NYU TANDON SCHOOL OF ENGINEERING

Robot Perception

Segmentation & 3D Deep Learning

Dr. Chen Feng

cfeng@nyu.edu

ROB-GY 6203, Fall 2023

References

- Sz: Ch 6.4
- Maturana, Daniel, and Sebastian Scherer. “Voxnet: A 3d convolutional neural network for real-time object recognition.” *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015.
- Su, Hang, et al. “Multi-view convolutional neural networks for 3d shape recognition.” *Proceedings of the IEEE international conference on computer vision*. 2015.
- Qi, C. R., Su, H., Mo, K., & 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).
- Yang, Y., Feng, C., Shen, Y., & Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 206-215).
- Ding, Li, and Chen Feng. "DeepMapping: Unsupervised map estimation from multiple point clouds." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

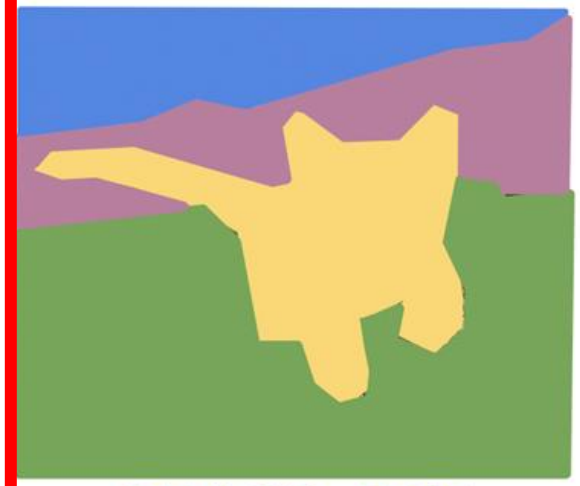
Semantic Segmentation: The Problem

Classification

CAT

No spatial extent

Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection

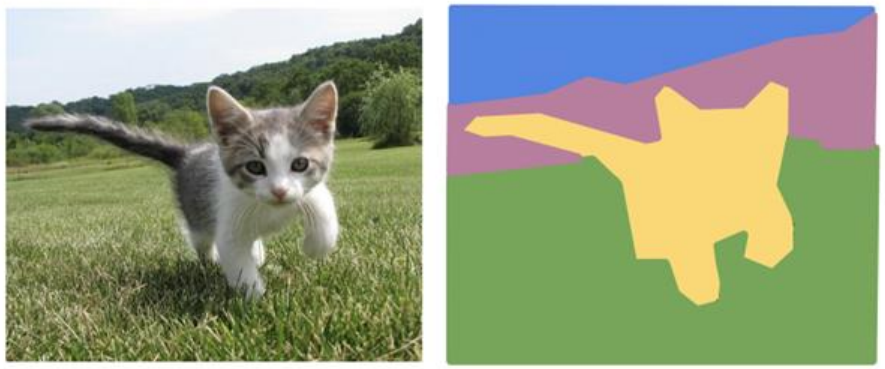
DOG, DOG, CAT

Instance Segmentation

DOG, DOG, CAT

Multiple Object

Semantic Segmentation: The Problem



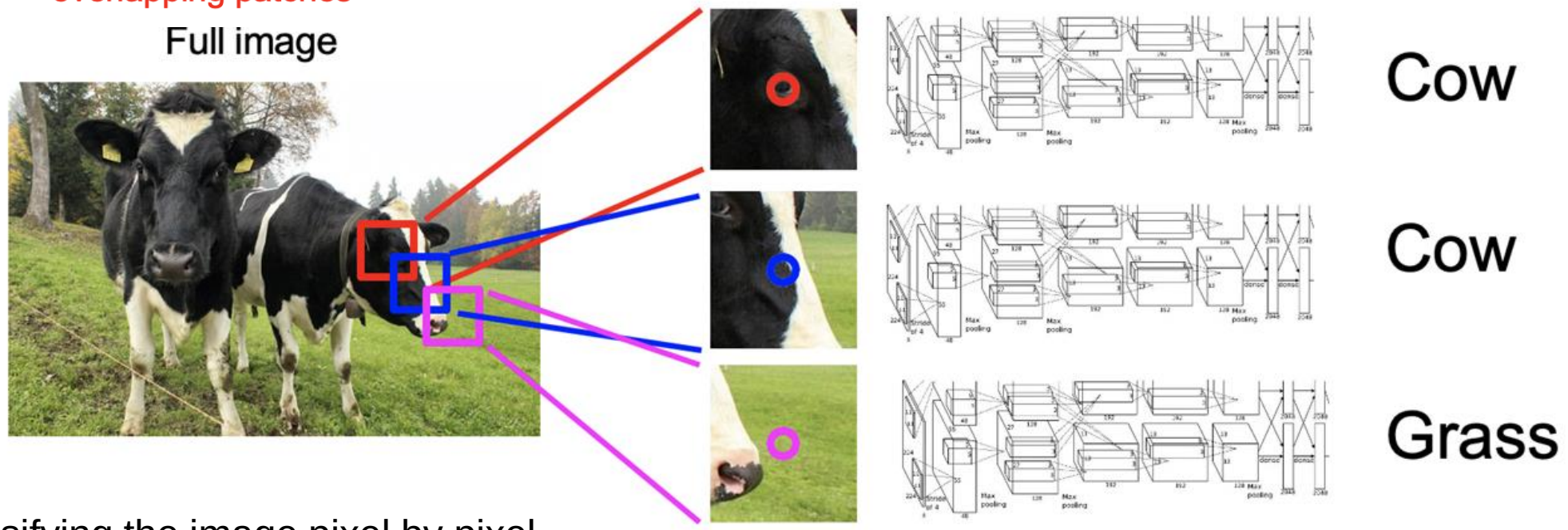
GRASS, CAT,
TREE, SKY, ...

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

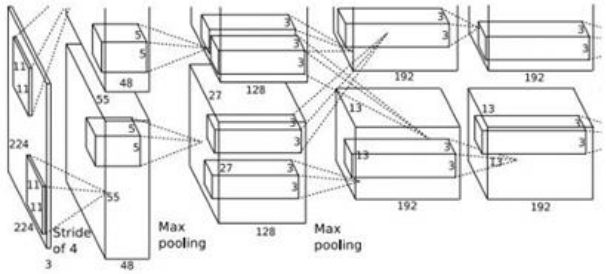


Classifying the image pixel by pixel is impossible without context. Processing small patches give some context for classifying

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

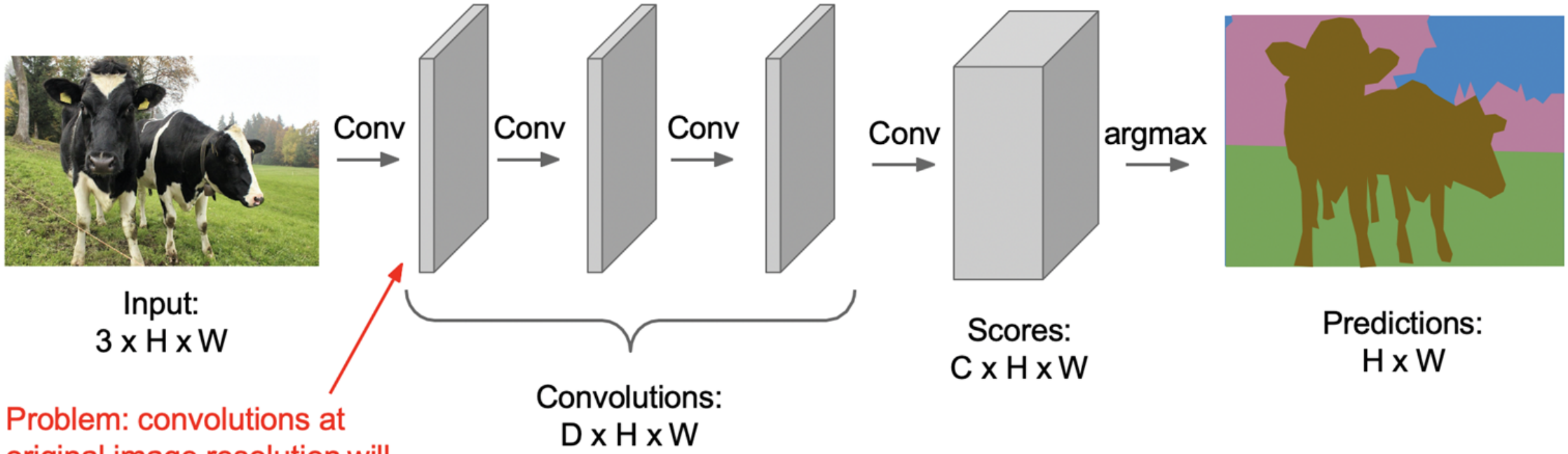


An intuitive idea: encode the entire image with conv net, and do semantic segmentation on top.

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!



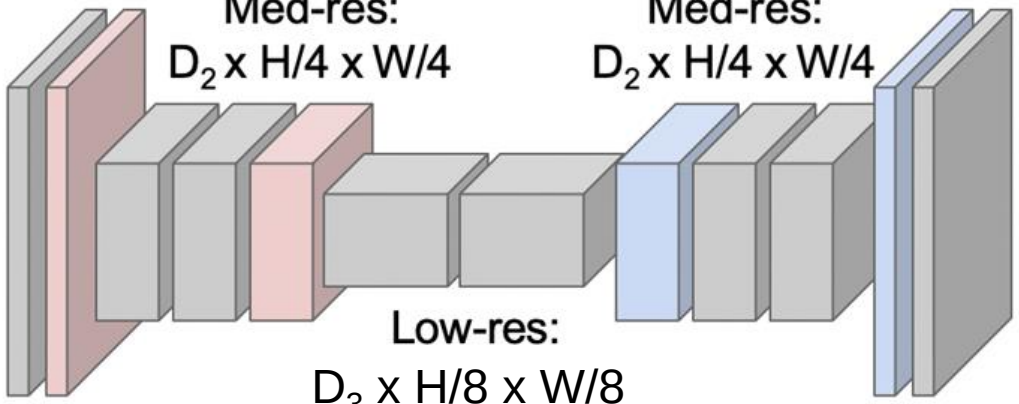
Semantic Segmentation Idea: Fully Convolutional

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



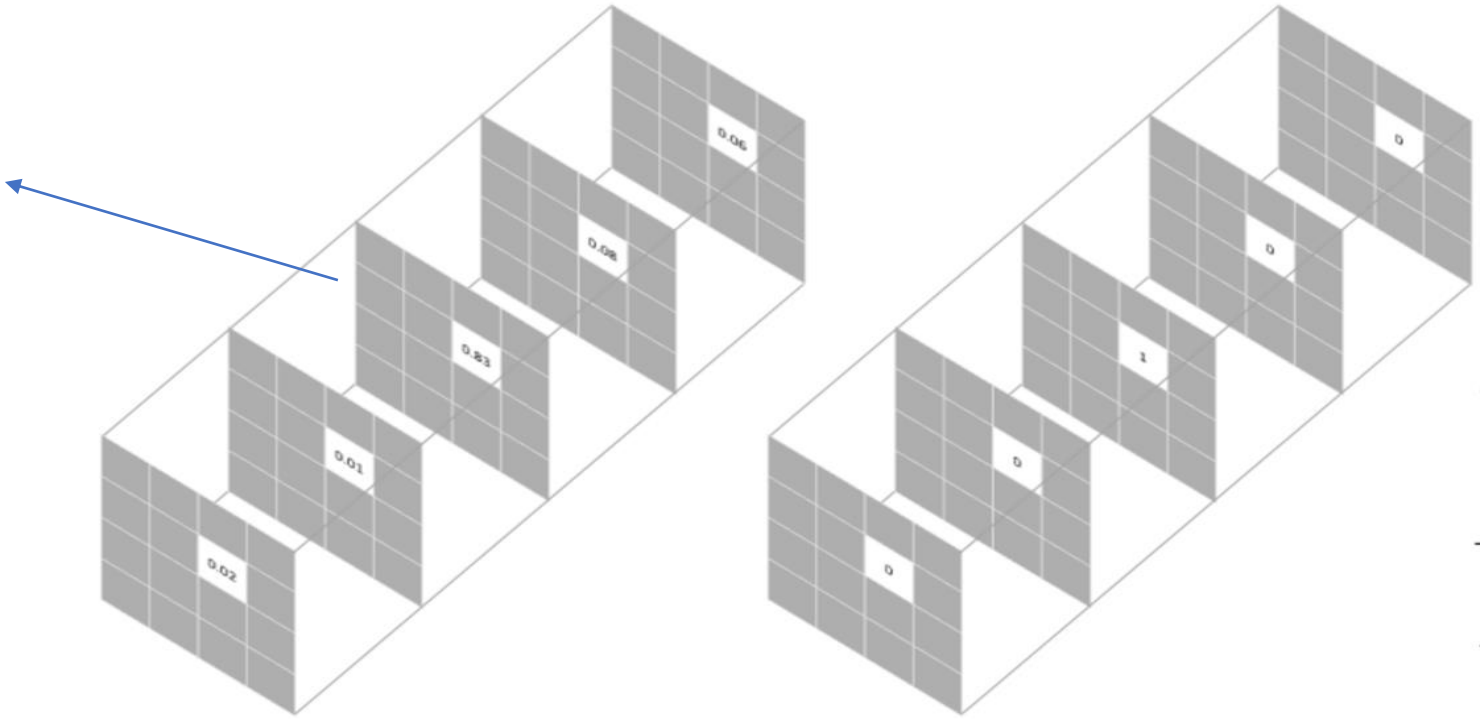
Predictions:
H x W

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 basically 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

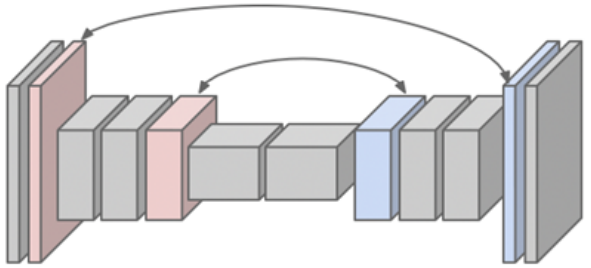
1	2
3	4

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



Upsampling – Transposed Convolution

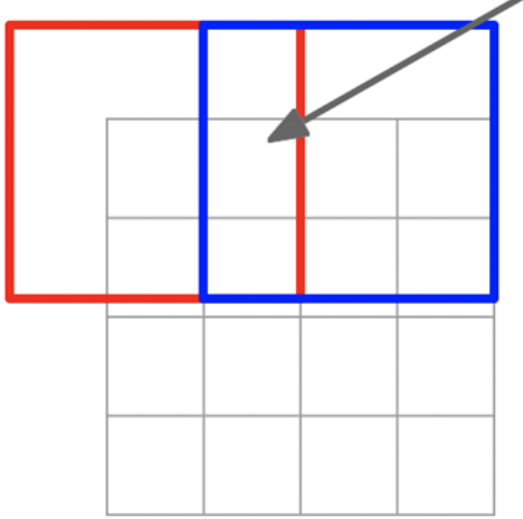
Other names:
-Deconvolution (bad)
-Upconvolution
-Fractionally strided convolution
-Backward strided convolution

Q: Why is it called transpose convolution?

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

Input: 2 x 2

Input gives weight for filter



Output: 4 x 4

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

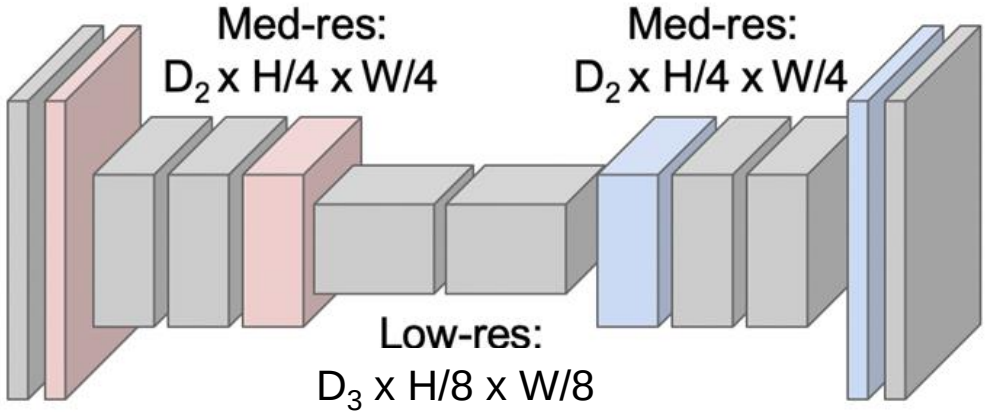
Semantic Segmentation Idea : Fully Convolutional

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

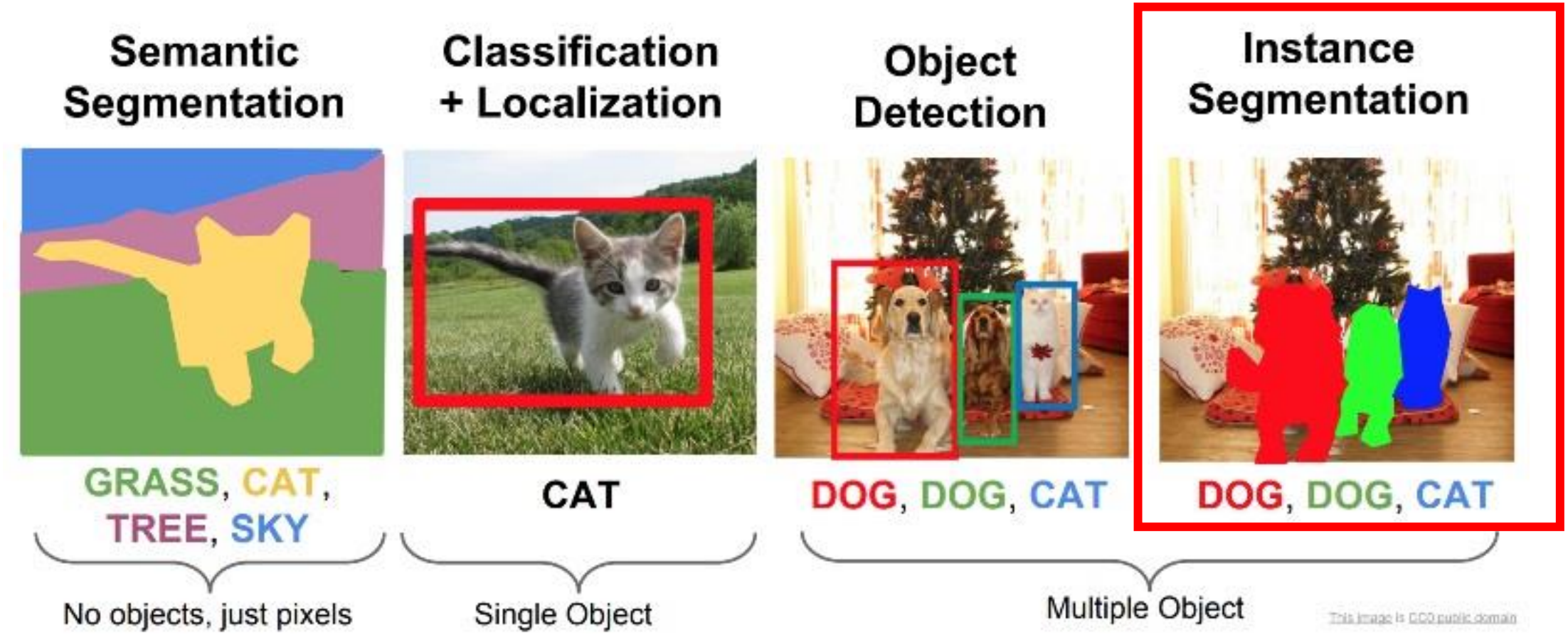
Semantic Segmentation : Evaluation Metrics

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:

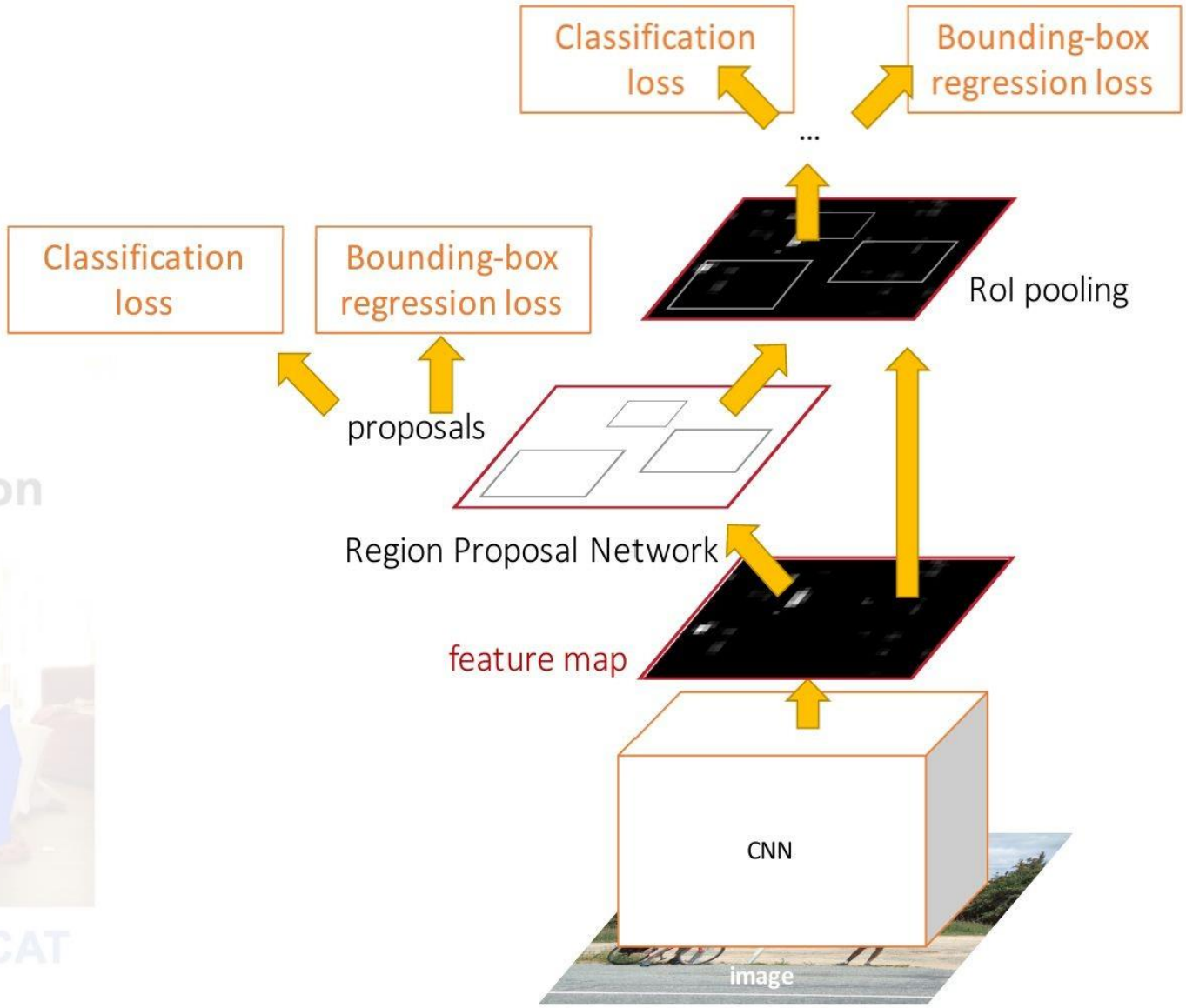
- pixel accuracy: $\sum_i n_{ii} / \sum_i t_i$
- mean accuraccy: $(1/n_{\text{cl}}) \sum_i n_{ii}/t_i$
- mean IU: $(1/n_{\text{cl}}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$
- frequency weighted IU: $\left(\sum_k t_k\right)^{-1} \sum_i t_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii}\right)$

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

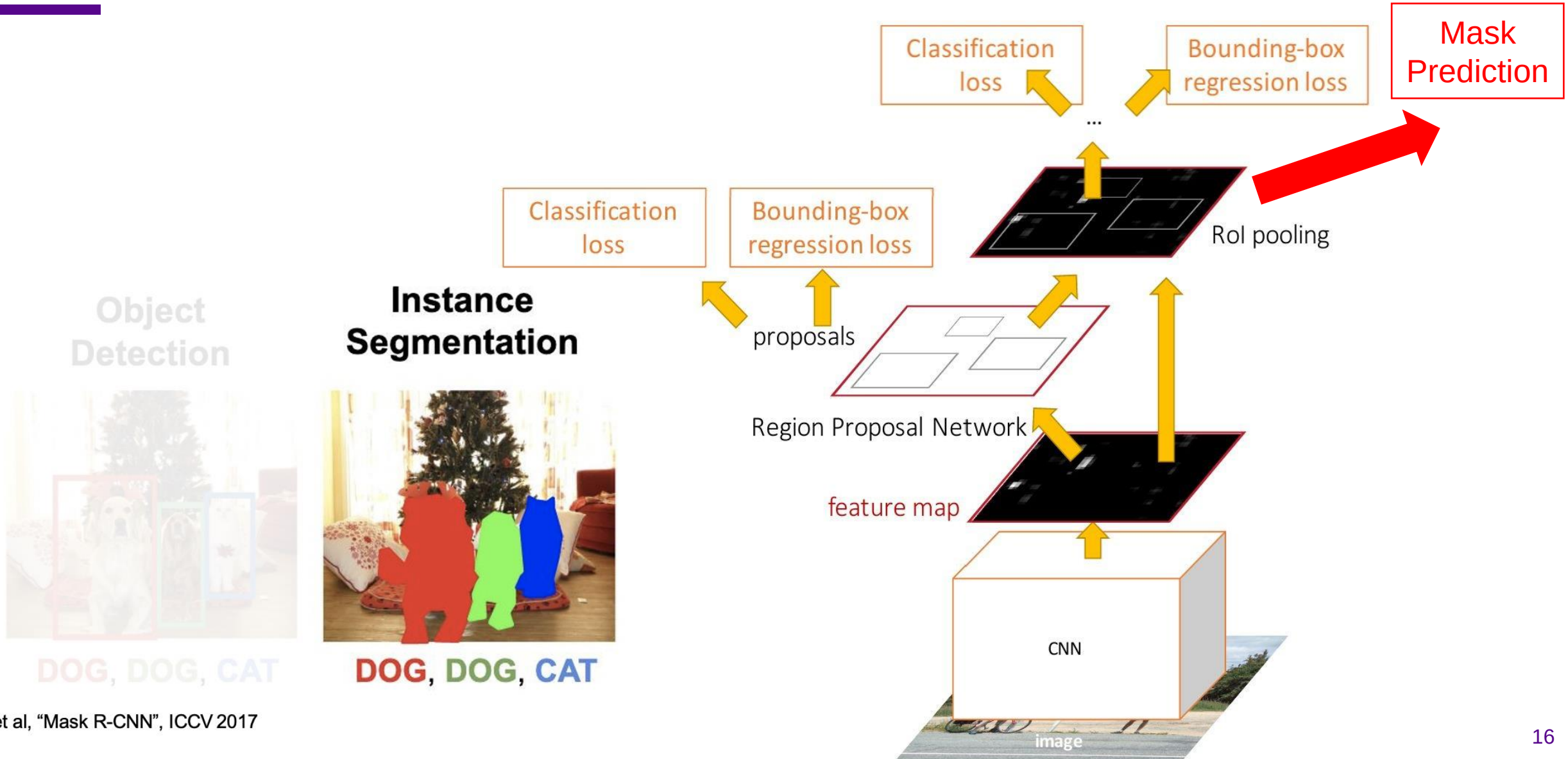
Instance Segmentation



Faster R-CNN- Recall



Instance Segmentation – Mask R-CNN

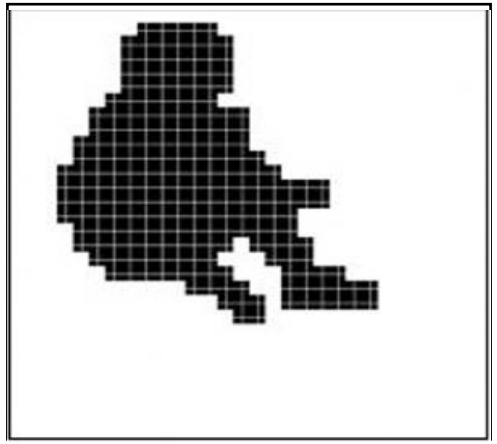


He et al, "Mask R-CNN", ICCV 2017

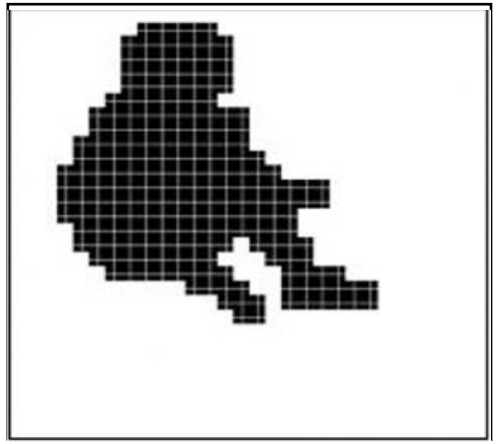
Instance Segmentation – Mask R-CNN Loss

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

Ground truth mask



Predicted mask

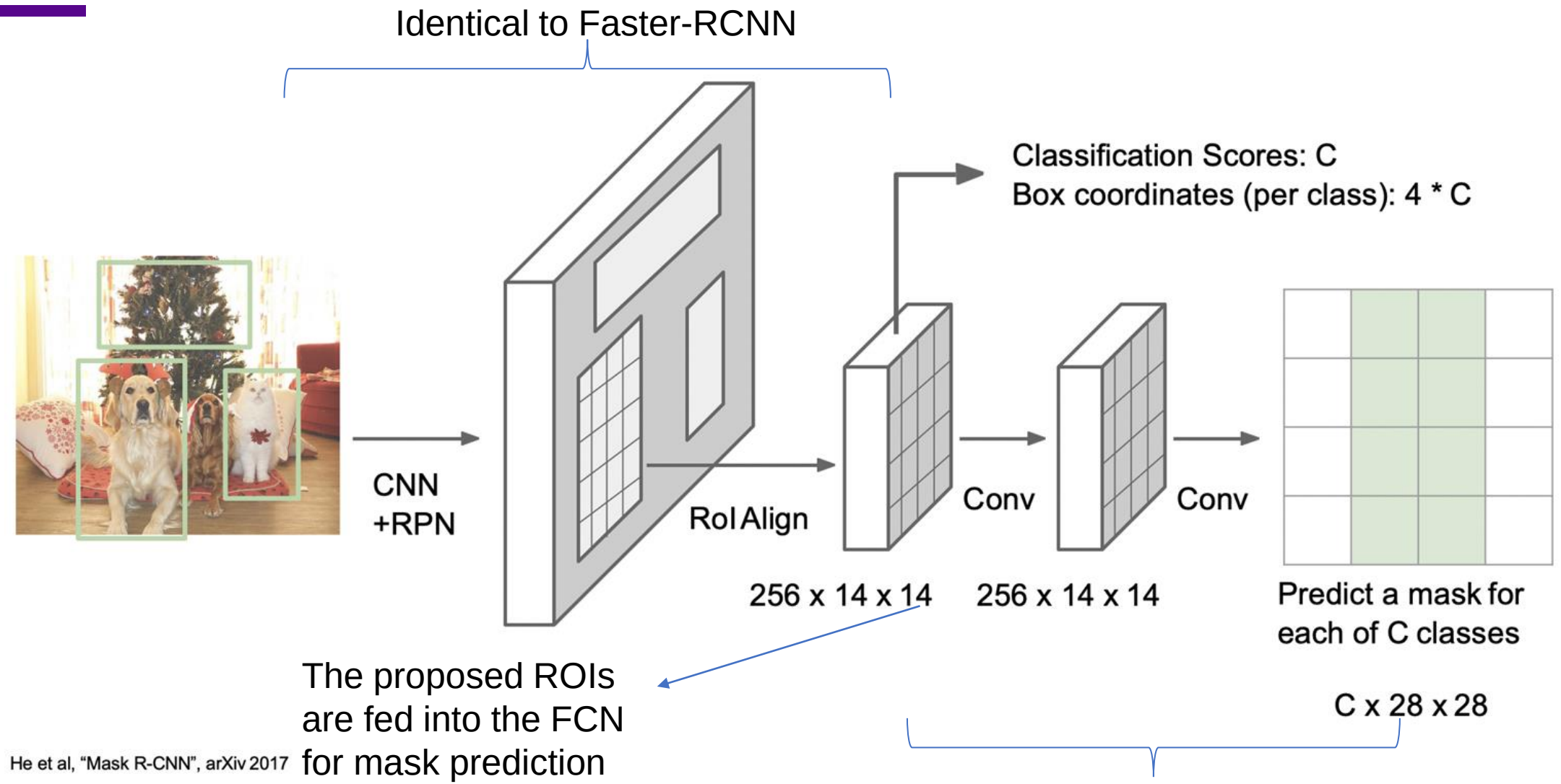


$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} \left[y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k) \right]$$

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.

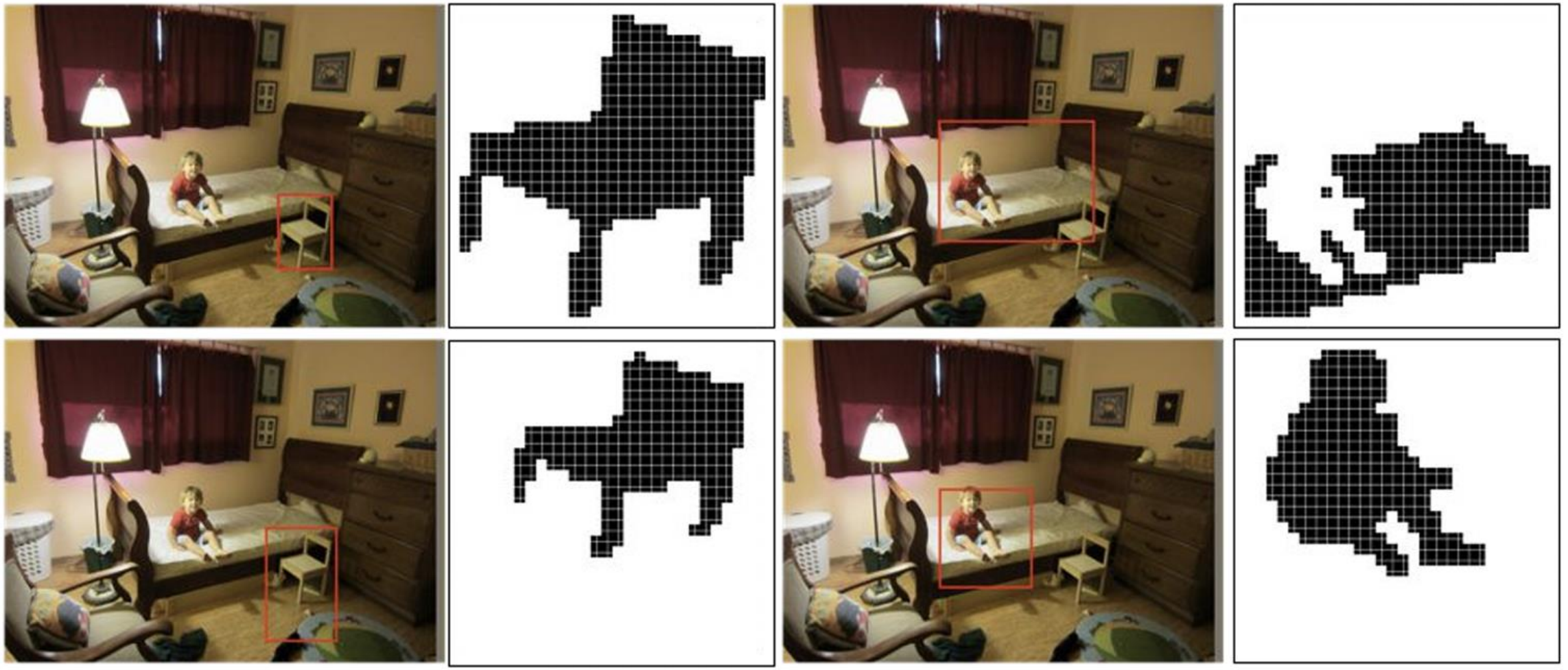
The remaining loss terms are exactly from Faster R-CNN (Lecture 10, Slide 77)

Mask R-CNN : Architecture



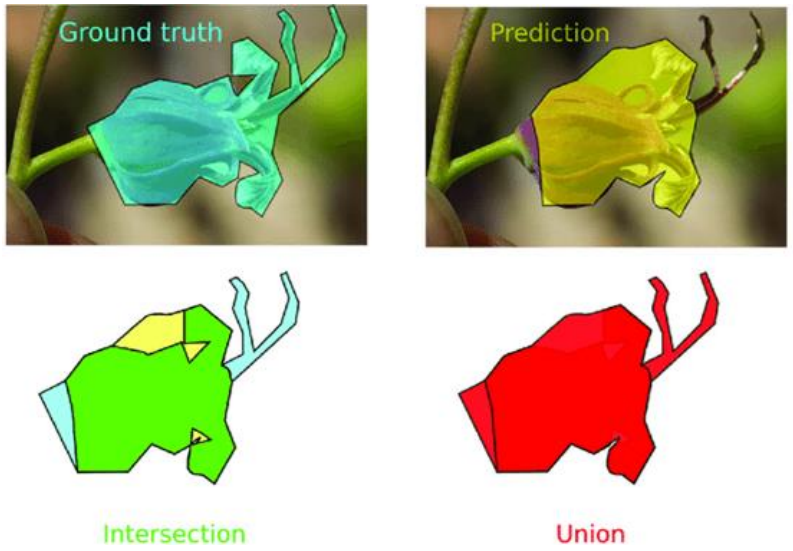
He et al, "Mask R-CNN", arXiv 2017

Mask R-CNN: Example Mask Training Targets



Mask R-CNN: Evaluation

- Mask IoU (Intersection over Union)

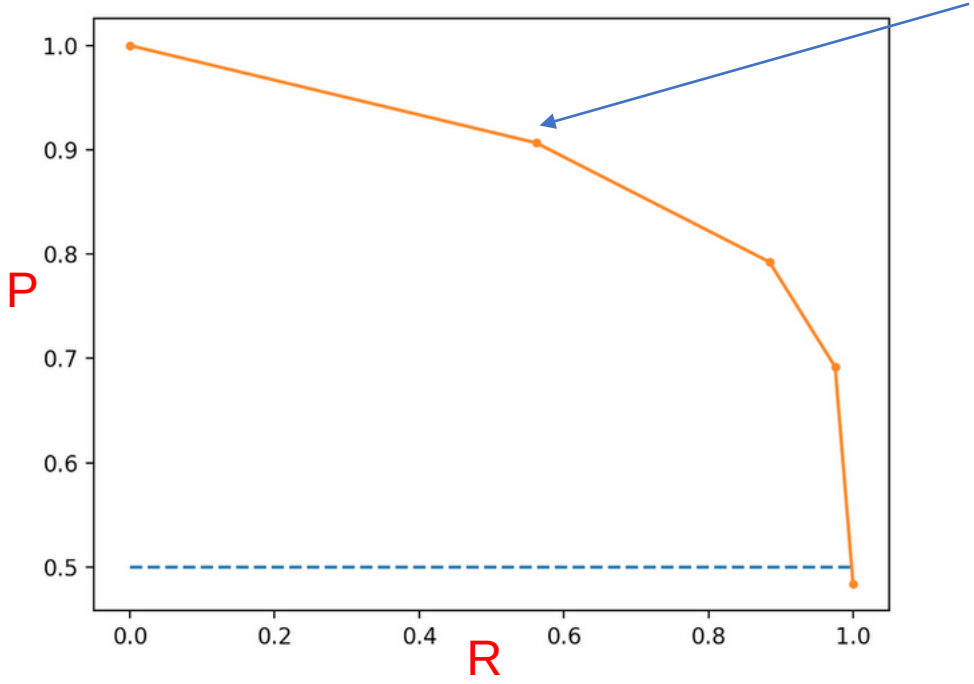


$$IoU = \frac{\text{Intersection}}{\text{Union}}$$

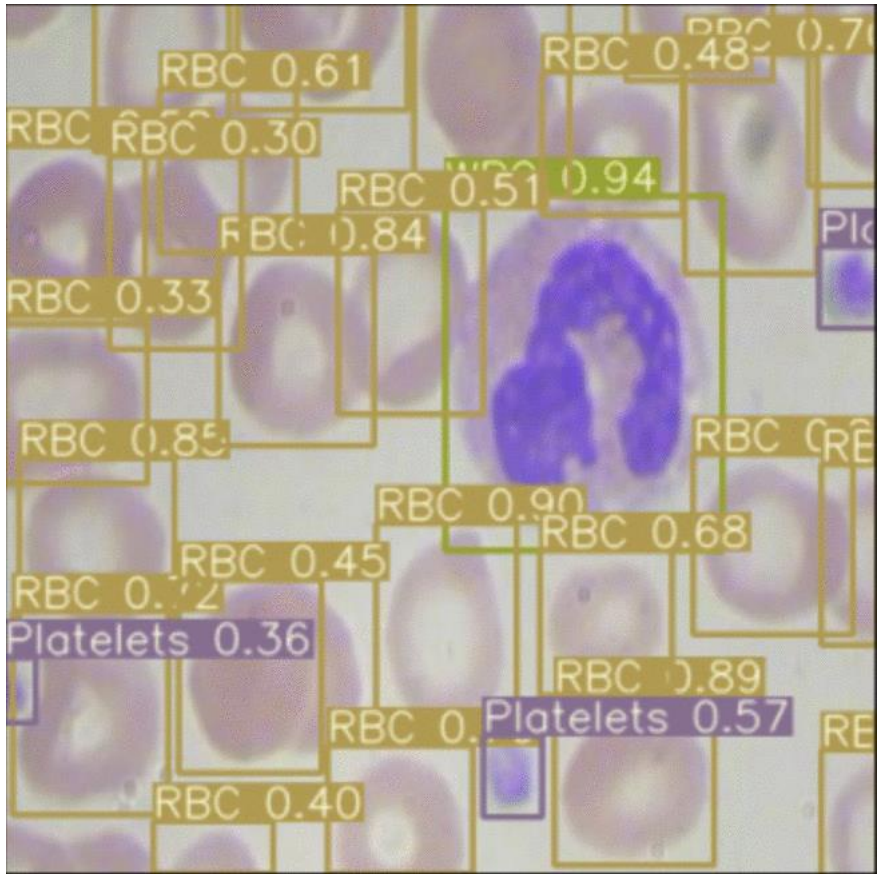
- mAP (Mean average precision)
  - averaged over categories and IoU thresholds

Mean Average Precision

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



Precision and recall is plotted for each confidence threshold

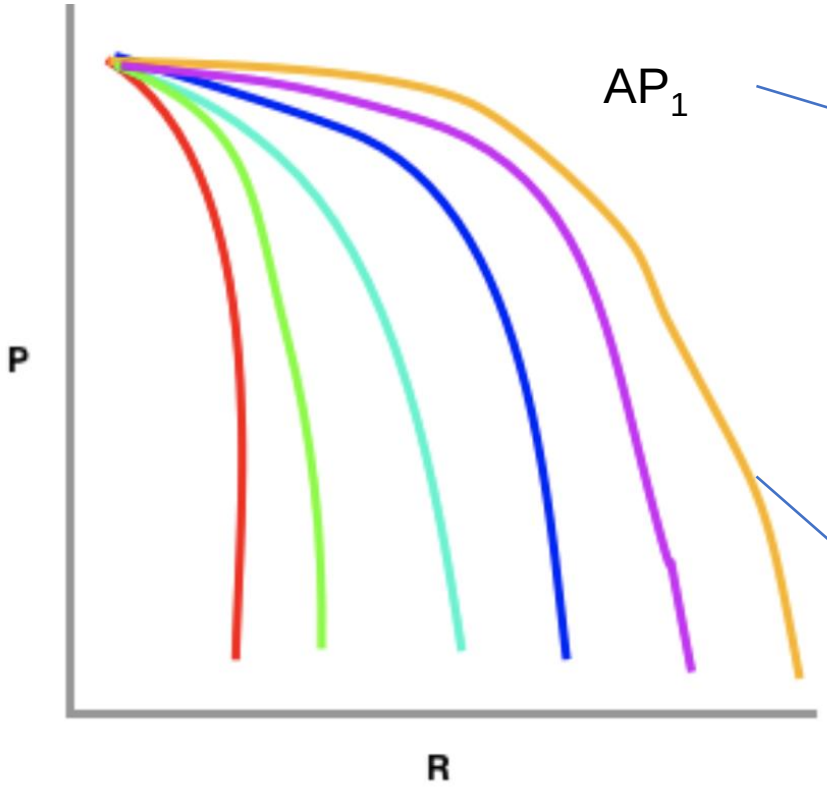


Average Precision can be computed by finding the Area Under the Curve (AUC) of the PR curve.

$$\mathrm{AP} = \int_0^1 p(r)dr$$

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

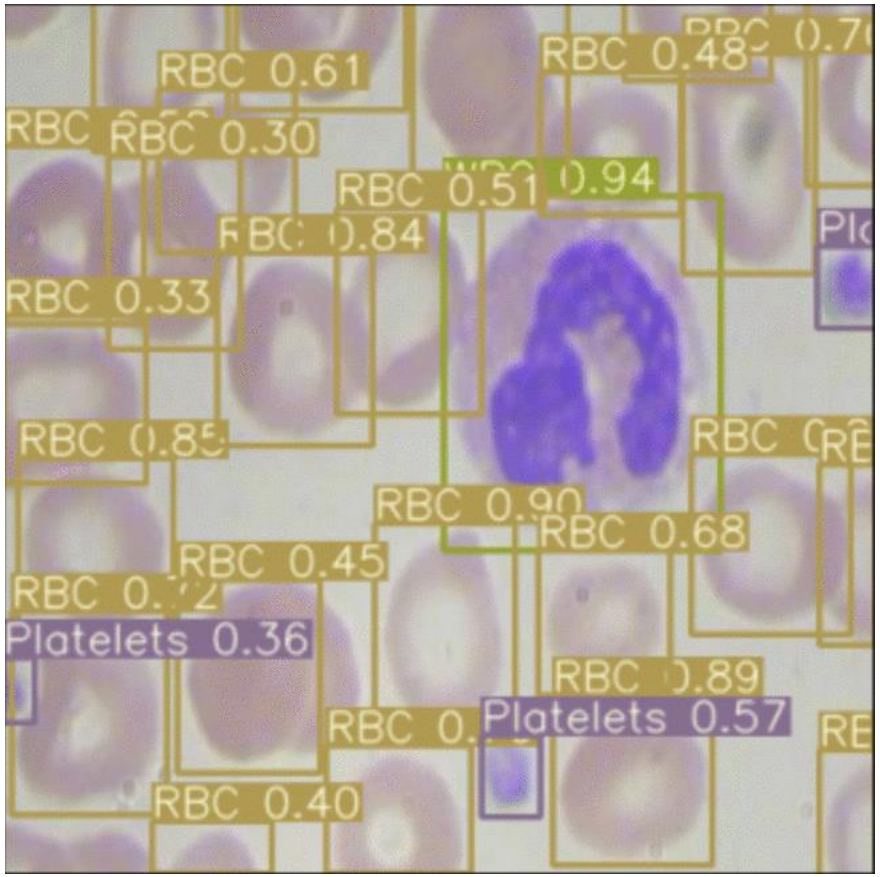


Class - A

AP_1

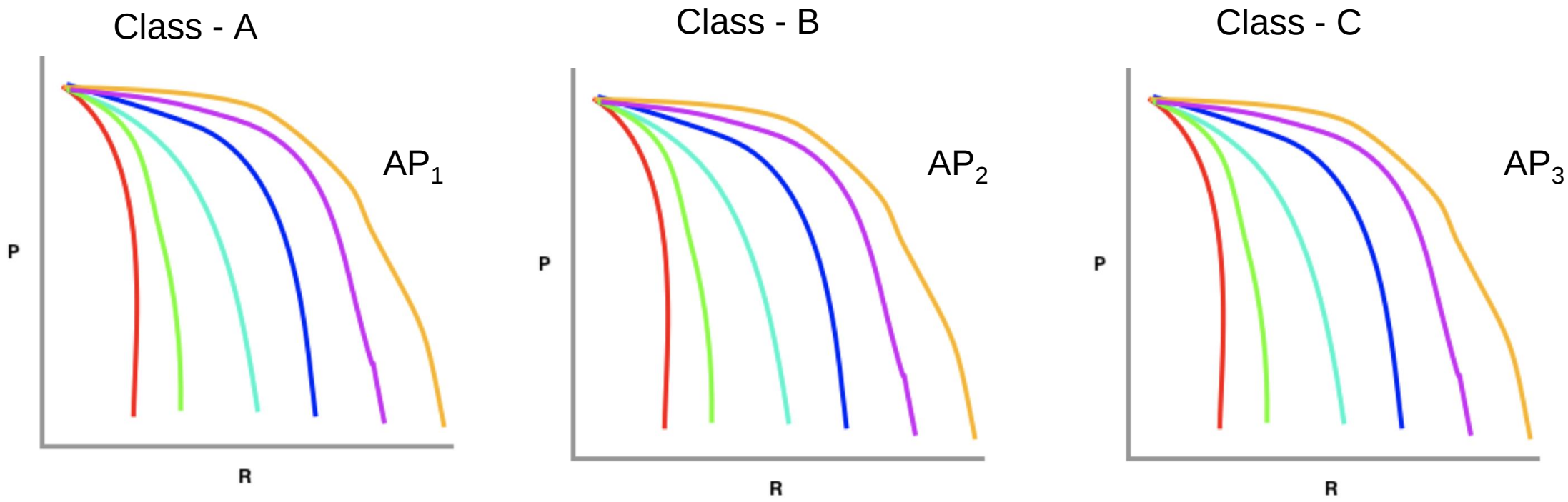
We can compute the AP for this class by averaging the Average Precisions for each IoU

Each curve corresponds to a IoU threshold



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

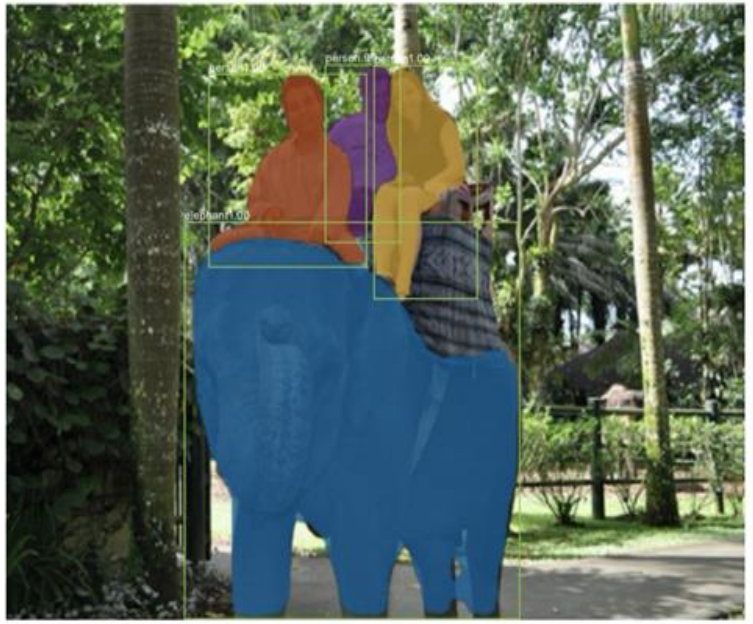
Mean Average Precision

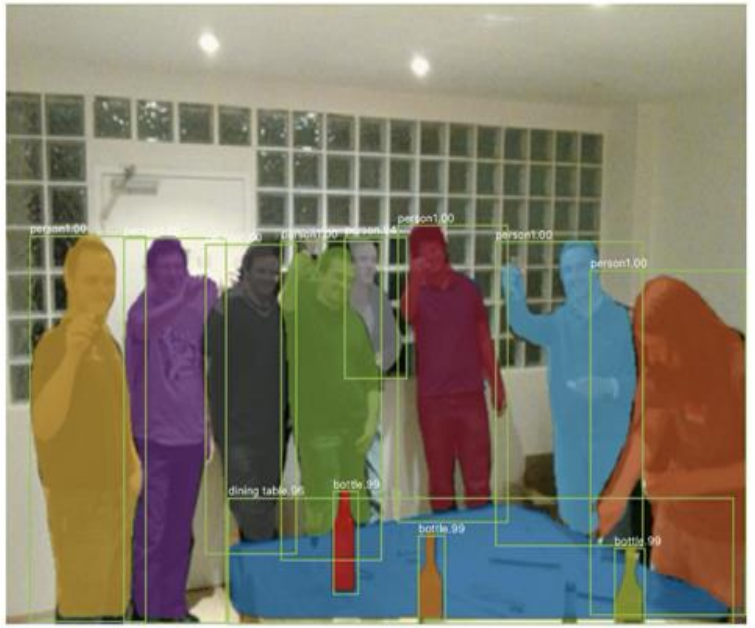


Draw multiple Precision recall curves for each class

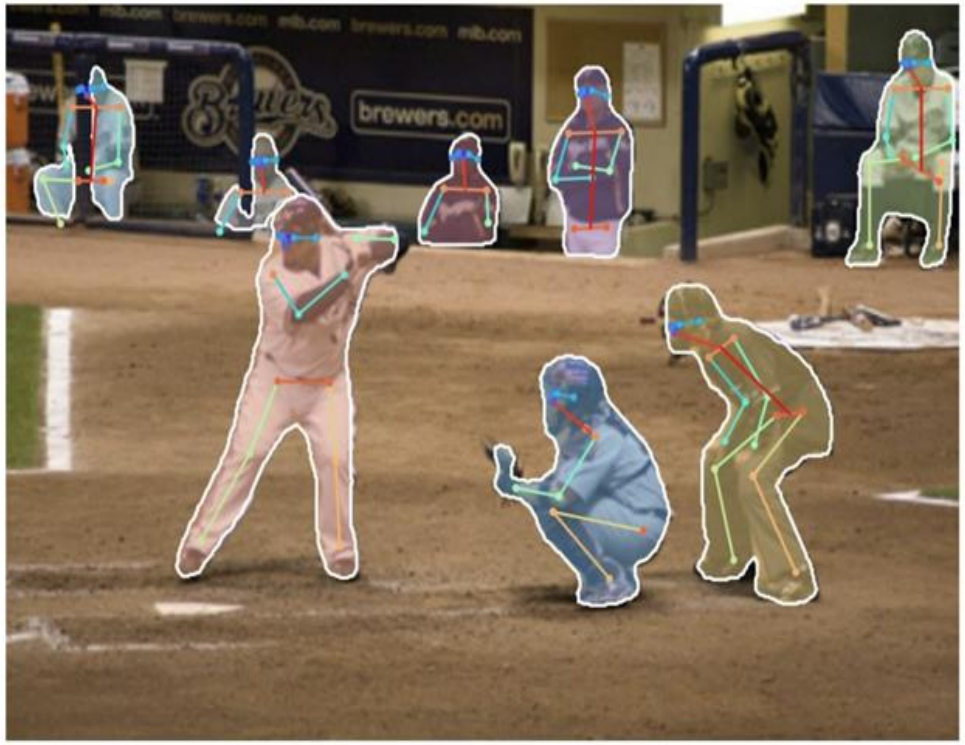
Mean Average Precision = Mean of Average Precisions of each class = 1/3 (AP_1 + AP_2 + AP_3)

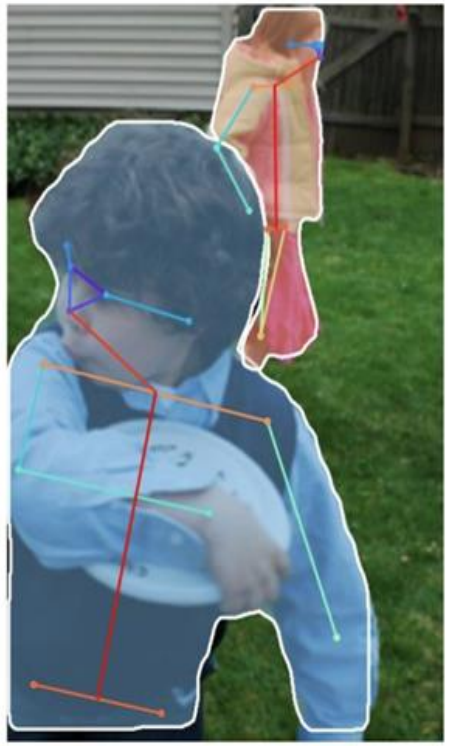
Mask R-CNN : Results





Mask R-CNN : Human Pose Estimation





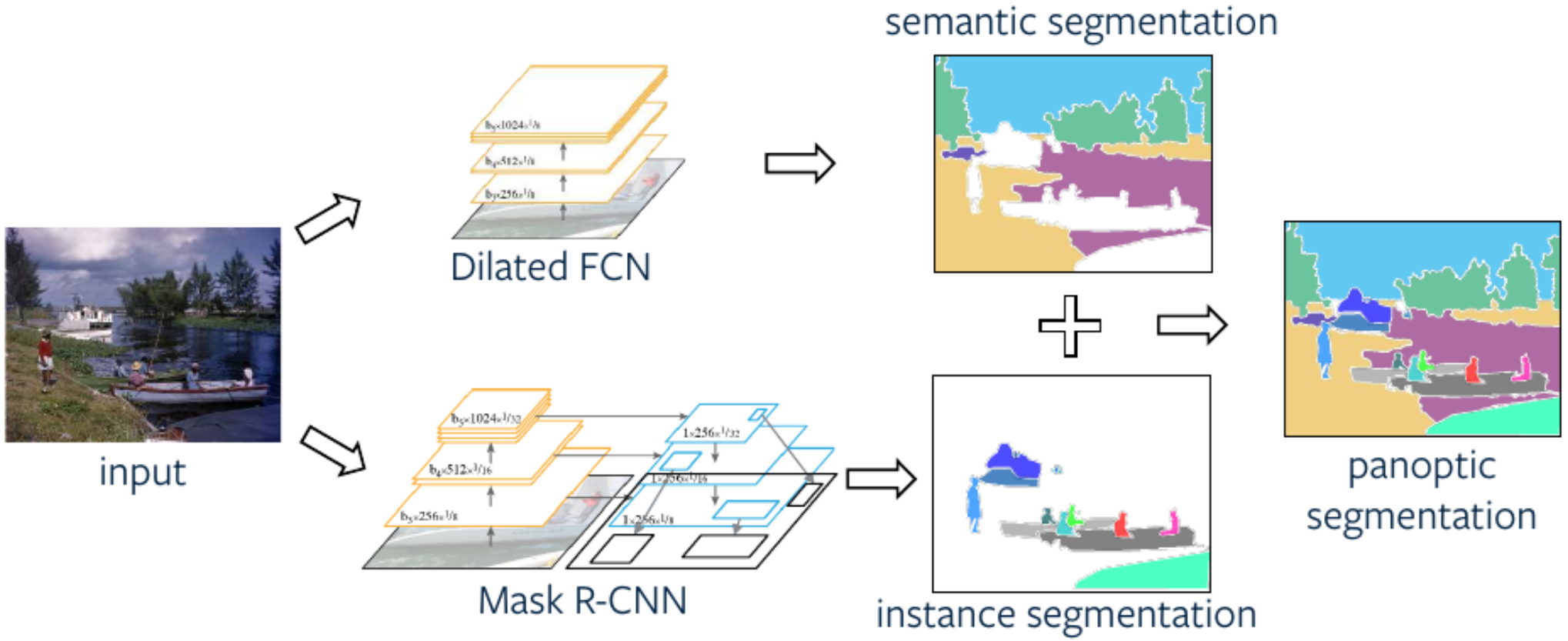
Panoptic Segmentation : Task

- 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

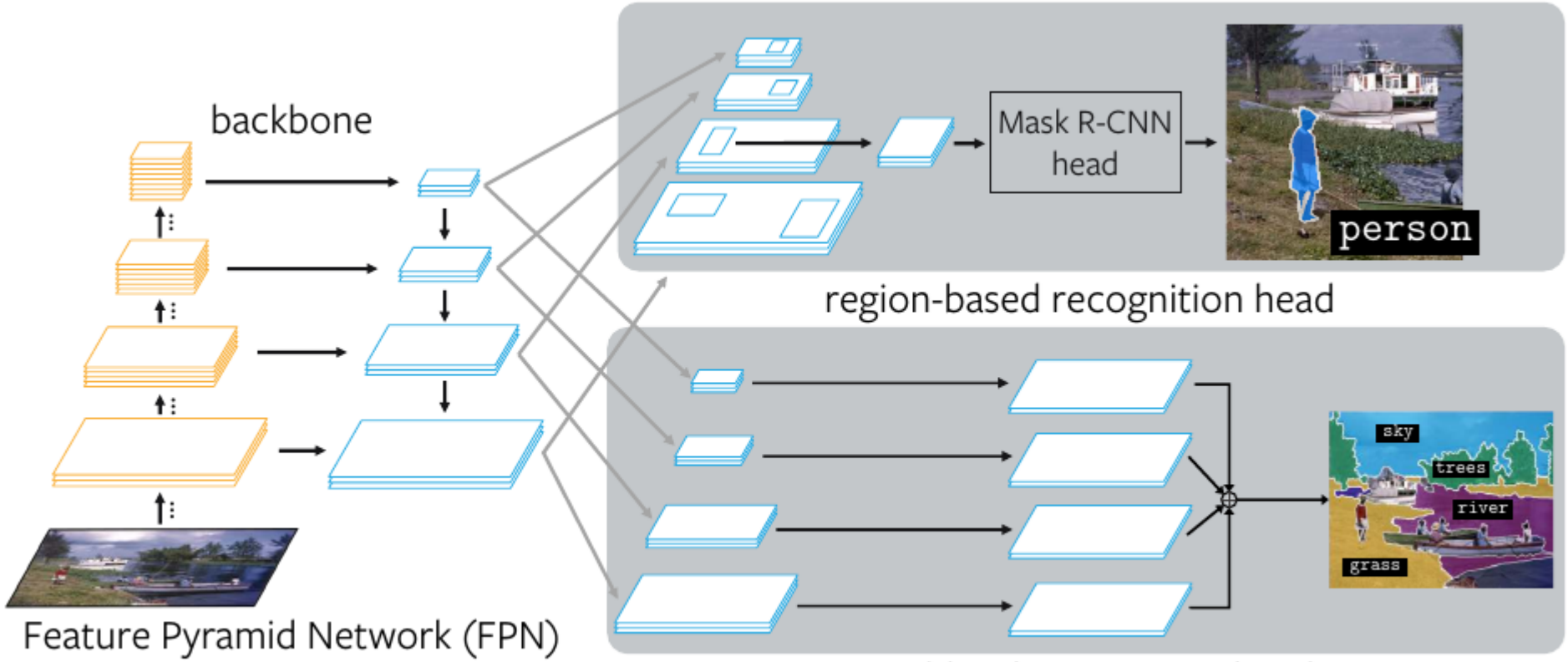
Panoptic Segmentation : Naïve Model

- 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

Panoptic Segmentation : Panoptic FPN Training

- During training, the instance segmentation branch has three losses **(classification loss), (bounding-box loss),** and **(mask loss)**. The semantic segmentation branch has semantic loss computed as the per-pixel cross-entropy between the predicted and the ground truth labels
- In addition, a weighted combination of the semantic and instance loss is used by adding two tuning parameters to get the panoptic loss.

$$L = \lambda_i(L_{cls} + L_{bbox} + L_{mask}) + \lambda_s L_s$$

- It is observed that the losses from these two branches have different scales and normalization policies. Simply adding them degrades the final performance for one of the tasks

Panoptic Segmentation : Evaluation metrics

- **Panoptic Quality (PQ)**

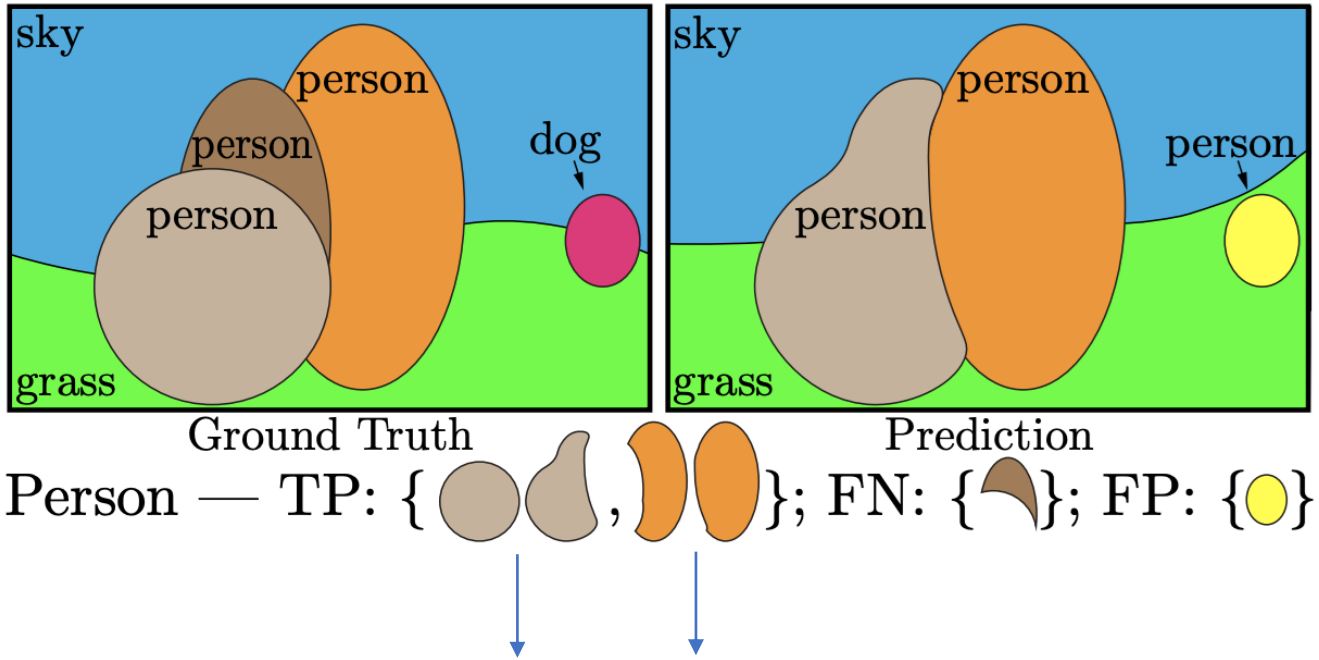
$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{recognition quality (RQ)}}$$

- Mean IoU (Intersection over Union)
- Mean Average Precision

Segmentation quality (SQ) is evaluating how closely matched our segments are with their ground truths

This metric is a combination of precision and recall, attempting to identify how effective our model is at getting a prediction right

Panoptic Quality - Example



$SQ_{\{Person\}}$ = (Iou1 + Iou2) / 2 = 0.775

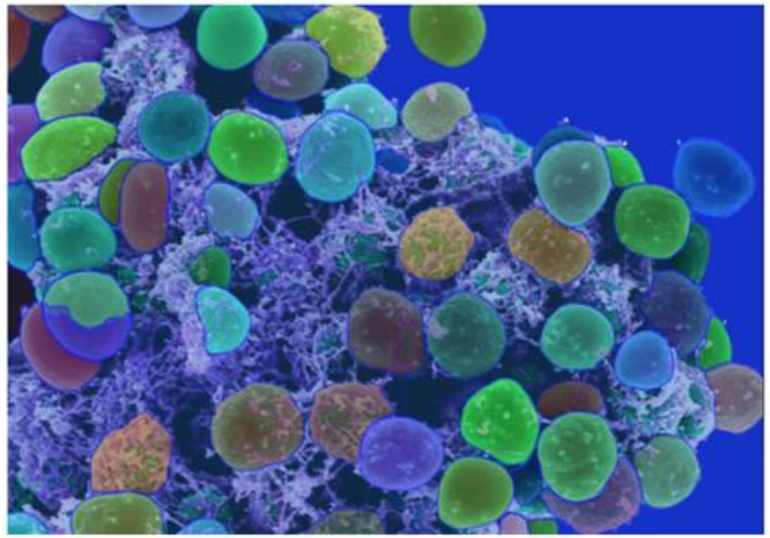
$RQ_{\{Person\}}$ = 2 / (2 + 0.5*1 + 0.5*1) = 0.66

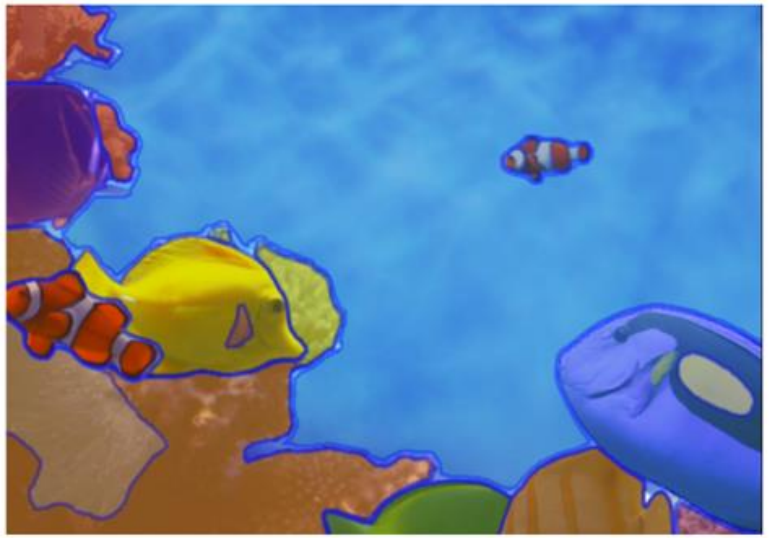
$PQ_{\{Person\}} = SQ_{\{Person\}} * RQ_{\{Person\}}$ = 0.516

$|TP|_{\{Person\}}$ = 2 $|FP|_{\{Person\}}$ = 1 $|FN|_{\{Person\}}$ = 1

Kirillov, A., He, K., Girshick, R., Rother, C. and Dollár, P., 2019. Panoptic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9404-9413).

State-of-the-Art: Segment Anything - Introduction





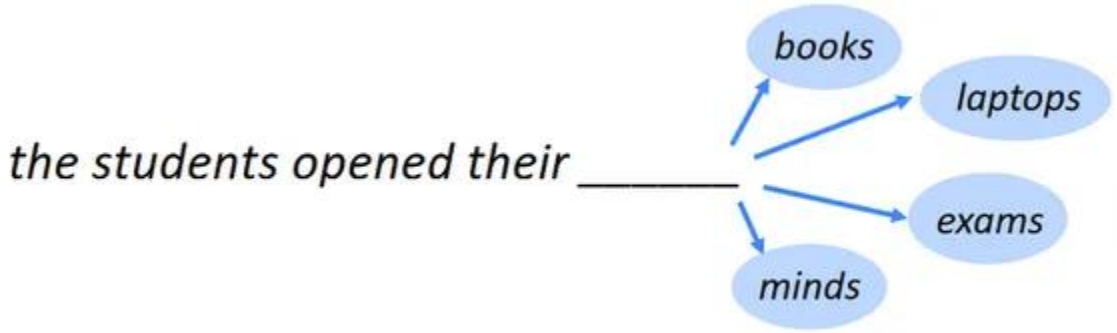
SOTA model called Segment Anything from Meta. Almost able to segment anything from any given image.

Segment Anything, Alexander Kirillov et al., 2023

State-of-the-Art: Segment Anything – The Idea

We need to talk about **GPT** (as in ChatGPT) before we talk about Segment Anything.

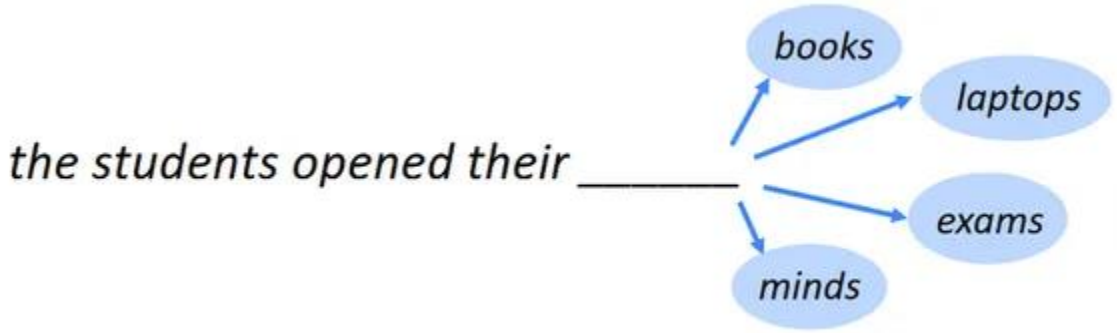
Researchers in Natural Language Processing noticed that, training a very large model with an insanely huge amount of data on the "**next-word prediction**" task make the said model grasp the "essence" of human languages.



State-of-the-Art: Segment Anything – The Idea

We need to talk about **GPT** (as in ChatGPT) before we talk about Segment Anything.

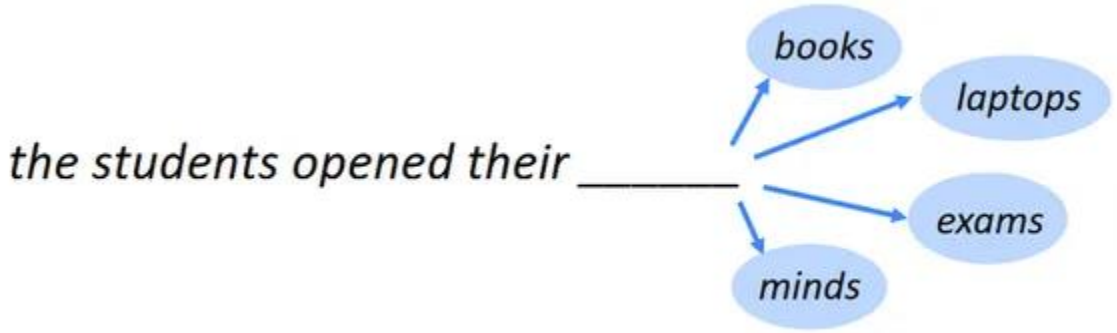
A model **pre-trained** this way can perform various language tasks given **a description or a few example** of the said tasks. Without the need for task-specific training data. Generative Pre-trained Transformer (GPT) is named after this.



State-of-the-Art: Segment Anything – The Idea

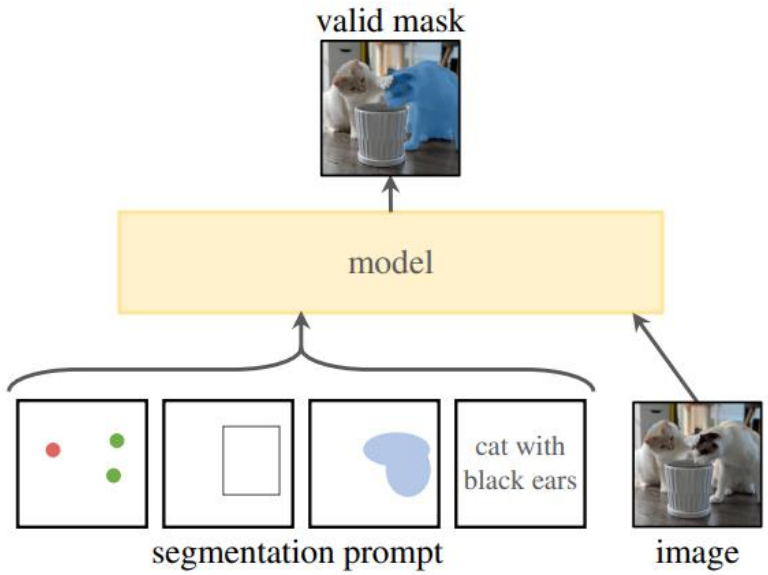
We need to talk about **GPT** (as in ChatGPT) before we talk about Segment Anything.

It would be great if we can **find a pre-training task** for perception/segmentation that is **analogous** to “next-word prediction”. Segment Anything is one step in this direction.



State-of-the-Art: Segment Anything – Methodology

In Segment Anything, the pre-training task designed is from “a set of foreground/background points, a rough box or mask, free-form text, or, in general, **any information indicating what to segment** in an image” to a **segmentation mask**.



Segment Anything, Alexander Kirillov et al., 2023

State-of-the-Art: Segment Anything – Methodology

In a sense, they **failed** to design a truly analogous pre-training task. This is still just supervised-learning.

But with **enough high-quality data**, Segment Anything is still able to achieve SOTA performance across many different segmentation tasks and be very generalizable across different domains and form of prompt

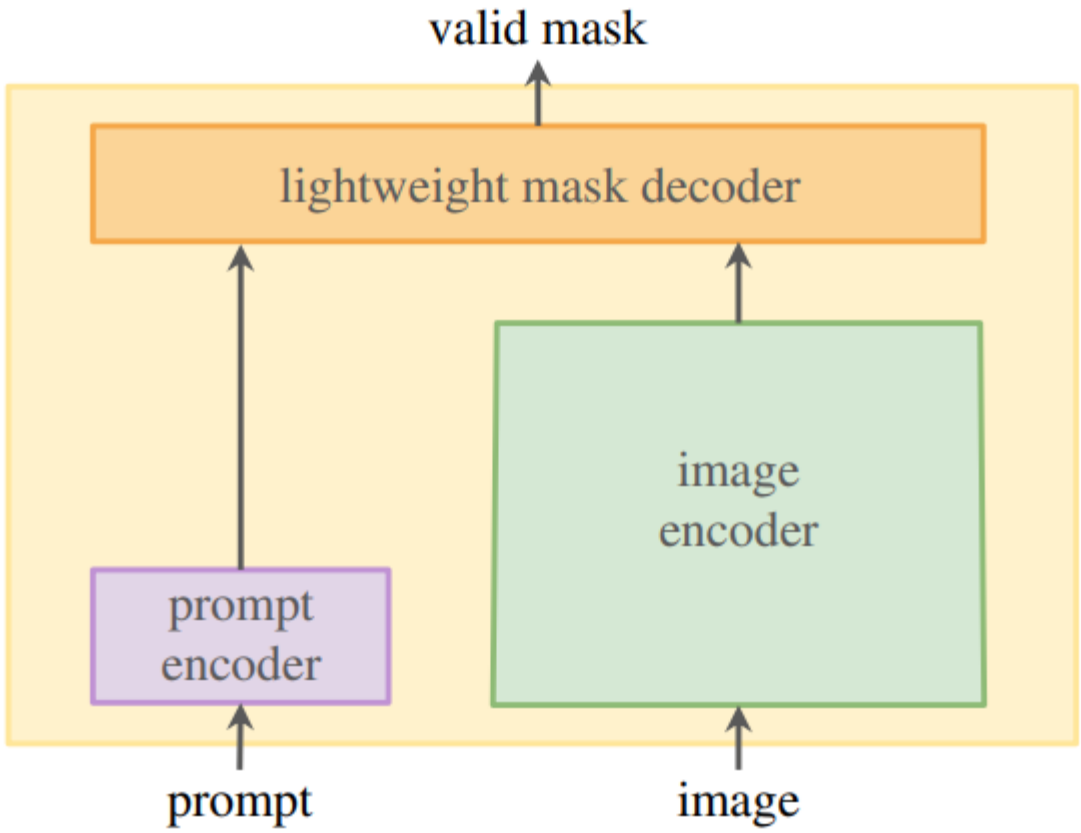
Segment Anything 1B (**SA-1B**):

- **1+ billion masks**
- 11 million images
- privacy respecting
- licensed images

Segment Anything, Alexander Kirillov et al., 2023

State-of-the-Art: Segment Anything – Methodology

The SAM model itself is not super sophisticated by today's standards.

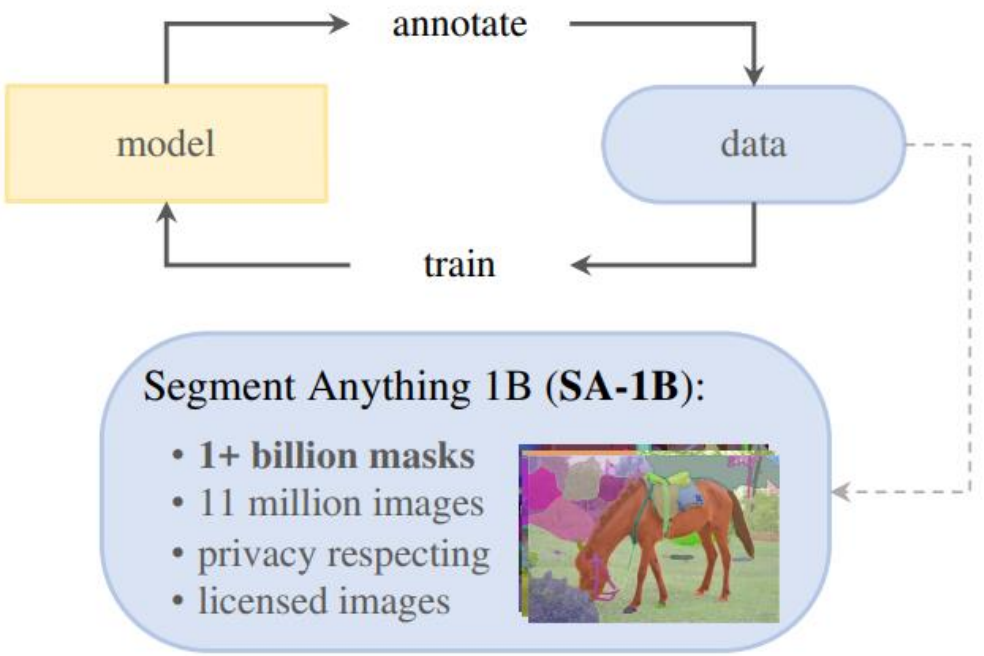


Segment Anything, Alexander Kirillov et al., 2023

State-of-the-Art: Segment Anything – Data Bootstrapping

They curate a high quality dataset full of segmentation mask by **bootstrapping**.

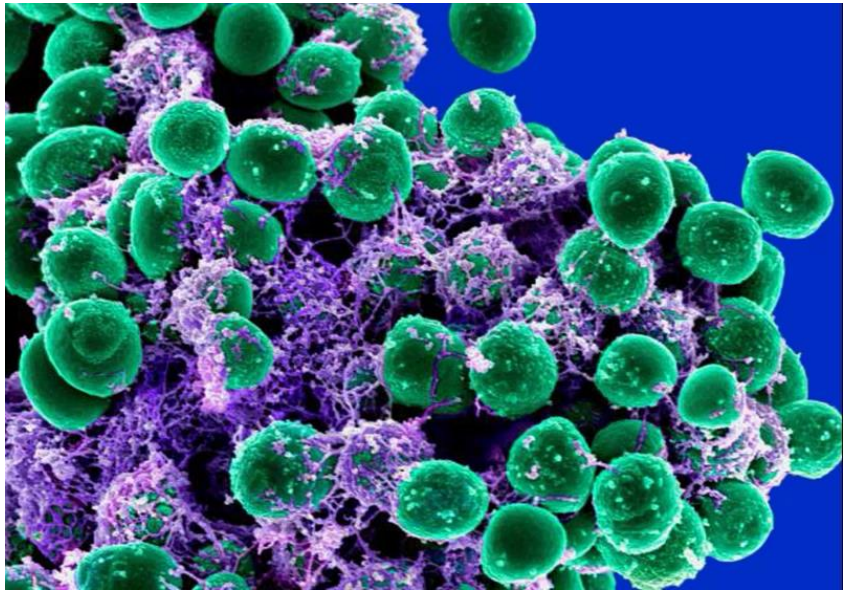
Hand annotated data first to train a small SAM model. Small SAM model to generate more segmentation mask data to train larger, more capable SAM model



Segment Anything, Alexander Kirillov et al., 2023

State-of-the-Art: Segment Anything – Results

Impressive and generalizable **zero-shot** performance.



Segment Anything, Alexander Kirillov et al., 2023

State-of-the-Art: Segment Anything – Results

Very flexible – prompt-able via gaze (for VR) and text too

Segment Anything, Alexander Kirillov et al., 2023

State-of-the-Art: Segment Anything – Results

Some particularly striking results!






Segment Anything, Alexander Kirillov et al., 2023

3D Deep Learning

3D Object Recognition via 2D Images

1. Multi-view Convolutional Neural Networks for 3D Shape Recognition

Dealing with Points Clouds

1. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition
2. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation
3. FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation
4. DeepMapping: Unsupervised Map Estimation from Multiple Point Clouds & DeepMapping 2
5. Semantic Occupancy Prediction

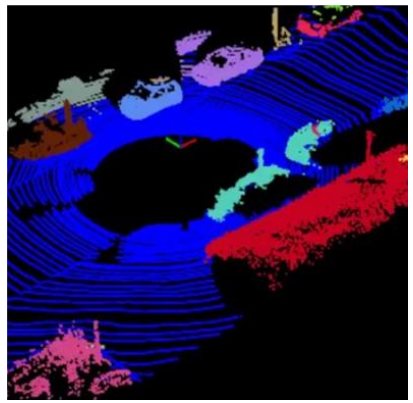
Neural Implicit 3D Segmentation

1. Panoptic Lifting for 3D Scene Understanding with Neural Fields

Why Deep Learning on 3D Data?

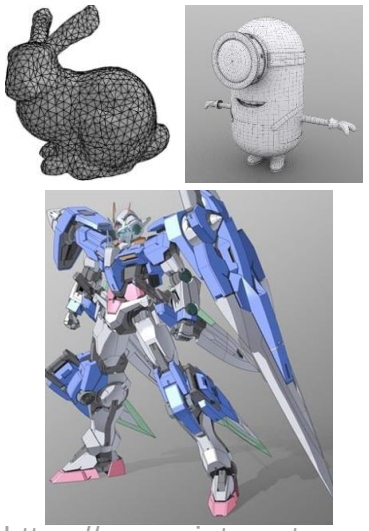
- How to enable robots to create maps and understand scenes in 3D?
- An important form of data – various application domains
- Intrinsically different than images – challenges for existing deep networks

Robotics



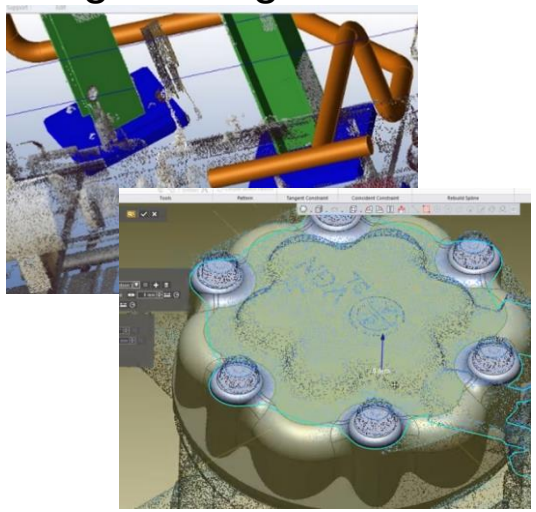
https://www.youtube.com/watch?v=7NNpvtdrHkU

Graphics/3DP



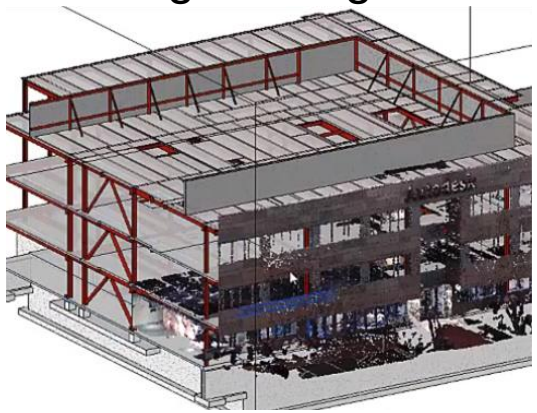
https://www.pinterest.com/pin/134756213823244639

Mechanical Engineering



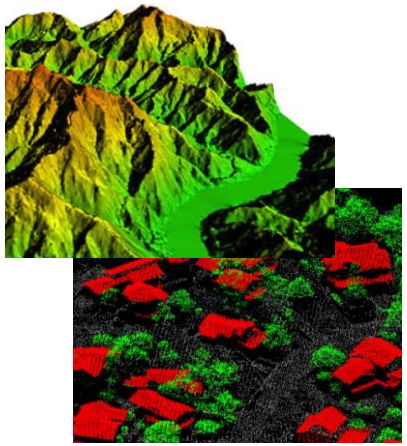
https://www.youtube.com/watch?v=UD4asn3gkNI

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

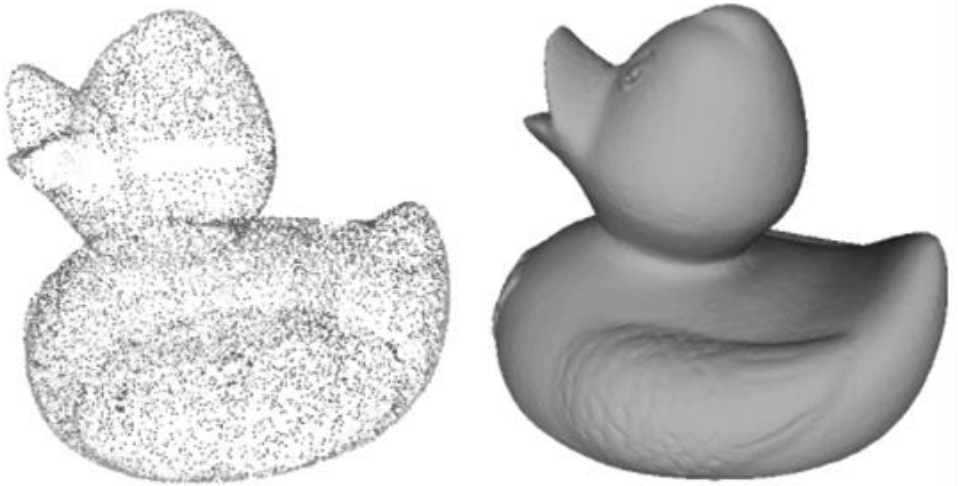
- ✓ 2D CNN
- • Generalize to points?
- × Large networks

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

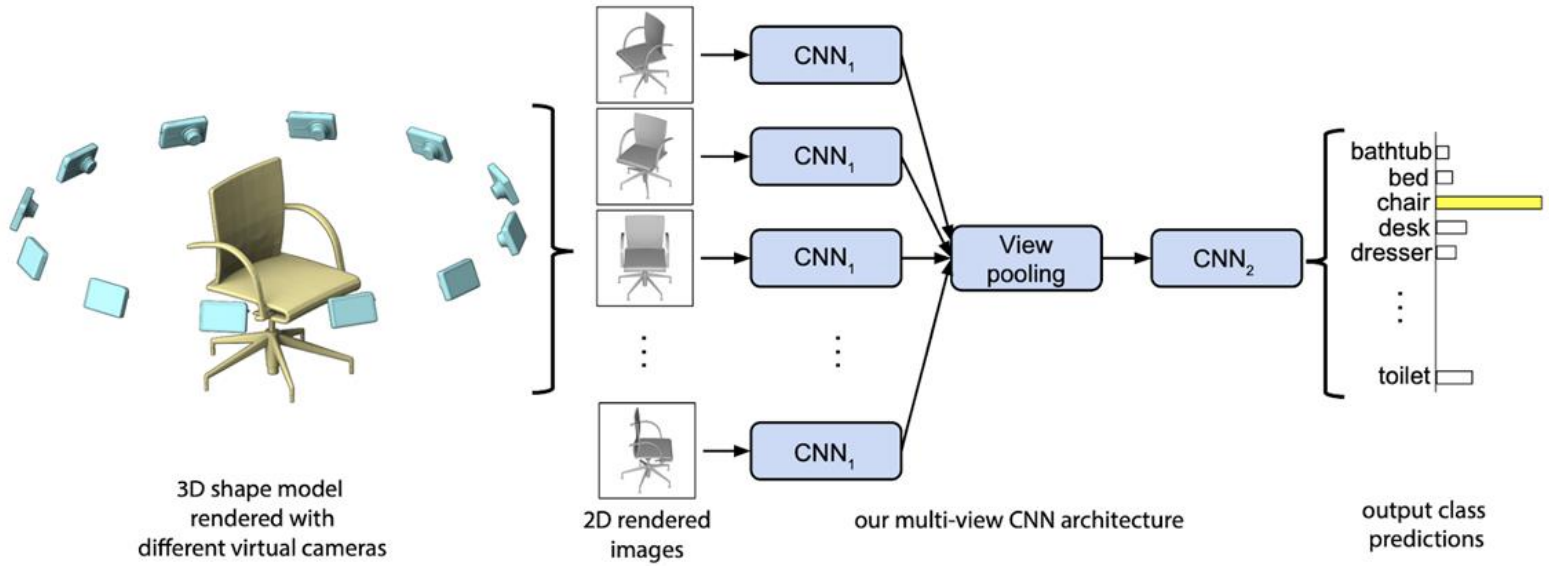


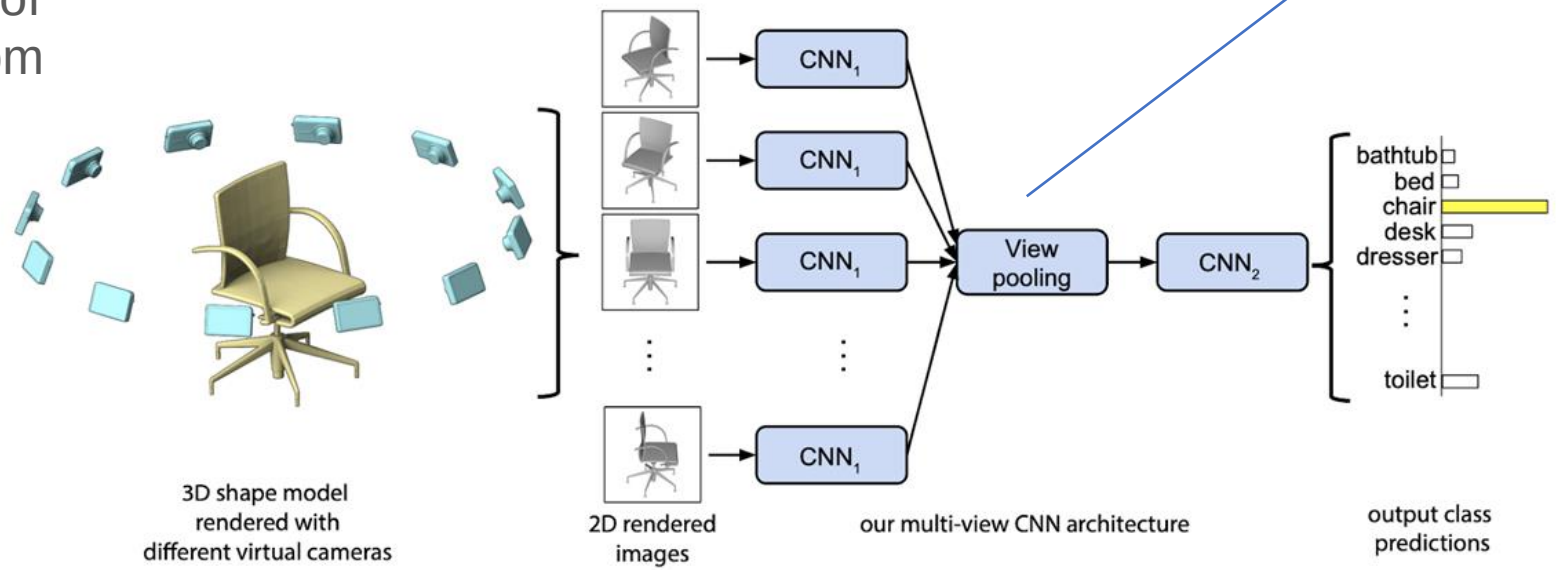
Figure 1. Multi-view CNN for 3D shape recognition (illustrated using the 1st camera setup). At test time a 3D shape is rendered from 12 different views and are passed thorough CNN_1 to extract view based features. These are then pooled across views and passed through CNN_2 to obtain a compact shape descriptor.

Multi-view Convolutional Neural Networks - Methodology

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

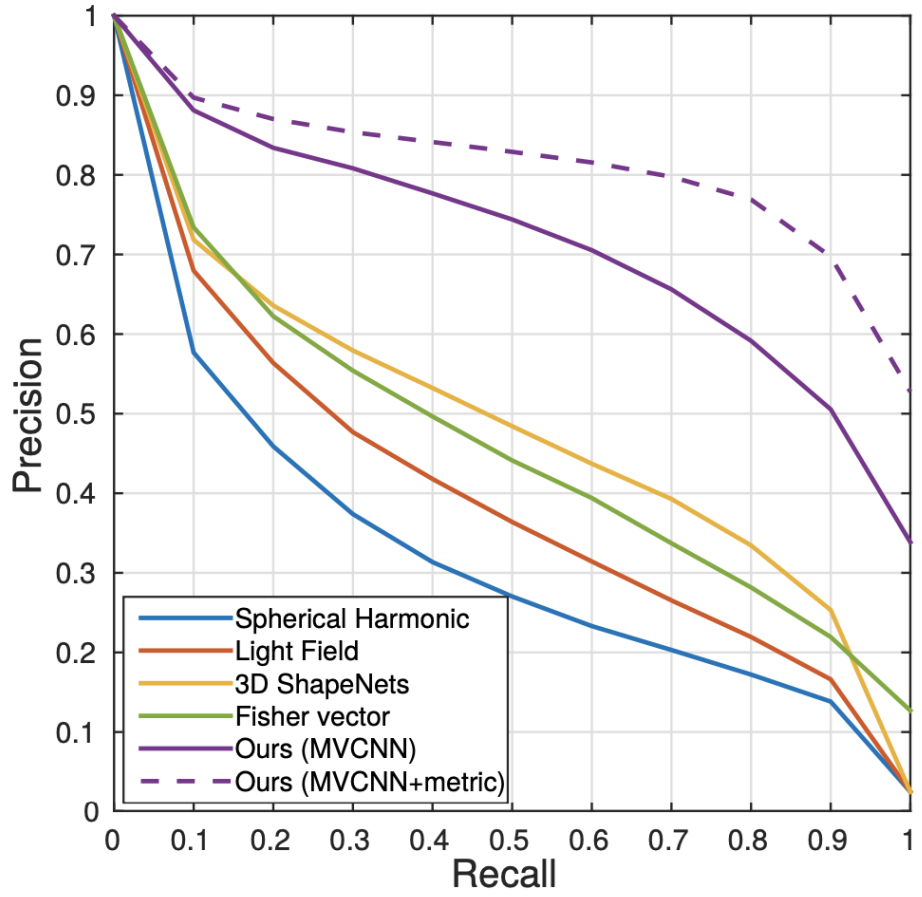


Figure 2. Precision-recall curves for various methods for 3D shape retrieval on the ModelNet40 dataset. Our method significantly outperforms the state-of-the-art on this task achieving 80.2% mAP.

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

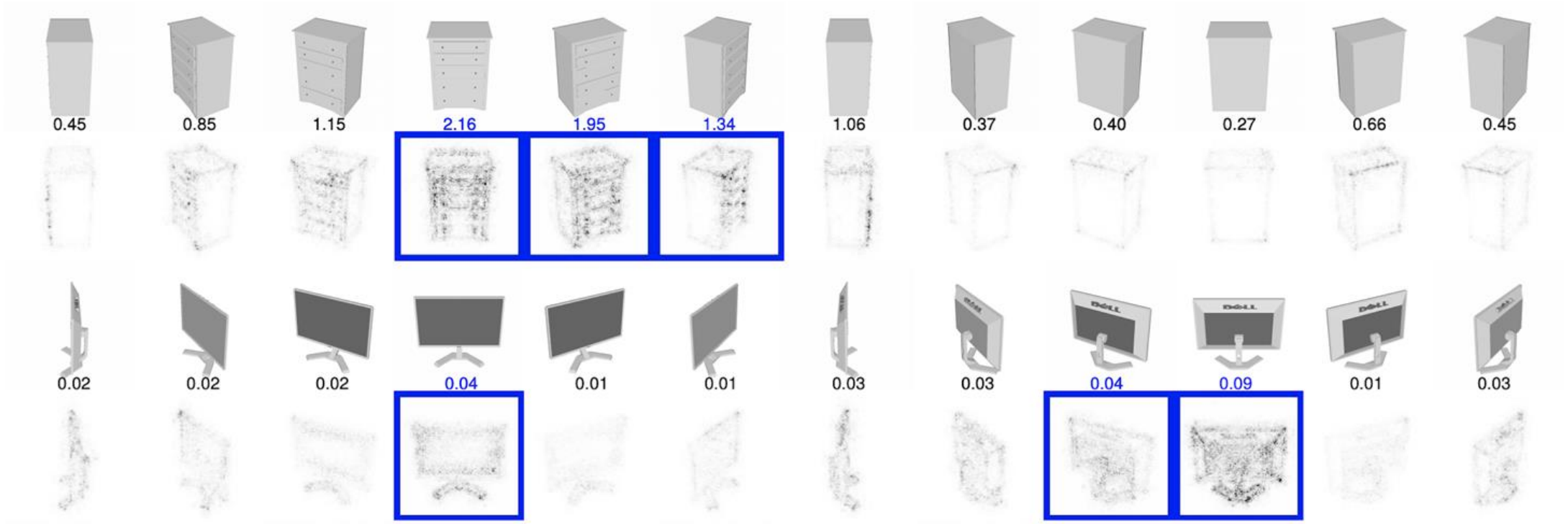


Figure 3. Top three views with the highest saliency are highlighted in blue and the relative magnitude of gradient energy for each view is shown on top. The saliency maps are computed by back-propagating the gradients of the class score onto the images via the view-pooling layer. Notice that the handles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

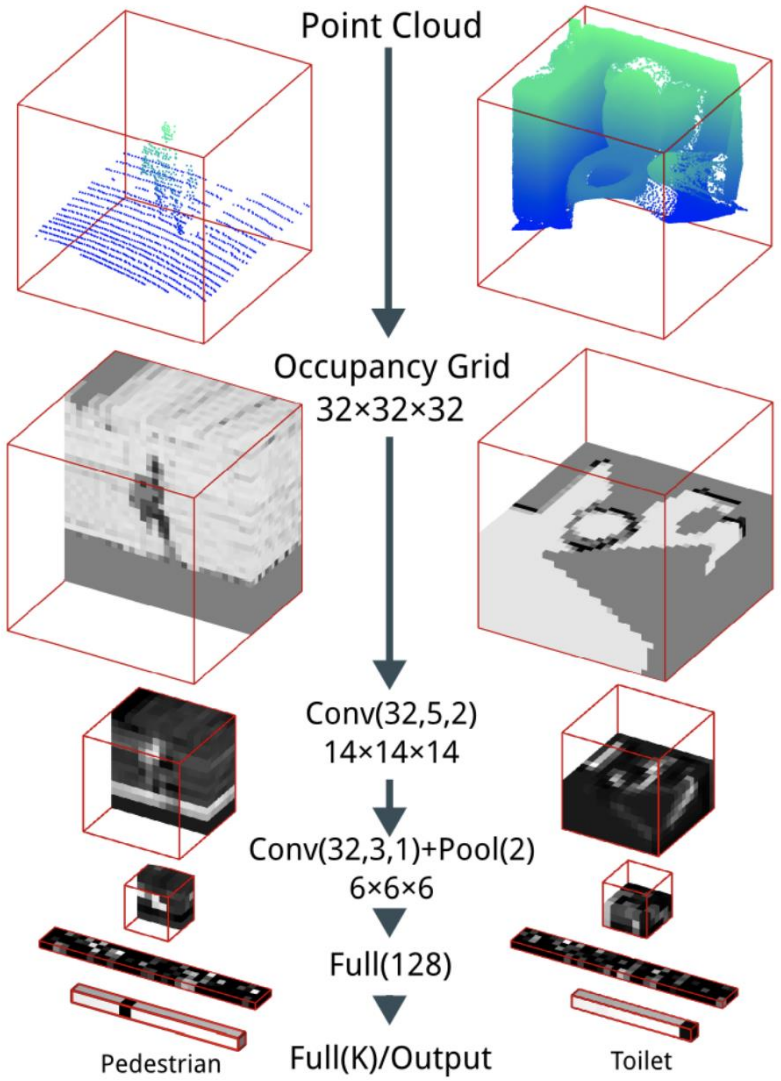
- Input Layer : This layer accepts a fixed-size grid of I×J×K voxels. In this work, we use I = J = K = 32

- 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 by 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

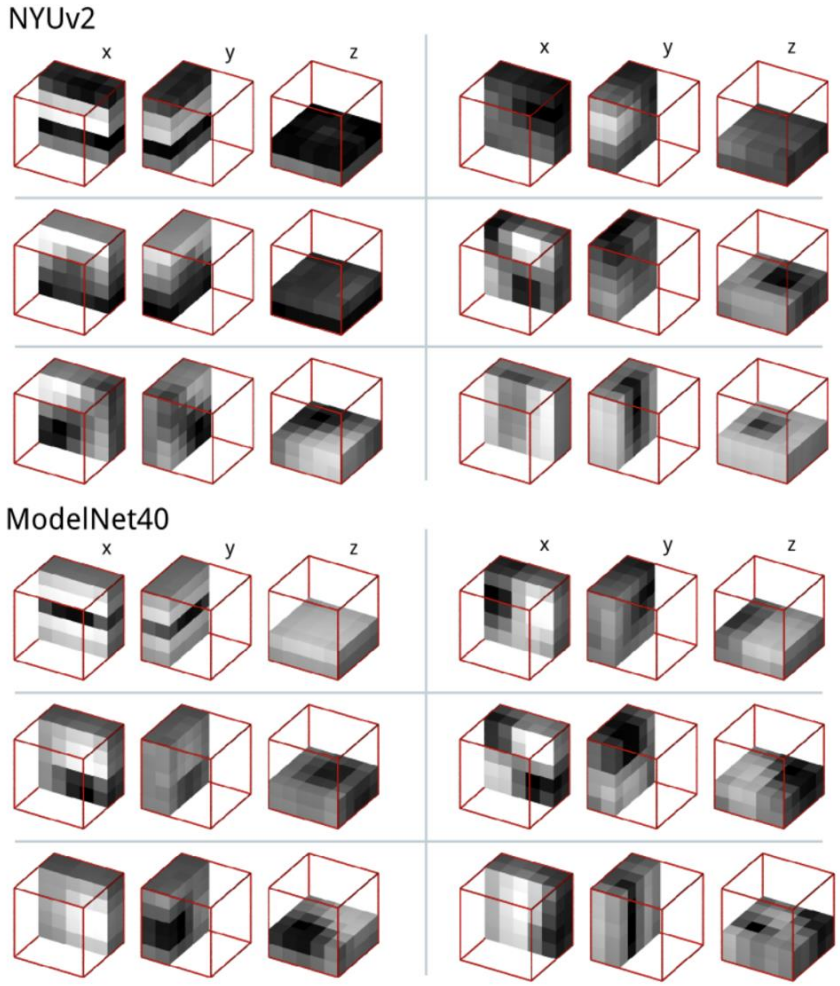


Fig. 4. Cross sections along the x, y and z axes of selected first layer filters learned in the ModelNet40 and NYUv2 datasets.

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 Point Cloud?

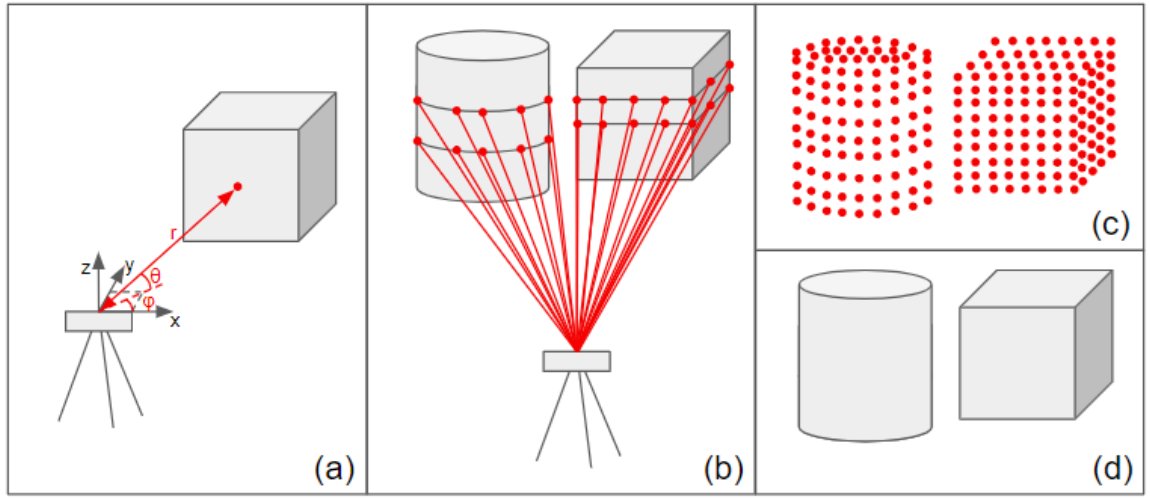


Figure 1: A point cloud from a laser scanner. See body text for more explanation.

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

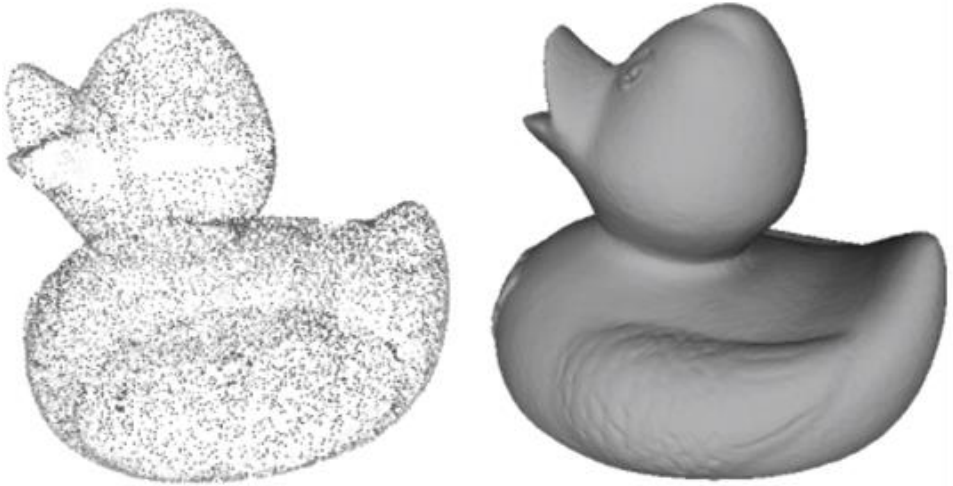
- A point cloud is nothing more than a collection of millions (sometimes billions) of points coming from a scanner. (Unorganized)
- The points only represent the surfaces of scanned objects
- Typical Scanners: 2D or 3D Lidar (Laser Scanner), Structure Sensor (1st Gen Kinect)

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

PointNet - What is Point Cloud?

Point Cloud/Mesh

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



```
ply
format ascii 1.0
element vertex 6
property float x
property float y
property float z
property uchar red
property uchar green
property uchar blue
element face 8
property list uchar int vertex_index
end_header
1.000000 0.000000 0.000000 255 0 0
0.000000 1.000000 0.000000 0 255 0
0.000000 0.000000 1.000000 0 0 0
-1.000000 0.000000 0.000000 0 0 255
0.000000 -1.000000 0.000000 255 0 255
0.000000 0.000000 -1.000000 255 255 255
3 0 1 2
3 1 3 2
3 3 4 2
3 4 0 2
3 1 0 5
3 3 1 5
3 4 3 5
3 0 4 5
```

Header begins

Number of vertices

Vertex coordinates data type

Data type for color property

Number of faces

How the vertex indices are listed to define faces

Vertex coordinates (x,y,z)

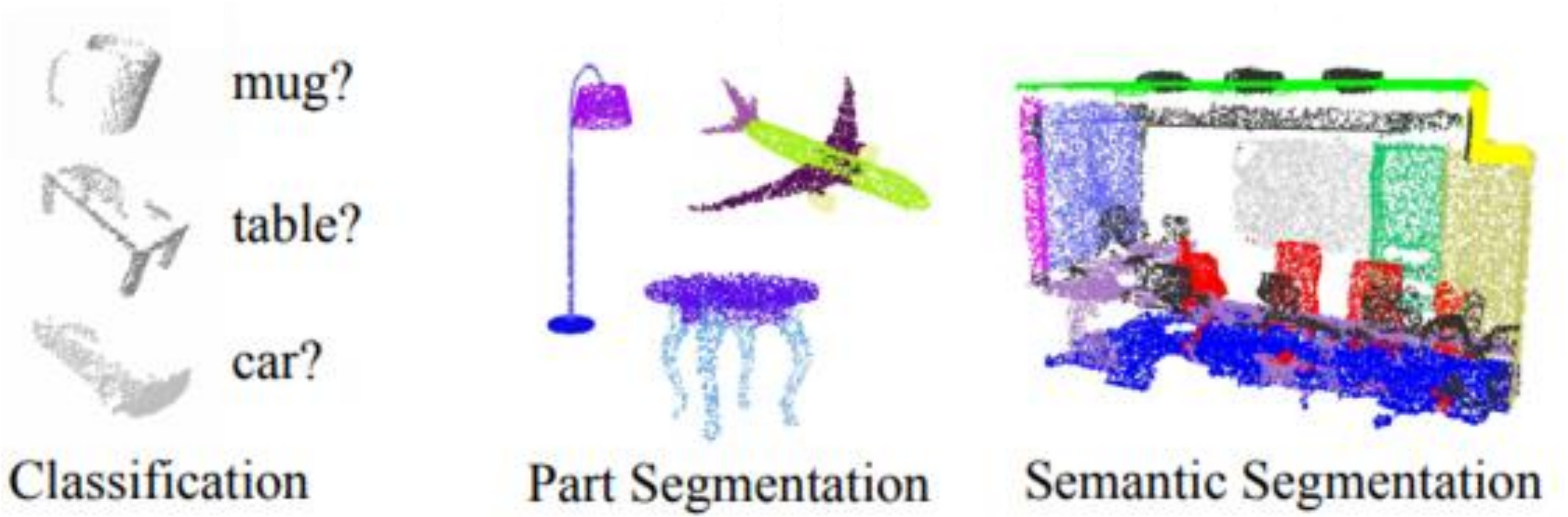
RGB triplet to define vertex colors (0 to 255)

Number of vertices that define this face

Definition of faces from vertex indices (1st vertex is 0)

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

Machine Learning Tasks on Point Cloud Data



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

PointNet: A Framework to Process Point Cloud with MLP

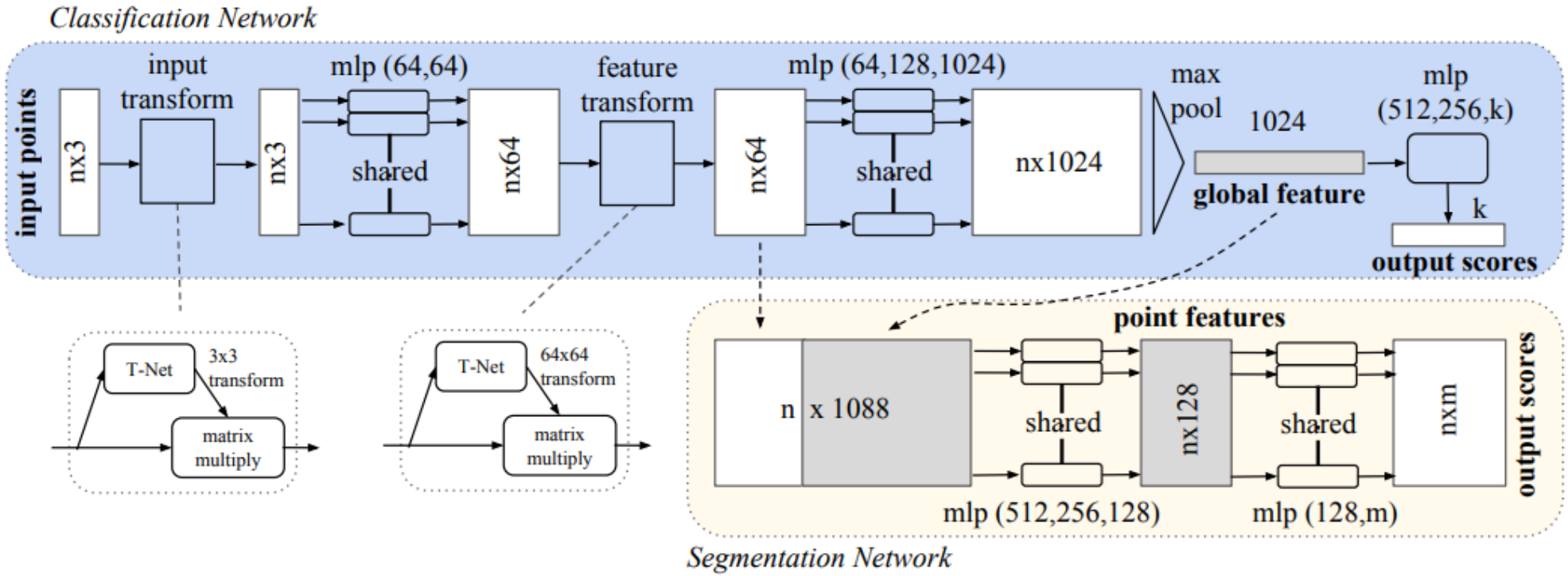


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. 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.

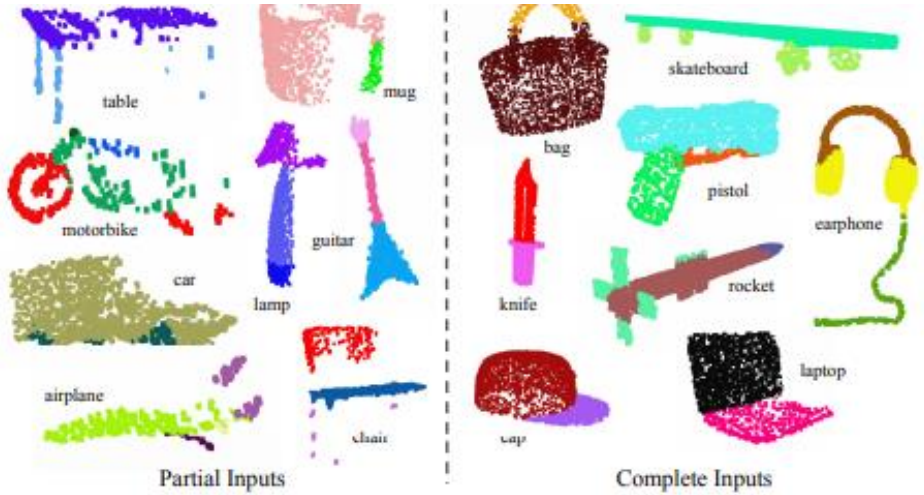


Figure 3. **Qualitative results for part segmentation.** We visualize the CAD part segmentation results across all 16 object categories. We show both results for partial simulated Kinect scans (left block) and complete ShapeNet CAD models (right block).

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**

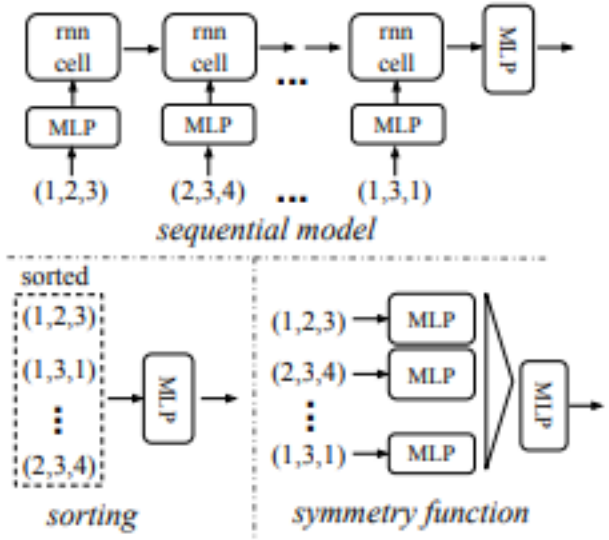
Table 2. **Segmentation results on ShapeNet part dataset.** Metric is mIoU(%) on points. We compare with two traditional methods [27] and [29] and a 3D fully convolutional network baseline proposed by us. Our PointNet method achieved the state-of-the-art in mIoU.

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

Critical point sets: those contributed to the max pooled feature, summarizing the skeleton of the shape.

Upper-bound shapes: the largest possible point cloud that give the same global feature shape.

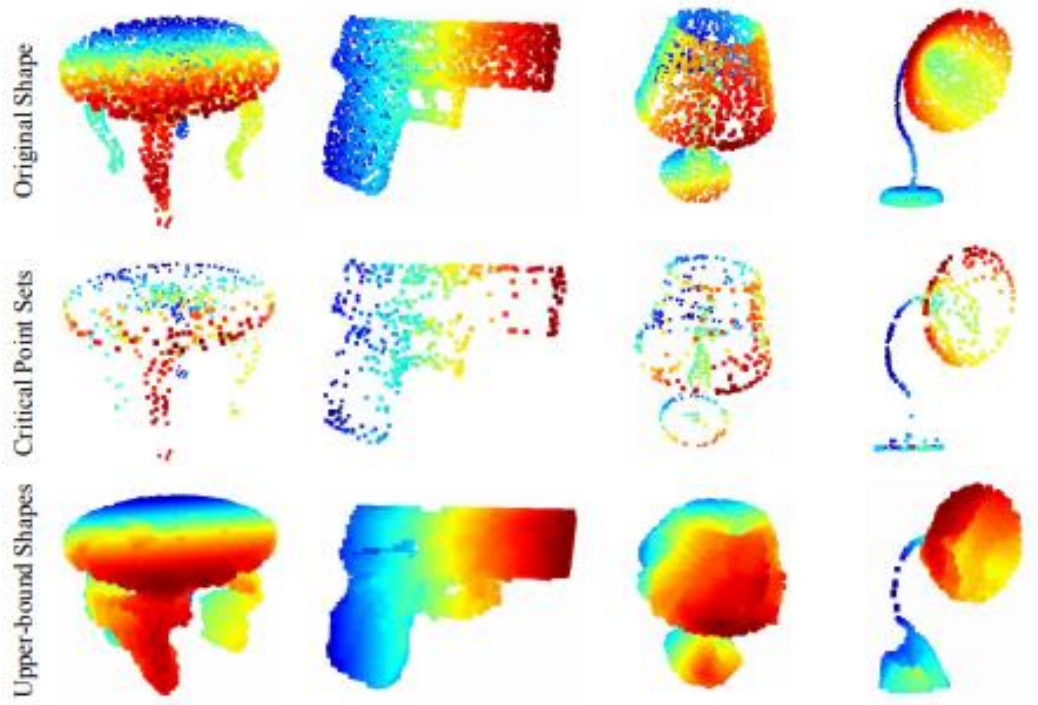


Figure 7. **Critical points and upper bound shape.** While critical points jointly determine the global shape feature for a given shape, any point cloud that falls between the critical points set and the upper bound shape gives exactly the same feature. We color-code all figures to show the depth information.

PointNet++

PointNet++ leverages neighborhoods at multiple scales to achieve both robustness and detail capture.

Introduces a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set.

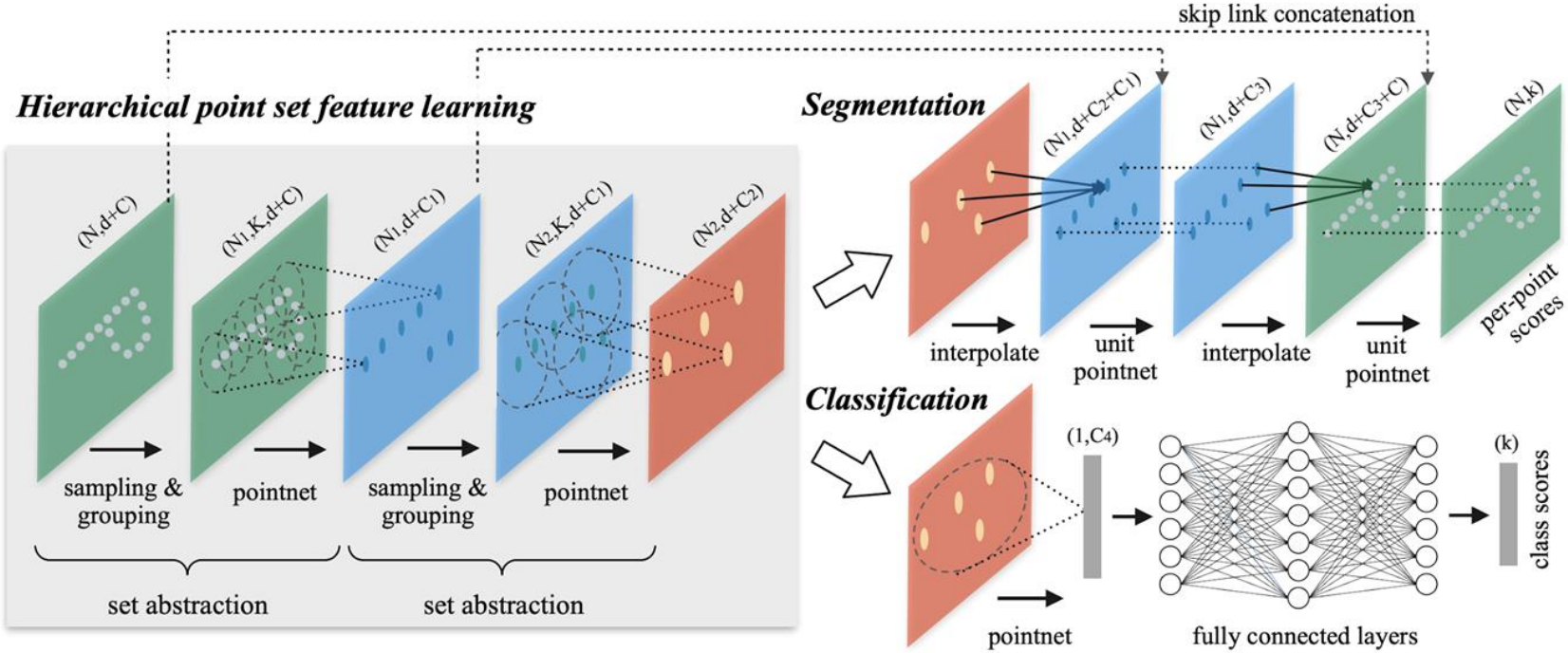
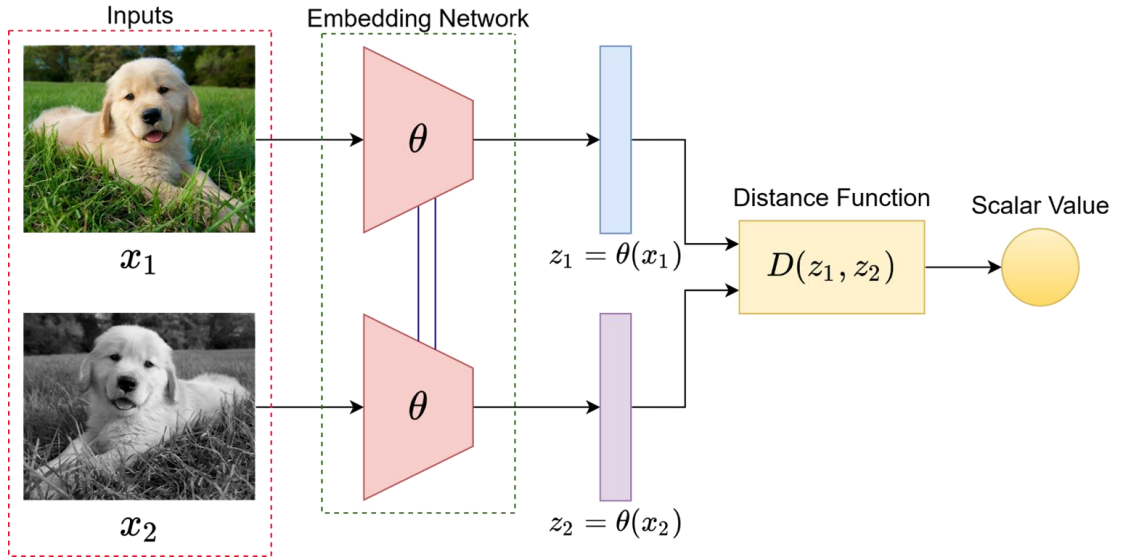


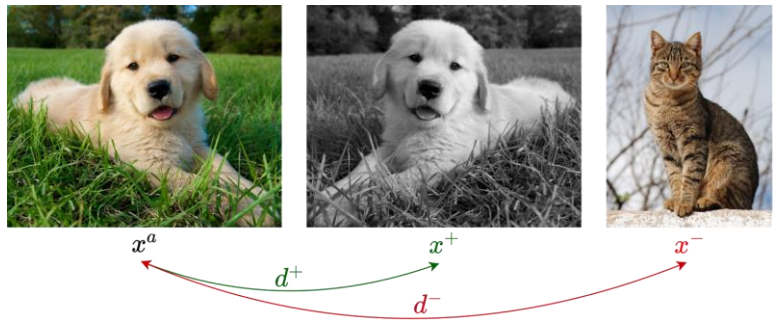
Figure 2: Illustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example. Single scale point grouping is visualized here. For details on density adaptive grouping, see Fig. 3

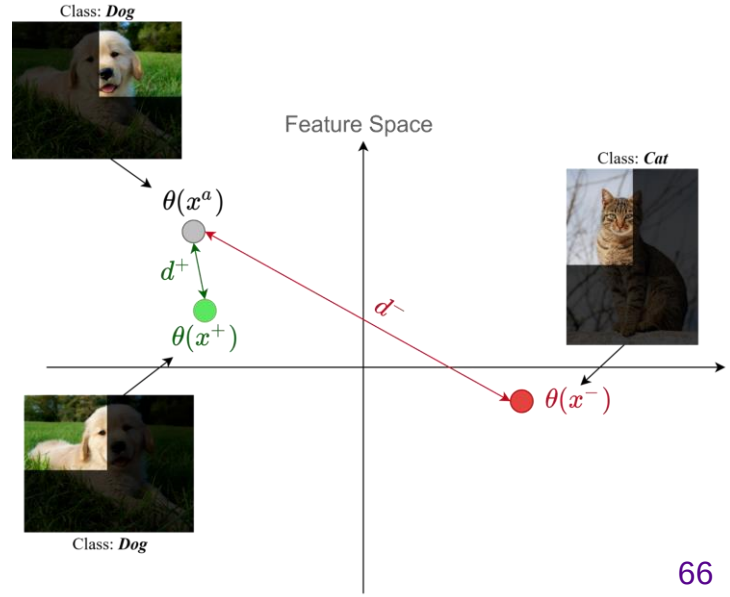
How to Learn Representations without (External) Supervision

- Self-supervised learning (SSL)
- Benefit: scalability

- Two (very coarse) categorization
  - Contrastive
    - MoCo
    - SimCLR
    - SwAV
  - Reconstructive / Generative
    - VAE
    - Mask Autoencoder (MAE)
    - Cycle-consistency

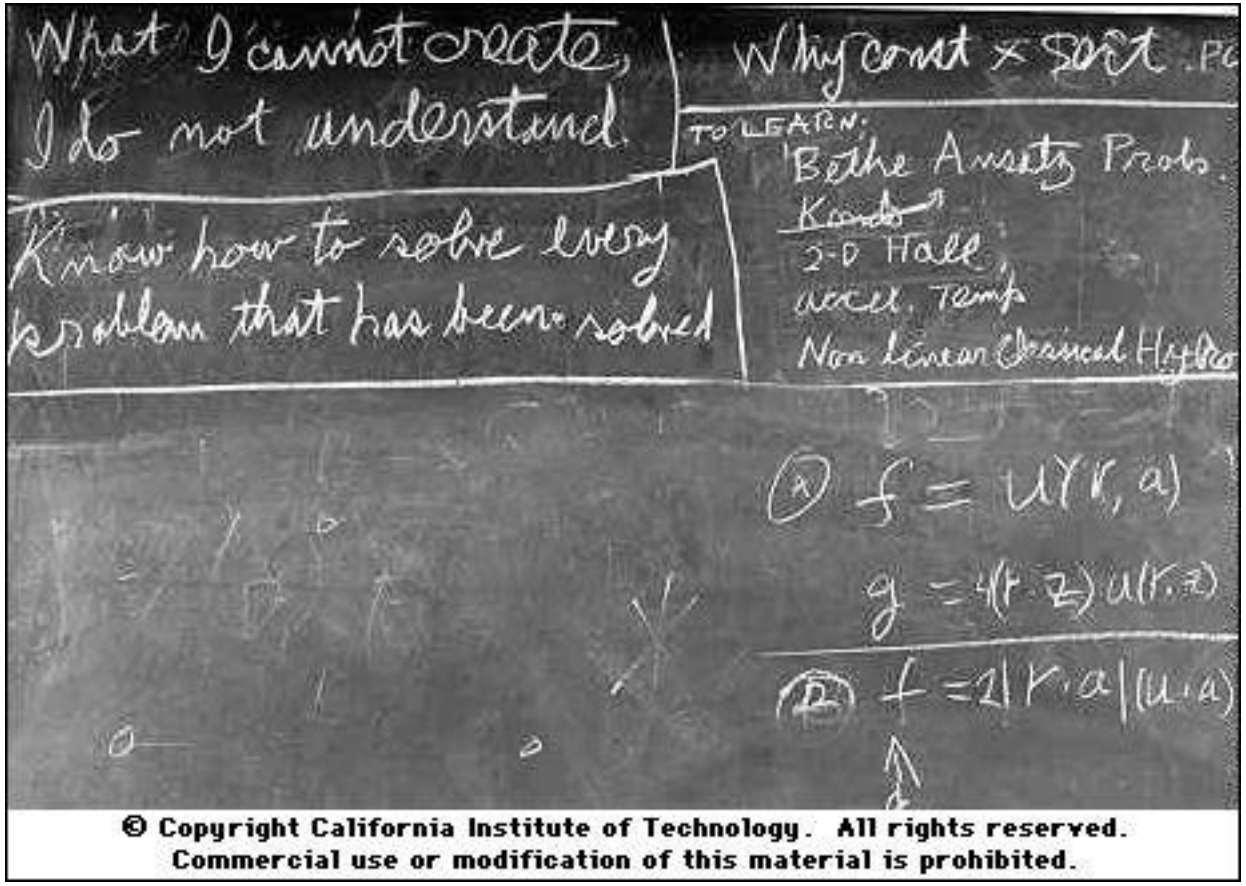






https://www.v7labs.com/blog/self-supervised-learning-guide#h1

Richard Feynman Quote

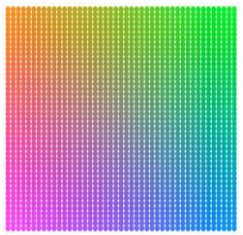
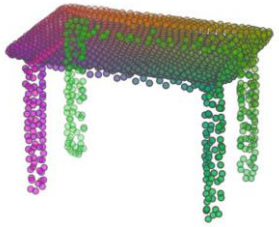


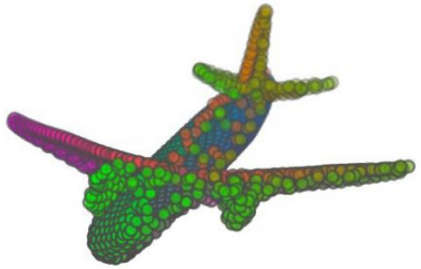
"What I cannot create, I do not understand."

Richard Feynman (1918-1988)
Theoretical Physicist
Won 1965 Nobel Prize in Physics




Can Neural Networks Learn Paper Folding?

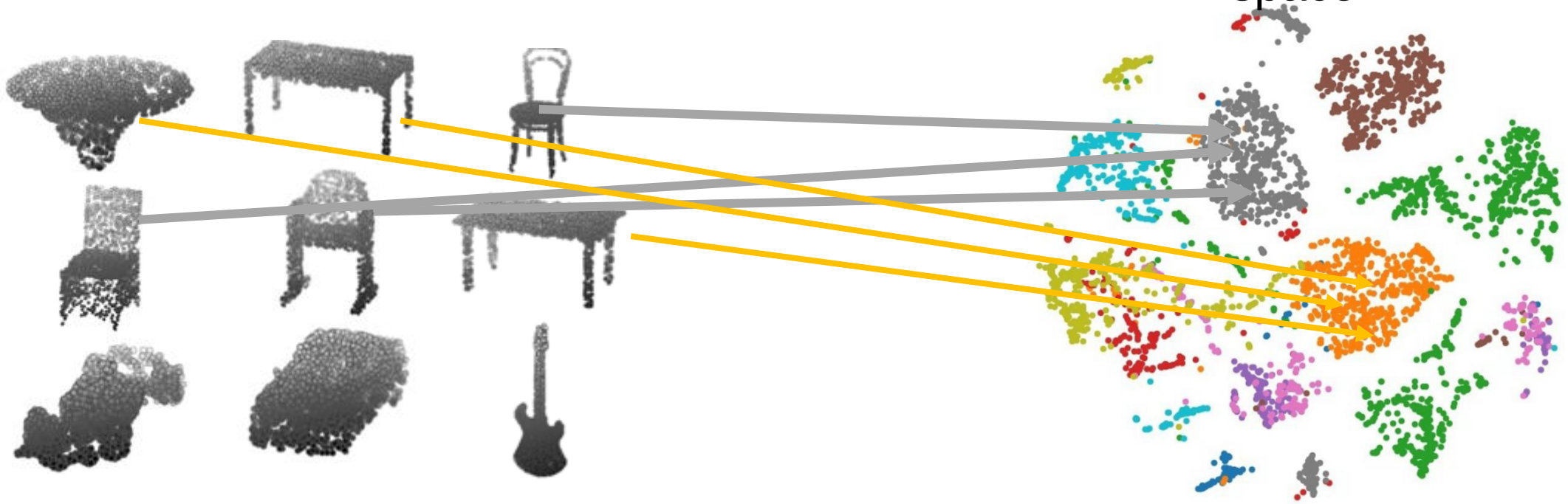





Representation Learning for Point Clouds

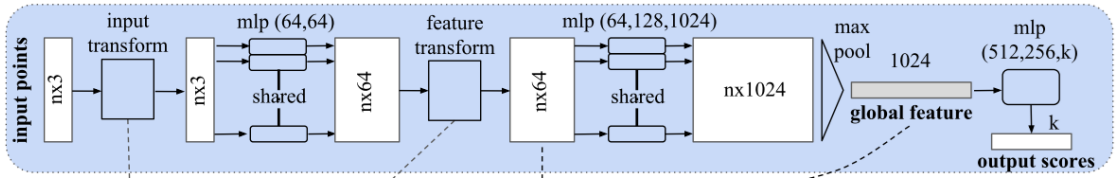
3D Data (Point Clouds)

Latent space



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

Baseline Point Cloud Auto-encoder



[Qi et al., CVPR'17]

Input

Encoder

1xK

codeword

Decoder (MLP)

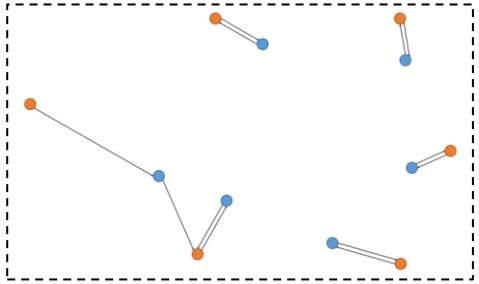
[Achlioptas et al., arxiv'17]

3Nx1

Reshape

Nx3

Output



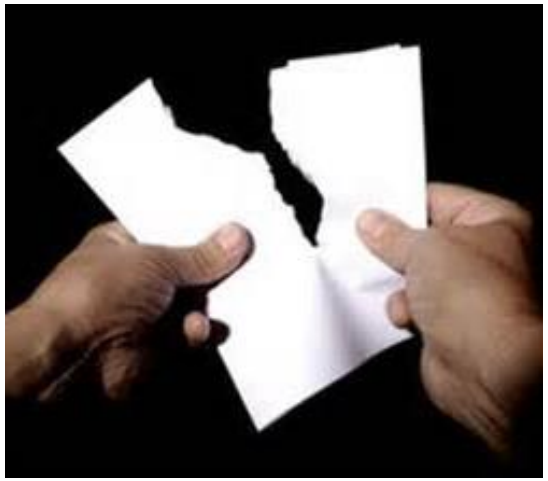
Chamfer Distance, Earth-Mover's Distance, etc.

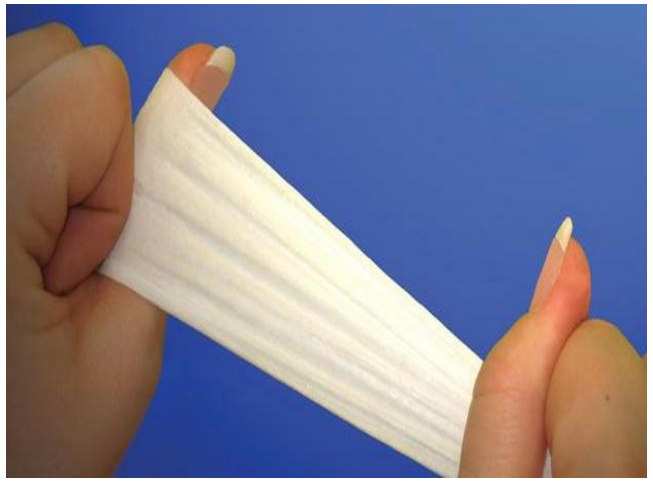
$$\max\left\{\frac{1}{|S|}\sum_{\mathbf{x}\in S}\min_{\hat{\mathbf{x}}\in\hat{S}}\|\mathbf{x}-\hat{\mathbf{x}}\|_2,\ \frac{1}{|\hat{S}|}\sum_{\hat{\mathbf{x}}\in\hat{S}}\min_{\mathbf{x}\in S}\|\hat{\mathbf{x}}-\mathbf{x}\|_2\right\}$$

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

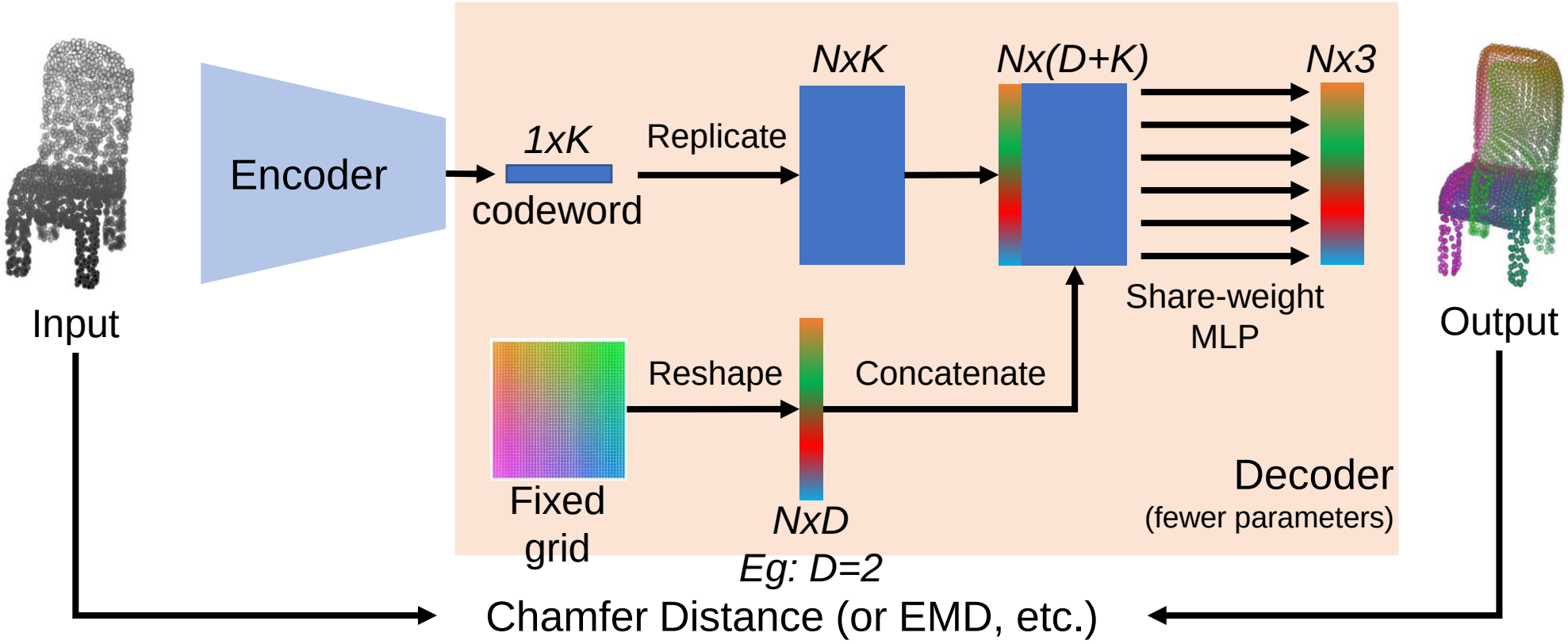

Stretch


Glue

FoldingNet Auto-encoder Framework

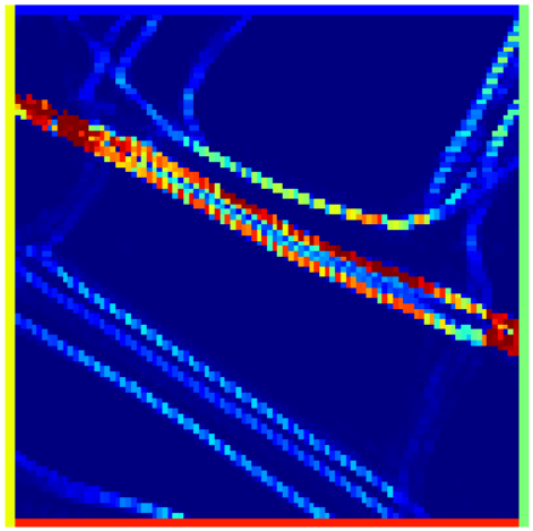
- FoldingNet decoder is a *Deep Parametric Surface*:

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

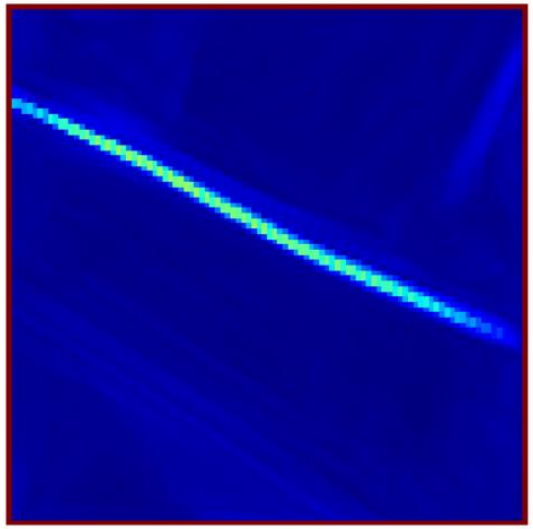


Learned Folding Profile - Sofa

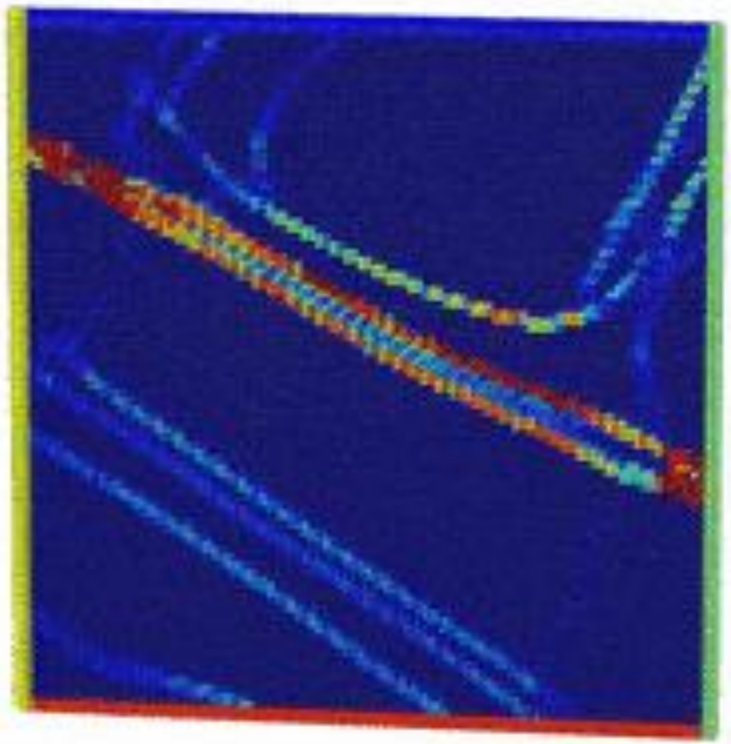
Folding Creases
(Curvature)



Tear/Stretch
(Neighbor Distance)



Folding Animation: Sofa
(colored by curvature)



Training Process Visualization

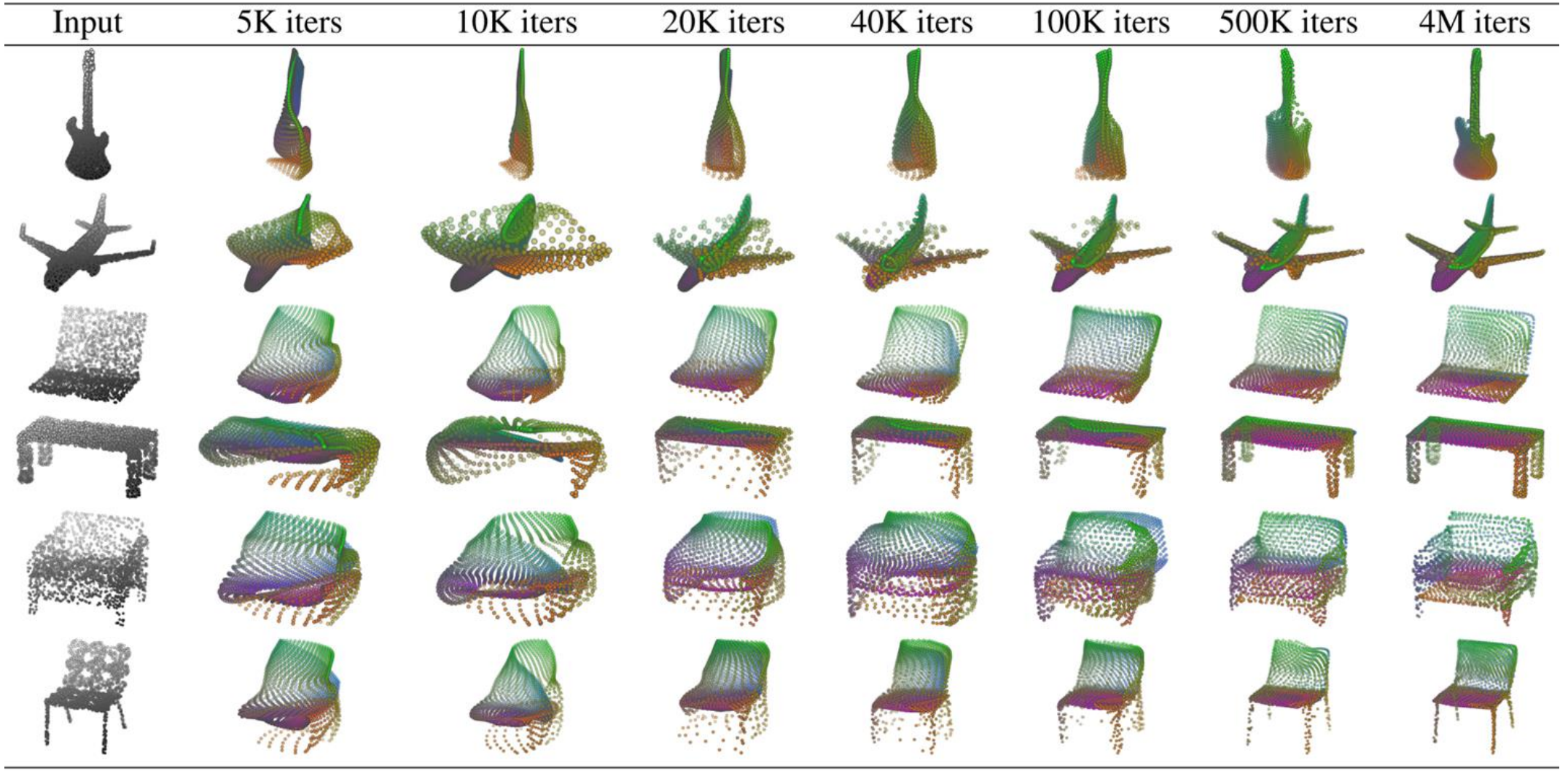


Table 2. Illustration of the training process. Random 2D manifolds gradually transform into the surfaces of point clouds.

Codeword Space Visualization

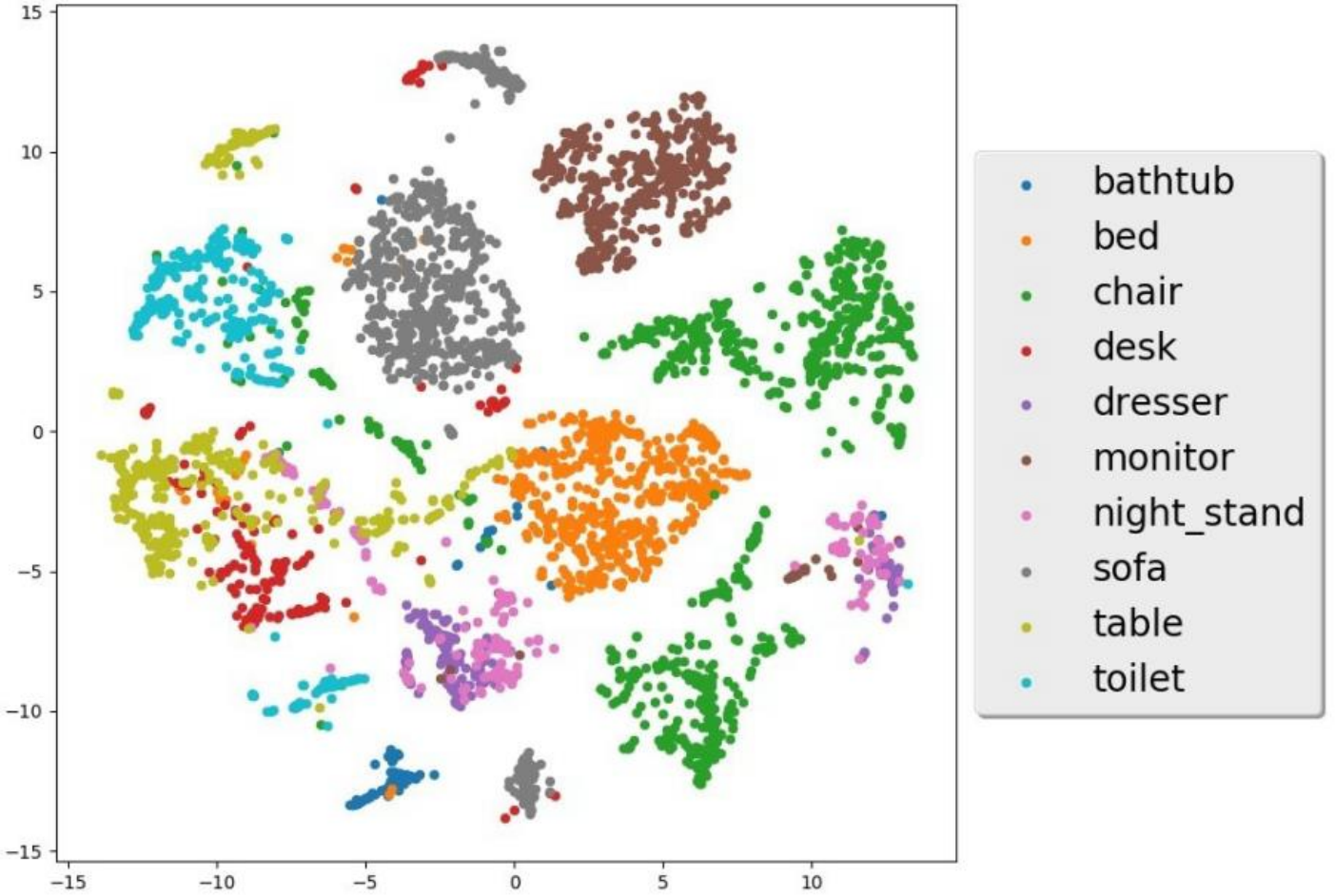
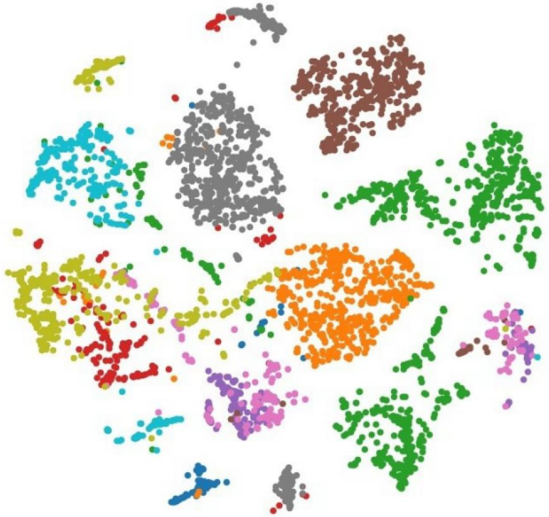


Figure 2. The T-SNE clustering visualization of the codewords obtained from FoldingNet auto-encoder.

Shape Interpolation



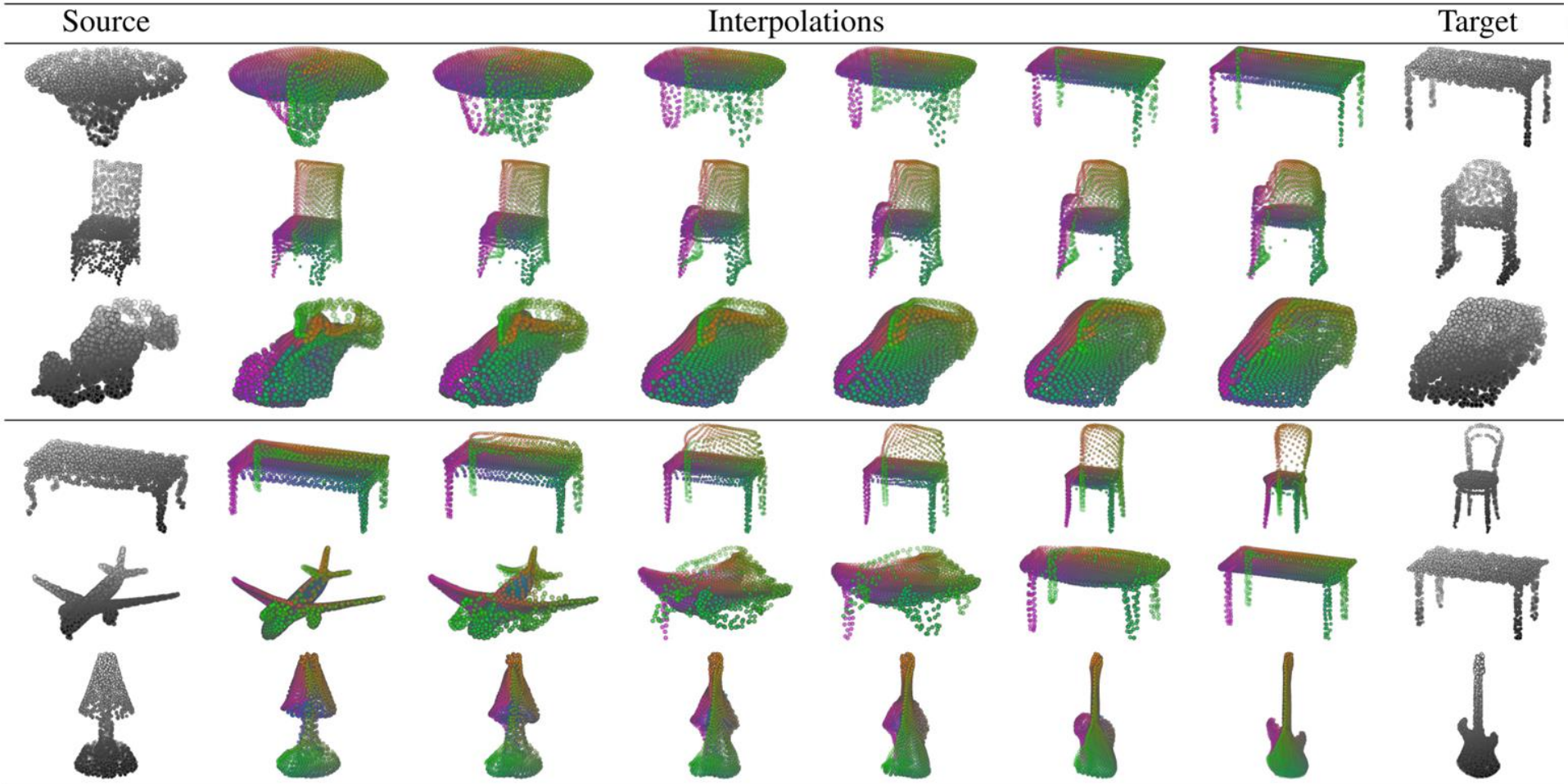


Table 3. Illustration of point cloud interpolation. The first 3 rows: intra-class interpolations. The last 3 rows: inter-class interpolations.

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

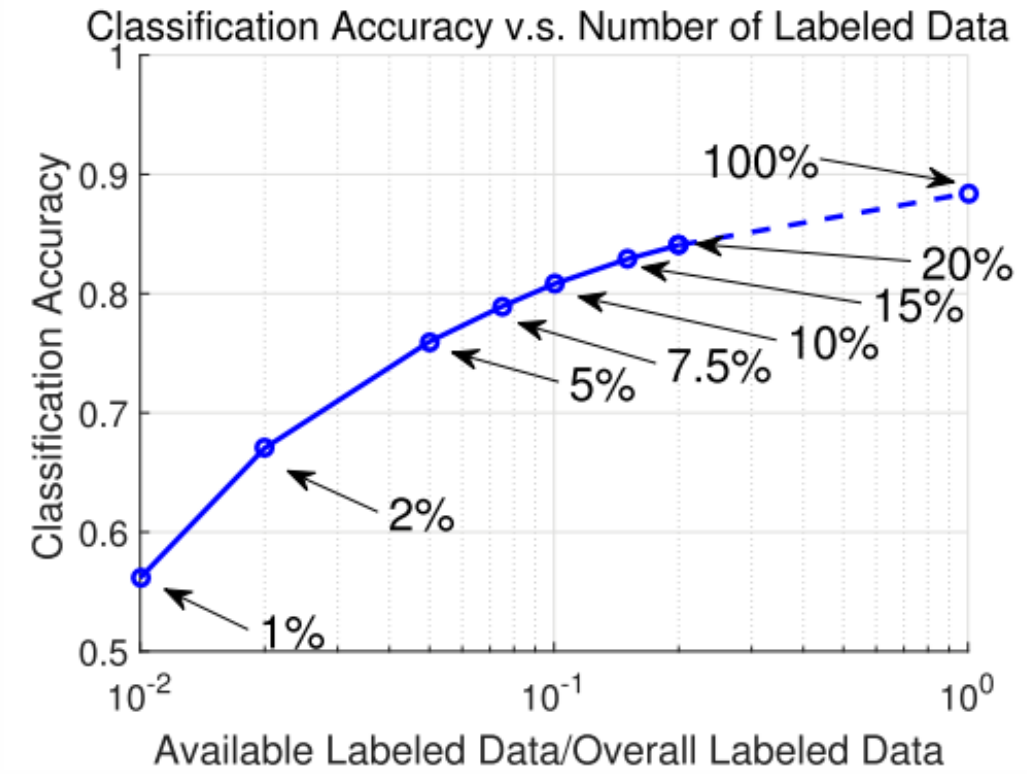


Figure 4. Linear SVM classification accuracy v.s. percentage of available labeled training data in ModelNet40 dataset.

Ablation: Decoder Variations

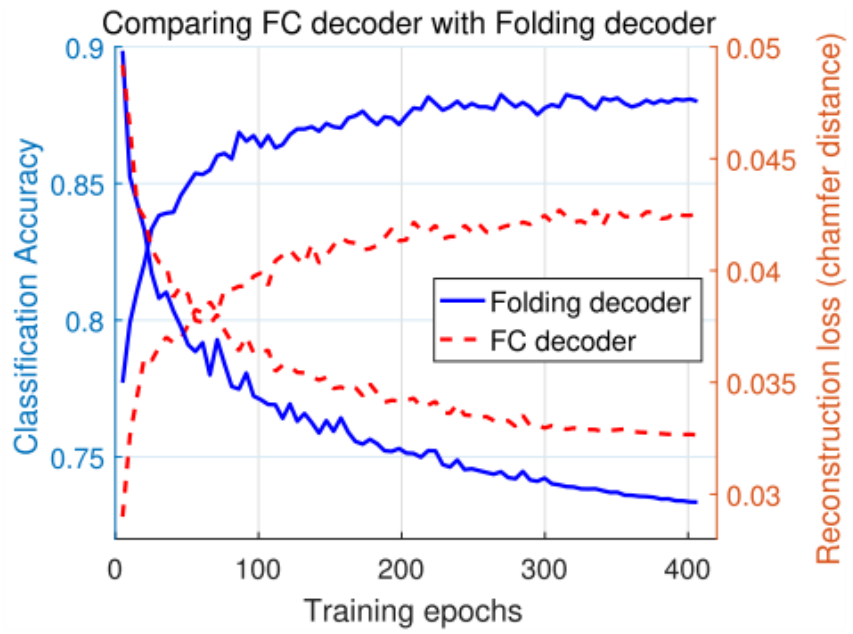


Figure 5. Comparison between the fully-connected (FC) decoder in [1] and the folding decoder on ModelNet40.

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

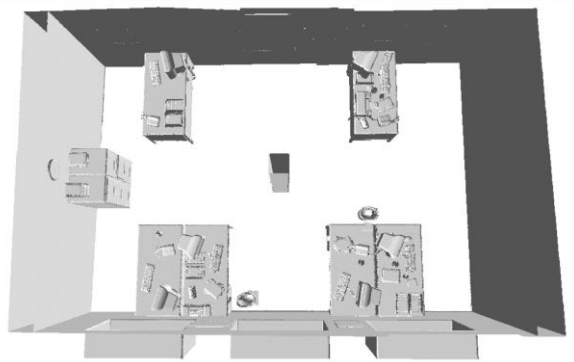
Table 6. Comparison between different FoldingNet decoders. "Uniform": the grid is uniformly random sampled. "Regular": the grid is regularly sampled with fixed spacings.

	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 Do 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.]

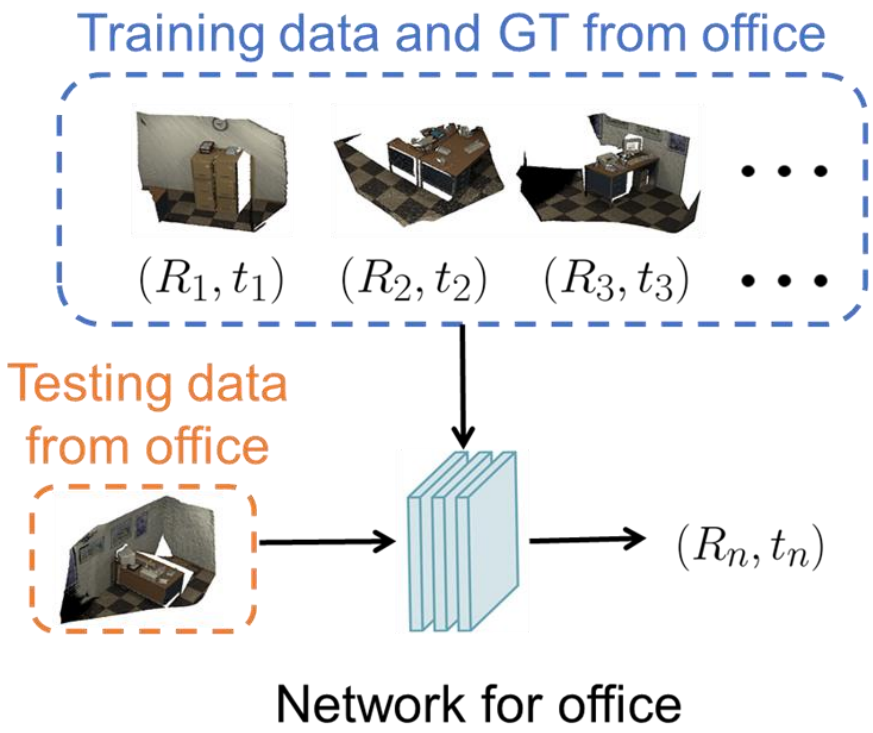
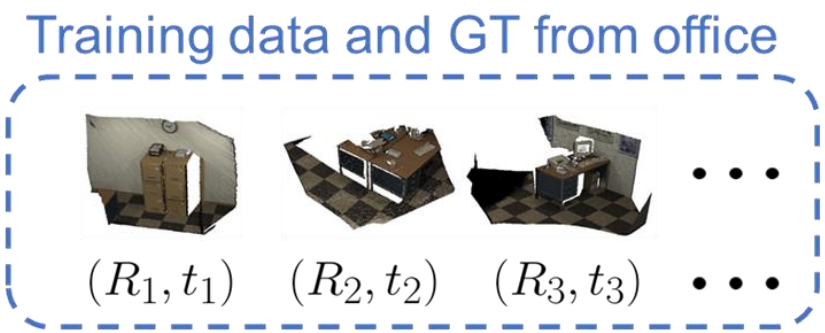
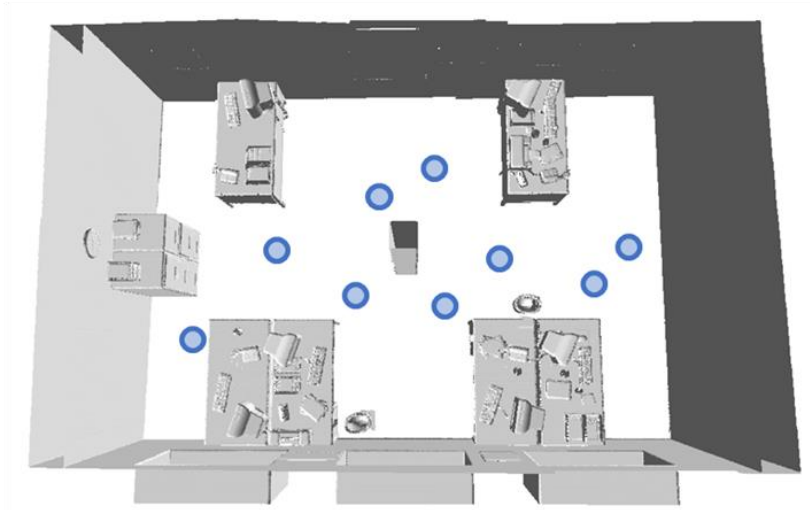


Image source: S. Choi, et. al. CVPR, 2015.

Motivation: Deep Learning for Geometric Vision Problems

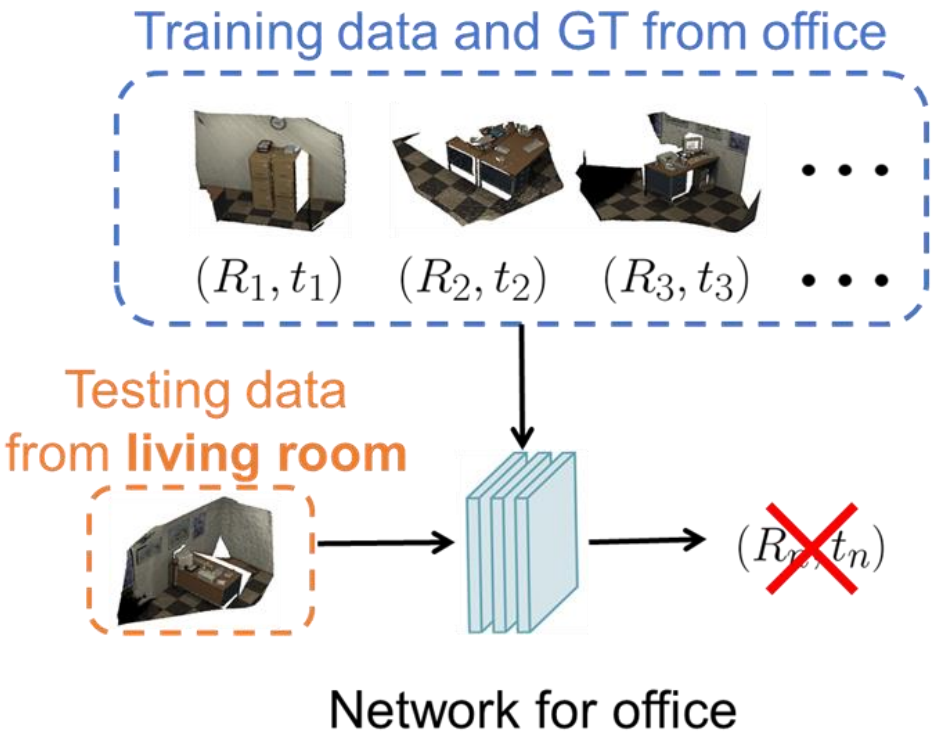
- Issues of these supervised approaches
  - Collect training dataset and ground truth poses
    - Ground truth is not always easy to obtain: SLAM, surveying, …



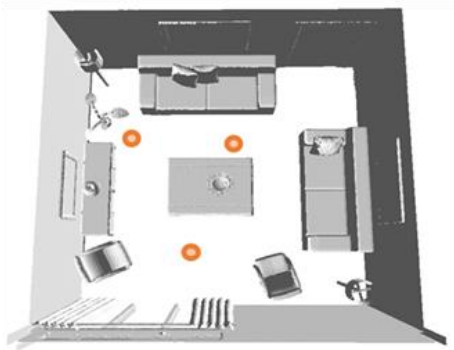

Office

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”

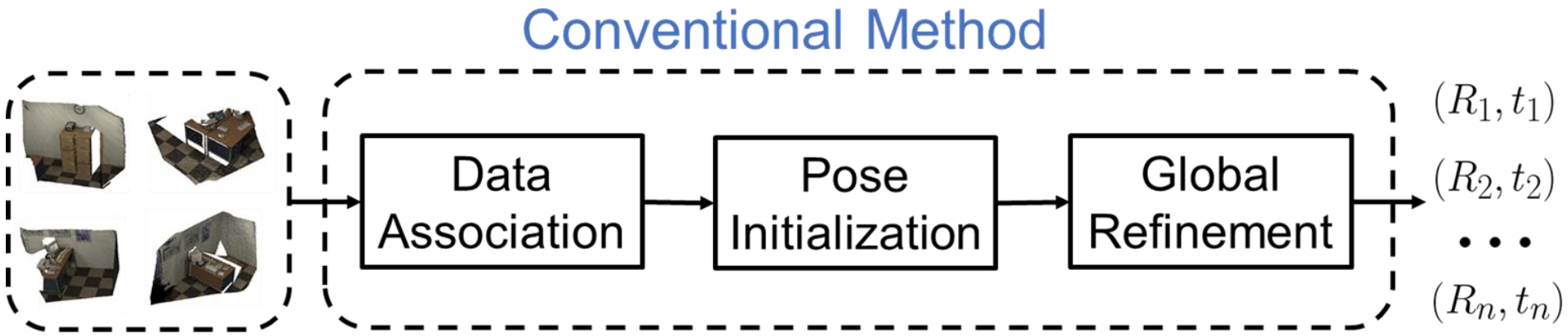


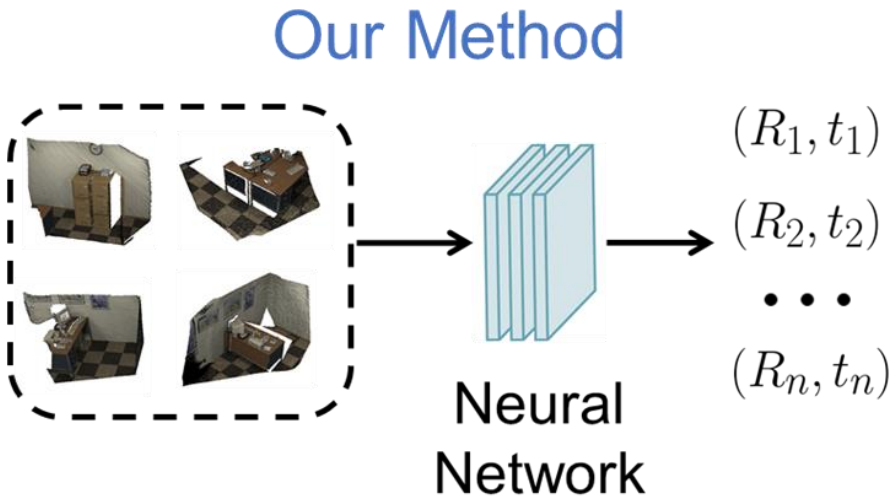

Office


Living room

Our Problem Formulation

- Use neural network to model traditional registration process



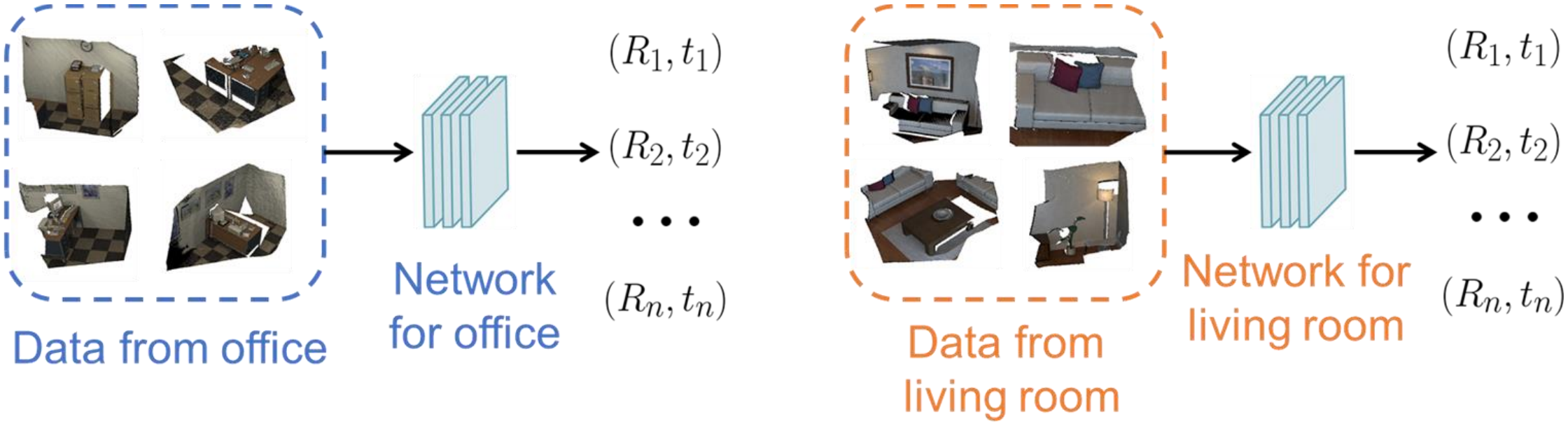


Instead of the common train-and-test pipeline, we propose a new formulation where training the networks is equivalent to solving the mapping problem.

We first use NN to model the complicated registration process that traditionally involves data association, sensor pose initialization, and global refinement.

Our Problem Formulation

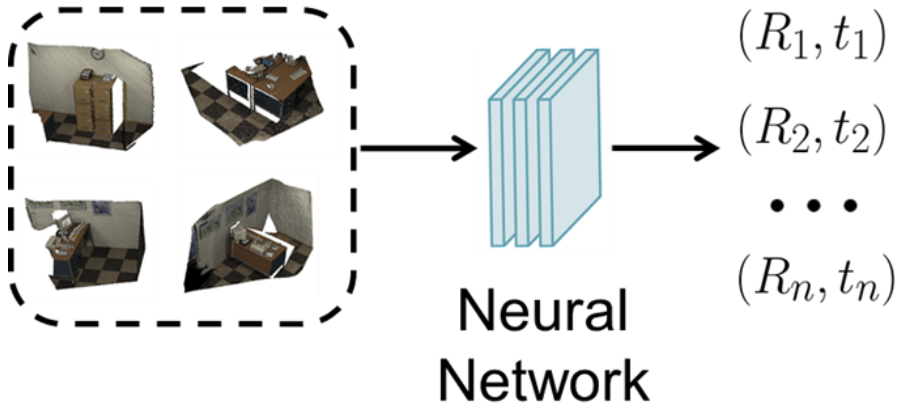
- Use neural network to model traditional registration process
- Train a neural network for each registration problem

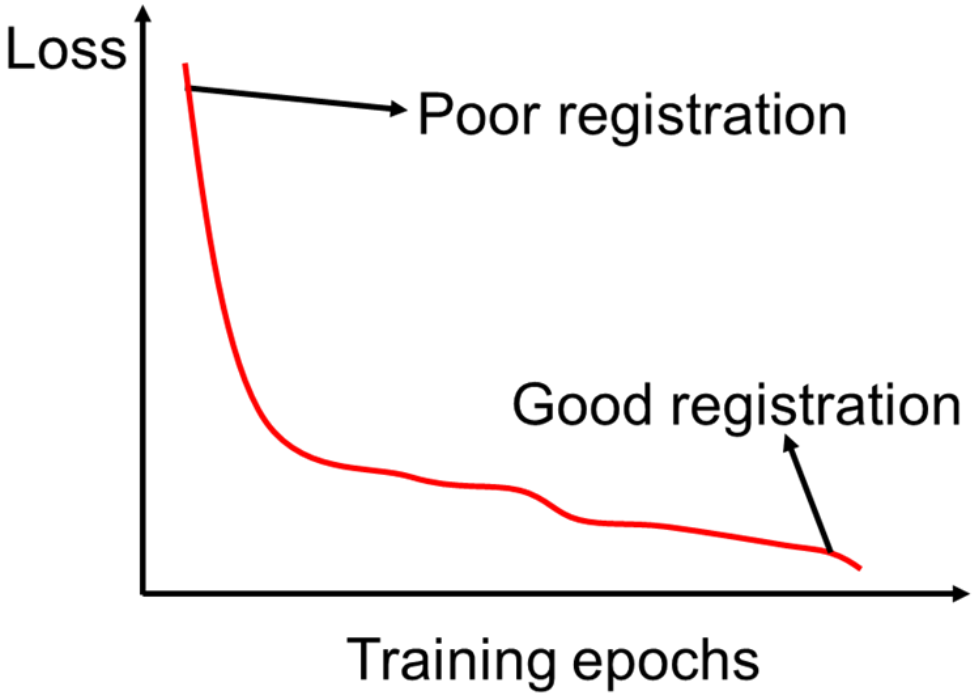


By giving up the expectation of the network should be generalizable, we hope to get something in return: a fully automatic mapping process.

Our Problem Formulation

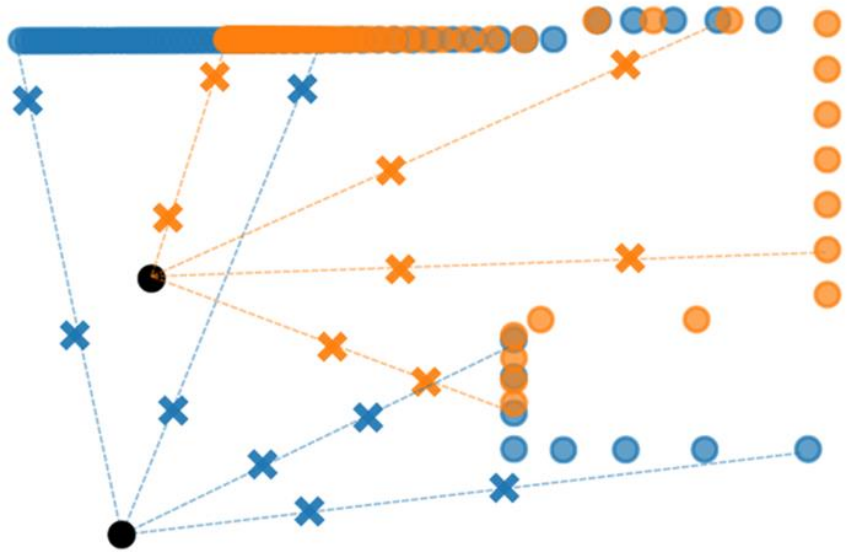
- Solve registration as an unsupervised “training” of neural network
  - Do not rely on ground truth data
  - Unsupervised loss measures registration quality
    - Minimize loss = encourage alignment



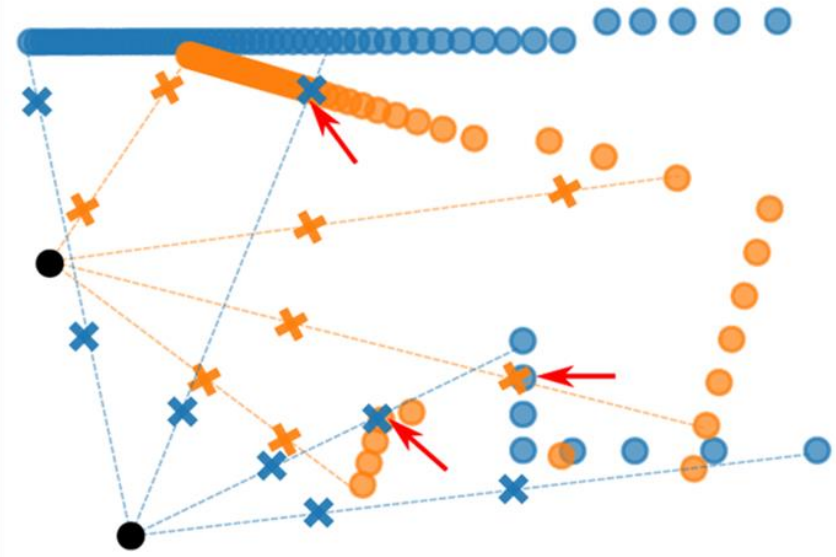


DeepMapping Loss

- 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

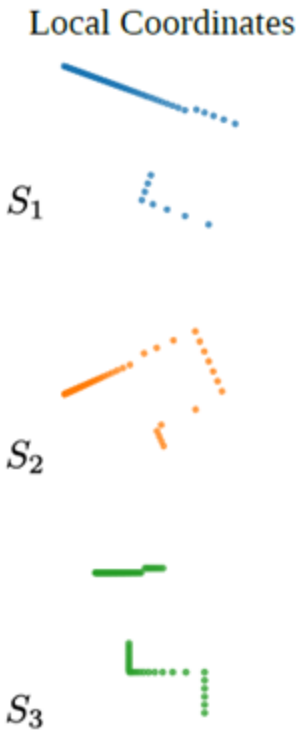


Correctly aligned
No free-space inconsistency



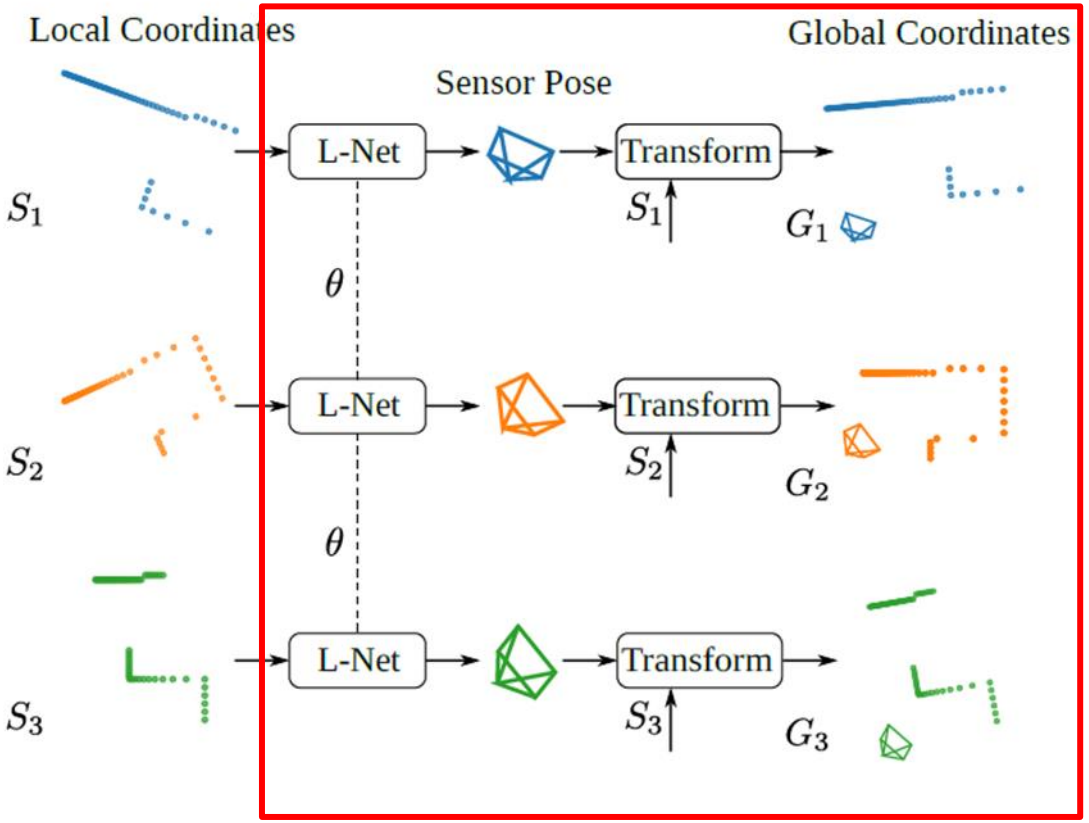
Mis-aligned
Free-space inconsistency

DeepMapping Pipeline



This leads us to the final DM pipeline. The input to DM is a set of point clouds in their local sensor coordinate systems.

DeepMapping Pipeline: Localization Network

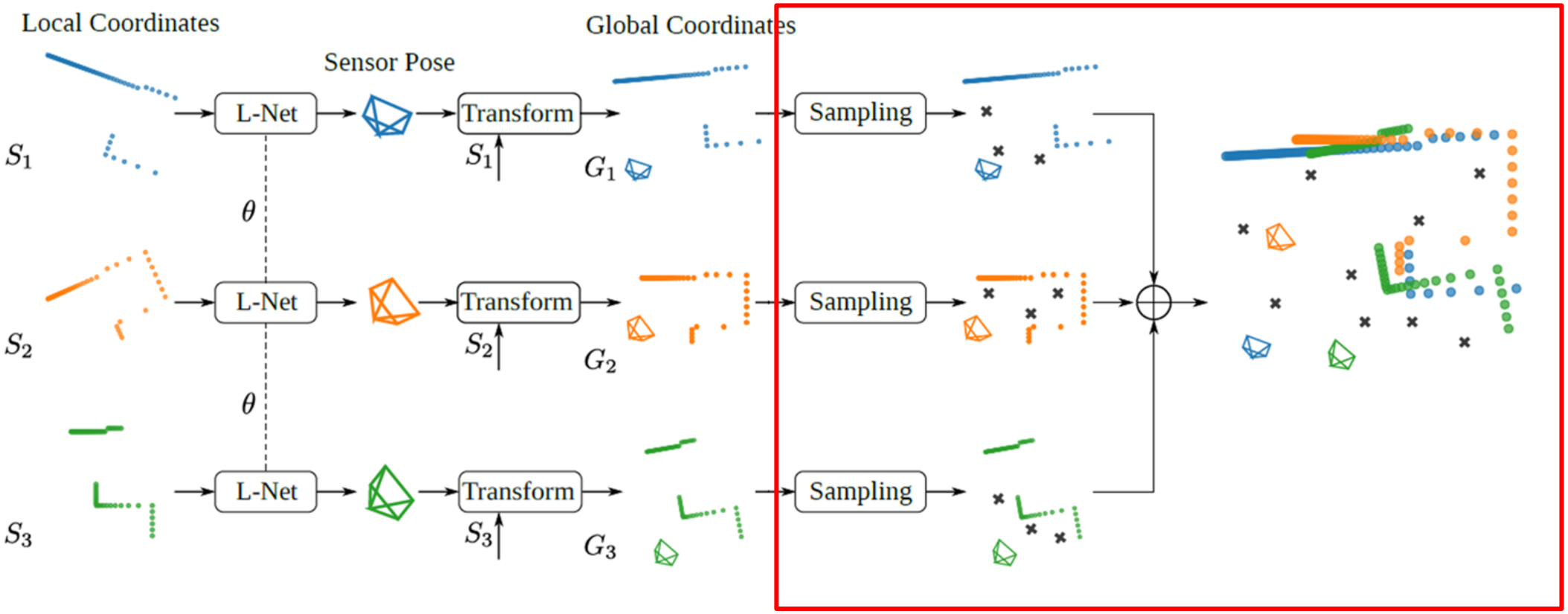


Localization Network (L-Net)

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

We first use a localization network or L-Net that outputs the sensor pose for each point cloud and then transform all local point clouds into a common global frame.

DeepMapping Pipeline: Free-Space Sampling

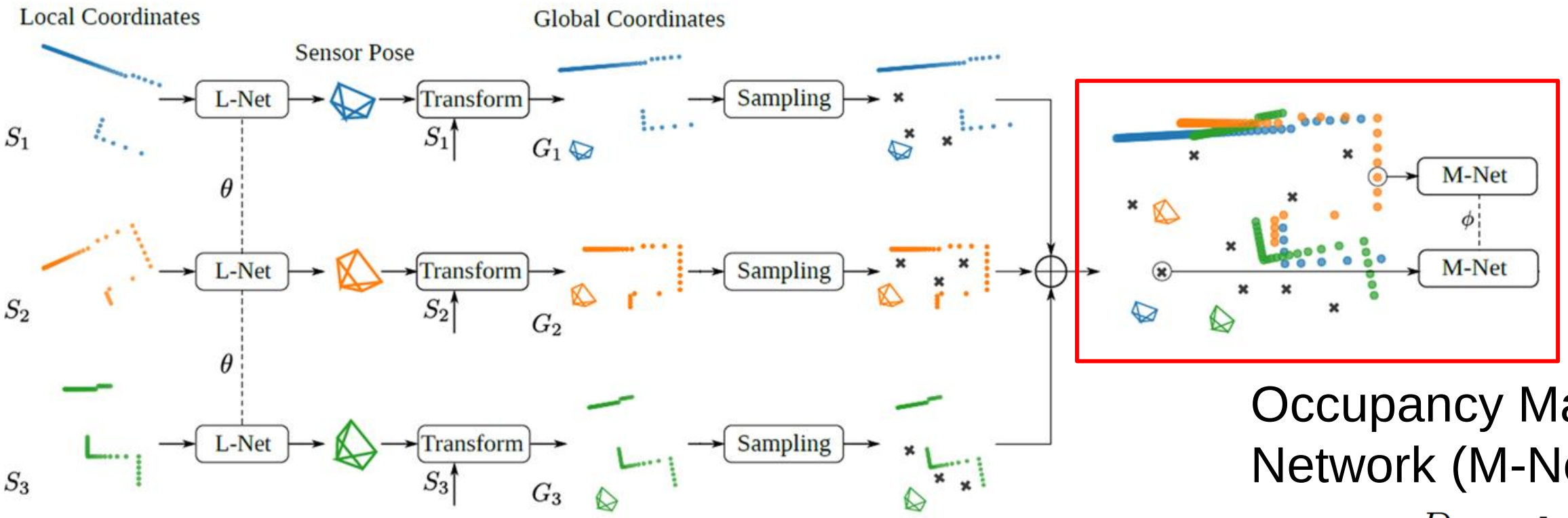


Localization Network (L-Net)

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

In the global frame, we can sample free-space points from the line-of-sight at each scanning center.

DeepMapping Pipeline: Occupancy Map Network



Occupancy Map Network (M-Net)

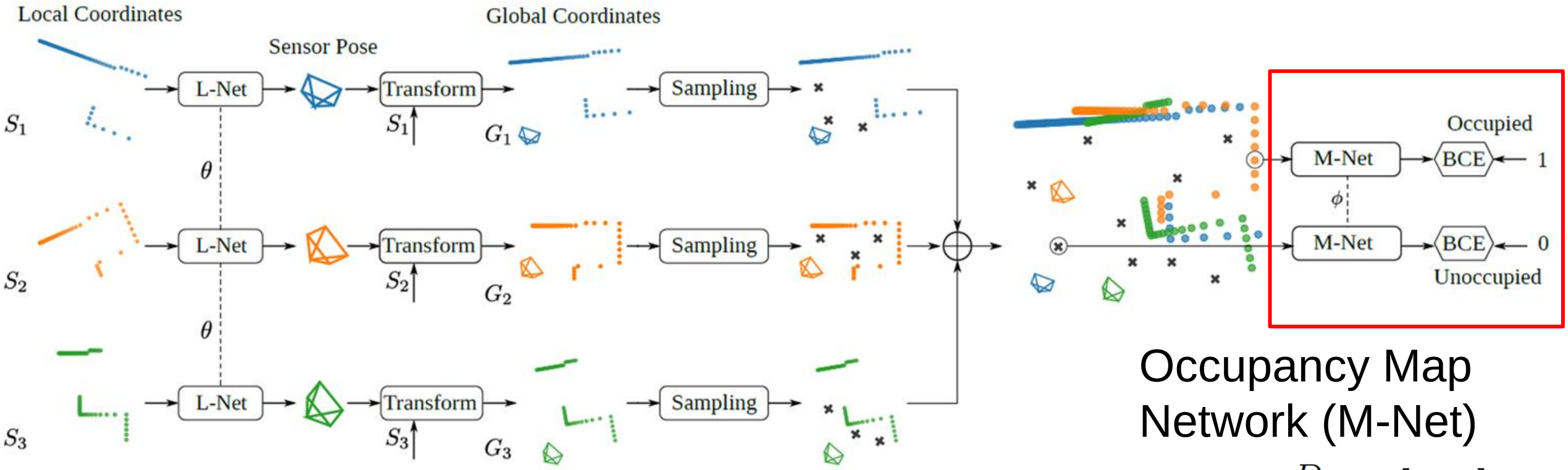
$$m_\phi : \mathbb{R}^D \rightarrow [0, 1]$$

Localization Network (L-Net)

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

Then we introduce the second network Occupancy Map network M-Net, which is a binary classification network that predicts probabilities of input locations being occupied.

DeepMapping Pipeline: Mapping Using Binary Classification Loss



Occupancy Map Network (M-Net)

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

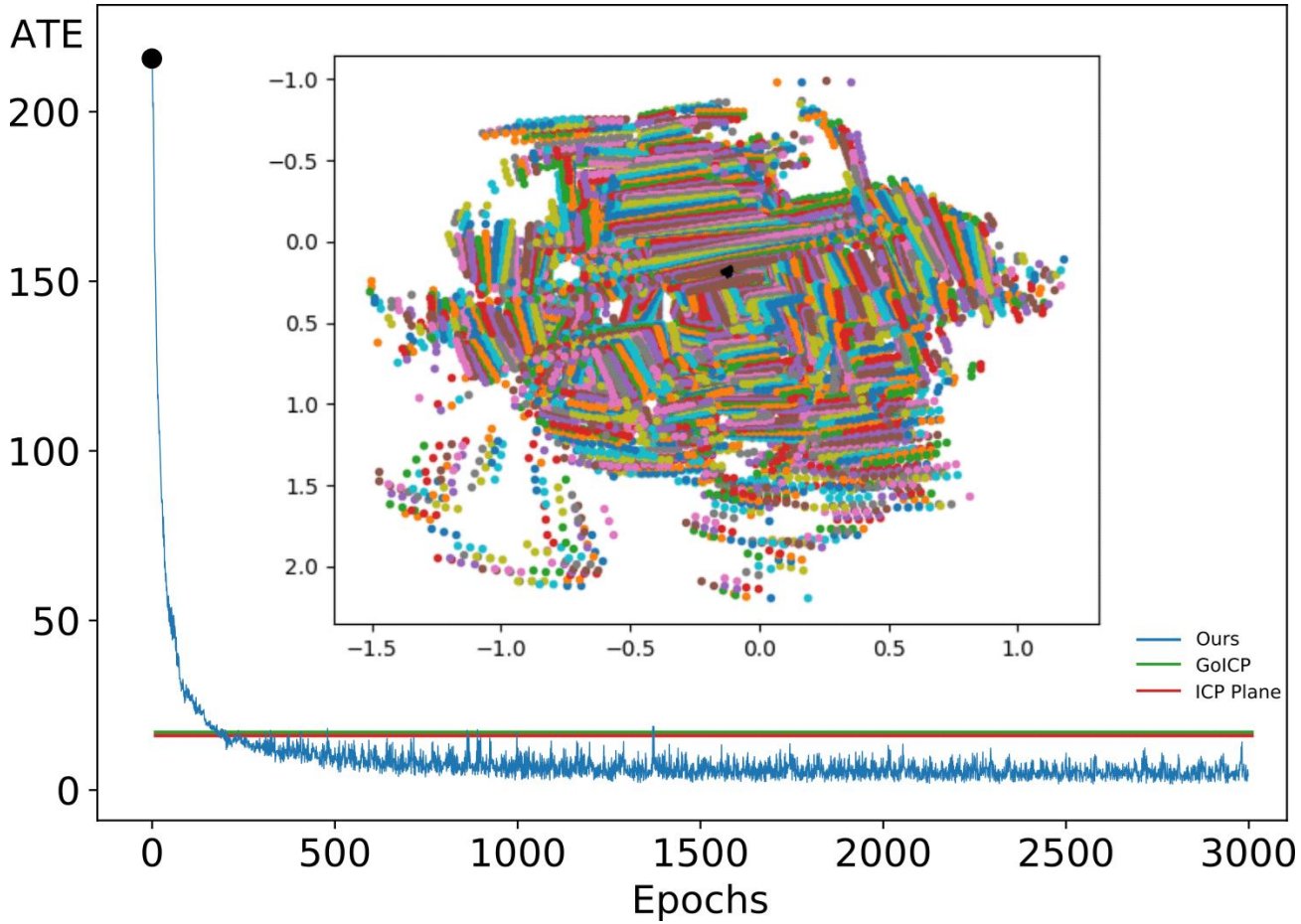
Localization Network (L-Net)

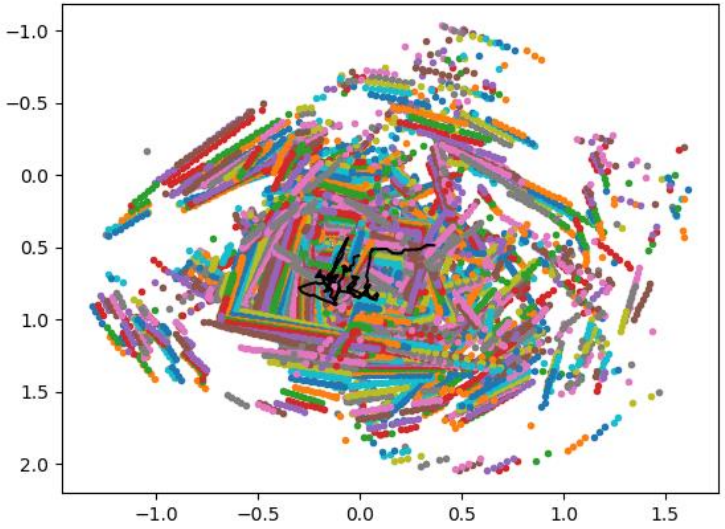
$$f_\theta \ : \ S_i \ \mapsto \ T_i$$

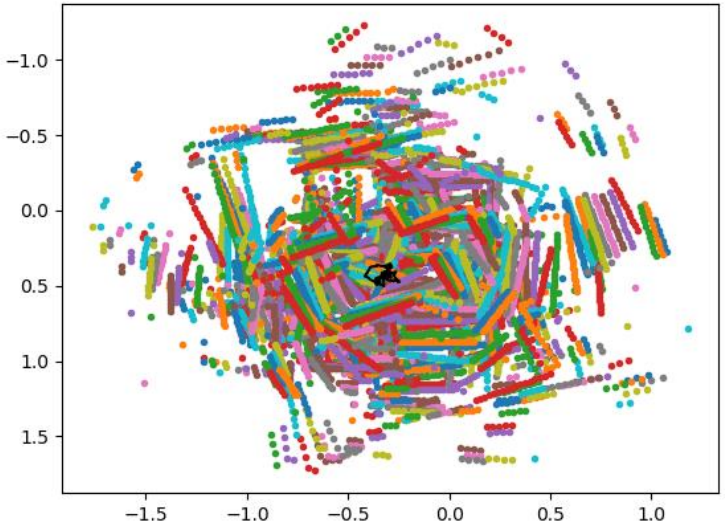
Those occupancy probabilities are used for computing the unsupervised loss, the BCE between predicted probability and occupancy labels. The DM is trained by minimizing the BCE loss and equivalently improve the registration quality.

2D Point Cloud Registration

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

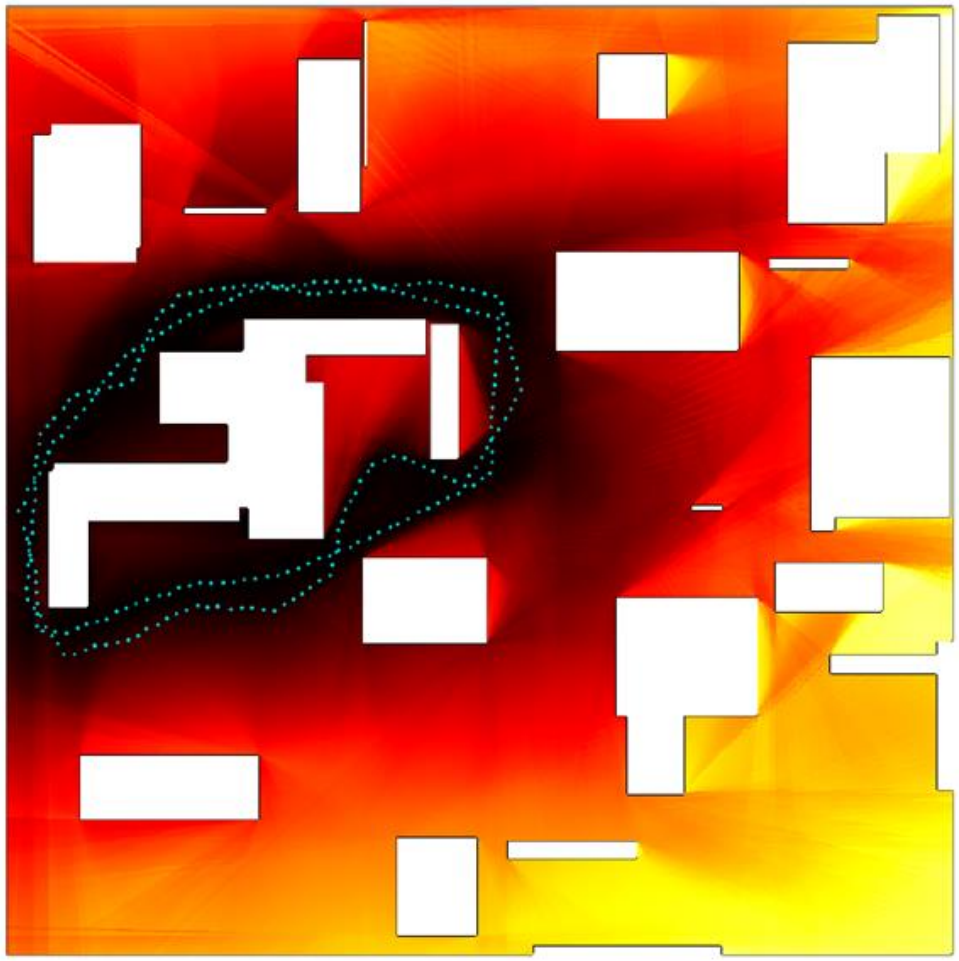


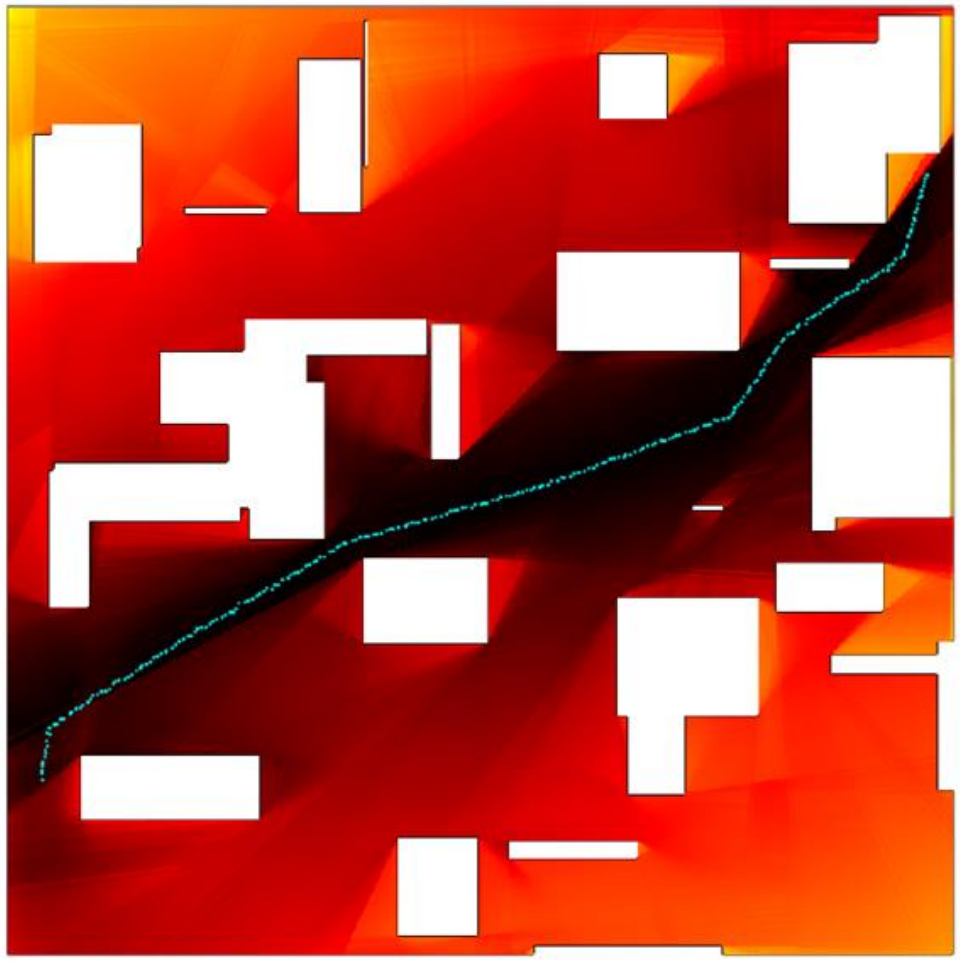




L-Net: Coarse Re-localization For Unseen Point Clouds

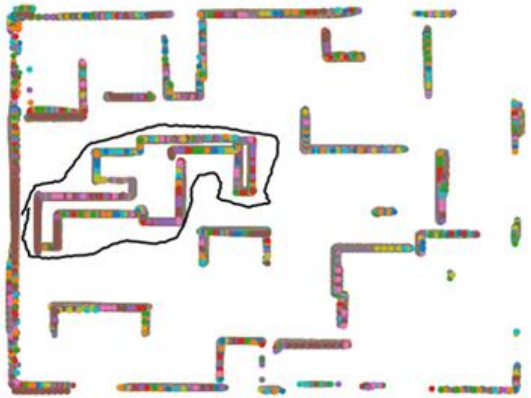
- Re-localization errors of two L-Nets “trained” on two trajectories
  - Darker is better



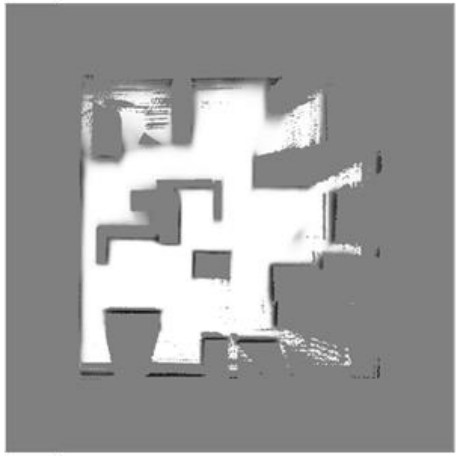


M-Net: Continuous Occupancy Map Representation

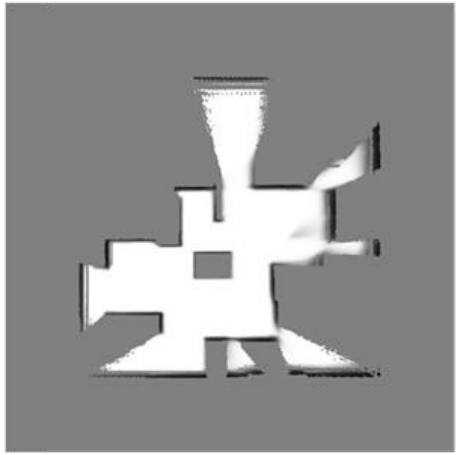
Aligned point clouds



M-Net output

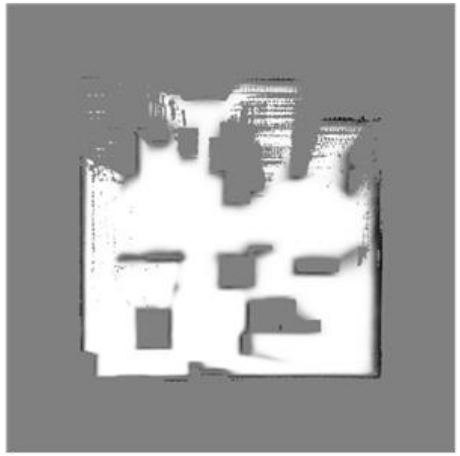


(Occupancy Map)

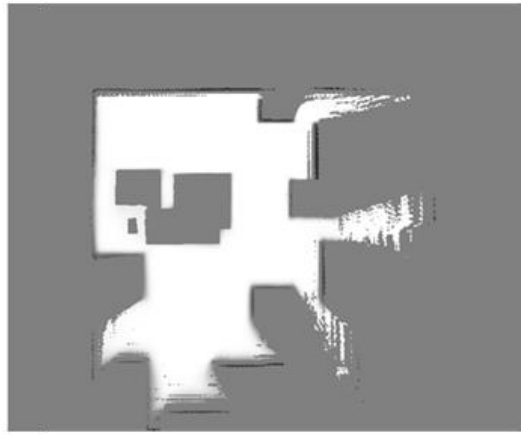


Aligned point clouds

M-Net output



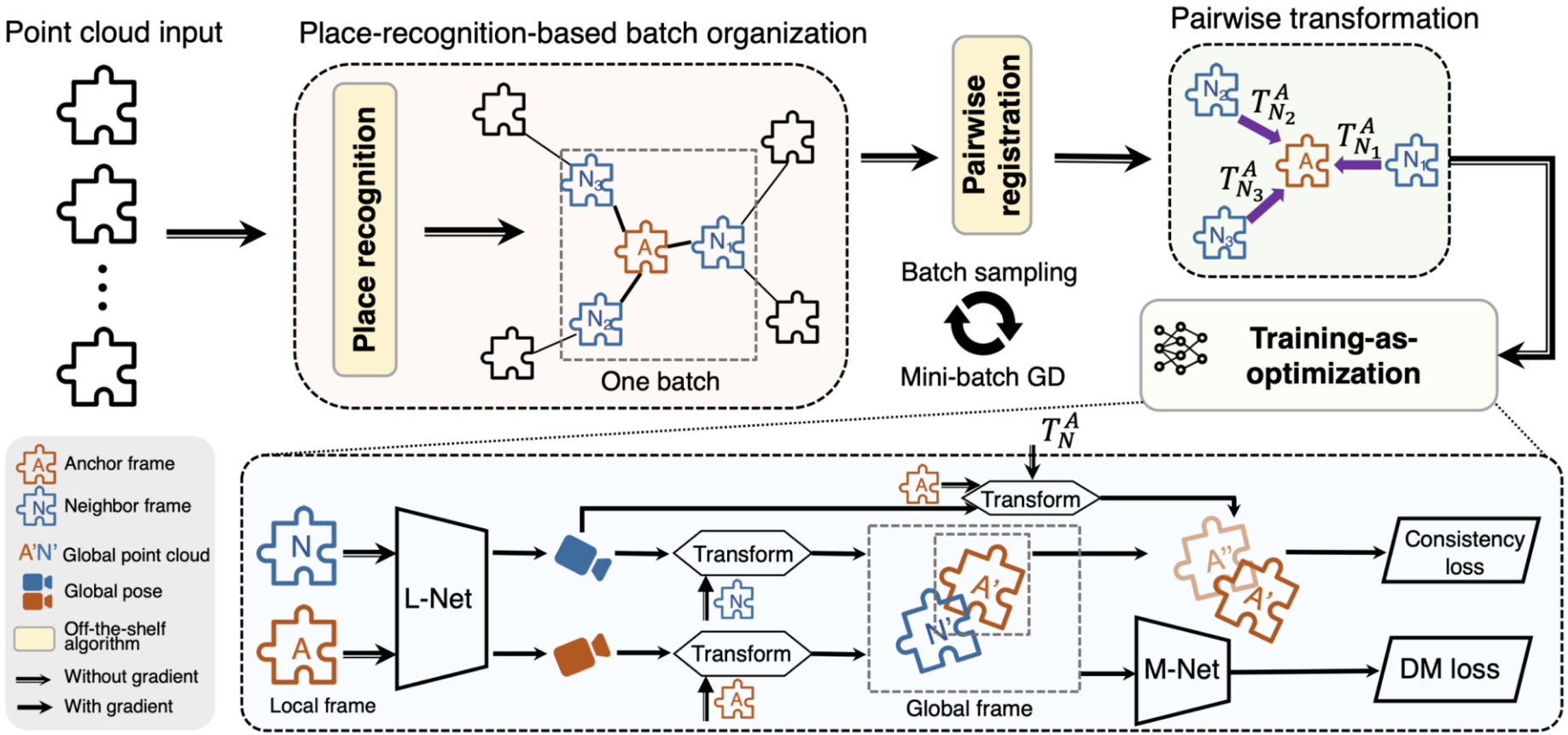
(Occupancy Map)



DeepMapping2: Self-Supervised Large-Scale LiDAR Map Optimization

Main Updates

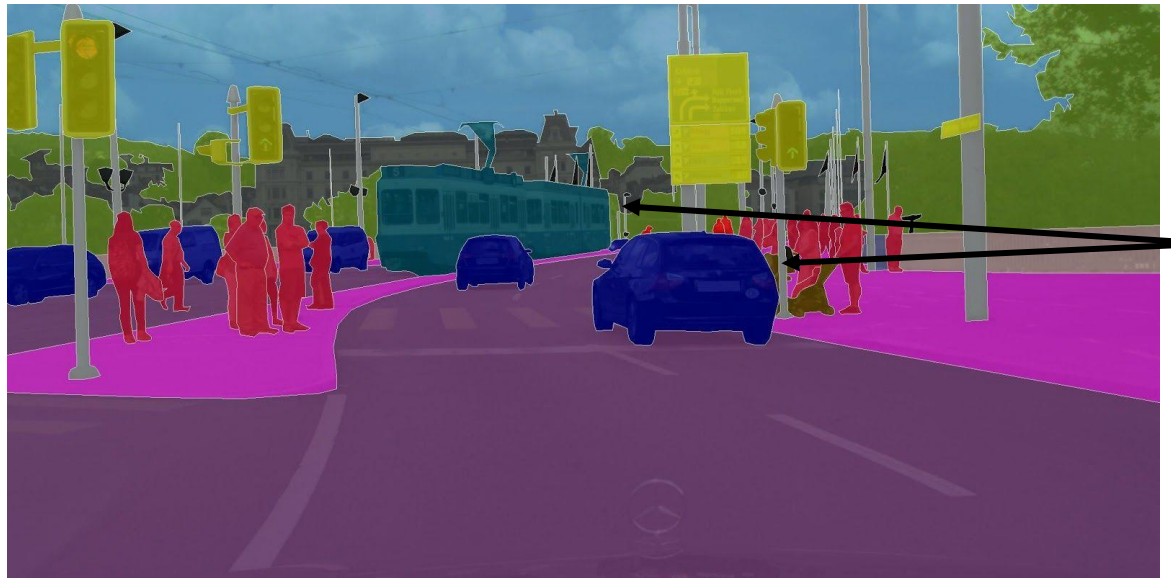
- Introduces technique for organization of training batch based on map topology from **loop closing**
- A self-supervised local-to-global point **consistency loss** via pairwise registration for better correspondences



Semantic Occupancy Prediction

Thought 🤔💭

- Let's turn our attention to another related task – Semantic Occupancy Prediction
- For task such as autonomous driving, a 2D segmentation map is not good enough! (hard to tell how far away the incoming obstacle really is)



How far?

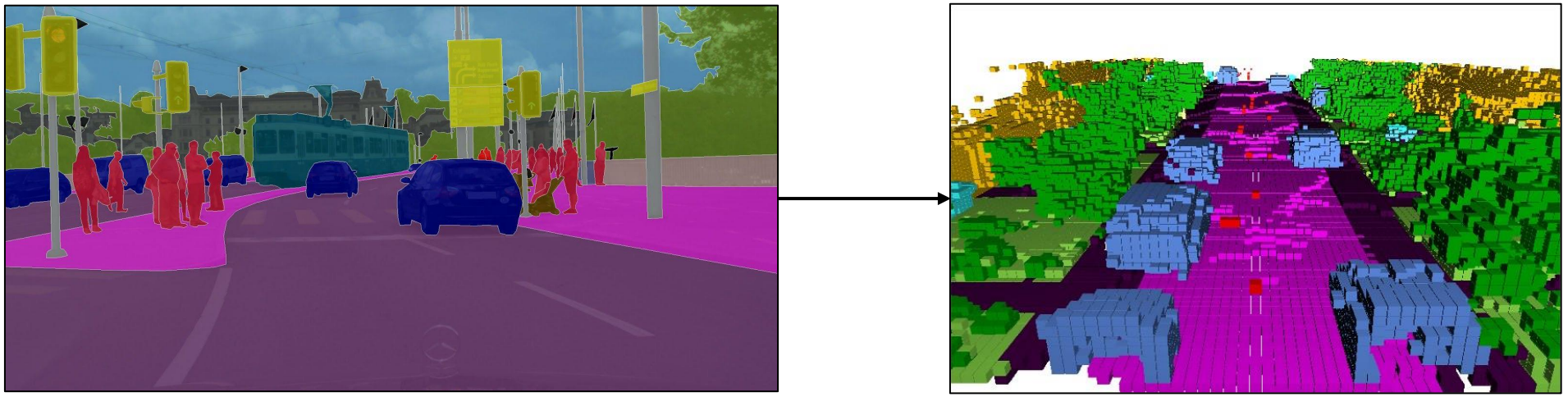
How much "space" it actually takes up?

Hard to tell from 2D images alone

Semantic Occupancy Prediction

Thought 🤔💭

- Point-cloud representation is not ideal for autonomous driving planning either.
- Perhaps we can predict and use voxelized occupancy grids instead?
- Building on earlier work such as what we have seen prior and Occupancy Network, it's later blown up by Tesla's keynote at CVPR 2022.



Semantic Occupancy Prediction – Quick Survey

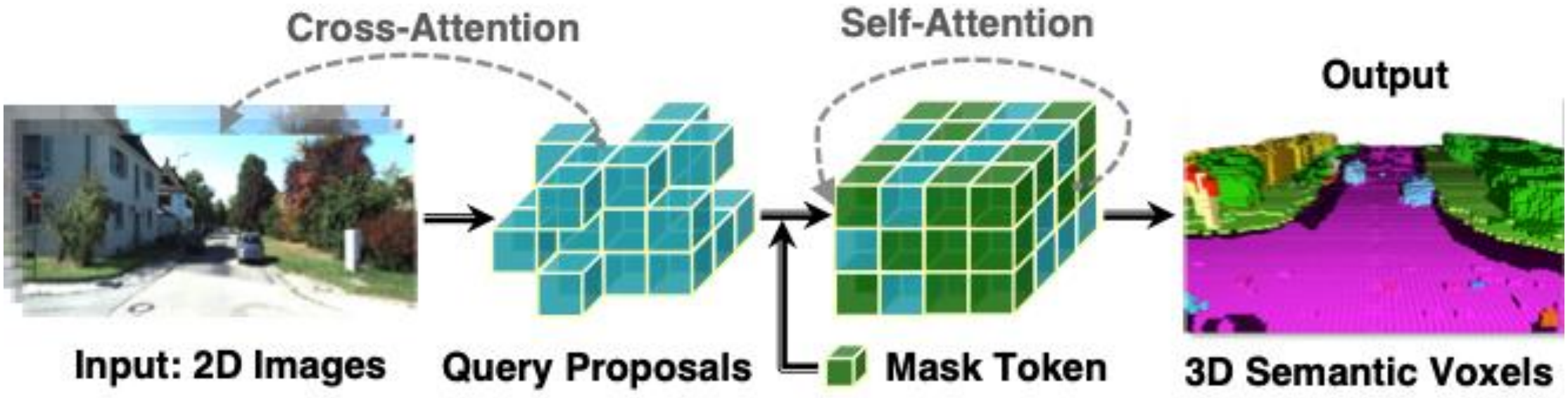
There has been *many* new work recently on this topic, with *many* approaches.

- Cascade Occupancy Network (CONet) (this is a good entry point)
- VoxFormer (from our lab! 😎)
- TPVFormer (a flavor with BEV)
- MonoScene (an ”older” work circa 2022)
- SSCNet (one of the first work on Semantic Scene Completion; not on self-driving)

Feel free to jump into this rabbit hole at your own discretion, but for now, let’s take a quick look at VoxFormer as an example!

VoxFormer - Overview

Unlike conventional formulation, VoxFormer only takes monocular RGB images as input; does not need any 3D input!



VoxFormer: Sparse Voxel Transformer for Camera-based 3D Semantic Scene Completion, Li et al., 2023

VoxFormer - Methodology

A 2-stage setup!

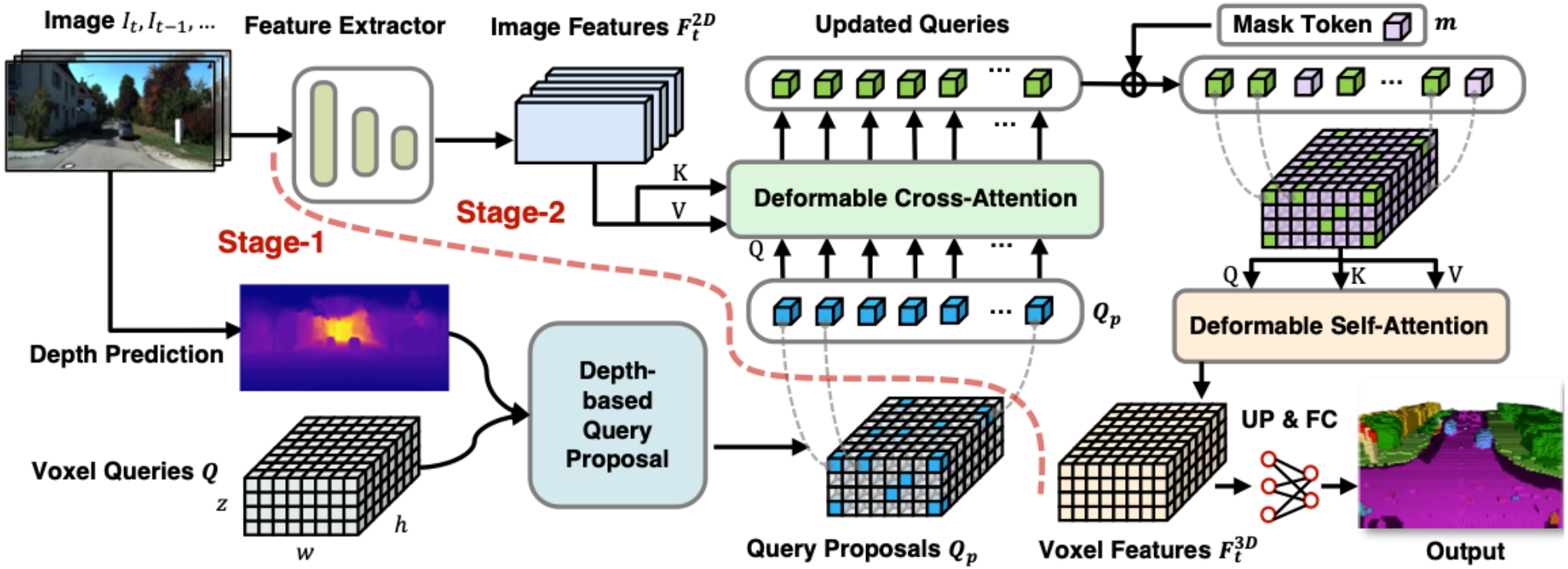


Figure 2. **Overall framework of VoxFormer.** Given RGB images, 2D features are extracted by ResNet50 [61] and the depth is estimated by an off-the-shelf depth predictor. The estimated depth after correction enables the class-agnostic query proposal stage: the query located at an occupied position will be selected to carry out deformable cross-attention with image features. Afterwards, mask tokens will be added for completing voxel features by deformable self-attention. The refined voxel features will be upsampled and projected to the output space for per-voxel semantic segmentation. Note that our framework supports the input of single or multiple images.

VoxFormer: Sparse Voxel Transformer for Camera-based 3D Semantic Scene Completion, Li et al., 2023

VoxFormer - Results

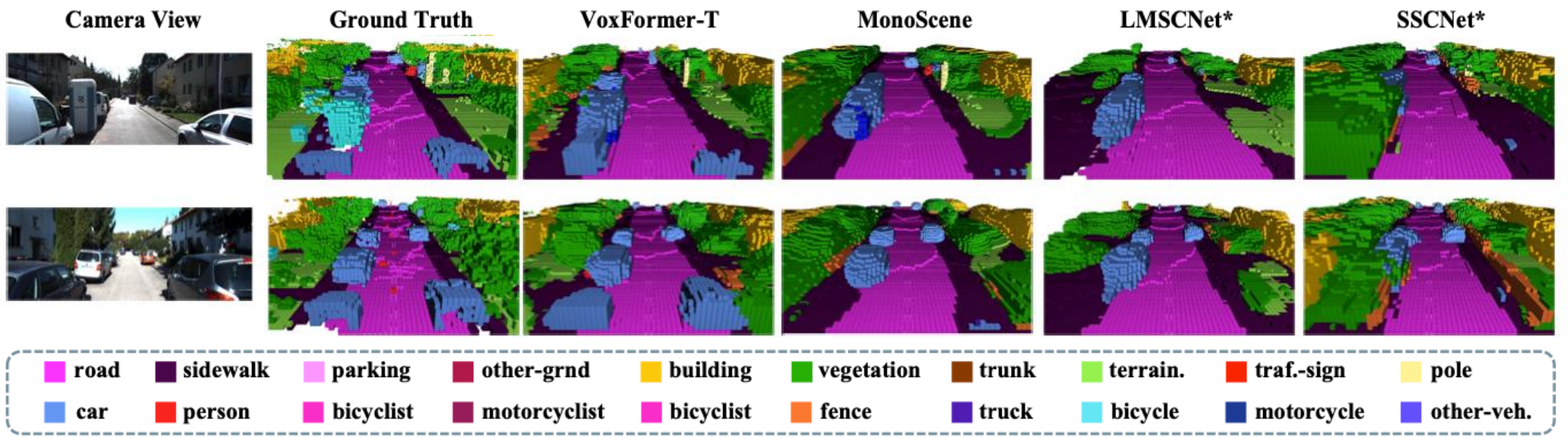


Figure 3. **Qualitative results of our method and others.** VoxFormer better captures the scene layout in large-scale self-driving scenarios. Meanwhile, VoxFormer shows satisfactory performances in completing small objects such as trunks and poles.

Methods	VoxFormer-T (Ours)			VoxFormer-S (Ours)			MonoScene [4]			LMSCNet* [6]			SSCNet* [1]		
Range	12.8m	25.6m	51.2m	12.8m	25.6m	51.2m	12.8m	25.6m	51.2m	12.8m	25.6m	51.2m	12.8m	25.6m	51.2m
IoU (%)	65.38	57.69	44.15	65.35	57.54	44.02	38.42	38.55	36.80	65.52	54.89	38.36	59.51	53.20	40.93
Precision (%)	76.54	69.95	62.06	77.65	70.85	62.32	51.22	51.96	52.19	86.51	82.21	77.60	65.38	59.13	48.77
Recall (%)	81.77	76.70	60.47	80.49	75.39	59.99	60.60	59.91	55.50	72.98	62.29	43.13	86.89	84.15	71.80

Table 1. **Quantitative comparison** against the state-of-the-art **camera-based** SSC methods. We report the performances inside three volumes, *i.e.*, $12.8\times12.8\times6.4m^3$, $25.6\times25.6\times6.4m^3$, and $51.2\times51.2\times6.4m^3$. The first two volumes are introduced for assessing the SSC performance in safety-critical nearby locations. The top three performances are marked by **red**, **green**, and **blue** respectively.

VoxFormer: Sparse Voxel Transformer for Camera-based 3D Semantic Scene Completion, Li et al., 2023

Neural Implicit 3D Segmentation

Thought 🤔💭

- What we have seen so far seems to have a close relation with NeRF if you think about it.
- Can you think of a way to simplify difficult task like 3D segmentation with ideas from NeRF somehow?
- **Idea**: Why not output segmentation mask / map with NeRF as well? But how?

Neural Implicit 3D Segmentation – Problem Formulation

Let's refine our idea 🧐💭

- **Question:** How to perform segmentation with NeRF as well?

- **Naive Idea**: Just add another prediction head and directly classify the class / instance id from 2D monocular inputs!

  - Problem! This is actually a very hard task! (Need lots of labelled 3D data)
  - Also, if you think about it, this is not what NeRF is about either!
  - With NeRF, we want to learn representation of a *single* scene, which means we don't have any "pre-training" to learn segmentation from!

Neural Implicit 3D Segmentation via Panoptic Lifting

Let's refine our idea 🧐💭

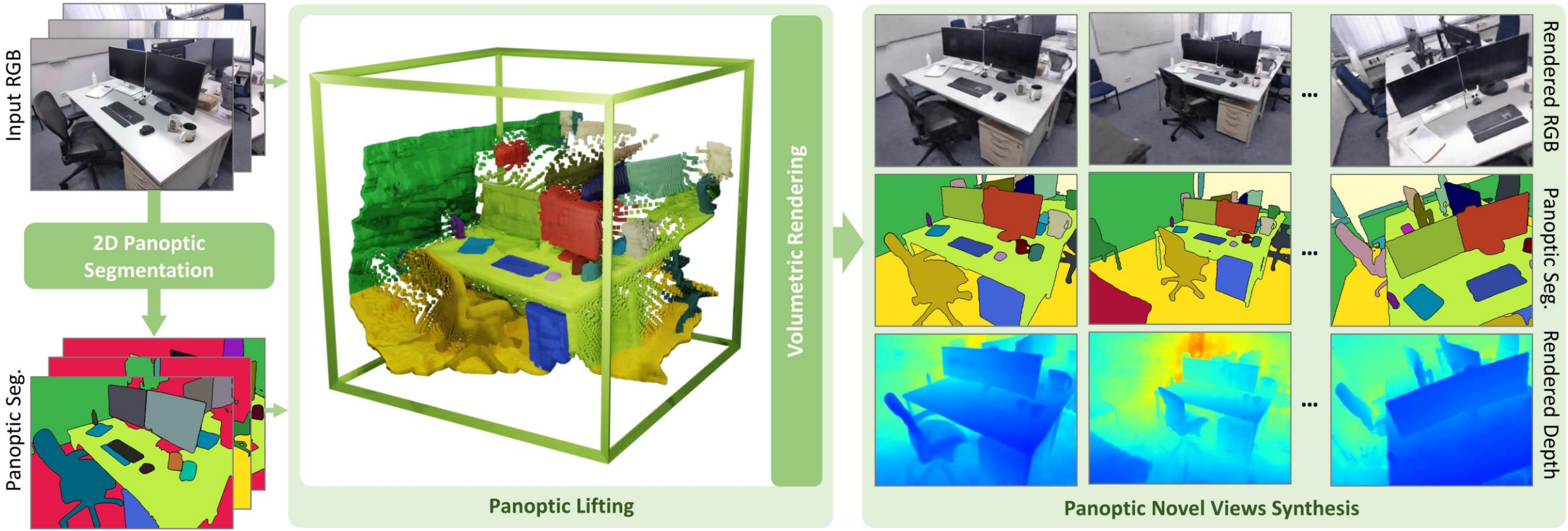
- **Solution**: Use off-the-shelf segmentation model to generate 2D map and feed this as input as well! (Just like RGB inputs to NeRF)

- Therefore, we're now going from
  2D RGB + Segmentation Map → 3D Segmentation!

- This method is **Panoptic Lifting** (literally lifting panoptic map from 2D → 3D)

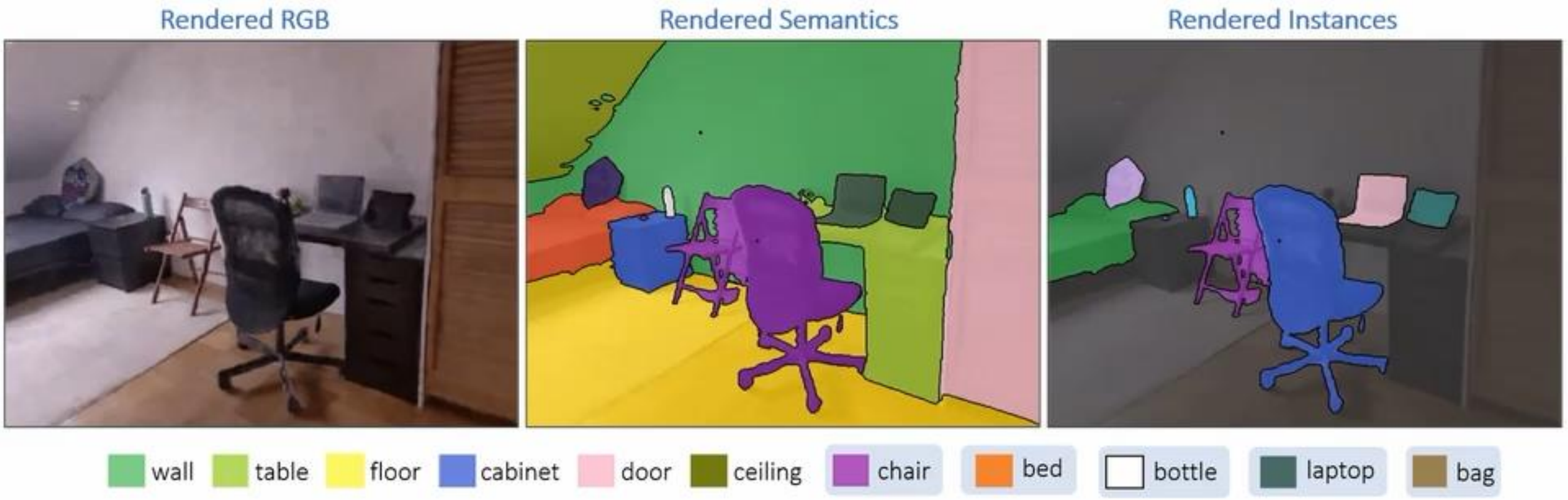
Applications?
VR / AR, 3D scene understanding, 3D scene editing, autonomous driving, etc

Panoptic Lifting - Overview



Panoptic Lifting for 3D Scene Understanding with Neural Fields, Yawar et al., 2023

Panoptic Lifting - Demo



Panoptic Lifting for 3D Scene Understanding with Neural Fields, Yawar et al., 2023

Panoptic Lifting - Demo


Original


Deletion


Duplication

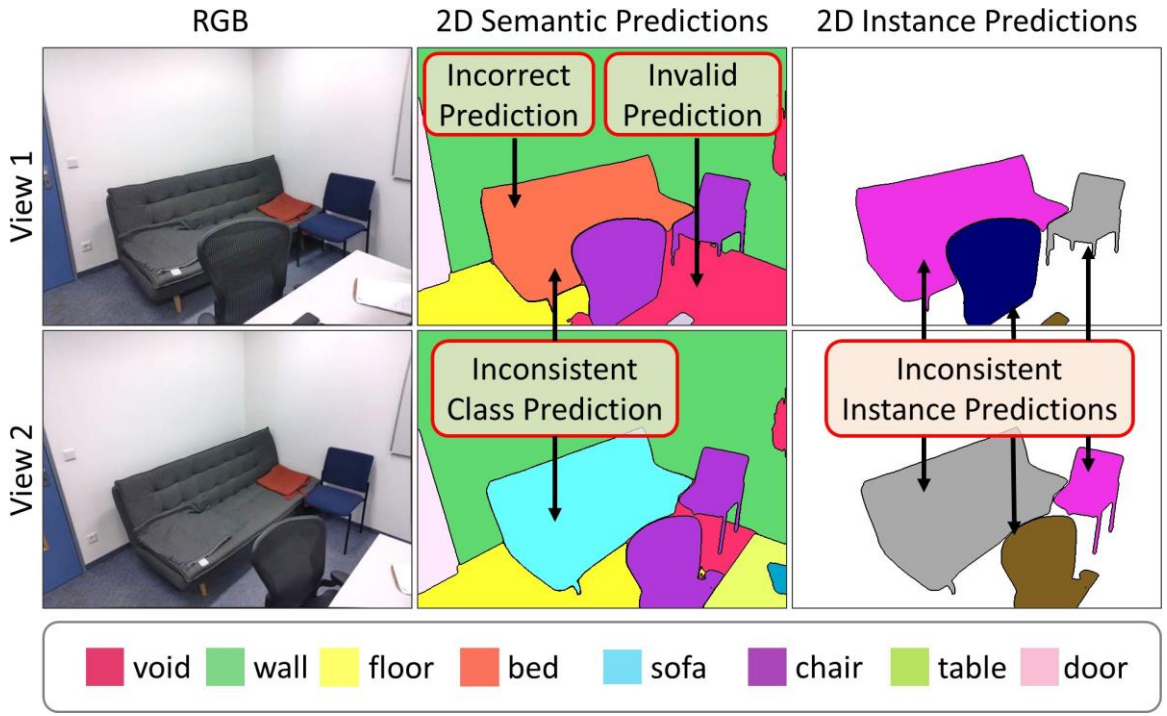

Manipulation

Panoptic Lifting for 3D Scene Understanding with Neural Fields, Yawar et al., 2023

Panoptic Lifting - Methodology

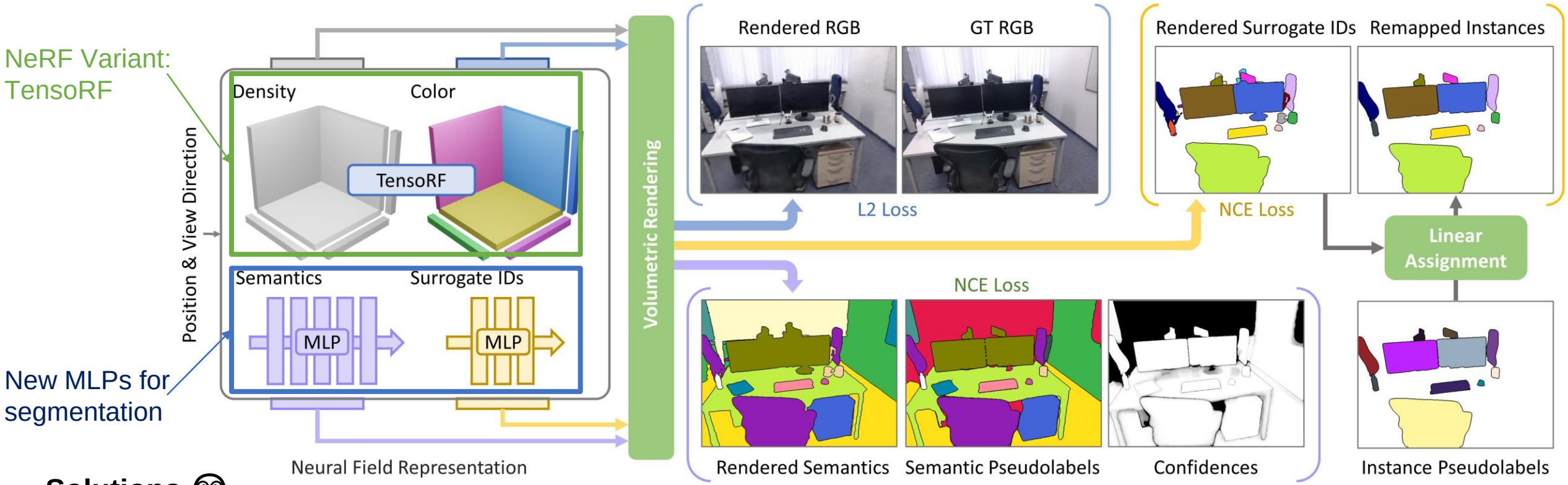
But wait 🧐

- If we just perform such task without additional changes, the result is rather bad!
  - **Problem 1:** Noisy class predictions as there are no enforced “object class consistency” for all reconstructed pixel of an object.
  - **Problem 2:** Also, the input pseudo-labels from off-the-shelf segmentation models are noisy and inconsistent as well! Garbage in, garbage out.



Panoptic Lifting for 3D Scene Understanding with Neural Fields, Yawar et al., 2023

Panoptic Lifting – Methodology Details



Solutions 🤓

1. Test-time augmentation: For each image, average multiple predictions with augmentation applied (e.g. flip)
2. Segment-consistency Loss: Enforce object-class consistency across rays sampled!
3. ”Additional contamination blockers”: Gradient-blocking & Bounded segmentation field (e.g. prevent model from simply changing NeRF geometry to satisfy class label!)

Panoptic Lifting for 3D Scene Understanding with Neural Fields, Yawar et al., 2023

Panoptic Lifting – Results

Method	HyperSim [29]			Replica [34]			ScanNet [9]		
	mIoU↑	PQ^{scene} ↑	PSNR↑	mIoU↑	PQ^{scene} ↑	PSNR↑	mIoU↑	PQ^{scene} ↑	PSNR↑
Mask2Former [6]	53.9 (-13.9)	–	–	52.4 (-14.8)	–	–	46.7 (-18.5)	–	–
SemanticNeRF [45]	58.9 (-8.9)	–	26.6	58.5 (-8.7)	–	24.8	59.2 (-6)	–	26.6
DM-NeRF [40]	57.6 (-10.2)	51.6 (-8.5)	28.1	56.0 (-11.2)	44.1 (-13.8)	26.9	49.5 (-15.7)	41.7 (-17.2)	27.5
PNF [18]	50.3 (-17.5)	44.8 (-15.3)	27.4	51.5 (-15.7)	41.1 (-16.8)	29.8	53.9 (-11.3)	48.3 (-10.6)	26.7
PNF + GT Bounding Boxes	58.7 (-9.1)	47.6 (-12.5)	28.1	54.8 (-12.4)	52.5 (-5.4)	**31.6**	58.7 (-6.5)	54.3 (-4.6)	26.8
Panoptic Lifting	**67.8**	**60.1**	**30.1**	**67.2**	**57.9**	29.6	**65.2**	**58.9**	**28.5**

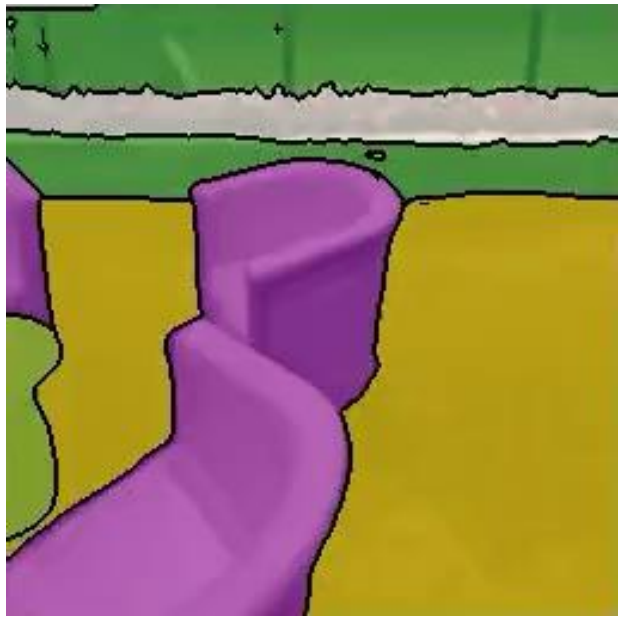
Table 1. Quantitative comparison on novel views from the test set. We outperform both 2D and 3D NeRF methods on semantic and panoptic segmentation tasks. Note that, compared to other methods, *PNF+GT Bounding Boxes* is given the advantage of using ground-truth 3D detections. Mask2Former does not predict scene-level object instances, thus it can't be evaluated for PQ^{scene}.

Segment Consistency	TTA	Bounded Segm. Field	Gradient Blocking	mIoU↑	PQ^{scene} ↑	PSNR↑
✗	✗	✗	✗	57.3	47.9	27.1
✗	✓	✓	✓	60.9	54.3	28.3
✓	✗	✓	✓	63.1	55.2	28.4
✓	✓	✗	✓	62.9	52.5	28.4
✓	✓	✓	✗	61.6	53.7	27.2
✓	✓	✓	✓	**65.2**	**58.9**	**28.5**

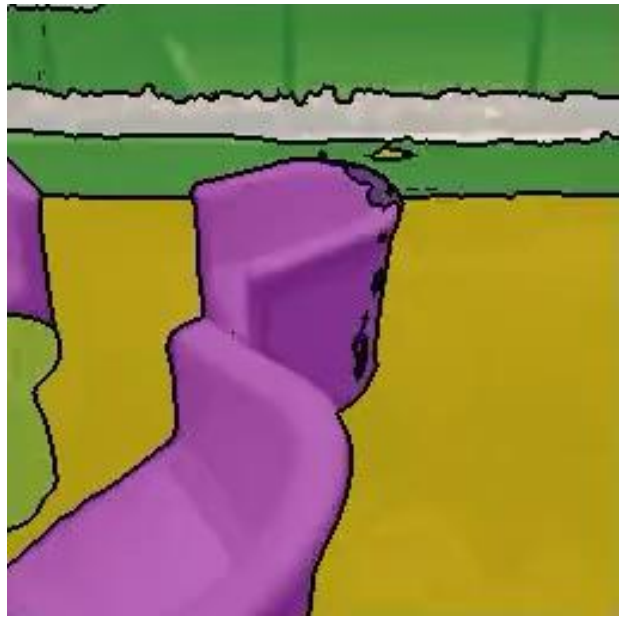
Table 2. Ablations of our design choices on ScanNet [9]. The segment consistency loss, test-time augmentations (TTA), probability field, and blocked segmentation gradients all contribute to robustness against real-world noisy labels.

Works pretty well! 

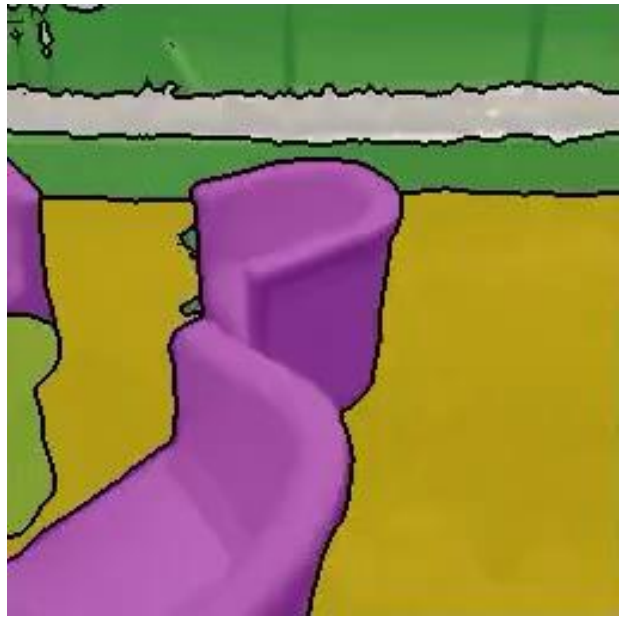
Panoptic Lifting – Ablation Visualization



Proposed



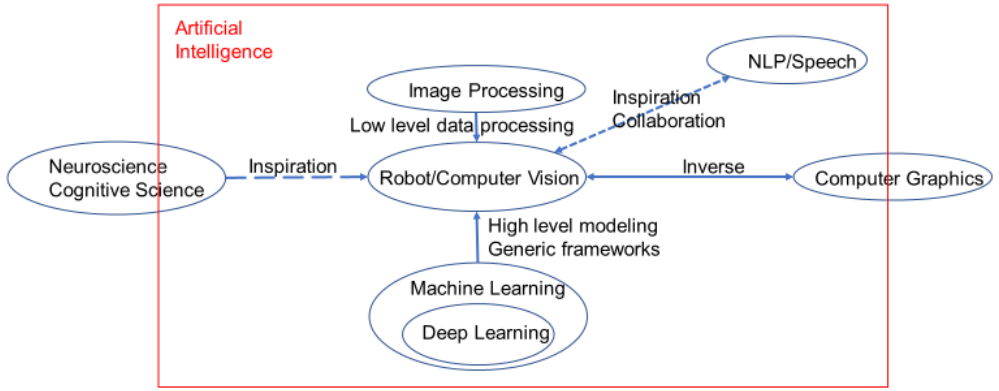
w/o Segment Consistency



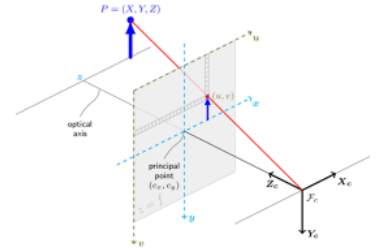
w/o Bounded Seg. Field

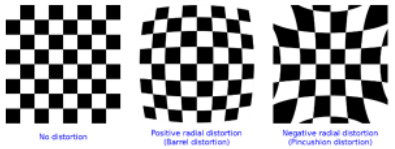
w/o Gradient Blocking

Related Disciplines



Everything Together in OpenCV (Full Model)



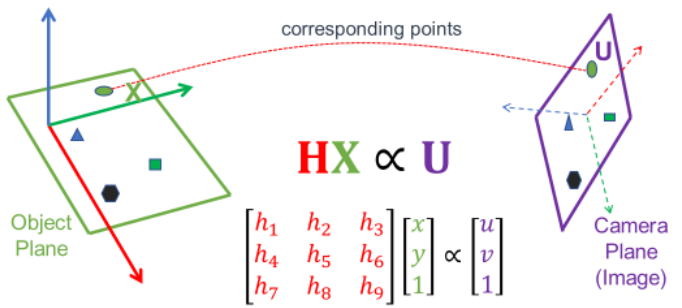


$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$
$$y' = y/z$$
$$x'' = x' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x' y' + p_2(r^2+2x'^2)$$
$$y'' = y' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + p_1(r^2+2y'^2) + 2p_2 x' y'$$
$$\text{where} \quad r^2 = x'^2 + y'^2$$
$$u = f_x * x'' + c_x$$
$$v = f_y * y'' + c_y$$

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

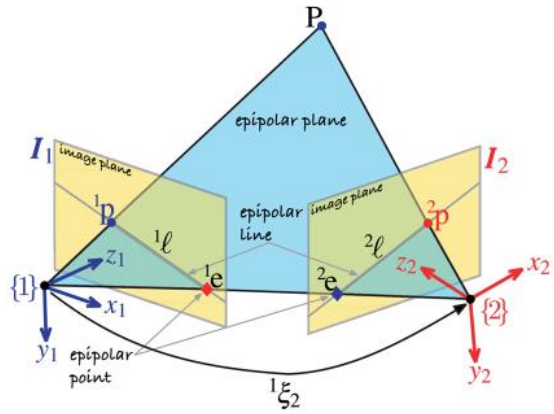
How to Estimate a Homography?



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

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

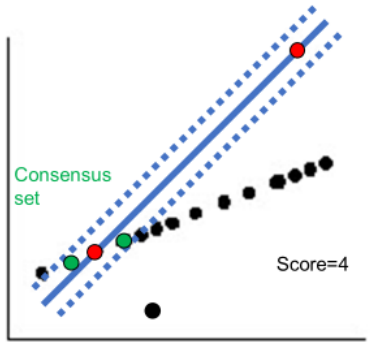
Two-view Geometry



Corke 2011

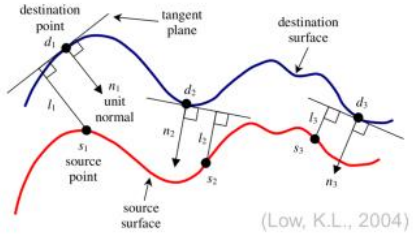
Hypothesis and Test

- Randomly select two points to generate a hypothesis line
- Test how good the current hypothesis is
  - Consensus set = {supporting points}
  - Score by #(supporting points)= |Consensus set|



ICP variants - Change Cost Function

- Point-to-Point: $C(R,t) = \sum_i \|D_i - RS_i - t\|^2$
- Point-to-Plan: $C(R,t) = \sum_i \left\|N_i^T (D_i - RS_i - t)\right\|^2$, N_i being $D_i's$ unit normal
  - Often significantly better convergence rate than point-to-point ICP
  - Linearization under small rotation assumption



(Low, K.L., 2004)

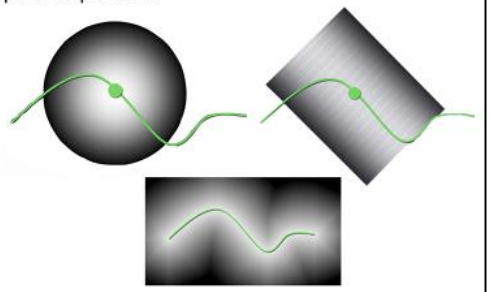
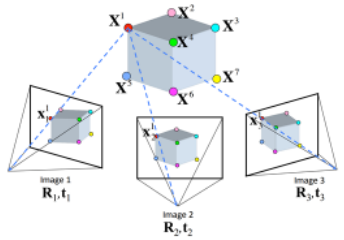


Image from Hao Li.

Yang Chen, and Gerard Medioni. *Object Modeling by Registration of Multiple Range Images.* International Journal of Image and Vision Computing, 10(3), pp. 145–155, 1992.

Writing in Math



	Point 1	Point 2	Point 3
Image 1	$\mathbf{x}_1^1 = \mathbf{K}[\mathbf{R}_1\|\mathbf{t}_1]\mathbf{X}^1$	$\mathbf{x}_1^2 = \mathbf{K}[\mathbf{R}_1\|\mathbf{t}_1]\mathbf{X}^2$	
Image 2	$\mathbf{x}_2^1 = \mathbf{K}[\mathbf{R}_2\|\mathbf{t}_2]\mathbf{X}^1$	$\mathbf{x}_2^2 = \mathbf{K}[\mathbf{R}_2\|\mathbf{t}_2]\mathbf{X}^2$	$\mathbf{x}_2^3 = \mathbf{K}[\mathbf{R}_2\|\mathbf{t}_2]\mathbf{X}^3$
Image 3	$\mathbf{x}_3^1 = \mathbf{K}[\mathbf{R}_3\|\mathbf{t}_3]\mathbf{X}^1$		$\mathbf{x}_3^3 = \mathbf{K}[\mathbf{R}_3\|\mathbf{t}_3]\mathbf{X}^3$

Slides modified from Jianxiong Xiao

Recap the SLAM process

Identify the state space X
- Qualify the domain
- Find a locally Euclidean parameterization

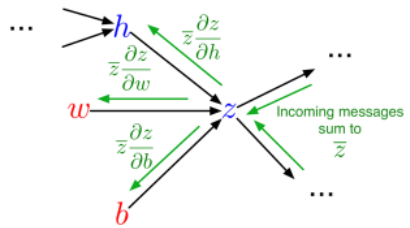
Identify the measurement space(s) Z
- Qualify the domain
- Find a locally Euclidean parameterization

Identify the prediction functions h(x)

Slide from Giorgio Grisetti 2016

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

Visual Place Recognition Pipeline

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

Summary – Deep Learning based Object Detection

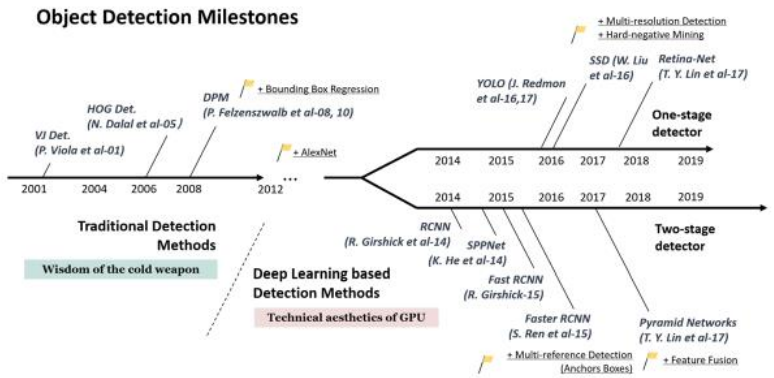


Fig. 2. A road map of object detection. Milestone detectors in this figure: VJ Det. [10, 11], HOG Det. [12], DPM [13–15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20], SSD [21], Pyramid Networks [22], Retina-Net [23].

https://arxiv.org/pdf/1905.05055.pdf

KLT Tracker

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

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

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

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

I Wish You All the Brightest Future!