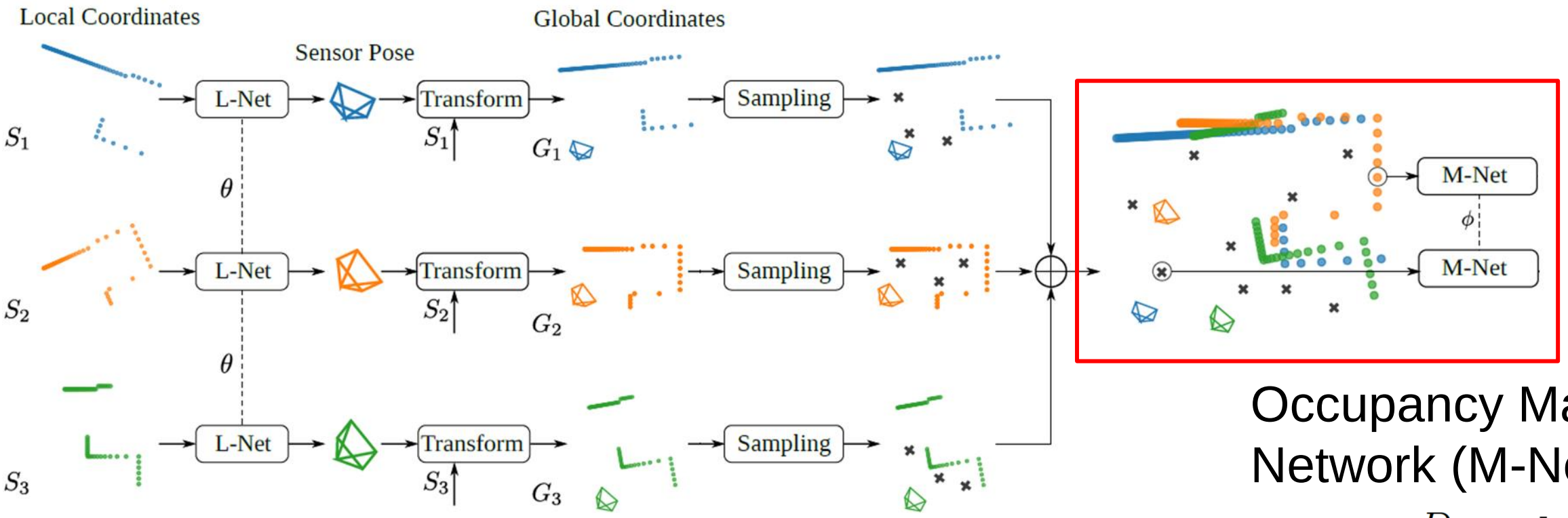
DeepMapping Pipeline: Free-Space Sampling



Localization Network (L-Net)

$$f_\theta : S_i \mapsto T_i$$

DeepMapping Pipeline: Occupancy Map Network



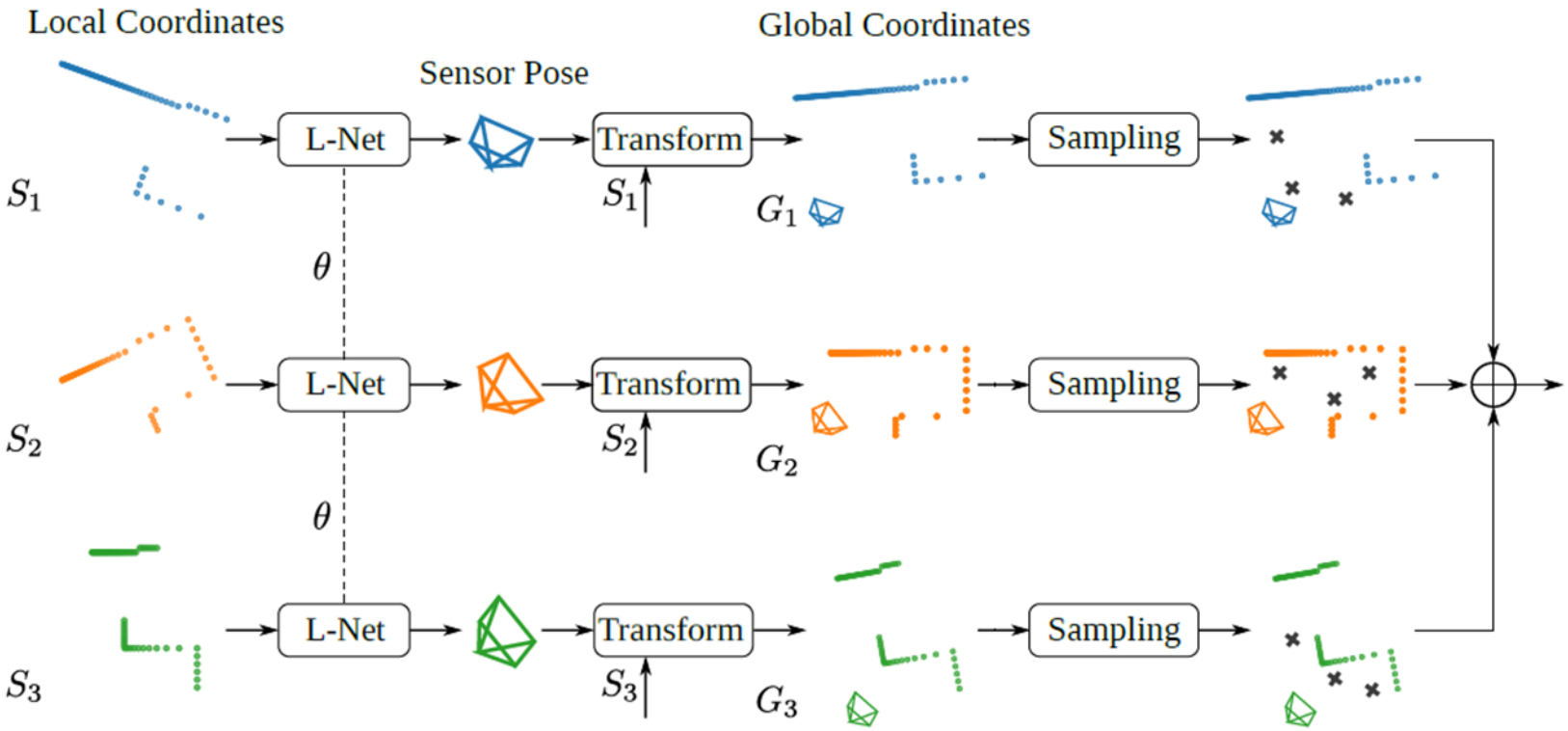
Occupancy Map Network (M-Net)

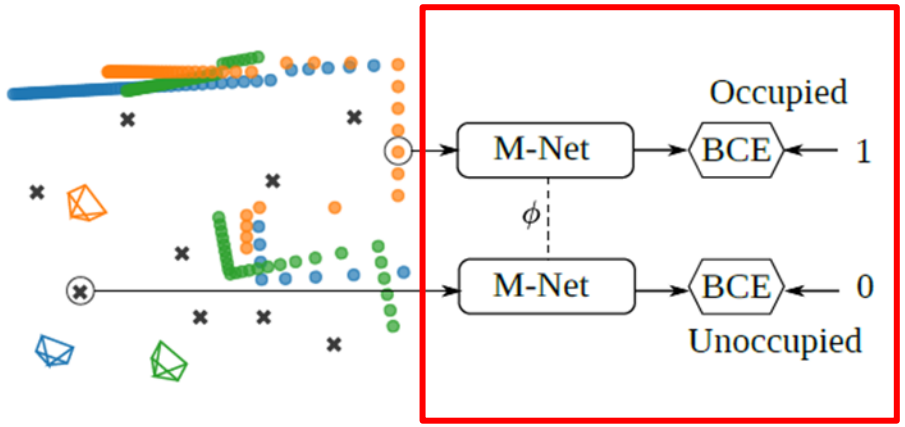
$$m_\phi : \mathbb{R}^D \rightarrow [0, 1]$$

Localization Network (L-Net)

$$f_\theta \ : \ S_i \ \mapsto \ T_i$$

DeepMapping Pipeline: Mapping Using Binary Classification Loss





Occupancy Map Network (M-Net)

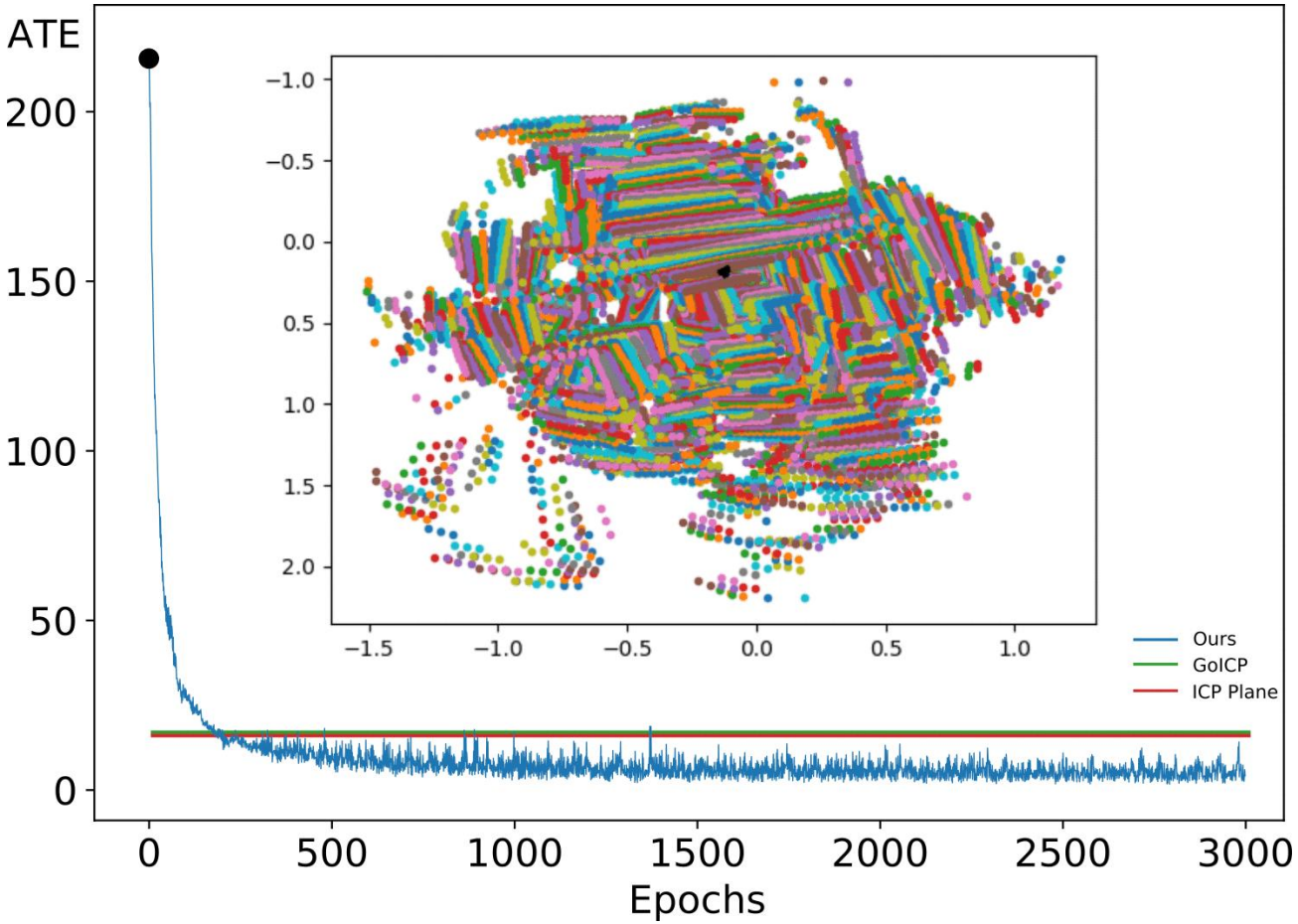
$$m_\phi : \mathbb{R}^D \rightarrow [0, 1]$$

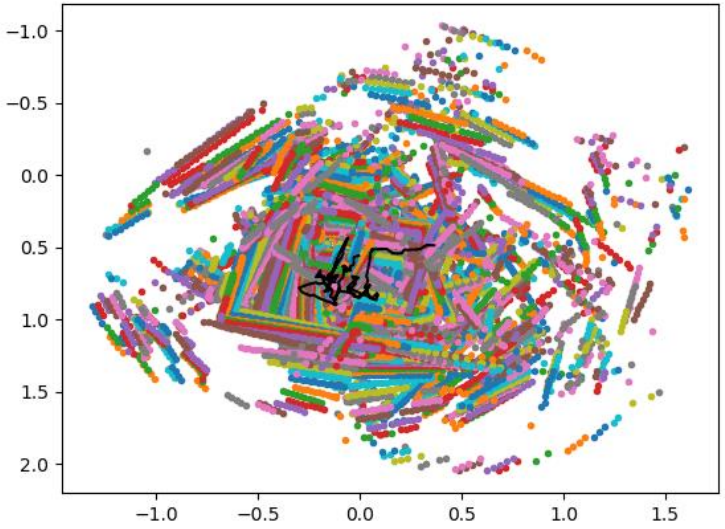
Localization Network (L-Net)

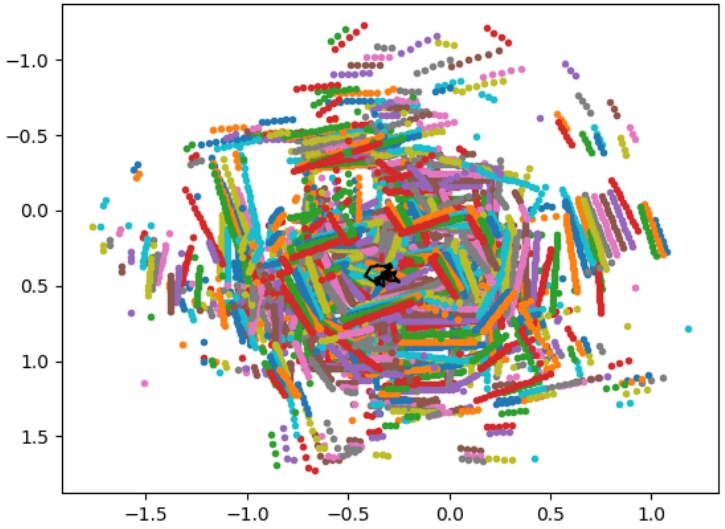
$$f_\theta : S_i \mapsto T_i$$

2D Point Cloud Registration

- Solve point cloud registration as “training” neural networks using unsupervised loss

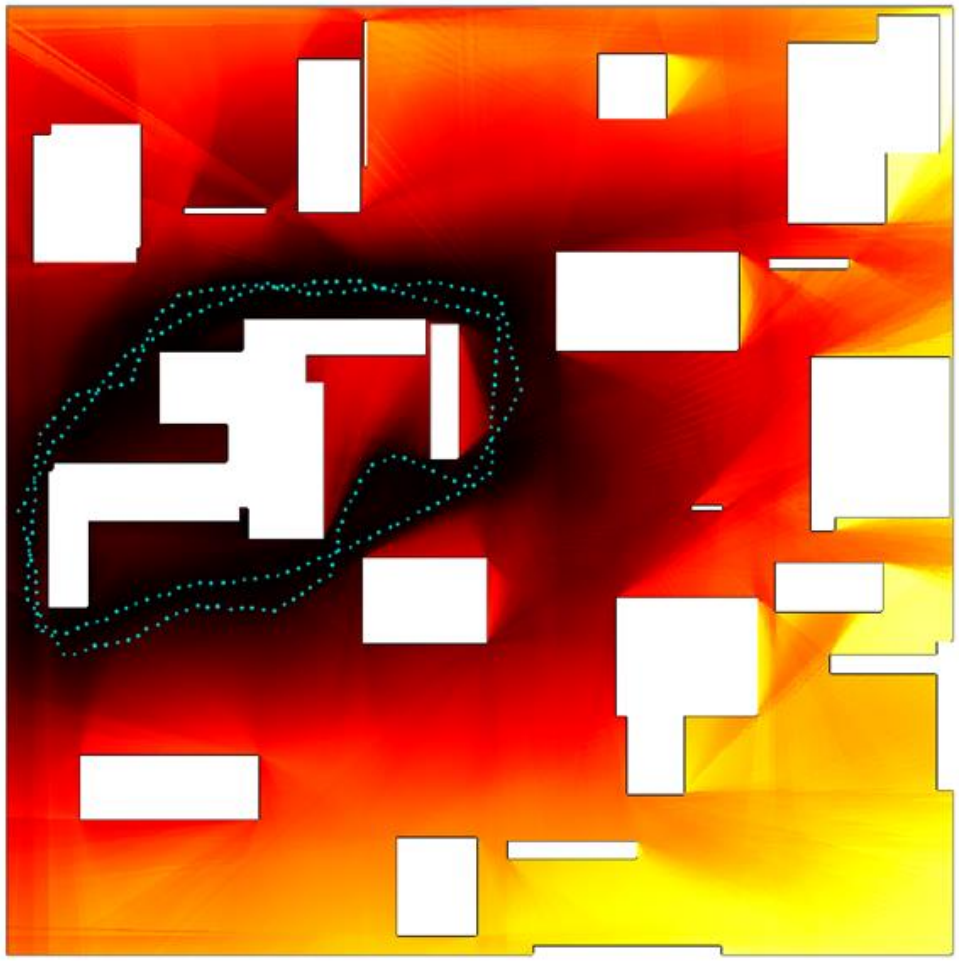


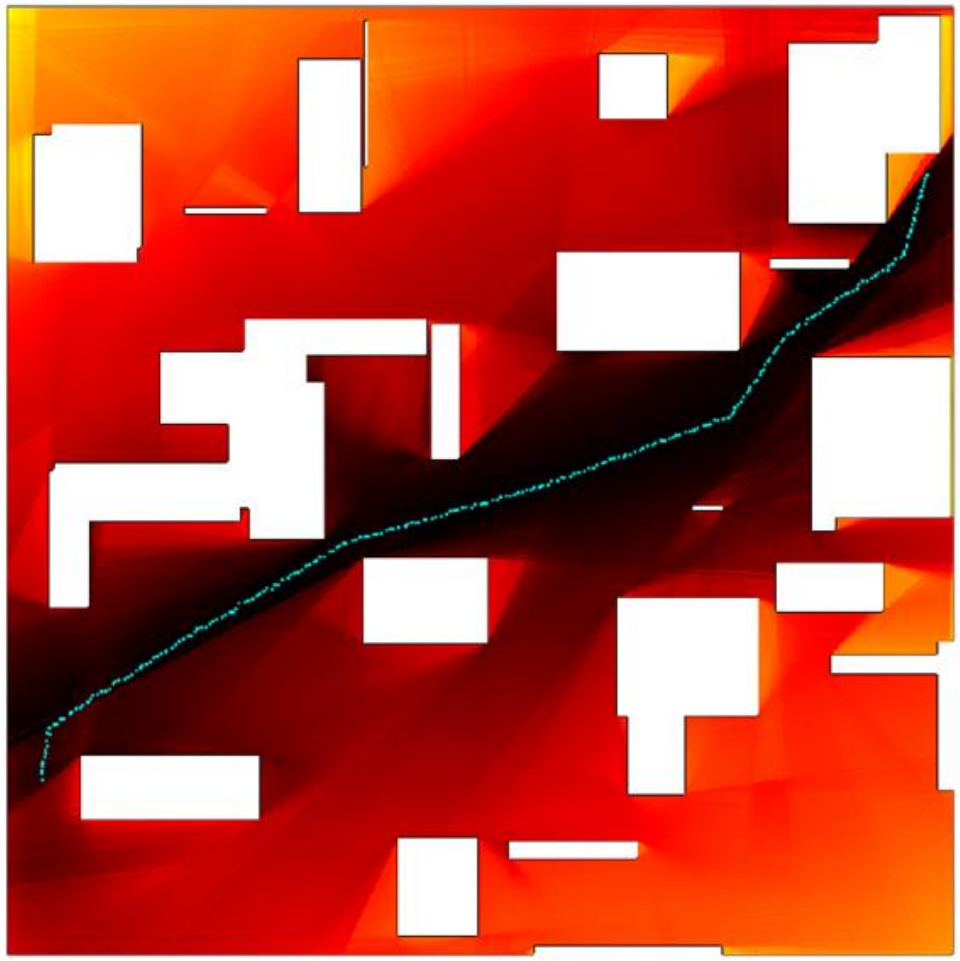




L-Net: Coarse Re-localization For Unseen Point Clouds

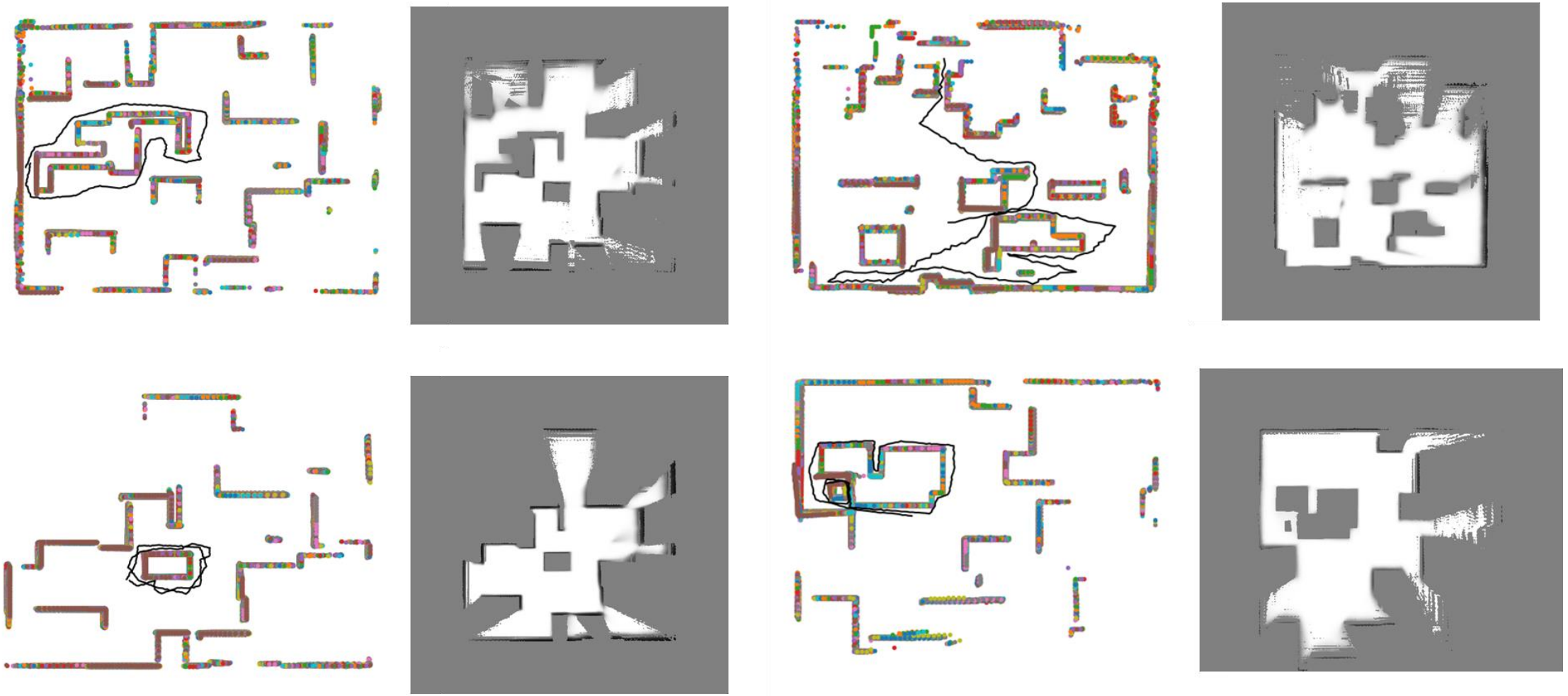
- Re-localization errors of two L-Nets “trained” on two trajectories
  - Darker is better





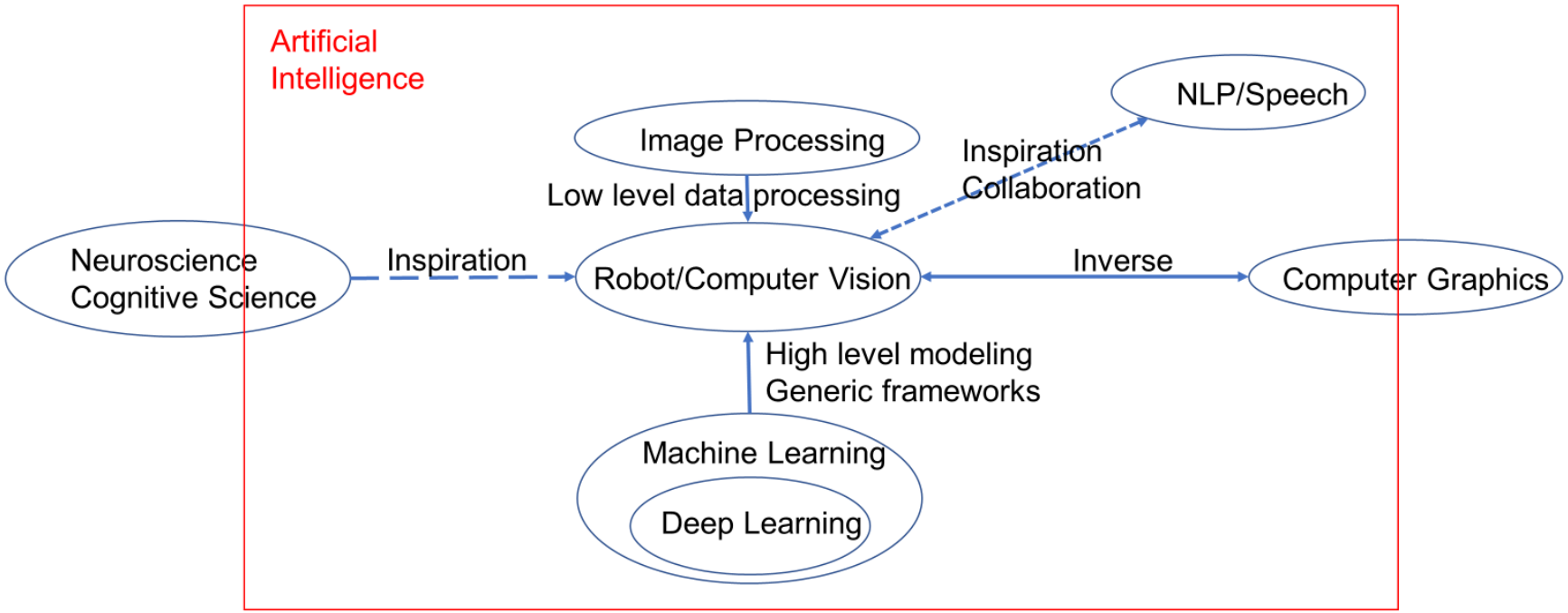
M-Net: Continuous Occupancy Map Representation

Aligned point clouds M-Net output Aligned point clouds M-Net output



Summary of The Semester – Introduction

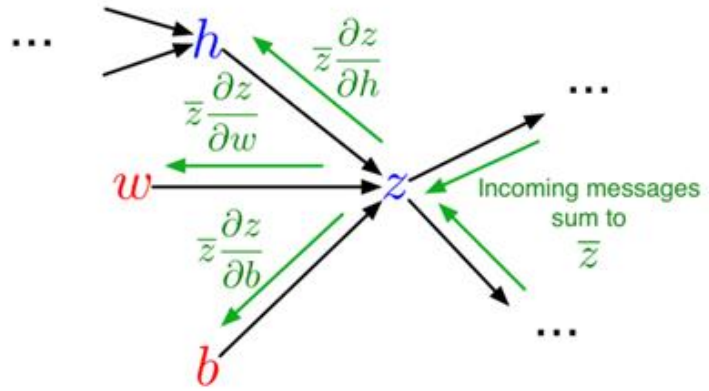
Related Disciplines



Summary of The Semester – Deep Learning

A Symbolic Computation Graph

Backprop as message passing:



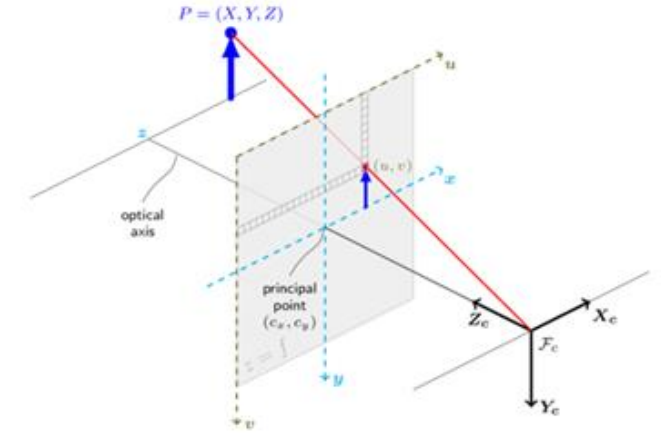
- Each node receives a bunch of messages from its children, which it aggregates to get its error signal. It then passes messages to its parents.
- This provides modularity, since each node only has to know how to compute derivatives with respect to its arguments, and doesn't have to know anything about the rest of the graph.

Slides from Roger Grosse

26

Summary of The Semester – Camera Model

Everything Together in OpenCV (Full Model)



$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x' y' + p_2(r^2 + 2x'^2)$$

$$y'' = y' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + p_1(r^2 + 2y'^2) + 2p_2 x' y'$$

where $r^2 = x'^2 + y'^2$

$$u = f_x * x'' + c_x$$

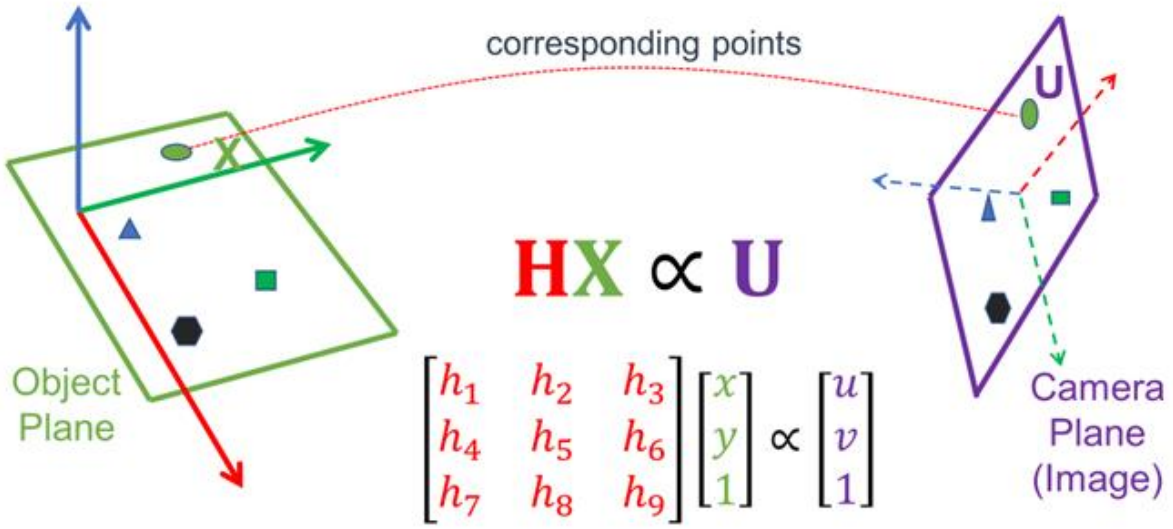
$$v = f_y * y'' + c_y$$

https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

35

Summary of The Semester – Homography

How to Estimate a Homography?



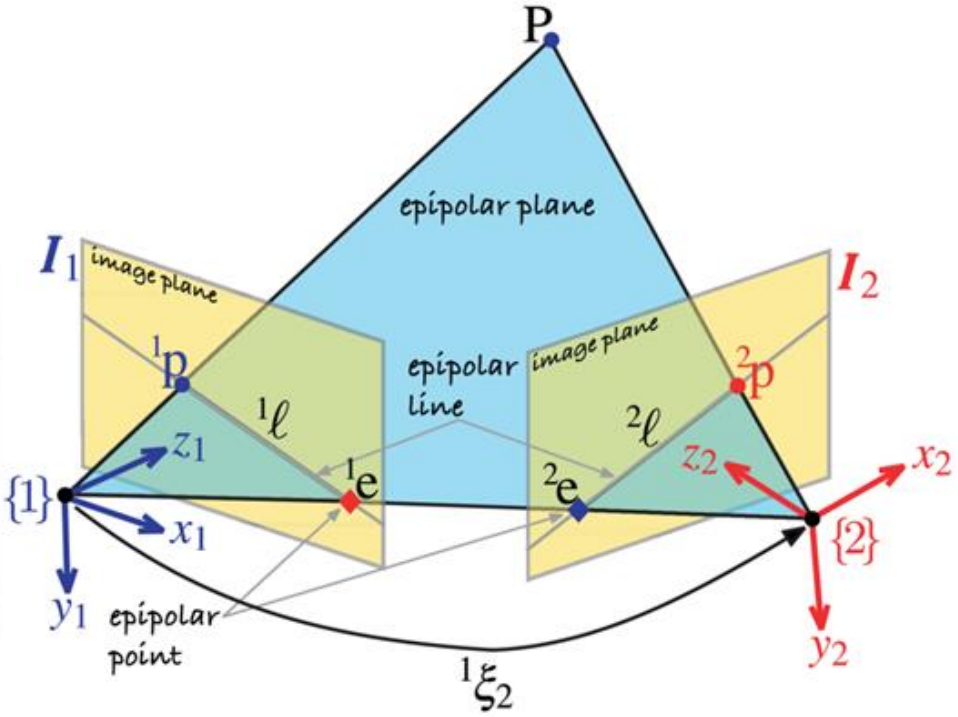
$$\mathbf{HX} \propto \mathbf{U}$$

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \propto \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

19

Summary of The Semester – Two-View Geometry

Two-view Geometry



Corke 2011

7

Summary of The Semester – SfM

Bundle Adjustment == Least Squares

A valid solution $[\mathbf{R}_1|\mathbf{t}_1],[\mathbf{R}_2|\mathbf{t}_2],[\mathbf{R}_3|\mathbf{t}_3]$ and $\mathbf{X}^1,\mathbf{X}^2,\mathbf{X}^3,\cdots$ must let the Re-projection close to the Observation, i.e. to minimize the reprojection error

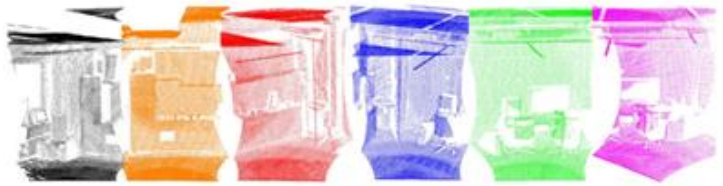
$$\min \sum_i \sum_j \left(\tilde{\mathbf{x}}_i^j - \mathbf{K}\left[\mathbf{R}_i|\mathbf{t}_i\right]\mathbf{X}^j\right)^2$$

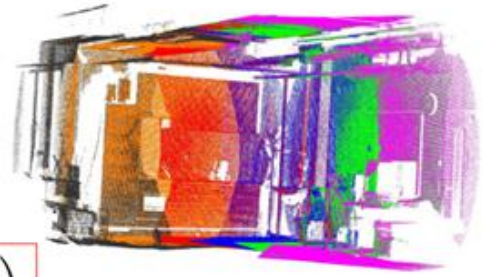
Slides from Jianxiong Xiao

23

Summary of The Semester – ICP

Point Cloud Mapping / Scan Registration

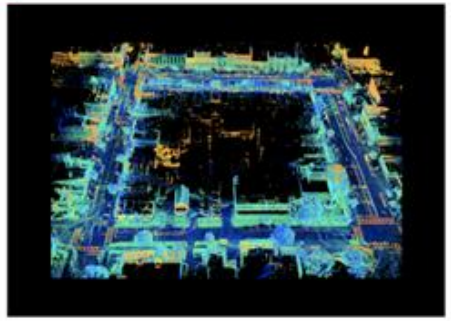




$$\underset{\{R_i, t_i\}}{\operatorname{argmin}} \sum_{i,j} \sum_{k} \rho\left(\min_{s} \left\| \begin{bmatrix} R_j & t_j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_s^j \\ 1 \end{bmatrix} - \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_k^i \\ 1 \end{bmatrix} \right\|^2 \right)$$

$$s.t. \quad R_i \in SO(3)$$

$$t_i, P_k^i \in \mathbb{R}^3, \ \rho(\cdot) \text{ is a robust loss function (e.g., Huber loss)}$$



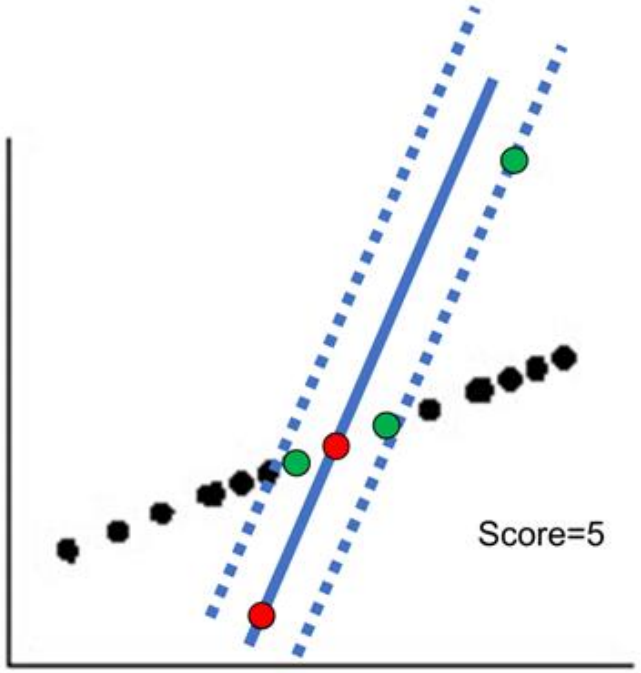
Images from http://pointclouds.org/documentation/tutorials/registration_api.php

5

Summary of The Semester – RANSAC

Hypothesis and Test

- Randomly select two points to generate a hypothesis line
- Test how good the current hypothesis is



Summary of The Semester – VPR

Visual Place Recognition Pipeline

Query image

Database images

Offline

1. Feature detection & description
2. Training visual vocabulary
3. Image description

4. Feature detection & description
5. Image description

6. Initial ranking
7. Re-ranking with geometric verification

Slides from Sattler et. al., 2015

14

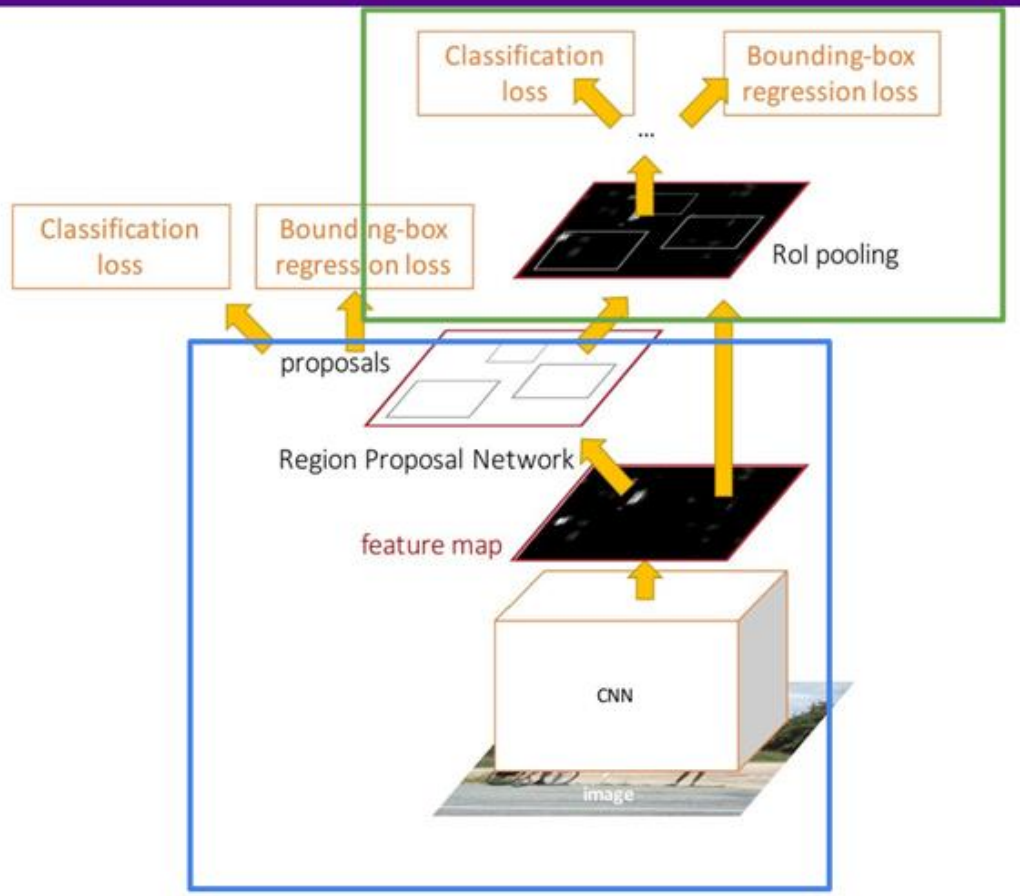
Summary of The Semester – Object Detection

Faster R-CNN

Faster R-CNN is a **Two-stage object detector**

First stage: Run once per image
- Backbone network
- Region proposal network

Second stage: Run once per region
- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset



66

Summary of The Semester – Object Tracking

KLT Tracker

– First assuming small displacement: 1st-order Taylor expansion inside SSD

$$\hat{d} = \underset{d}{\operatorname{argmin}} \sum_{p \in R(x)} \left| I^{(t+1)}(p) + \nabla I^{(t+1)}(p)^T d - I^{(t)}(p) \right|^2$$

$$\hat{d} = -\left(\sum_{p \in D(x)} \nabla I^{(t+1)}(p) \nabla I^{(t+1)}(p)^T \right)^{-1} \sum_{p \in D(x)} \nabla I^{(t+1)}(p) [I^{(t+1)}(p) - I^{(t)}(p)]$$

Solving Ax=b

- Solve by least square: $(A^T A)^{-1} A^T b$

For good conditioning, patch must be textured/structured enough:

- Uniform patch: no information
- Contour element: aperture problem (one dimensional information)
- Corners, blobs and texture: best estimate

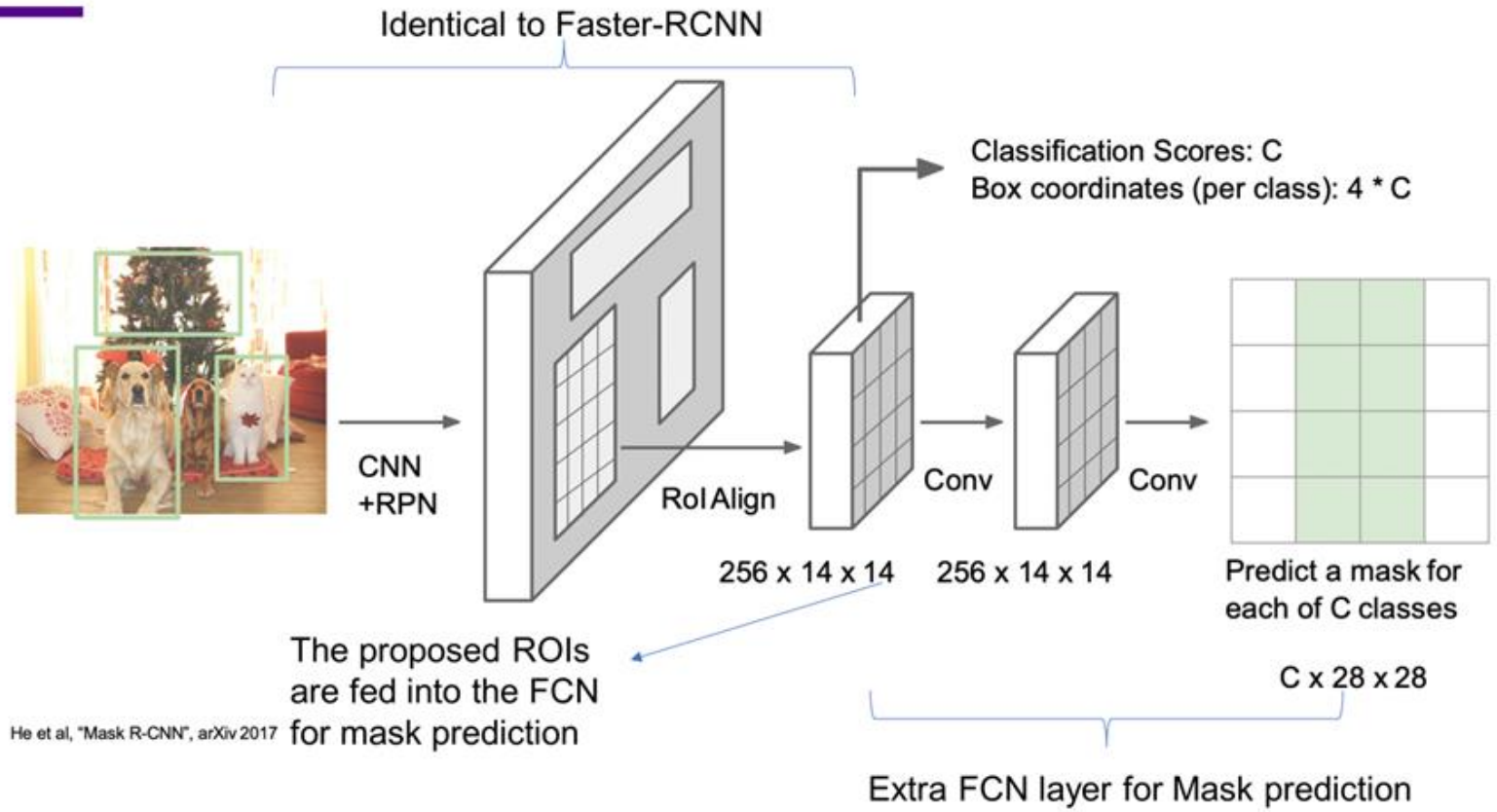
[Lucas and Kanda 1981][Tomasi and Shi, CVPR'94]

Slides from Jiri Matas 2016

23

Summary of The Semester – Segmentation & 3D Deep Learning

Mask R-CNN : Architecture



He et al, "Mask R-CNN", arXiv 2017

19

Thank you!
Questions?