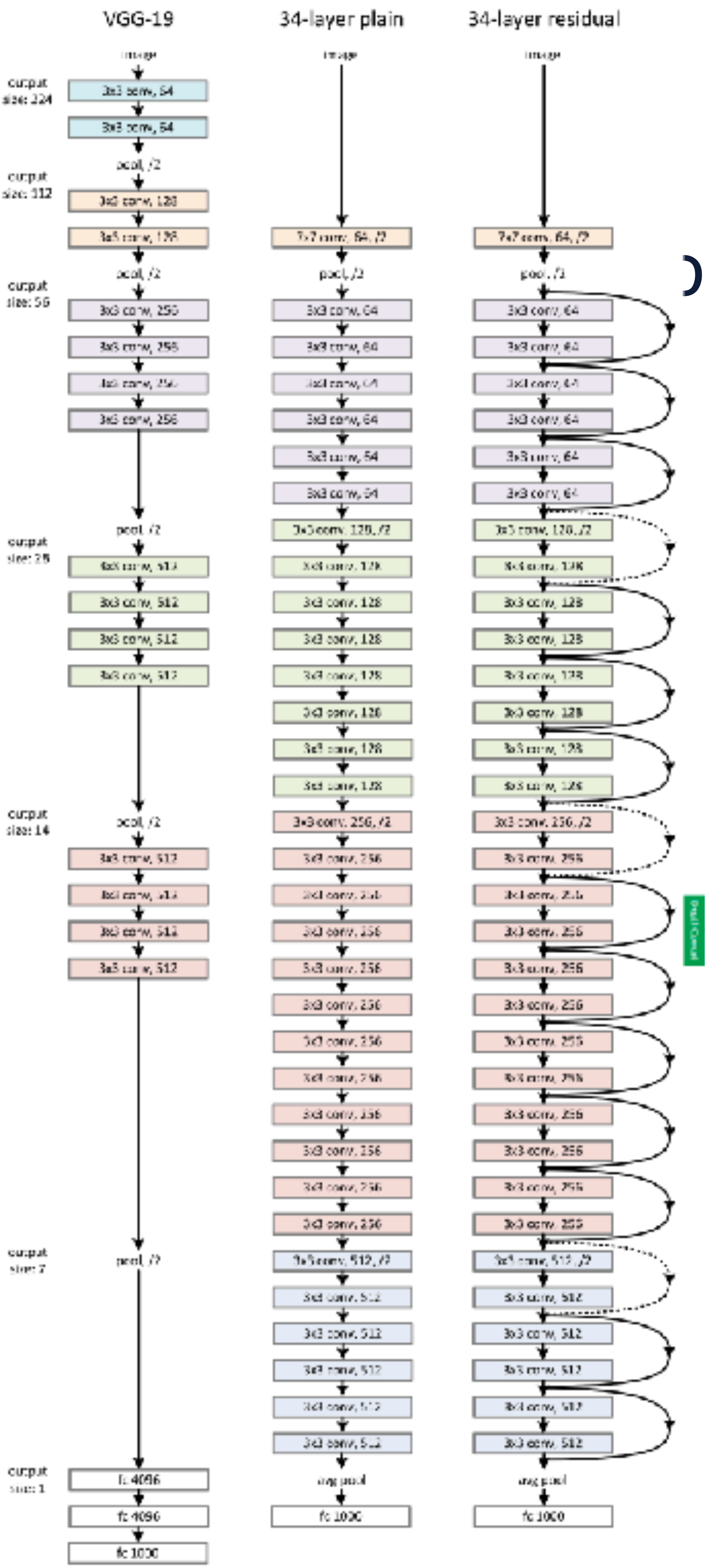
UC San Diego

Deep Learning for 3D Recognition

Instructor: Hao Su
Presenter: Jiayuan Gu

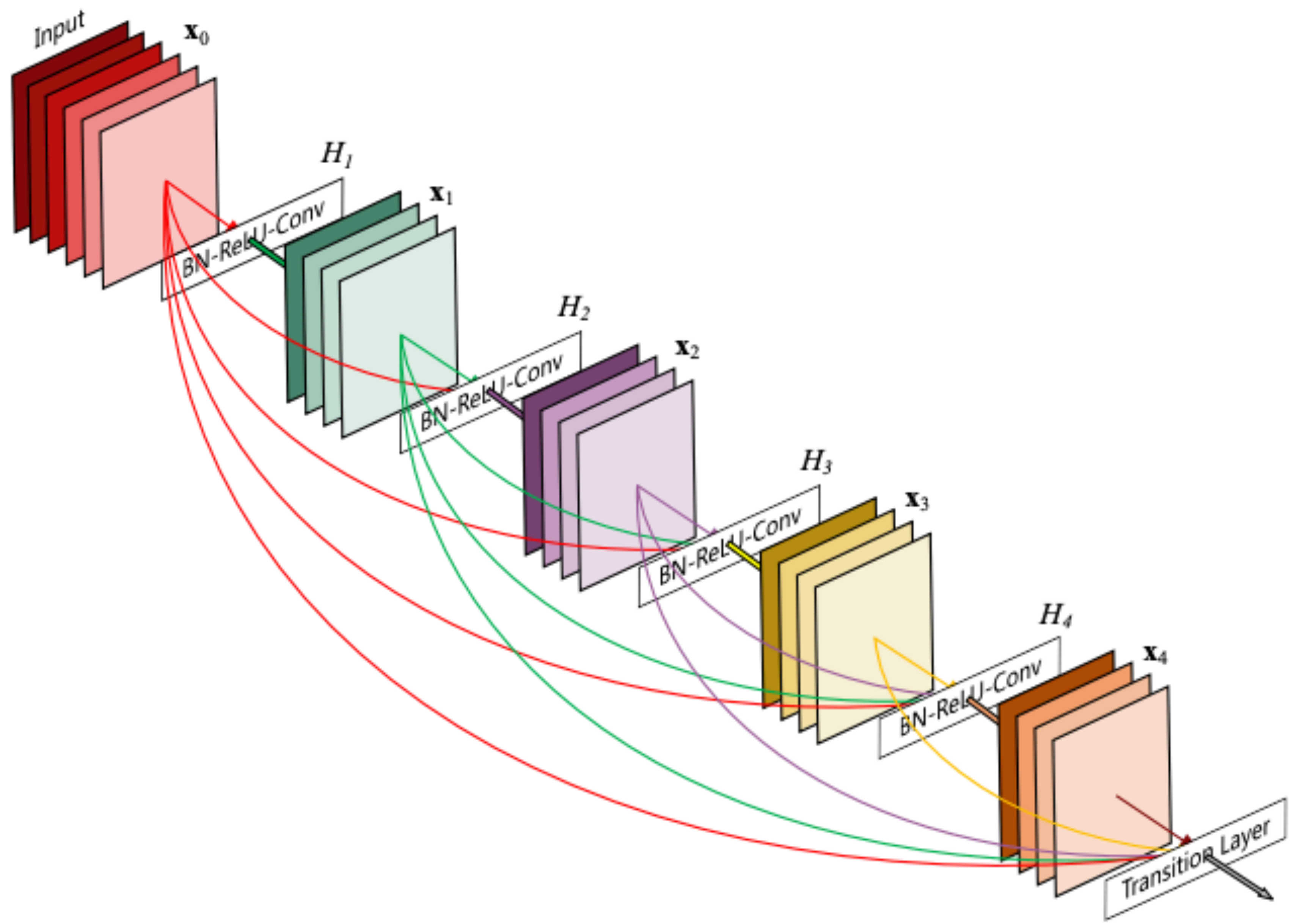
Goal

- Get an idea of the state-of-the-art 3D deep learning methods
- Learn the underlying mathematic

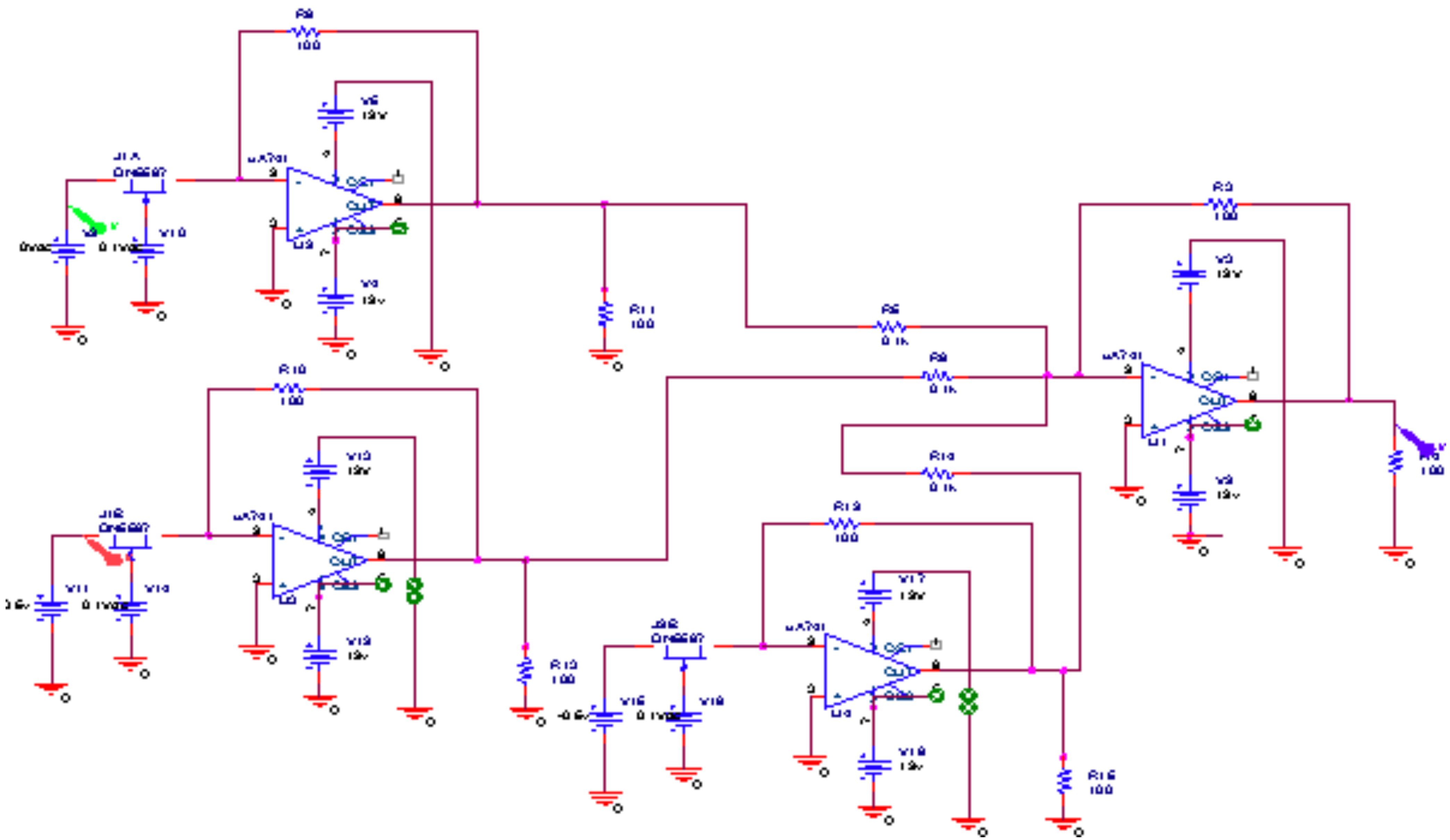


2D Vision

ominates current 2D computer vision?

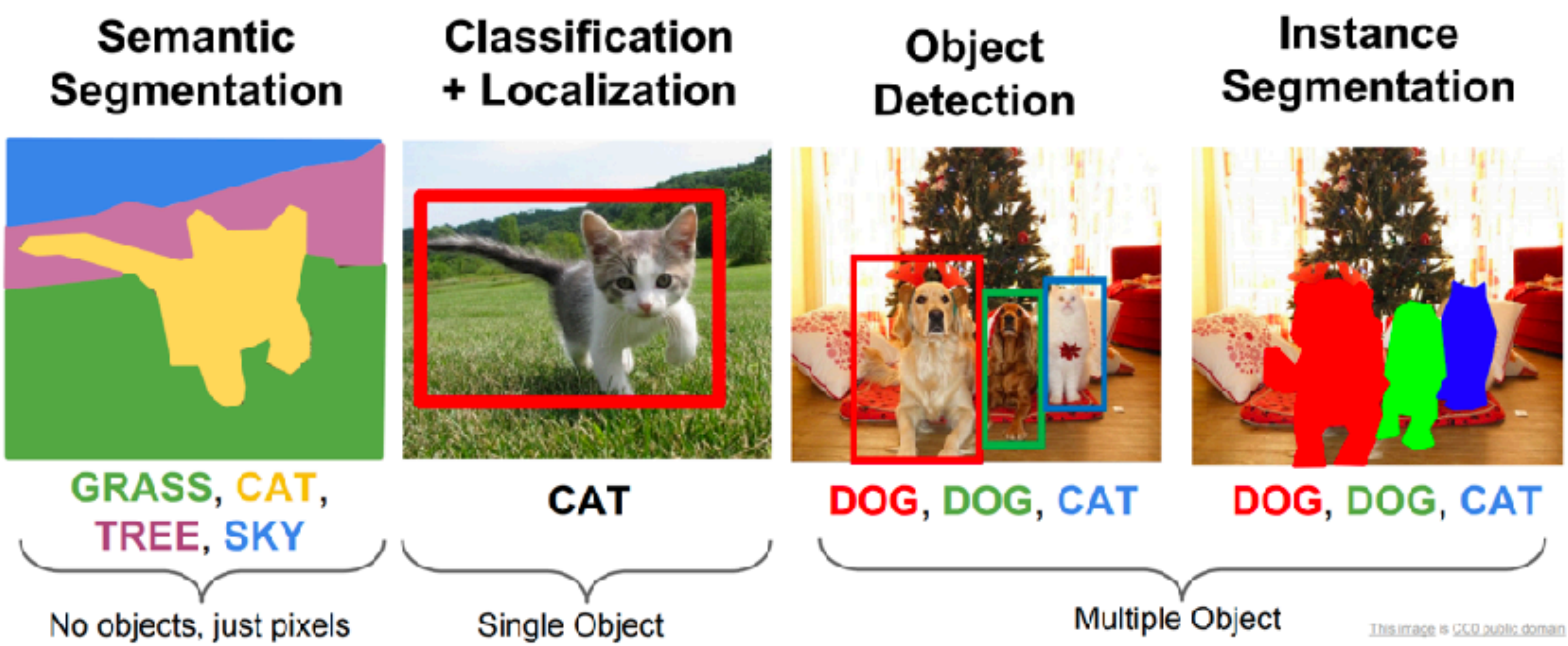


In the future…



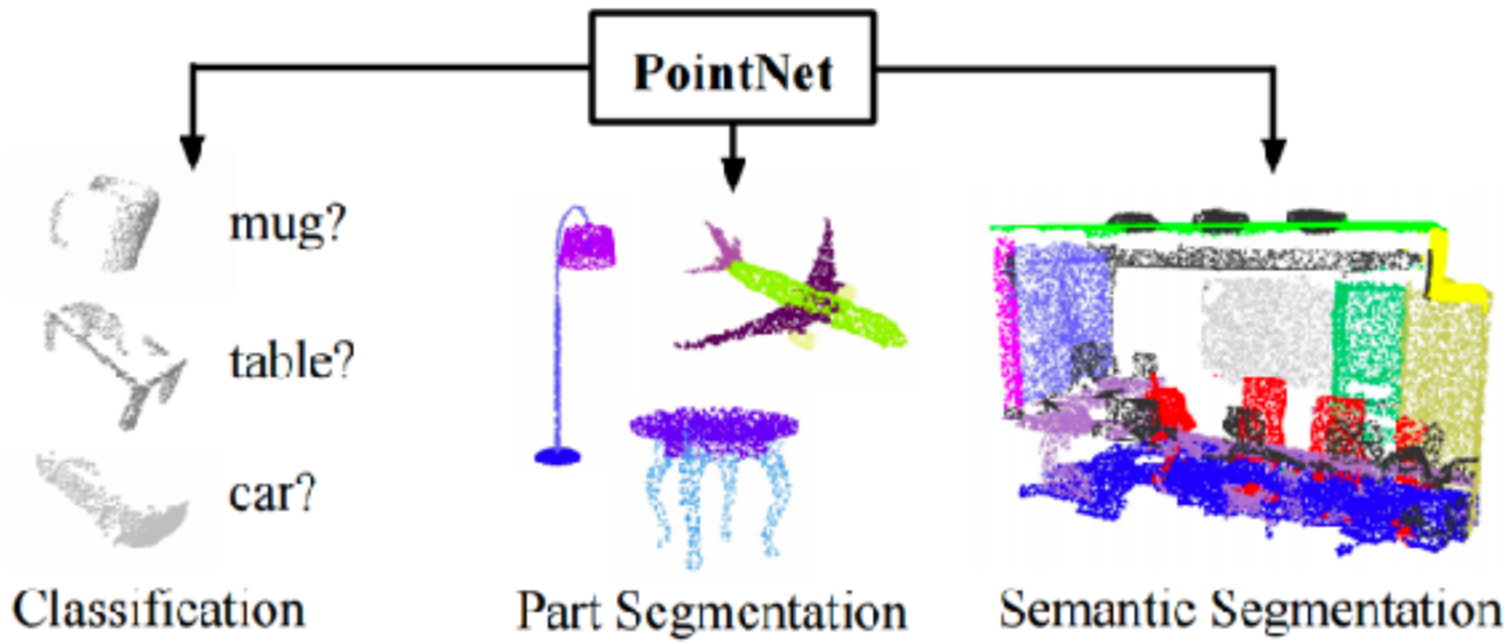
What tasks do 2D CNNs solve?

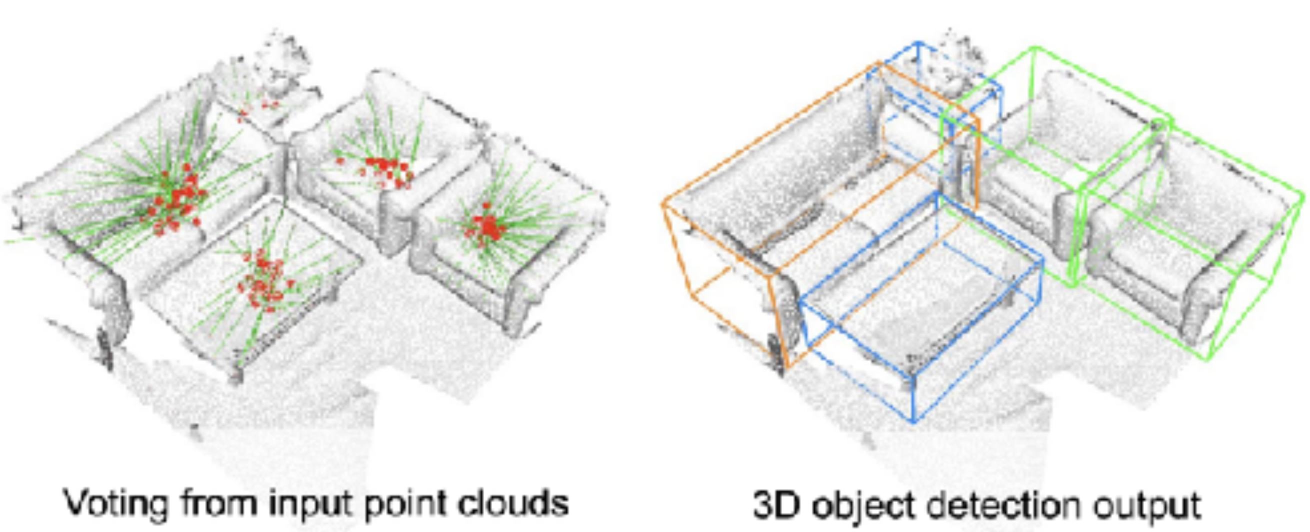
- Classification
- Detection
- Semantic Segmentation
- Instance Segmentation



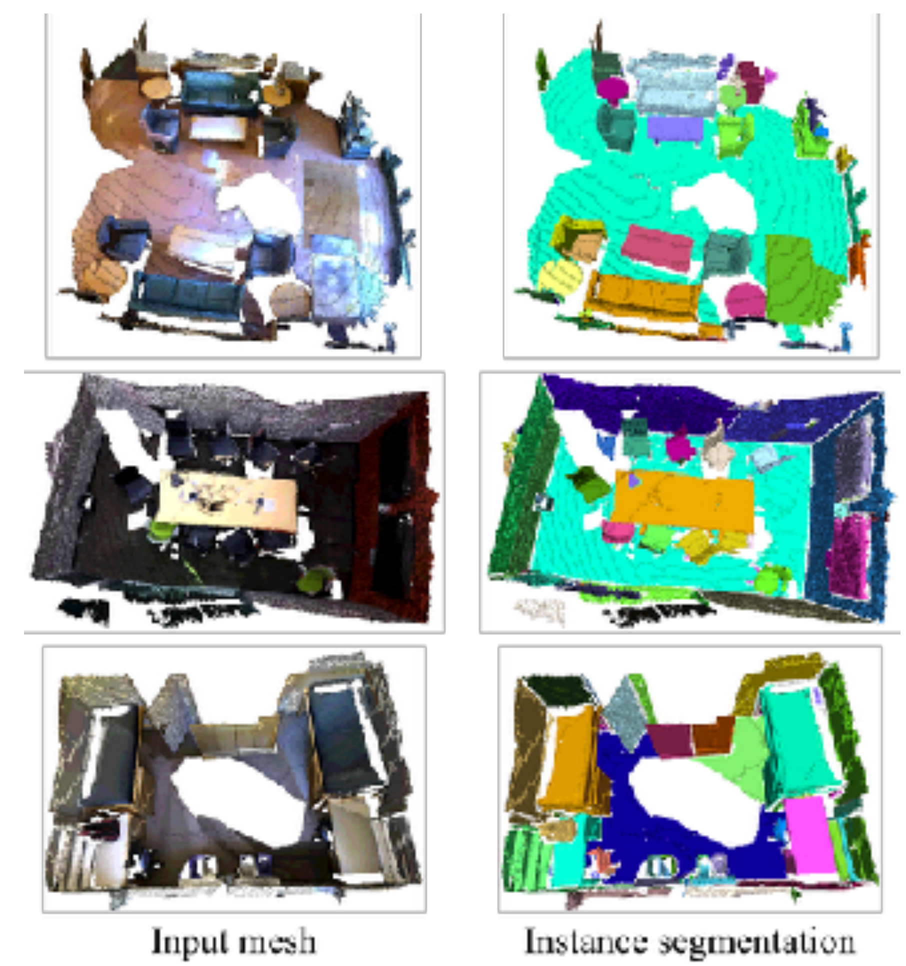
https://towardsdatascience.com/a-hitchhikers-guide-to-object-detection-and-instance-segmentation-ac0146fe8e11

3D tasks



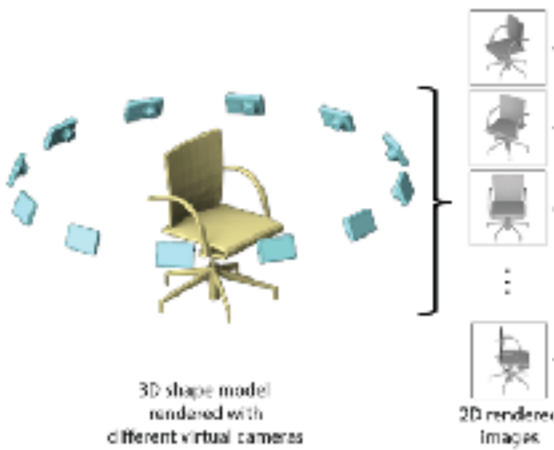


Detection

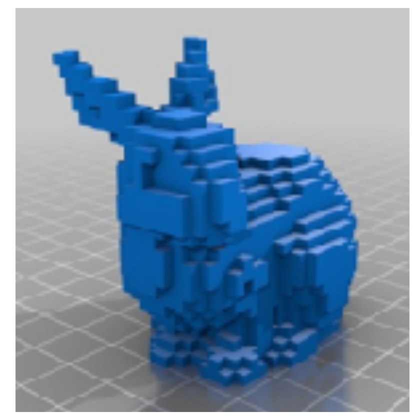


Instance segmentation

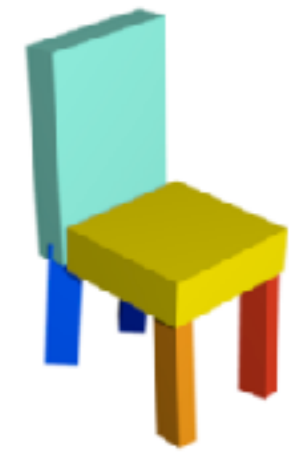
The Representation Challenge of 3D Deep Learning



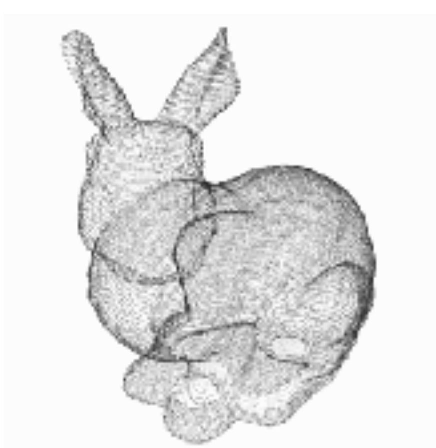
Multi-view



Volumetric



Part Assembly



Point Cloud

Mesh (Graph CNN)

$$F(x) = 0$$

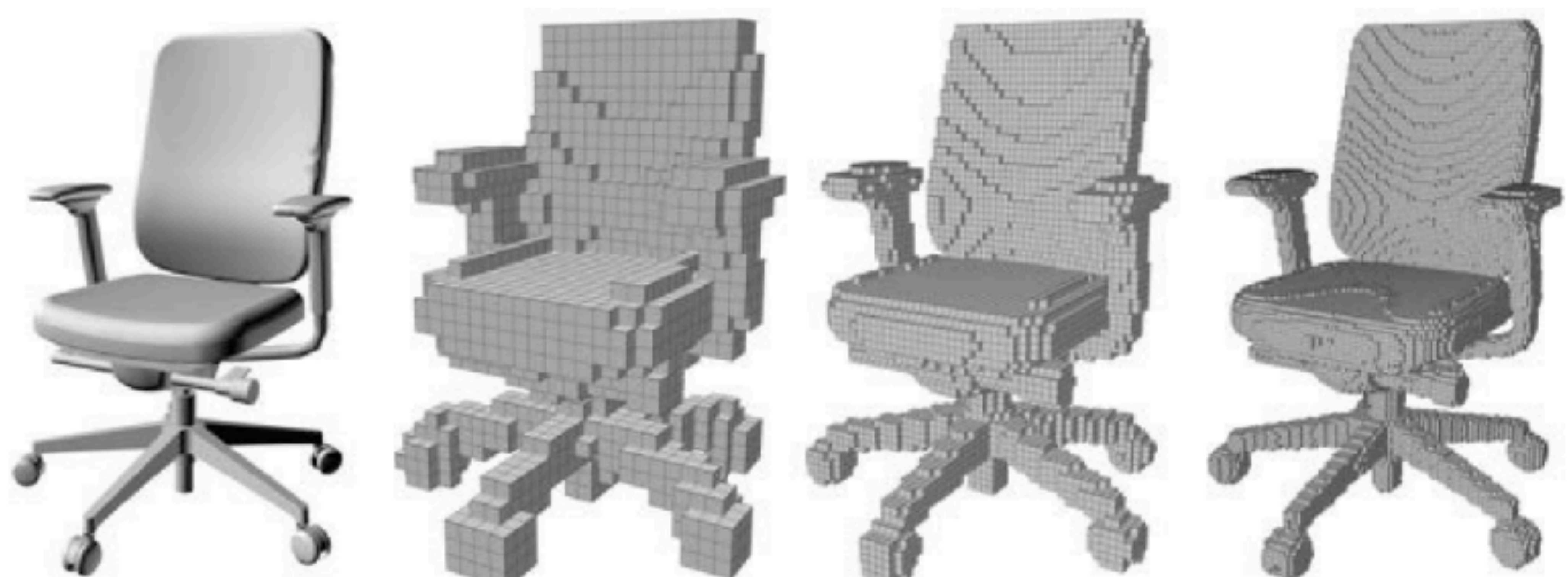
Implicit Shape

The Representation Challenge of 3D Deep Learning

Rasterized form (regular grids)

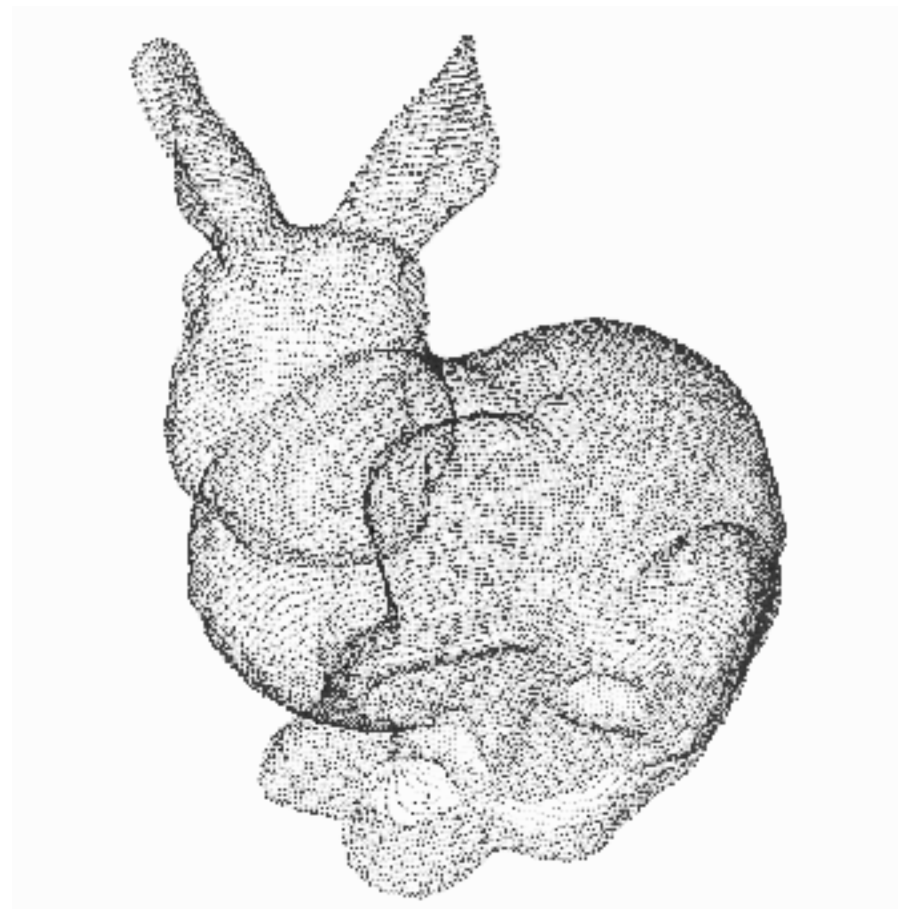
Geometric form (irregular)

Sparsity of 3D Shapes



Occupancy:	10.41%	5.09%	2.41%
Resolution:	32	64	128

It is computationally expensive to do 3D convolution!



Point cloud
(The most common 3D sensor data)

Agenda

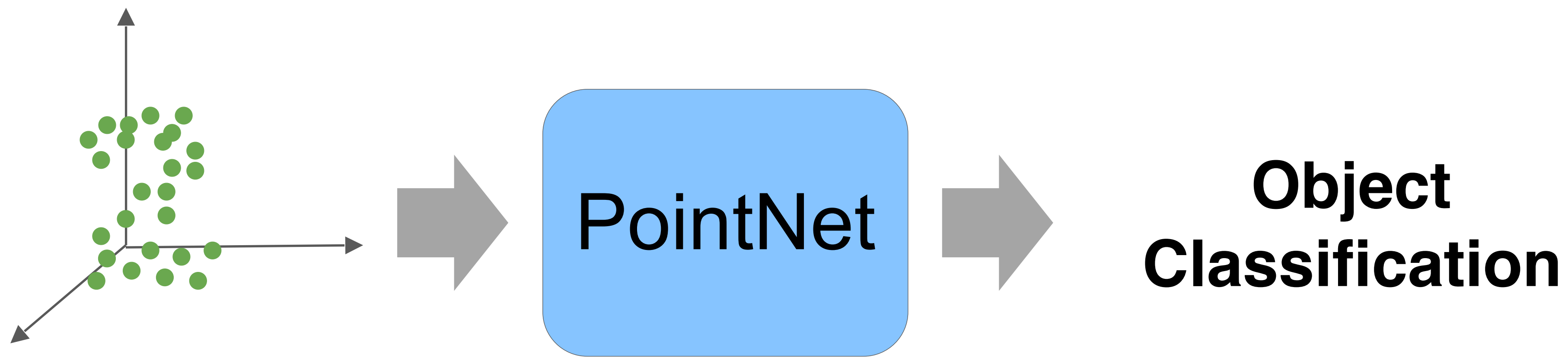
- 3D Classification
  - PointNet
- 3D Segmentation
  - SparseConv
- 3D Detection
  - VoteNet

POINTNET

Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

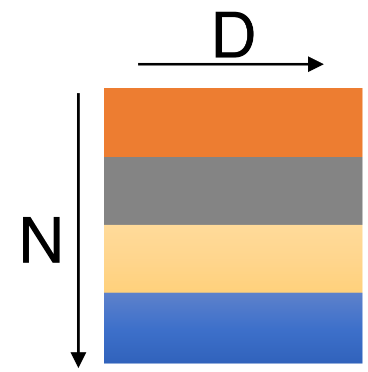
Directly Process Point Cloud Data

End-to-end learning for **unstructured, unordered** point data



Permutation invariance

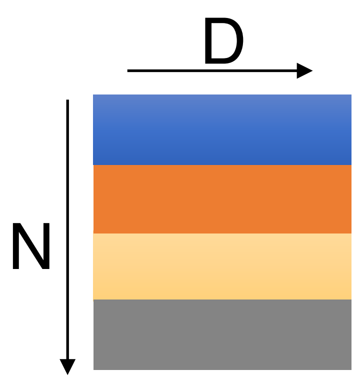
Point cloud: N **orderless** points, each represented by a D dim coordinate



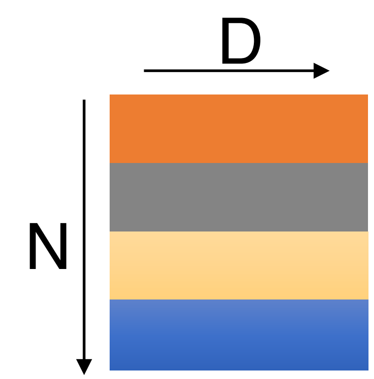
2D array representation

Permutation invariance

Point cloud: N **orderless** points, each represented by a D dim coordinate



represents the same **set** as



2D array representation

Symmetric Function

$$f(x_1, x_2, \cdots, x_n) = f(x_{i_1}, x_{i2}, \cdots, x_{i_n})$$

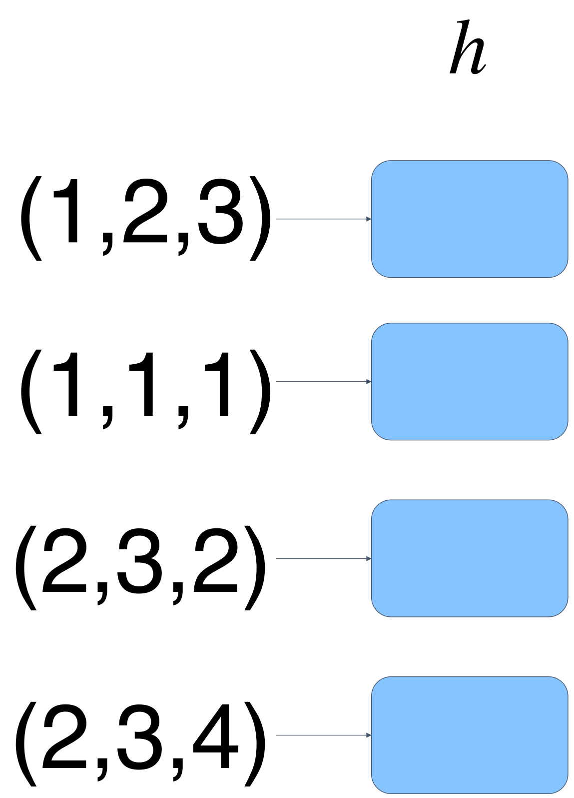
sum: $f(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{N} x_i$

max: $f(x_1, x_2, \cdots, x_n) = \max_{i=1}^{N} x_i$

Construct a Symmetric Function

Observe:

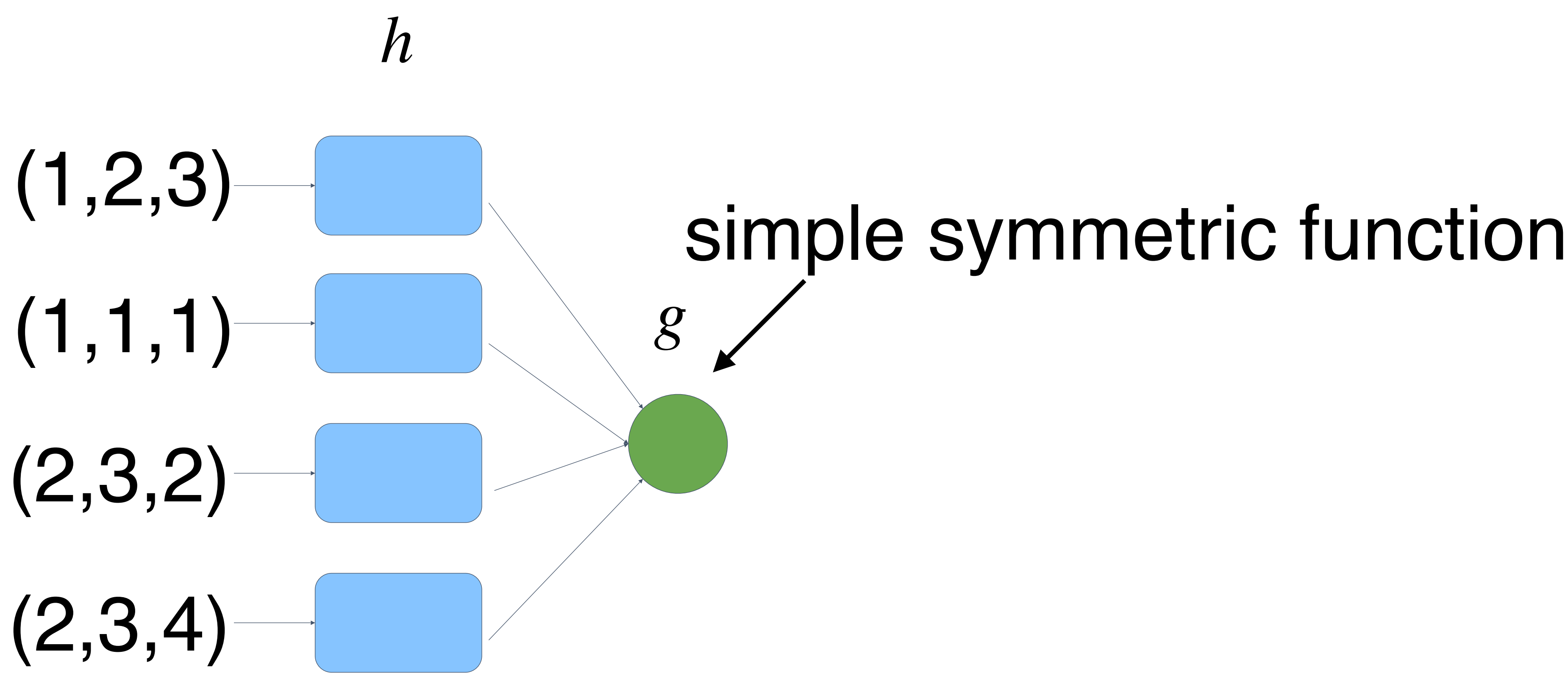
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Construct a Symmetric Function

Observe:

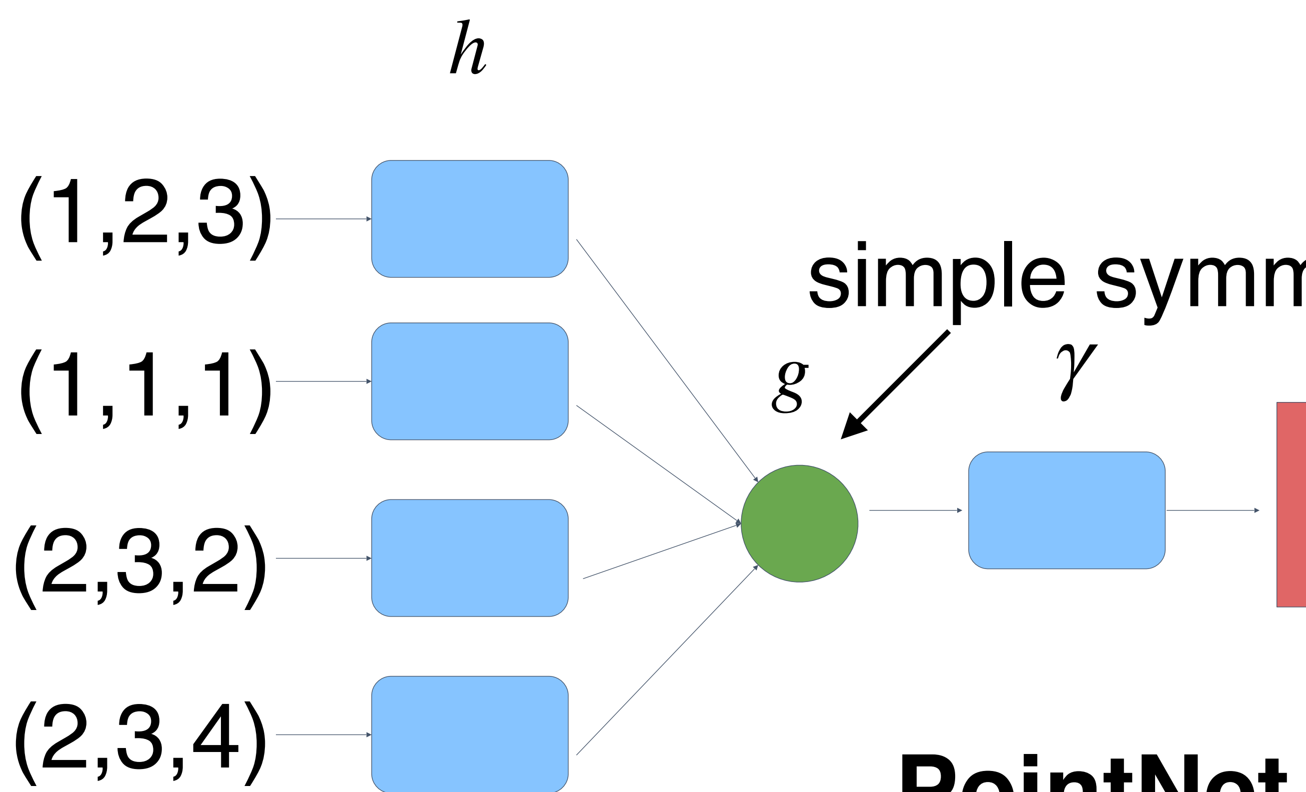
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Construct a Symmetric Function

Observe:

$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



PointNet (vanilla)

Advanced Topic

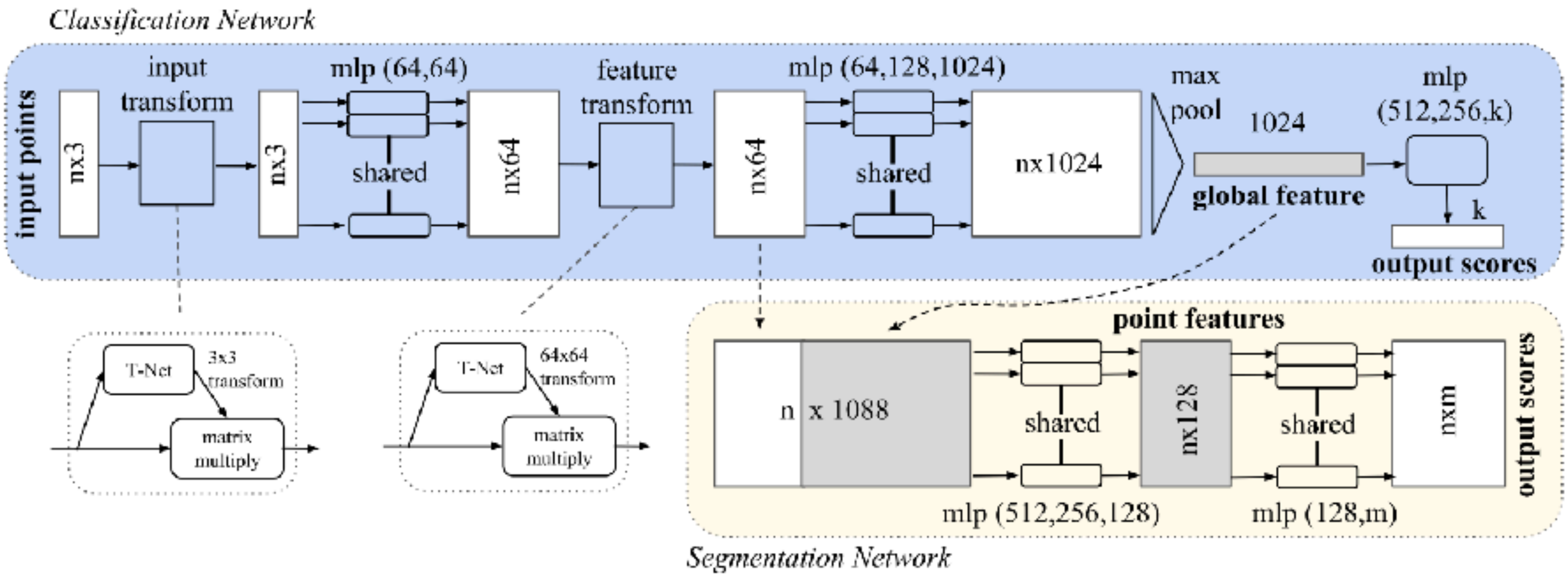
- Is PointNet a universal function approximator of any symmetric function?
- In other words, can we use PointNet to approximate any symmetric function as close as we want?

Theorem 1. *Suppose $f : \mathcal{X} \rightarrow \mathbb{R}$ is a continuous set function w.r.t Hausdorff distance $d_H(\cdot, \cdot)$. $\forall \epsilon > 0$, $\exists$ a continuous function h and a symmetric function $g(x_1, \ldots, x_n) = \gamma \circ MAX$, such that for any $S \in \mathcal{X}$,*

$$\left| f(S) - \gamma \left(\underset{x_i \in S}{MAX} \{h(x_i)\} \right) \right| < \epsilon$$

where $x_1, \ldots, x_n$ is the full list of elements in S ordered arbitrarily, γ is a continuous function, and MAX is a vector max operator that takes n vectors as input and returns a new vector of the element-wise maximum.

Beyond classification



Concatenate the global feature with each point feature
Do a point-wise classification = semantic segmentation

Robustness

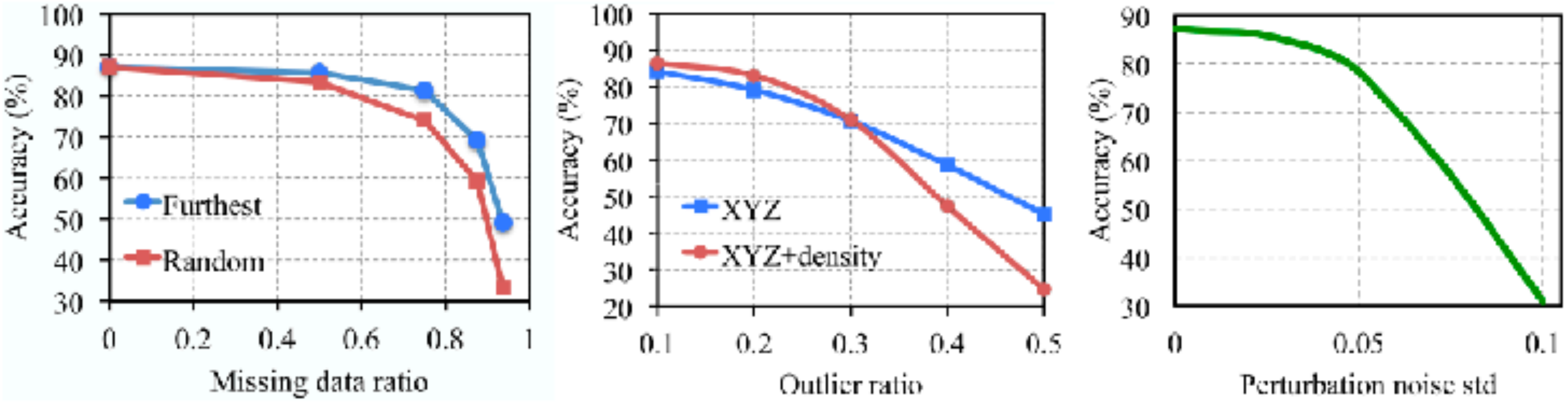
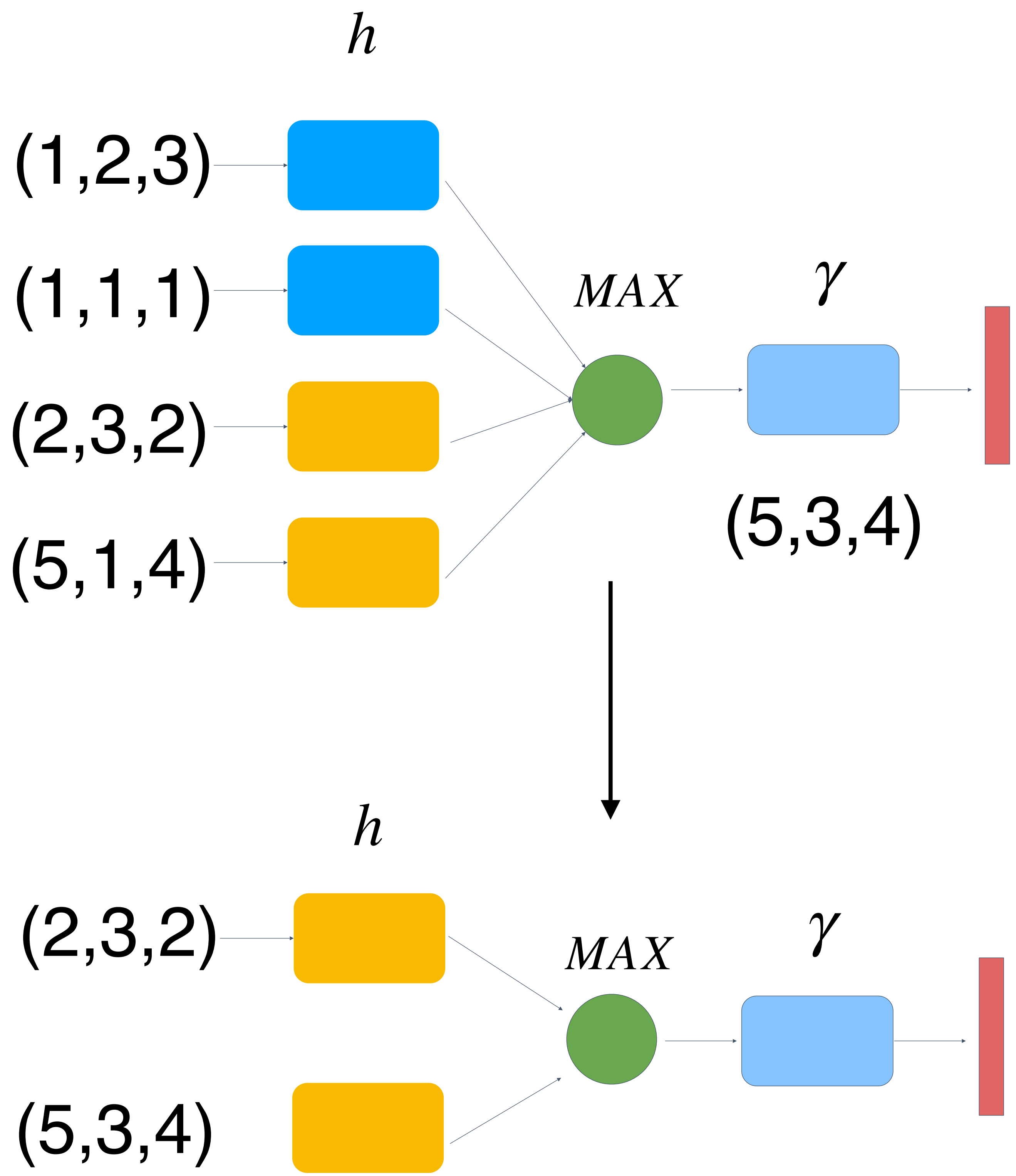
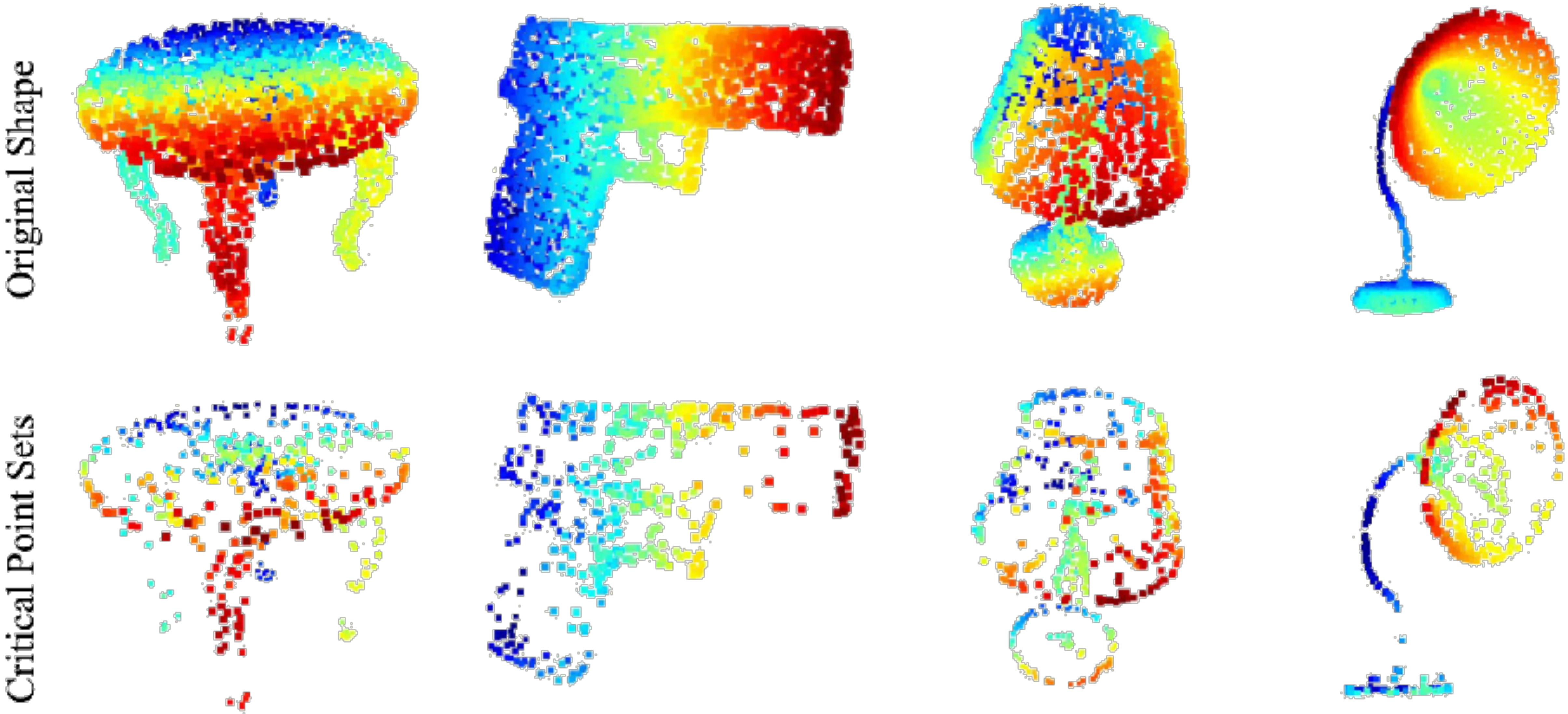


Figure 6. **PointNet robustness test.** The metric is overall classification accuracy on ModelNet40 test set. Left: Delete points. Furthest means the original 1024 points are sampled with furthest sampling. Middle: Insertion. Outliers uniformly scattered in the unit sphere. Right: Perturbation. Add Gaussian noise to each point independently.

Critical points

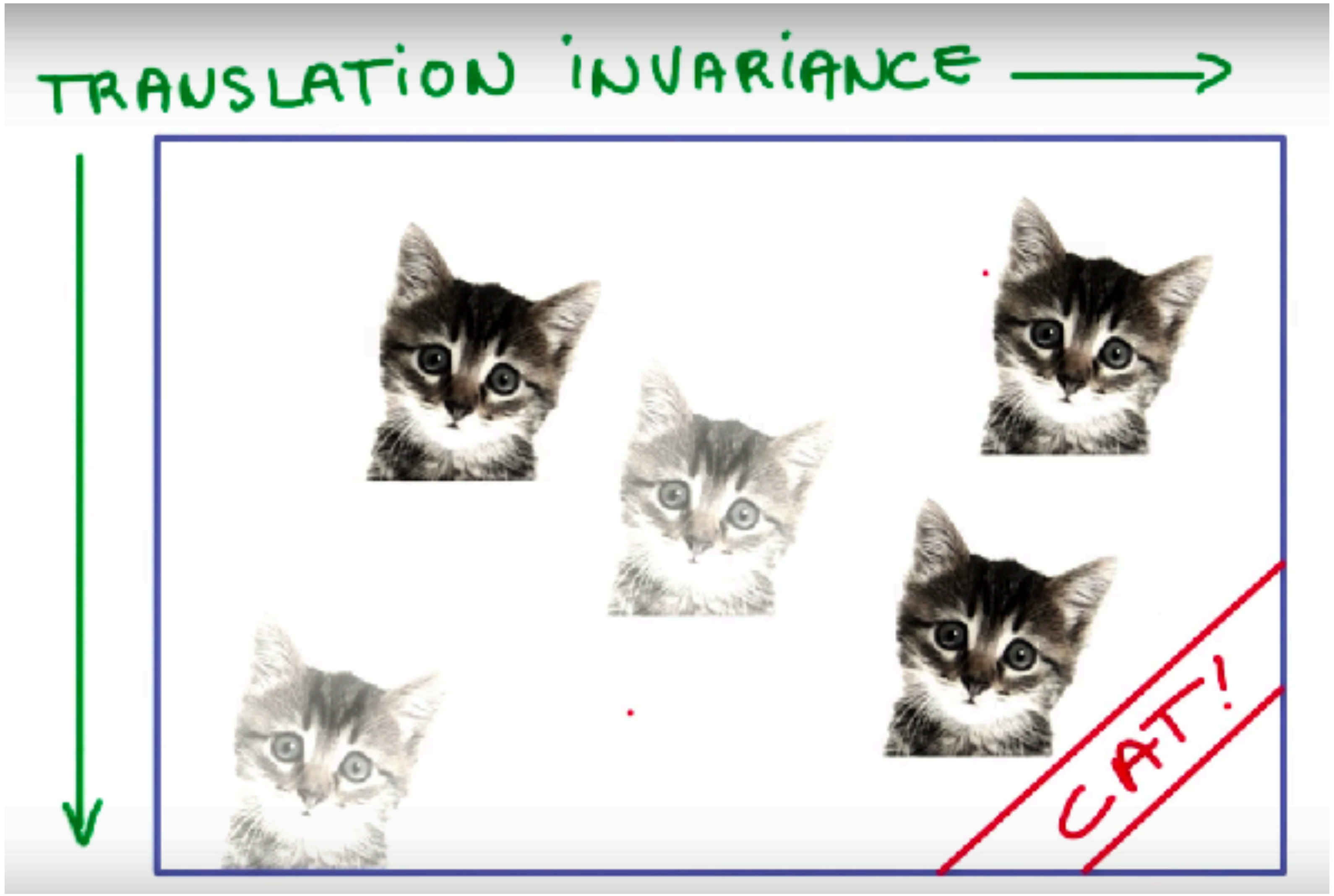


Visualize What is Learned by Reconstruction



Salient points are discovered!

Translation invariant?

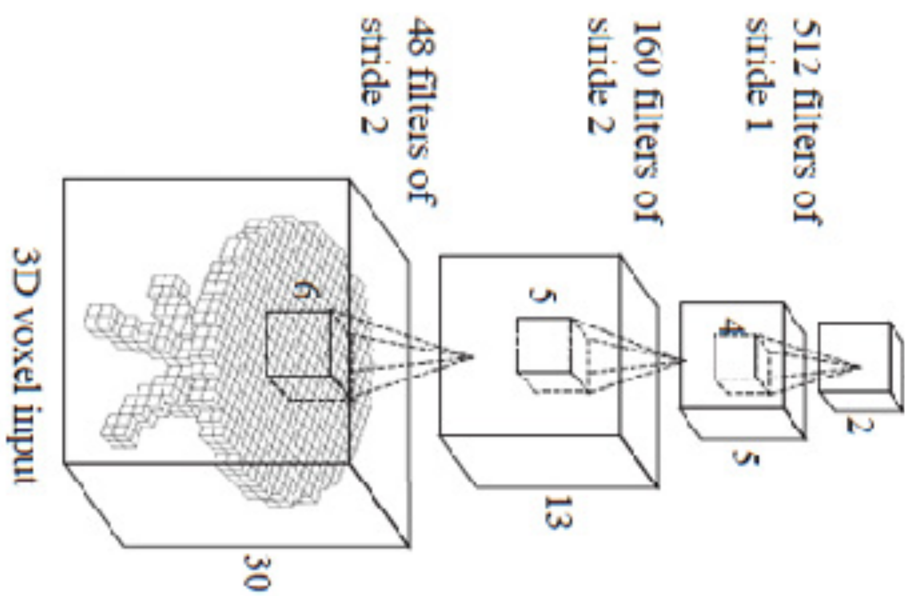


- Global feature depends on absolute coordinate. Hard to generalize to unseen scene configurations!

https://towardsdatascience.com/what-is-wrong-with-convolutional-neural-networks-75c2ba8fbd6f

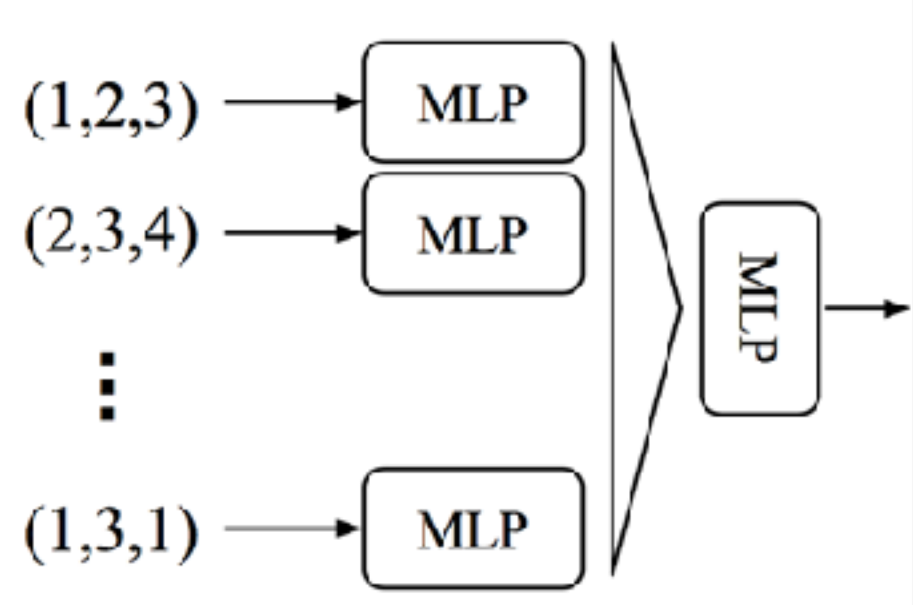
Limitations of PointNet

<u>Hierarchical</u> feature learning
<u>Multiple levels</u> of abstraction



3D CNN (Wu et al.)

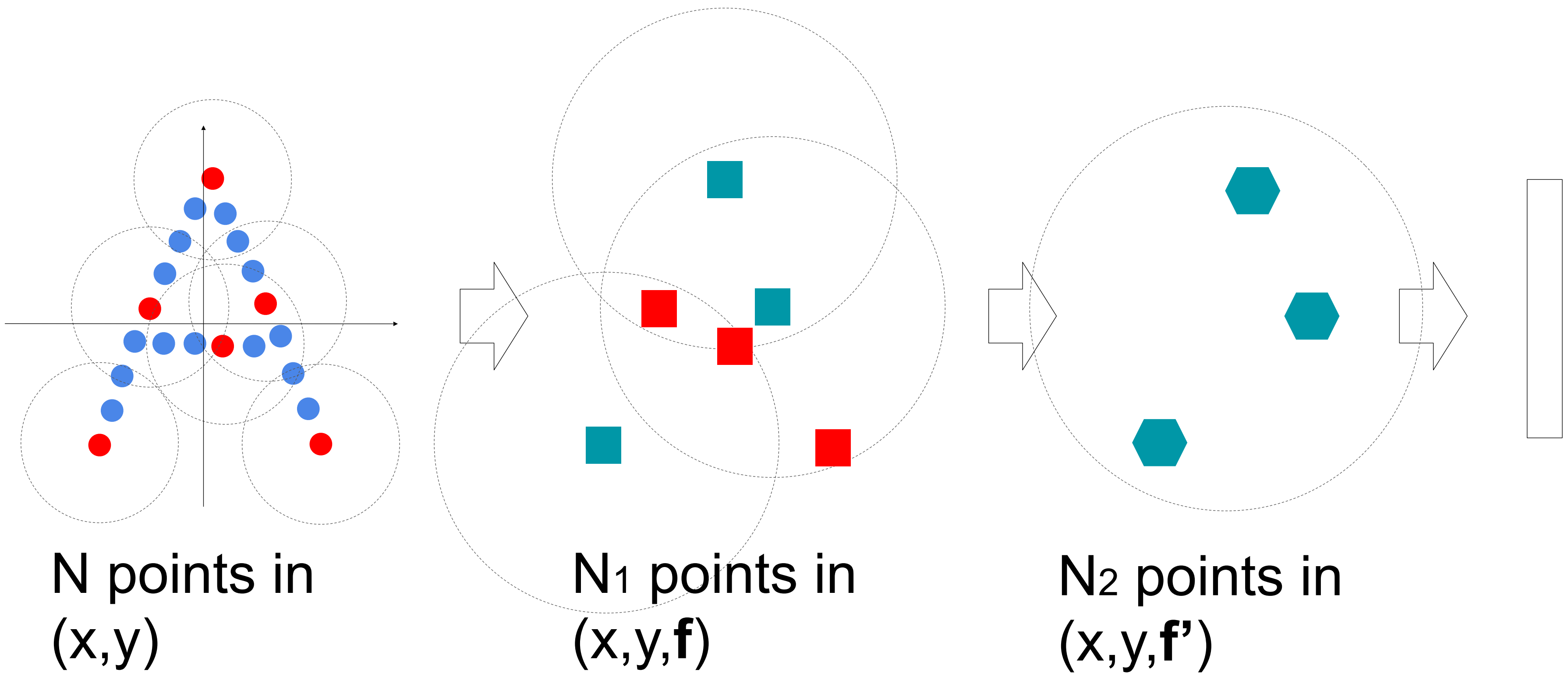
<u>Global</u> feature learning
Either <u>one</u> point or <u>all</u> points



PointNet (vanilla) (Qi et al.)

- No local context for each point!

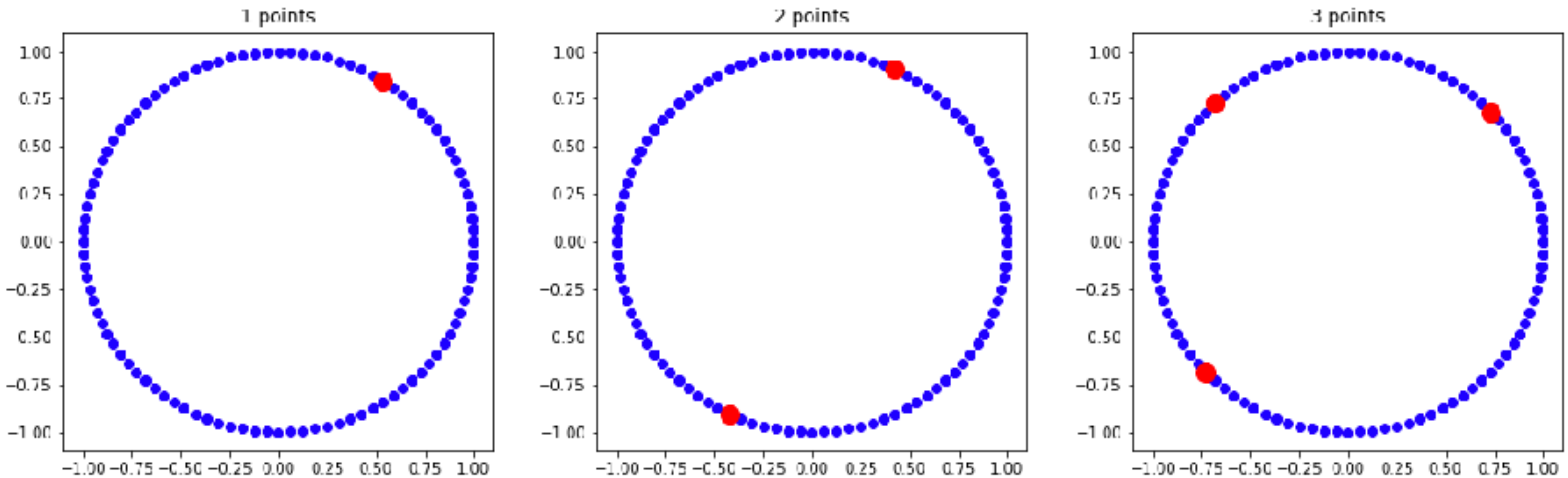
PointNet++: Multi-Scale PointNet

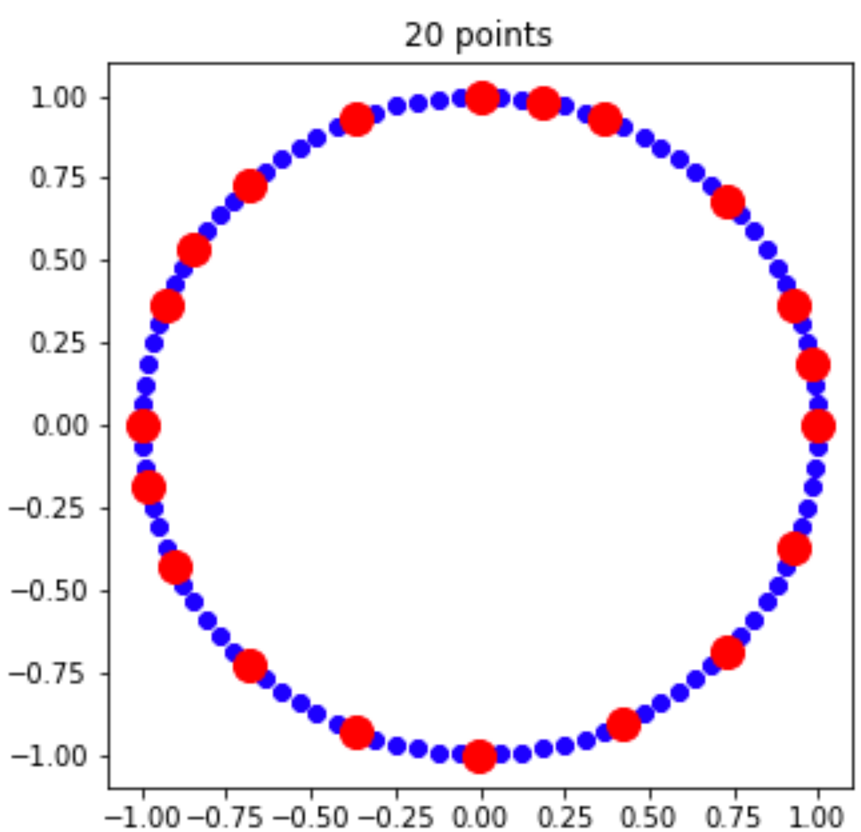


Repeat

- Sample anchor points
- Find neighborhood of anchor points
- Apply PointNet in each neighborhood to mimic convolution

Farthest point sampling

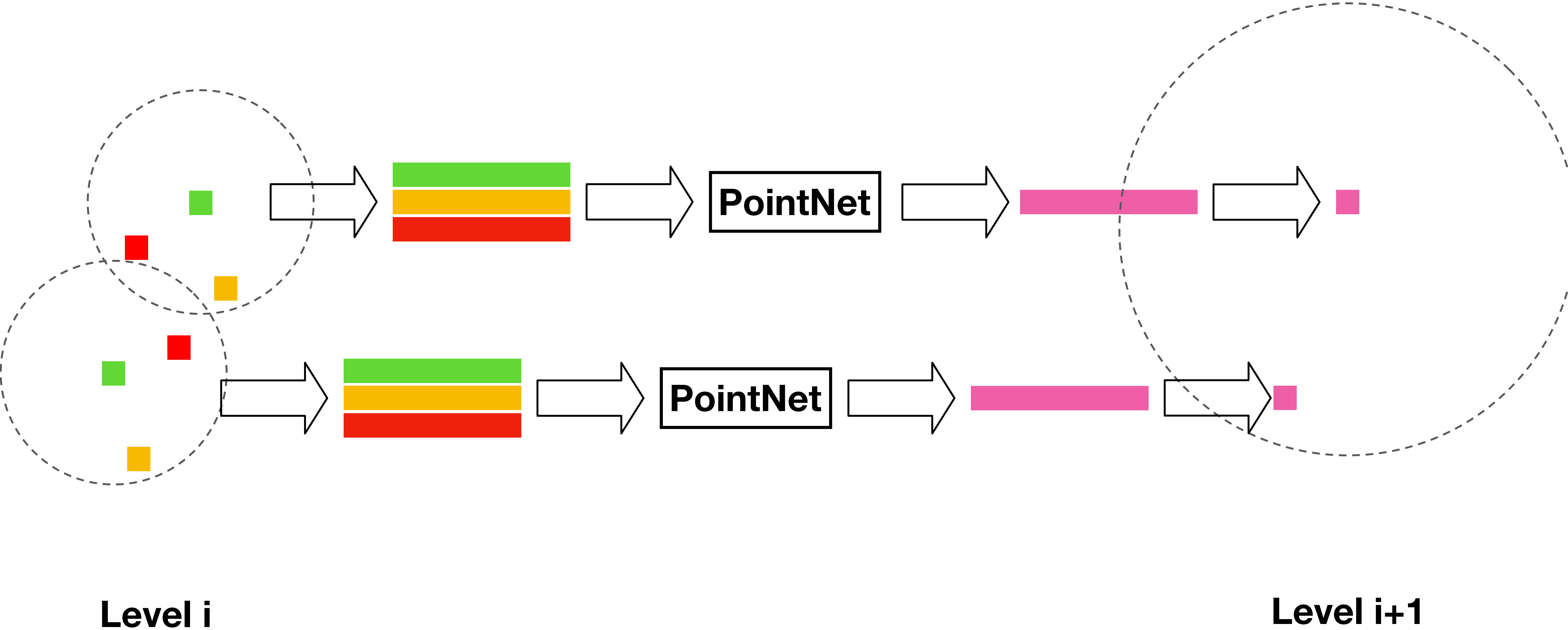




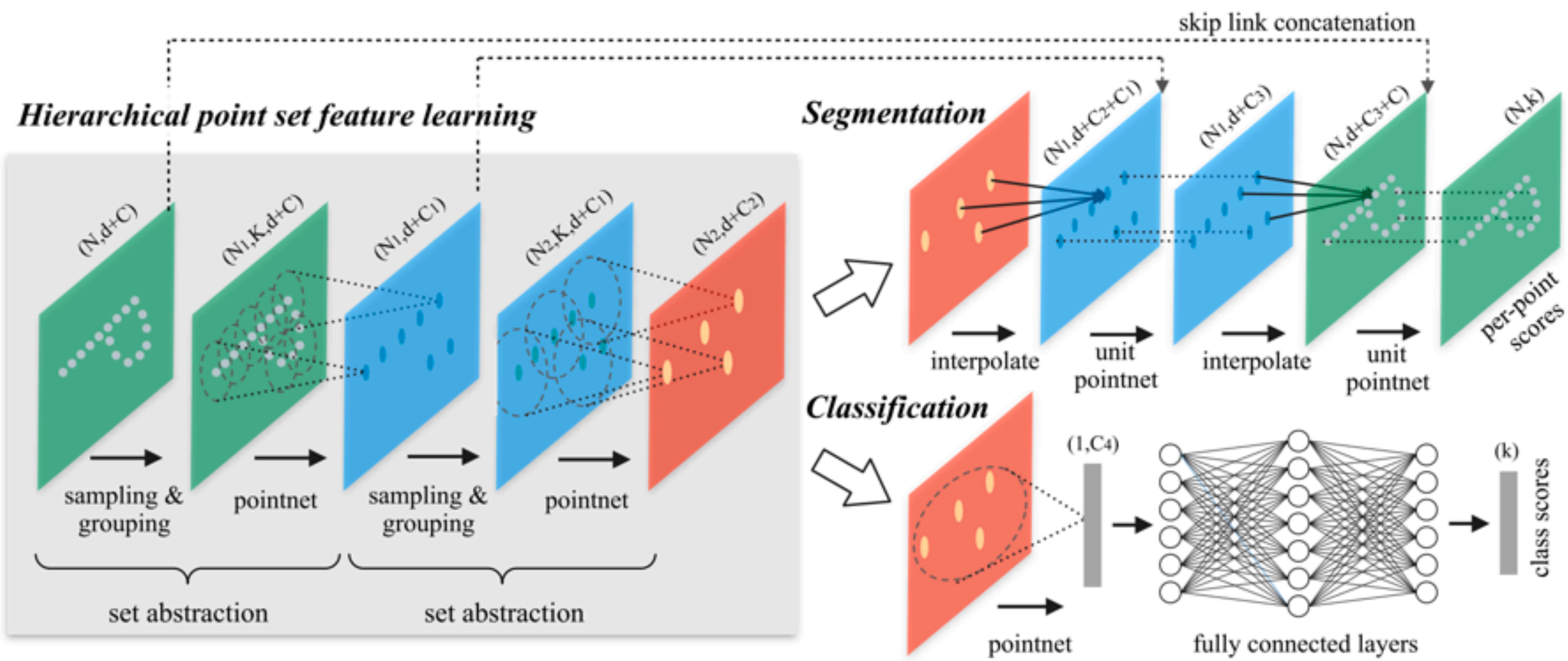
- choose one point at random as a starting point
- add the farthest point from the remaining points to the solution set
- do this until we have reached k points in the solution set

https://flothesof.github.io/farthest-neighbors.html

Ball query



Pipeline



skip link concatenation
Hierarchical point set feature learning
(N,d+C)
(N1,K,d+C)
(N1,d+C1)
(N2,K,d+C1)
(N2,d+C2)
sampling & grouping
pointnet
sampling & grouping
pointnet
set abstraction
set abstraction
Segmentation
(N1,d+C2+C1)
(N1,d+C3)
(N,d+C3+C)
(N,k)
interpolate
unit pointnet
interpolate
unit pointnet
per-point scores
Classification
(1,C4)
(k)
pointnet
fully connected layers
class scores

SPARSE-CONV

Graham, Benjamin, Martin Engelcke, and Laurens van der Maaten. "3d semantic segmentation with submanifold sparse convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

UC San Diego

Strong performance

3D Semantic label benchmark

This table lists the benchmark results for the 3D semantic label scenario.

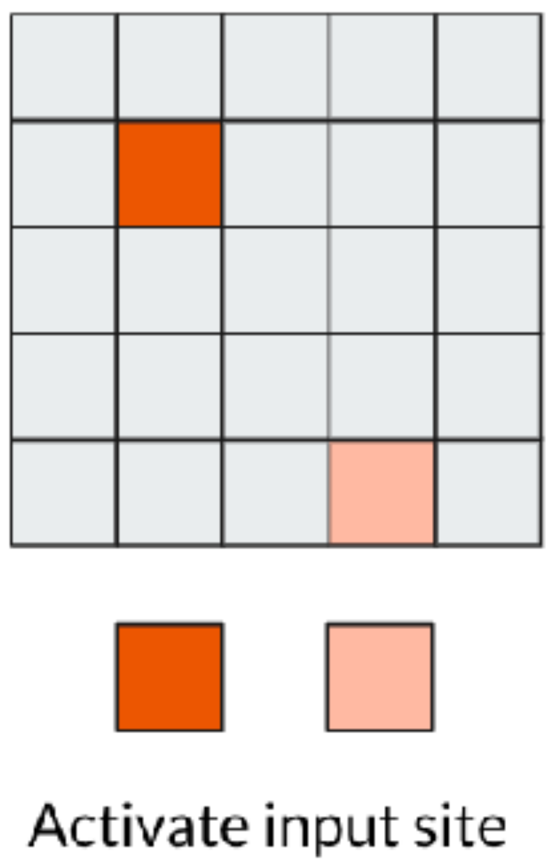
Method	Info	avg iou	bathtub	bed	bookshelf	cabinet	chair	counter	curtain	desk	door	floor	otherfurniture	picture	refrigerator	sho cur
OccuSeg+Semantic		**0.764** 1	0.758 12	0.796 3	0.839 2	**0.746** 1	**0.907** 1	**0.562** 1	0.850 2	**0.680** 1	**0.672** 1	**0.978** 1	**0.610** 1	**0.335** 1	**0.777** 1	0.8
MinkowskiNet	P	0.736 2	0.859 2	0.818 2	0.832 3	0.709 3	0.840 3	0.521 3	**0.853** 1	0.660 2	0.643 2	0.951 4	0.544 3	0.286 8	0.731 2	**0.8**
C. Choy, J. Gwak, S. Savarese: D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. CVPR 2019																
SparseConvNet		0.725 3	0.647 21	**0.821** 1	**0.846** 1	0.721 2	0.869 2	0.533 2	0.754 8	0.603 5	0.614 3	0.955 2	0.572 2	0.325 2	0.710 3	0.8
CU-Hybrid Net		0.693 4	0.596 24	0.789 4	0.803 5	0.677 4	0.800 10	0.469 8	0.846 3	0.554 12	0.591 6	0.948 10	0.500 4	0.316 3	0.609 8	0.8
KP-FCNN		0.684 5	0.847 4	0.758 10	0.784 7	0.647 6	0.814 7	0.473 6	0.772 6	0.605 4	0.594 4	0.935 20	0.450 10	0.181 24	0.587 8	0.8
H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette, L. Guibas.: KPConv: Flexible and Deformable Convolution for Point Clouds. ICCV 2019																
PointASNL		0.666 6	0.703 17	0.781 5	0.751 12	0.655 5	0.830 4	0.471 7	0.769 7	0.474 21	0.537 8	0.951 4	0.475 6	0.279 9	0.635 5	0.6
Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, Shuguang Cui: PointASNL: Robust Point Clouds Processing using Nonlocal Neural Networks with Adaptive Sampling. CVPR 2020																
PointConv	P	0.666 6	0.781 9	0.759 9	0.699 14	0.644 7	0.822 5	0.475 5	0.779 5	0.564 11	0.504 15	0.953 3	0.428 16	0.203 19	0.586 9	0.7
Wenxuan Wu, Zhongang Qi, Li Fuxin: PointConv: Deep Convolutional Networks on 3D Point Clouds. CVPR 2019																
DualMeshNet		0.658 8	0.778 10	0.702 14	0.806 4	0.619 9	0.813 8	0.468 9	0.693 14	0.494 18	0.524 11	0.941 15	0.449 11	0.298 4	0.510 15	0.8
Jonas Schult, Francis Engelmann, Theodora Kontogianni, Bastian Leibe: DualMeshNet: Joint Geodesic and Euclidean Convolutions for 3D Semantic Segmentation. CVPR 2020																
MVPNet	P	0.641 9	0.831 5	0.715 13	0.671 17	0.590 13	0.781 13	0.394 17	0.679 17	0.642 3	0.553 7	0.937 19	0.462 8	0.256 10	0.649 4	0.4
Maximilian Jaritz, Jiayuan Gu, Hao Su: Multi-view PointNet for 3D Scene Understanding. GMDL Workshop, ICCV 2019																

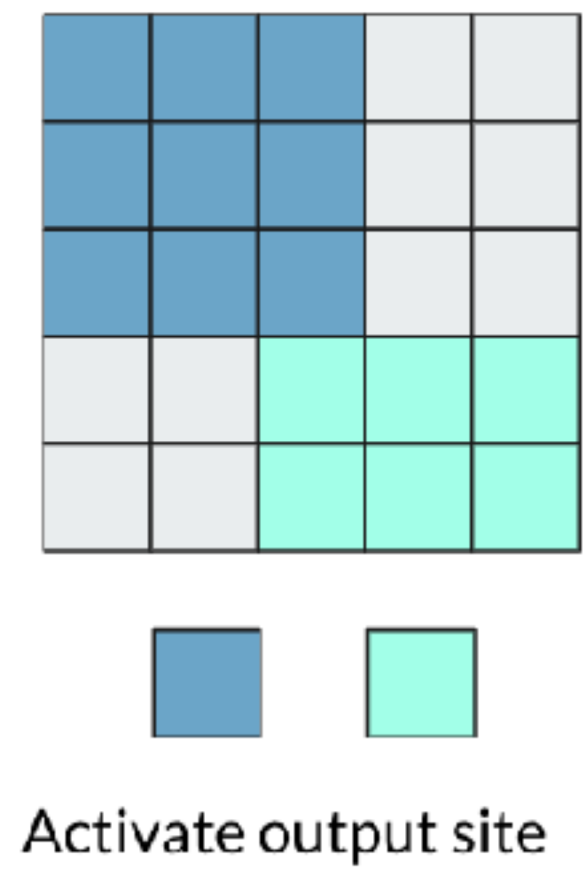
Top 3 entries are based on SparseConv

Terminology

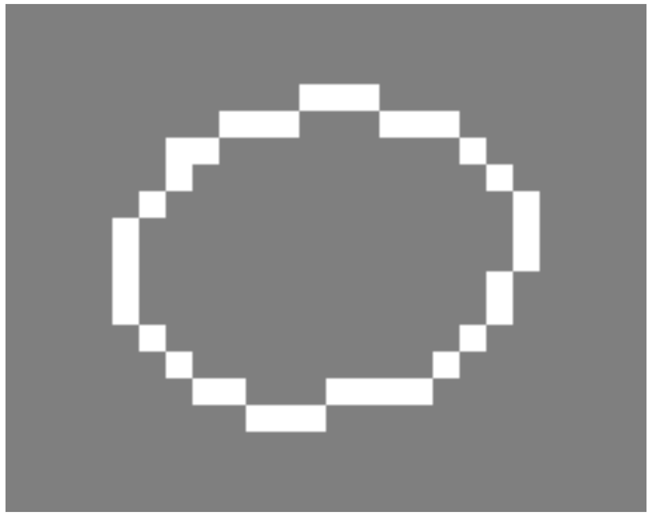
- d-dimension CNN with (d+1)-dimension input
  - e.g. 2D CNN with HxWxC image
- d-dimension site: associated with a feature vector
- active site: site associated with a non-zero feature
- ground state: zero feature vector

Sparse convolution (SC)

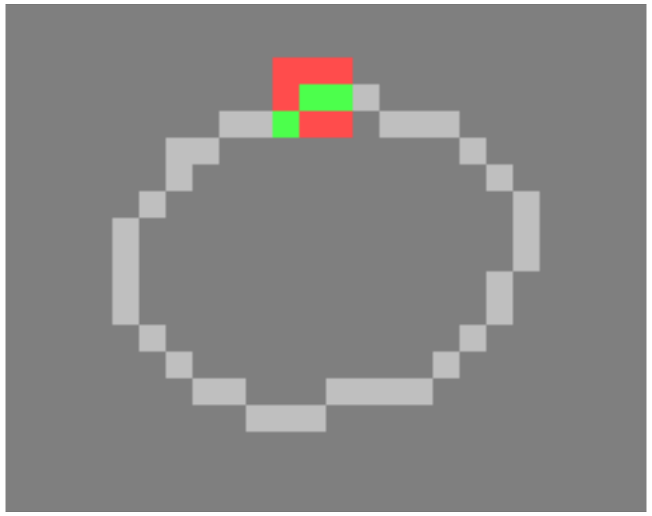




Dilated issues



SparseConv

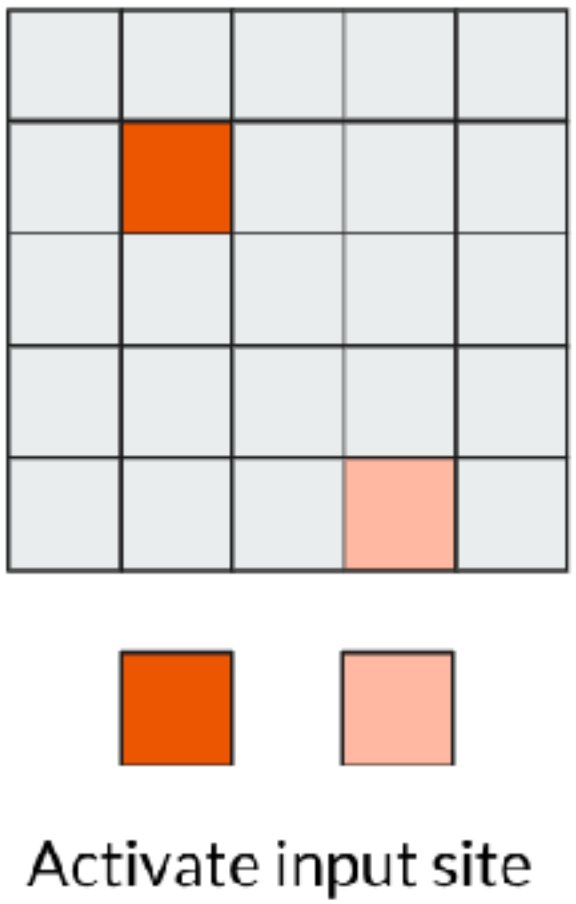


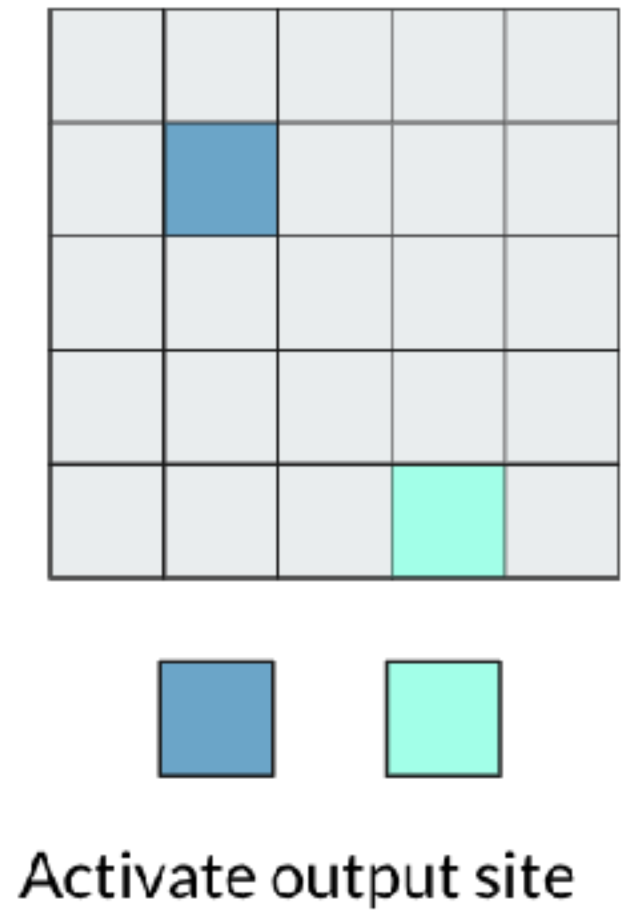
Submanifold SparseConv

Sparsity reduce
Computation increase

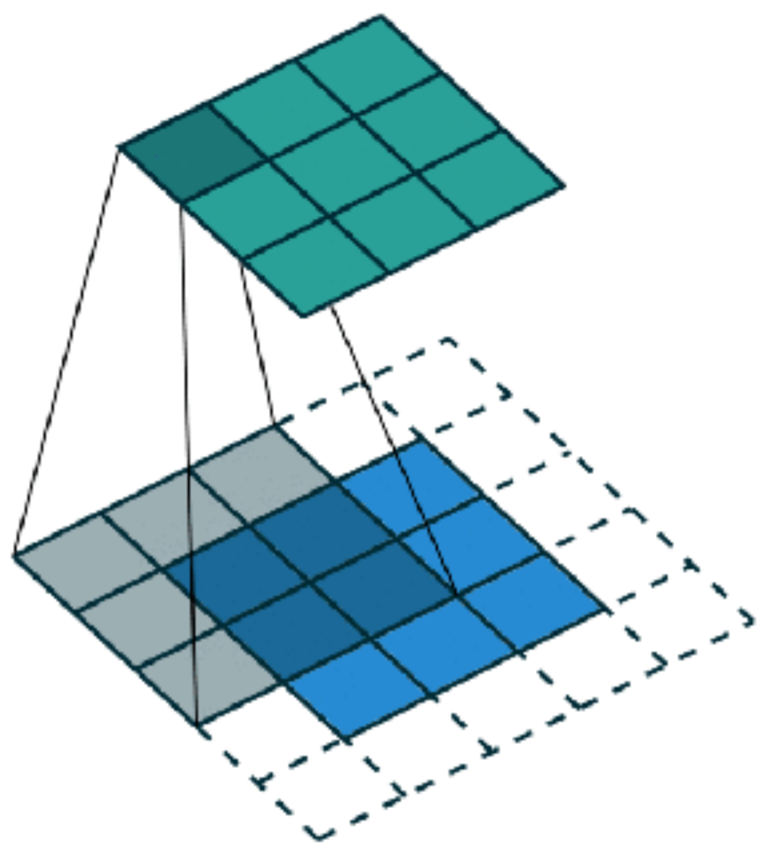
Sparsity keep
Computation keep

Submanifold Sparse convolution (SSC)

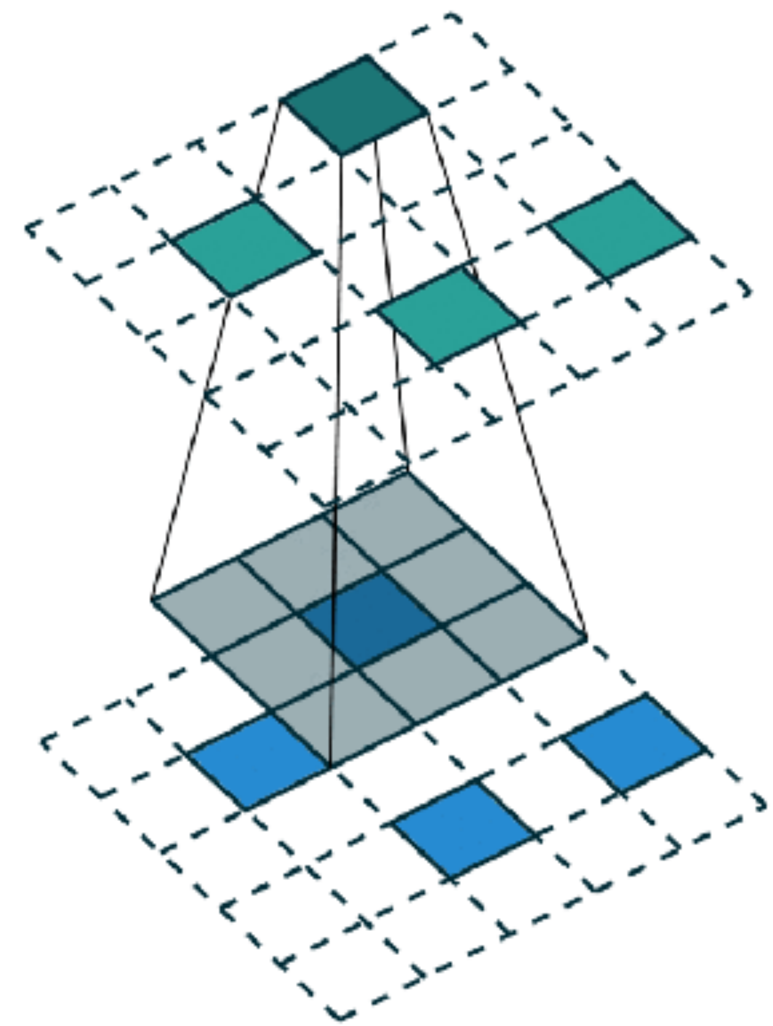




Dense vs. Sparse conv



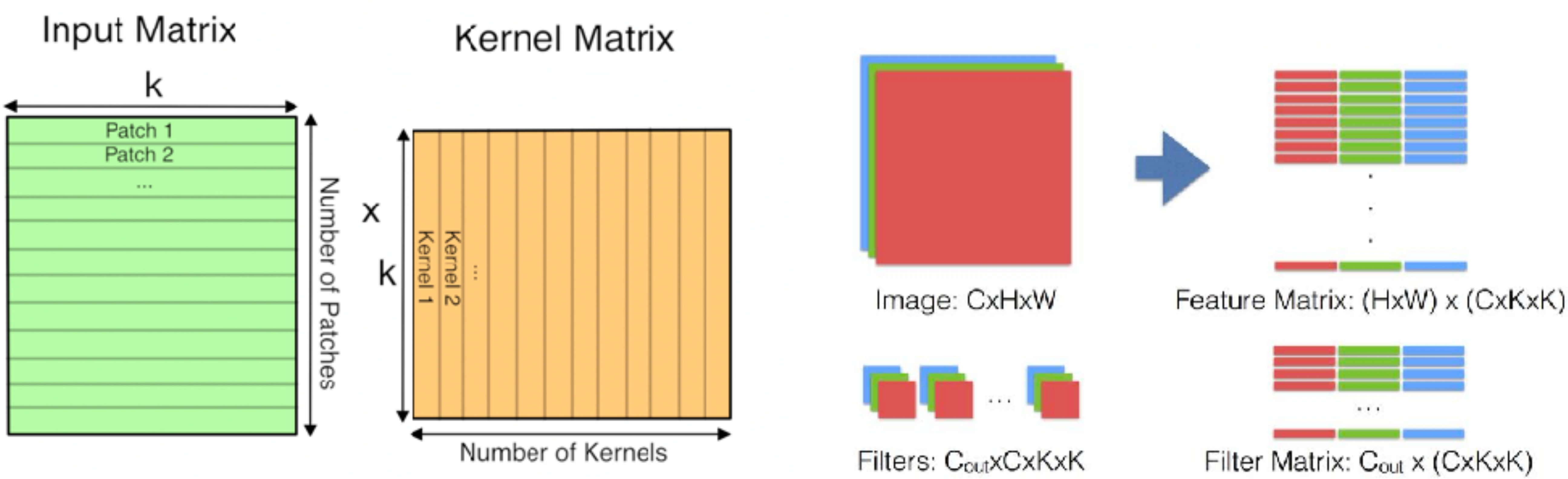
Dense Convolution



Submanifold Sparse Convolution

https://github.com/StanfordVL/MinkowskiEngine

Implementation (dense)

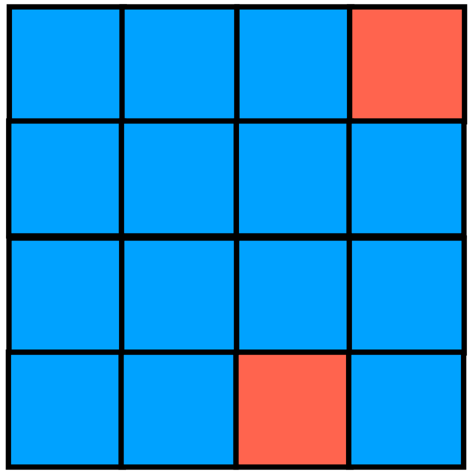


Convert convolution to matrix multiplication

https://petewarden.com/2015/04/20/why-gemm-is-at-the-heart-of-deep-learning/

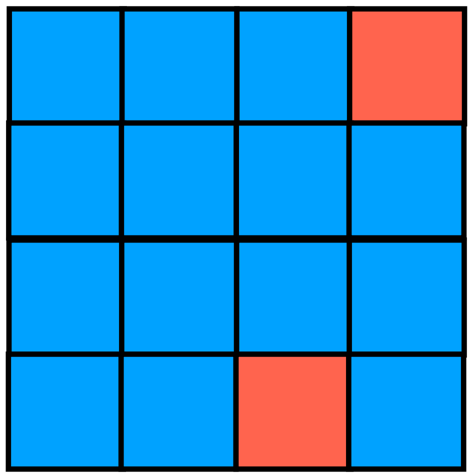
https://github.com/Yangqing/caffe/wiki/Convolution-in-Caffe:-a-memo

Implementation (sparse)



4x4 input feature map

Implementation (sparse)

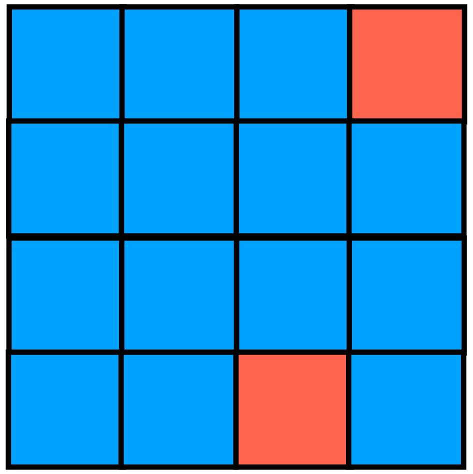


4x4 input feature map

Index	Loc	Feature
0	(0, 3)	
1	(3, 2)	

Input matrix:
build input hash table

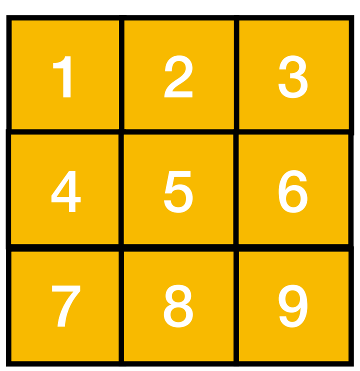
Implementation (sparse)



4x4 input feature map

Index	Loc	Feature
0	(0, 3)	
1	(3, 2)	

Input matrix:
build input hash table



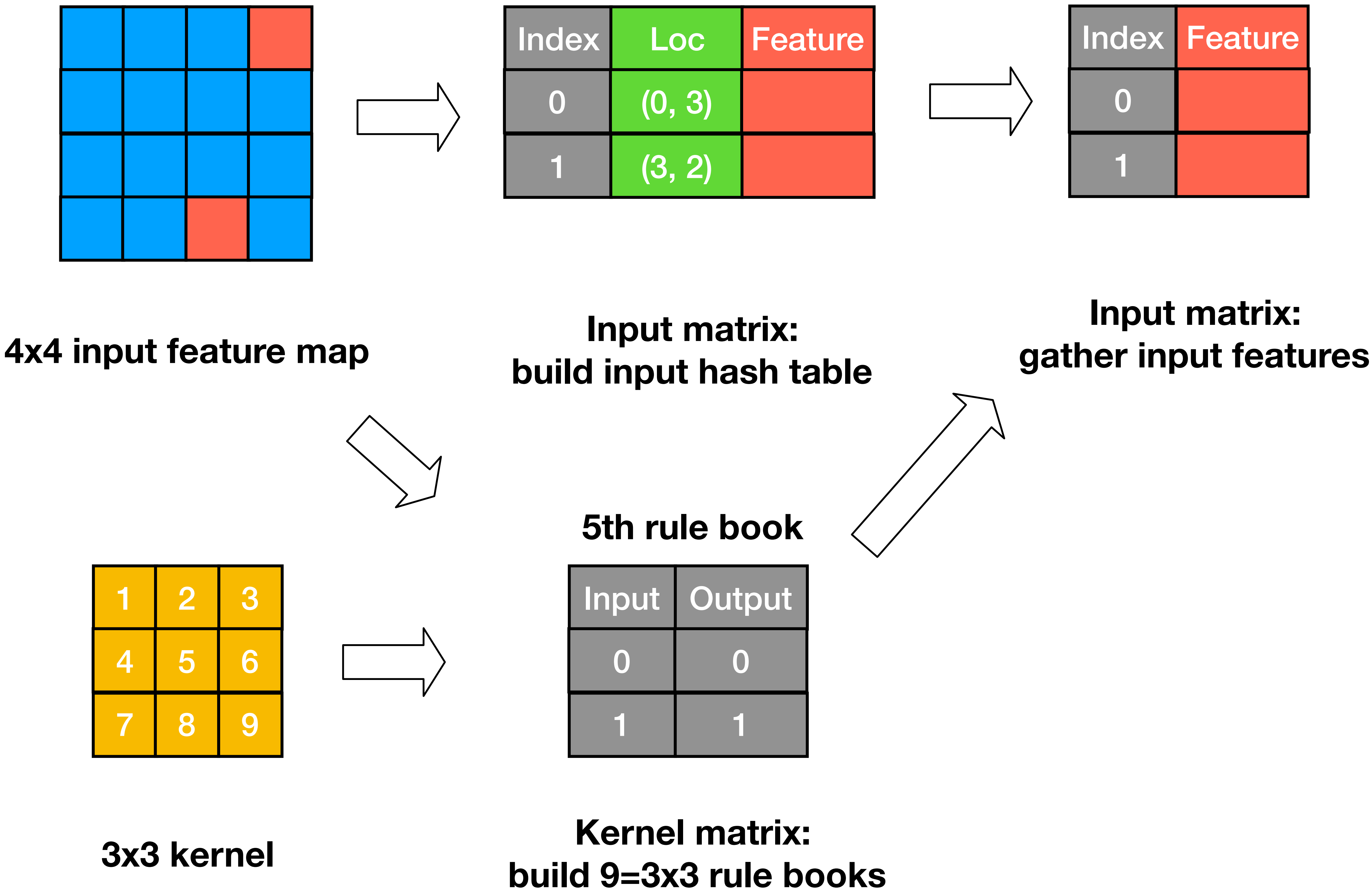
3x3 kernel

5th rule book

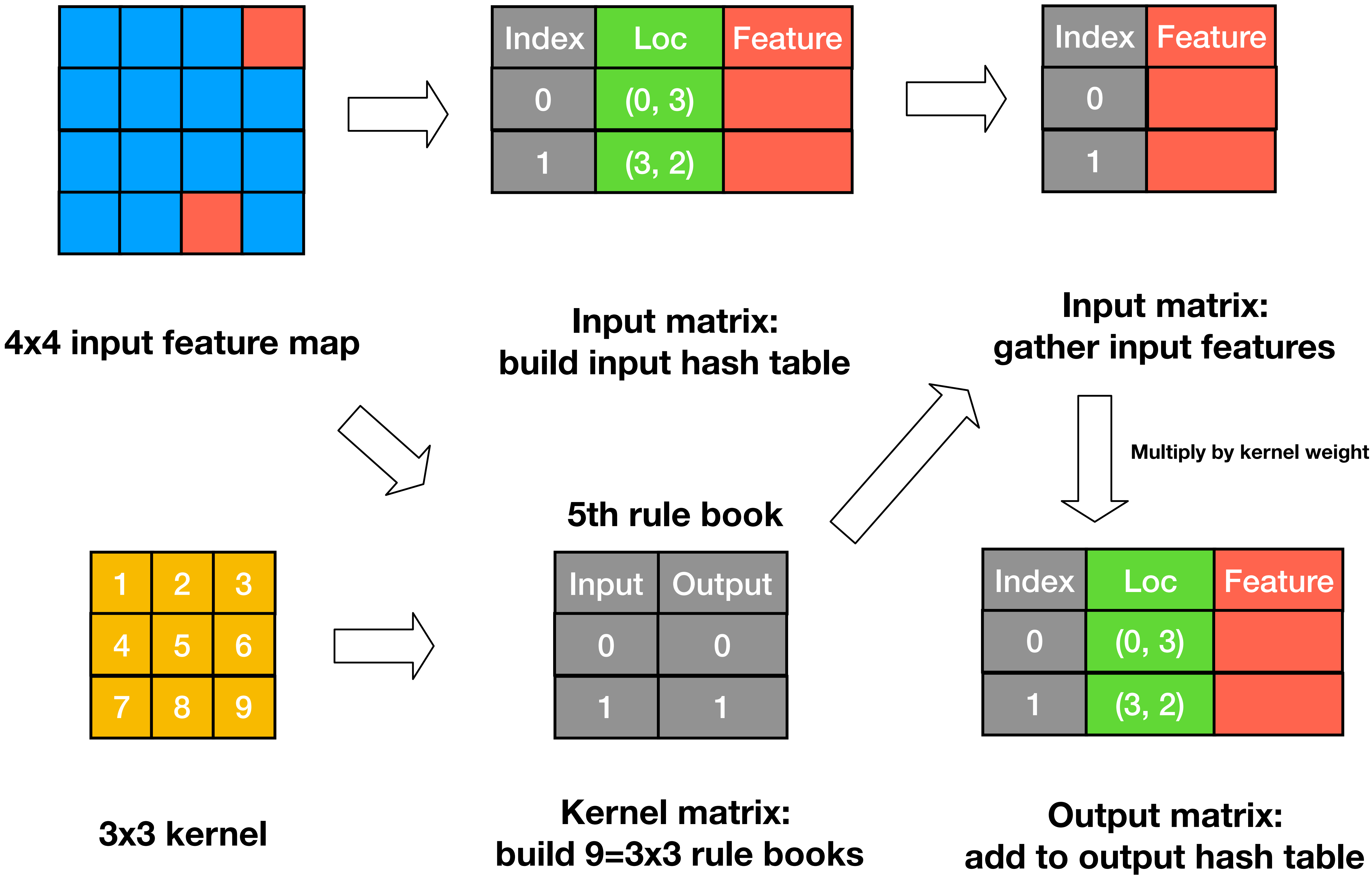
Input	Output
0	0
1	1

Kernel matrix:
build 9=3x3 rule books

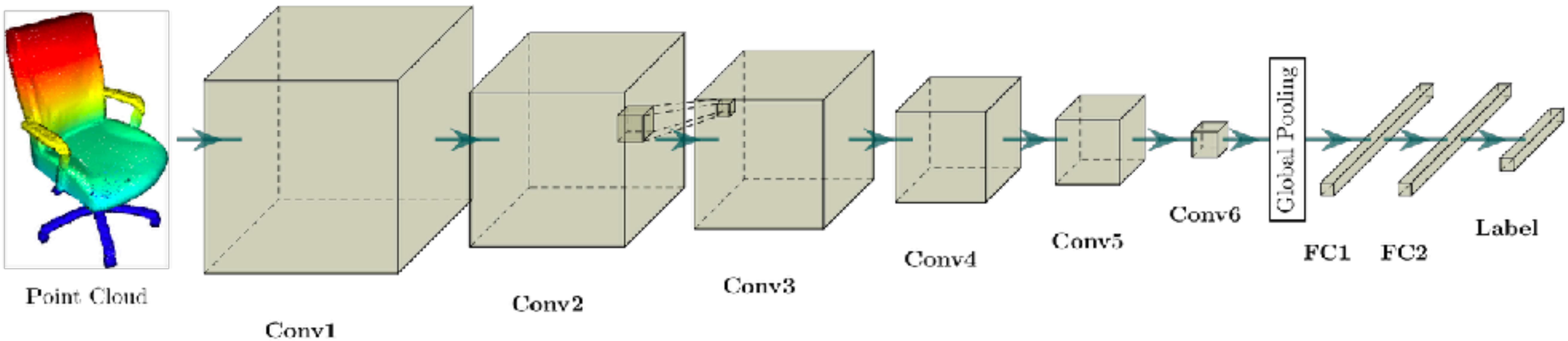
Implementation (sparse)



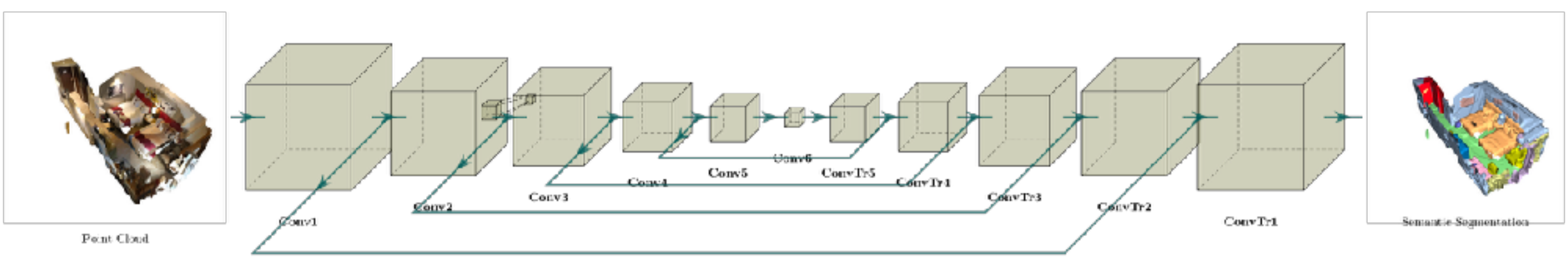
Implementation (sparse)



Application



Classification

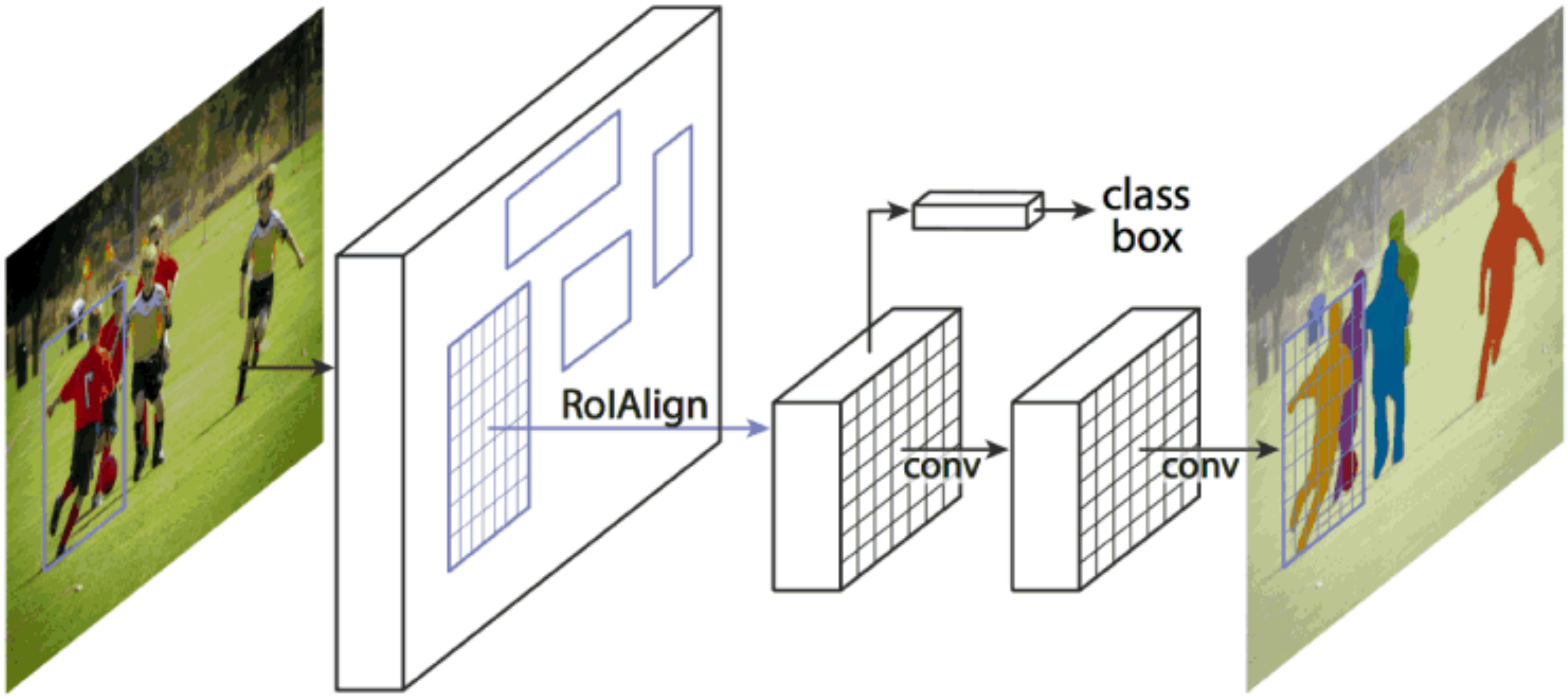


Semantic Segmentation

VOTENET

Qi, Charles R., et al. "Deep hough voting for 3d object detection in point clouds." *Proceedings of the IEEE International Conference on Computer Vision*. 2019.

3D R-CNN?

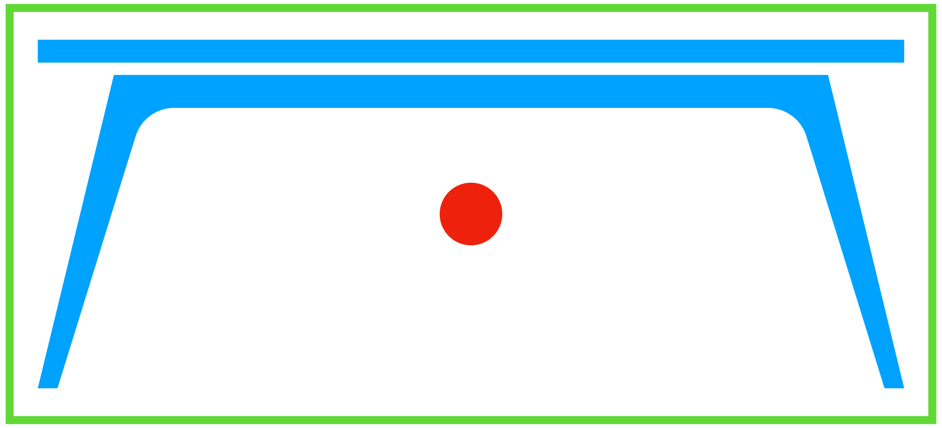


The Mask R-CNN framework for instance segmentation

https://www.techleer.com/articles/528-mask-r-cnn-mask-r-cnn-for-object-detection-and-instance-segmentation-on-keras-and-tensorflow/

Challenges for 3D detection

- For 2D, the center of the bounding box of an object is usually a local maximal of the activation.
- However, for 3D, the point cloud is located at the surface. Thus, the center of the 3D bounding box is not necessary to be located on the surface.
- It is still an open question how to represent 3D instances (bounding box or others)?



Generalized Hough transform

GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES*

D. H. BALLARD

Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A.

(*Received* 10 *October* 1979; *in revised form* 9 *September* 1980; *received for publication* 23 *September* 1980)

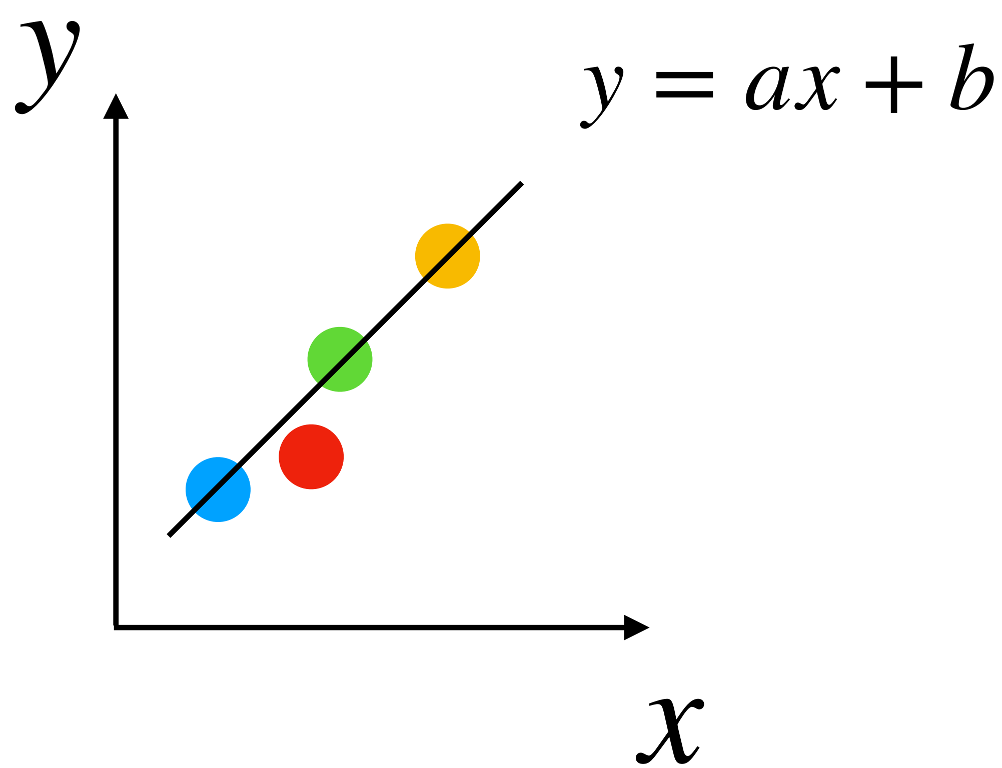
Abstract—The Hough transform is a method for detecting curves by exploiting the duality between points on a curve and parameters of that curve. The initial work showed how to detect both analytic curves[1,2] and non-analytic curves,[3] but these methods were restricted to binary edge images. This work was generalized to the detection of some analytic curves in grey level images, specifically lines,[4] circles[5] and parabolas.[6] The line detection case is the best known of these and has been ingeniously exploited in several applications.[7,8,9]

We show how the boundaries of an *arbitrary* non-analytic shape can be used to construct a mapping between image space and Hough transform space. Such a mapping can be exploited to detect instances of that particular shape in an image. Furthermore, variations in the shape such as rotations, scale changes or figure-ground reversals correspond to straightforward transformations of this mapping. However, the most remarkable property is that such mappings can be composed to build mappings for complex shapes from the mappings of simpler component shapes. This makes the generalized Hough transform a kind of universal transform which can be used to find arbitrarily complex shapes.

Image processing | Hough transform | Shape recognition | Pattern recognition | Parallel algorithms

Ballard, Dana H. "Generalizing the Hough transform to detect arbitrary shapes." *Readings in computer vision*. Morgan Kaufmann, 1987. 714-725.

Example: fit a line



In order to find a line passing the points,
we can check lines passing each points.

Example: fit a line

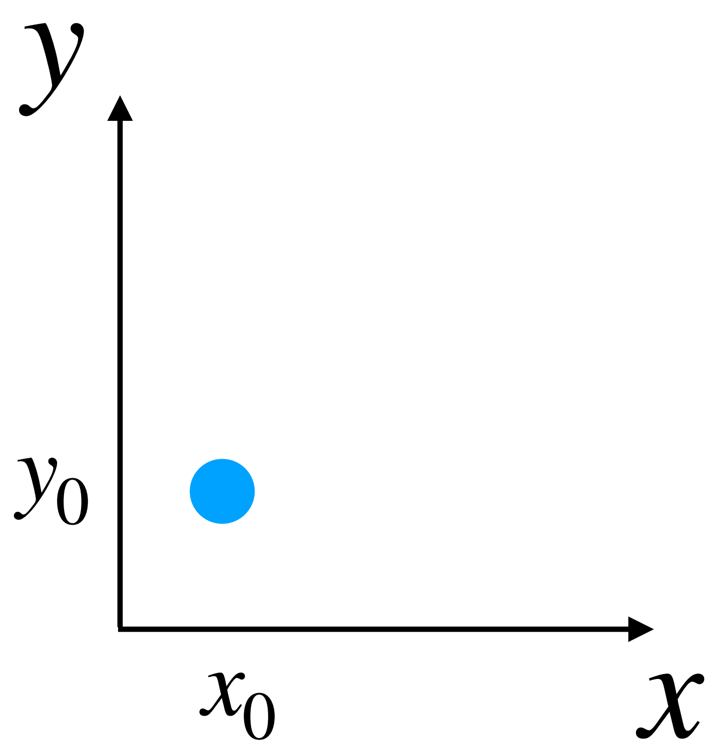
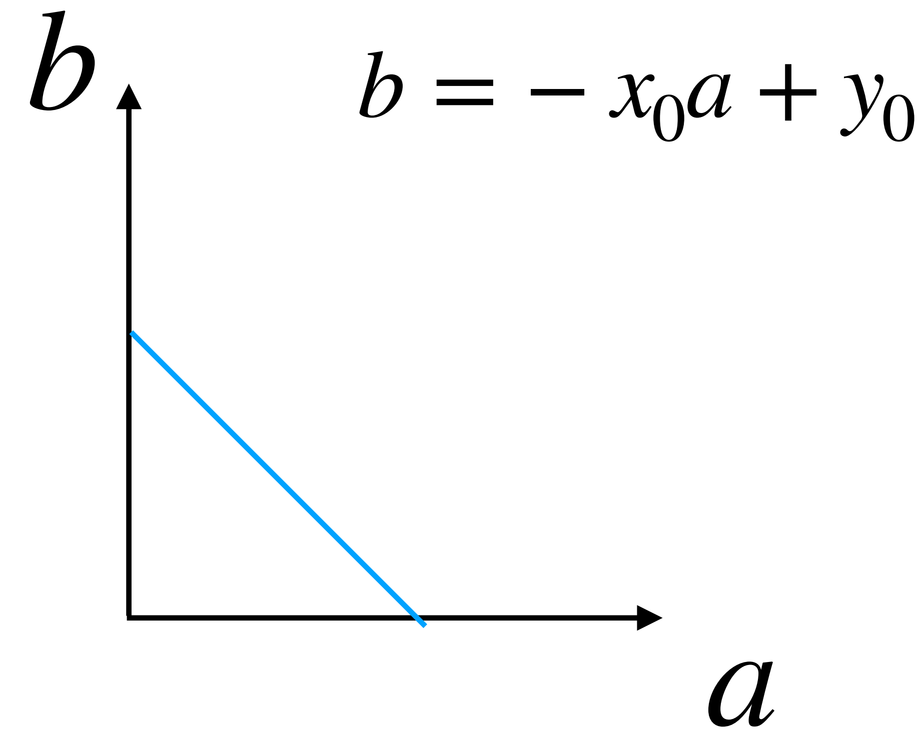
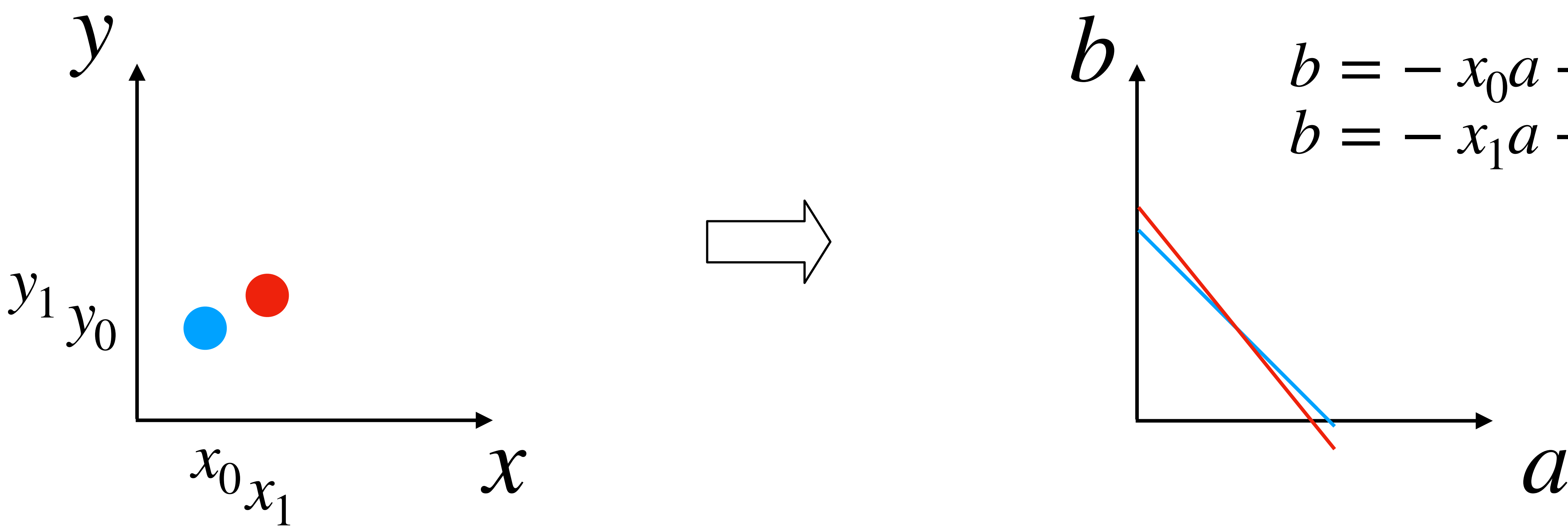


Image space



Parameter (Hough) space

Example: fit a line



$$b = -x_0 a + y_0$$
$$b = -x_1 a + y_1$$

Image space

Parameter (Hough) space

Example: fit a line

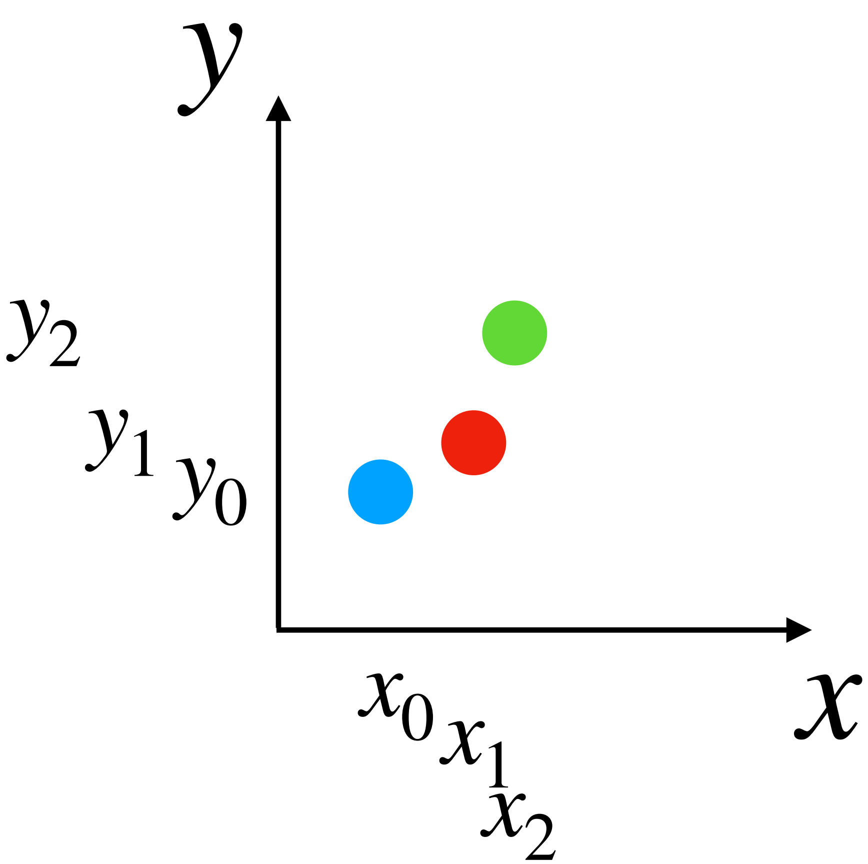
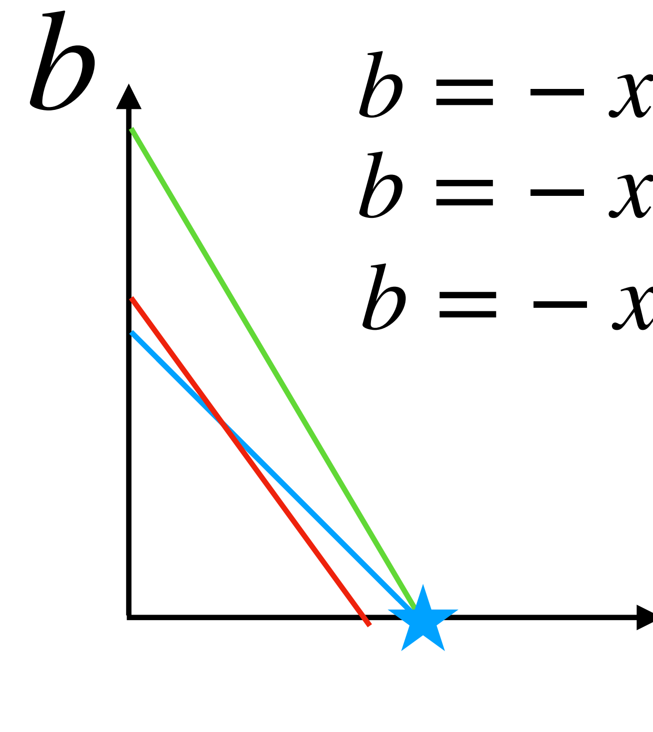


Image space

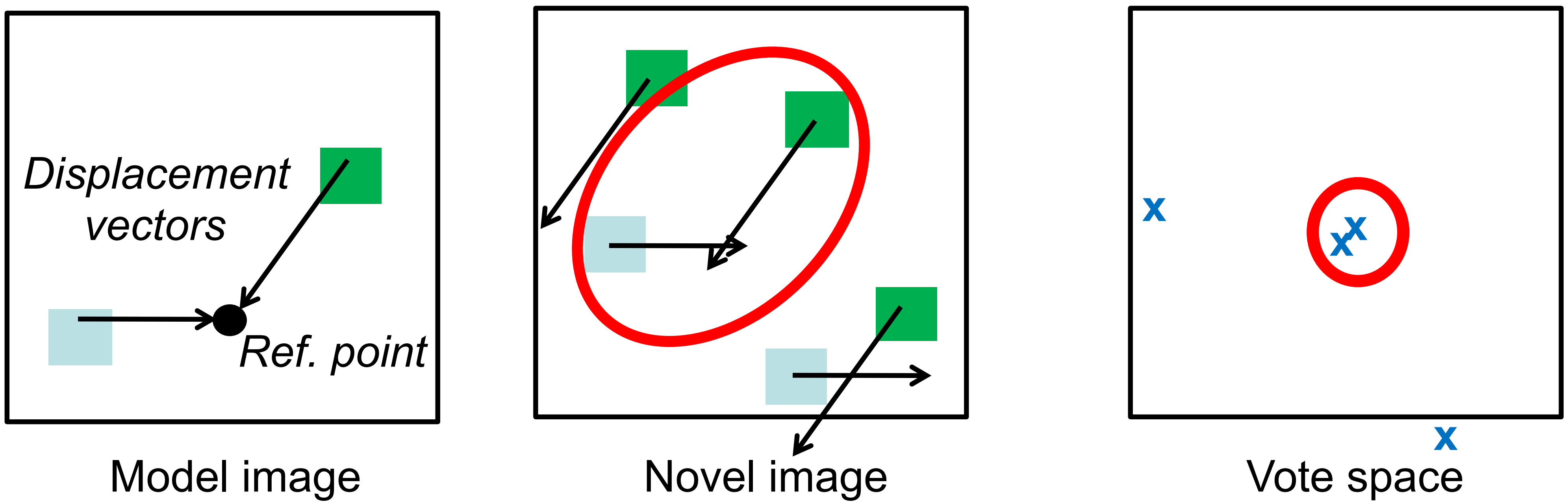


$$b = -x_0 a + y_0$$
$$b = -x_1 a + y_1$$
$$b = -x_2 a + y_2$$

Parameter (Hough) space

coordinate —> line parameter

Generalized hough transformation

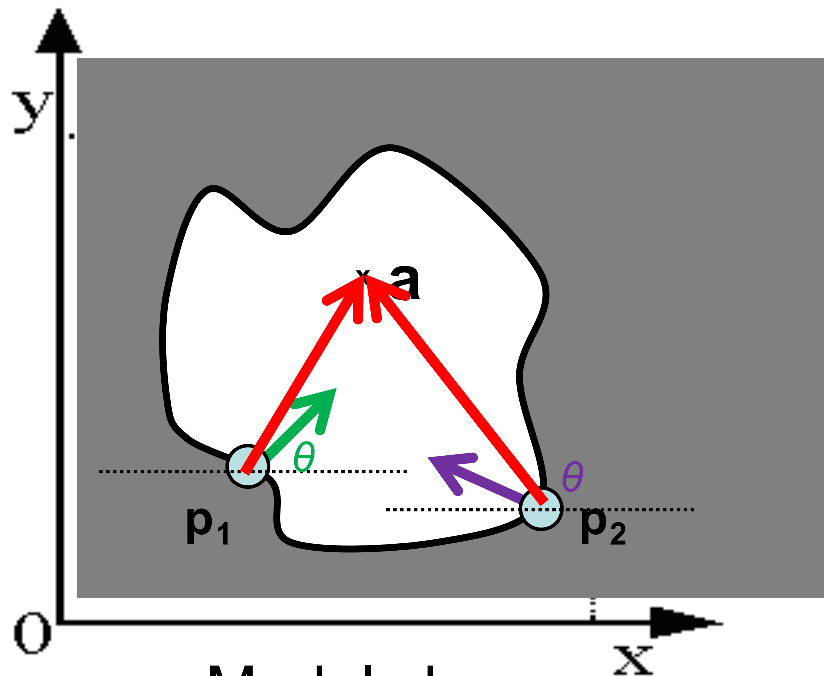


point info —> displacement to the reference point

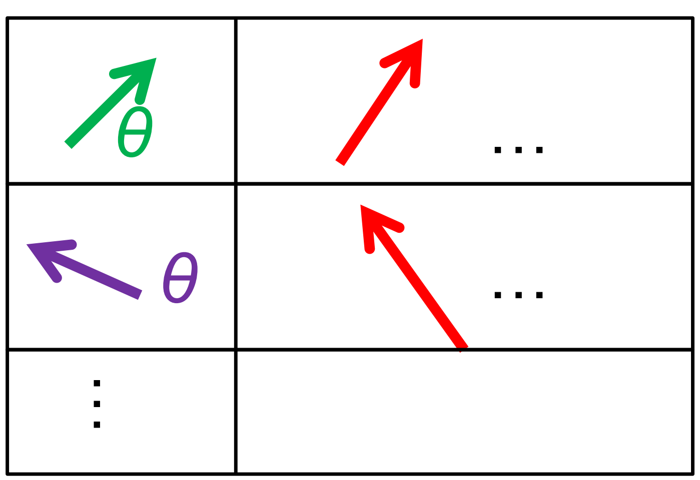
http://vision.cs.utexas.edu/376-spring2018/slides/lecture7-spring2018.pptx

Generalized hough transformation

- Define a model shape by its boundary points and a reference point.



Model shape



<u>Offline procedure:</u>

At each boundary point, compute displacement vector: $\mathbf{r} = \mathbf{a} - \mathbf{p}_i$.

Store these vectors in a table indexed by gradient orientation θ.

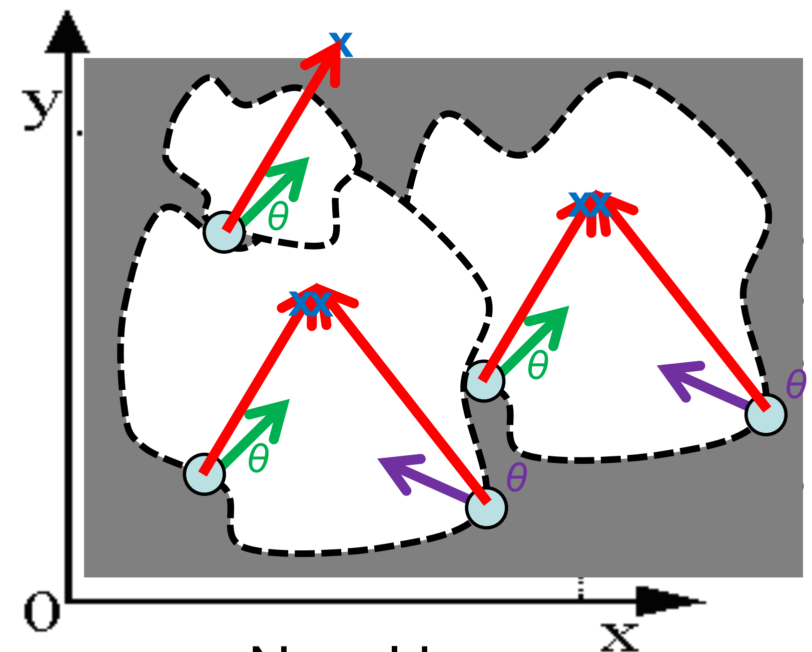
http://vision.cs.utexas.edu/376-spring2018/slides/lecture7-spring2018.pptx

Generalized hough transformation

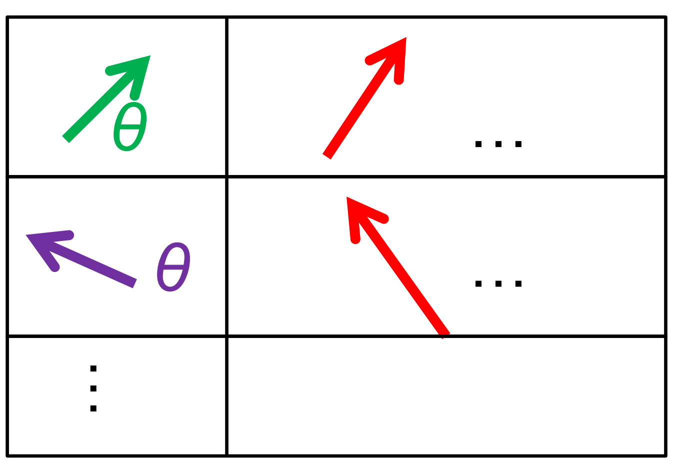
<u>Detection procedure:</u>

For each edge point:

- Use its gradient orientation θ to index into stored table
- Use retrieved **r** vectors to vote for reference point

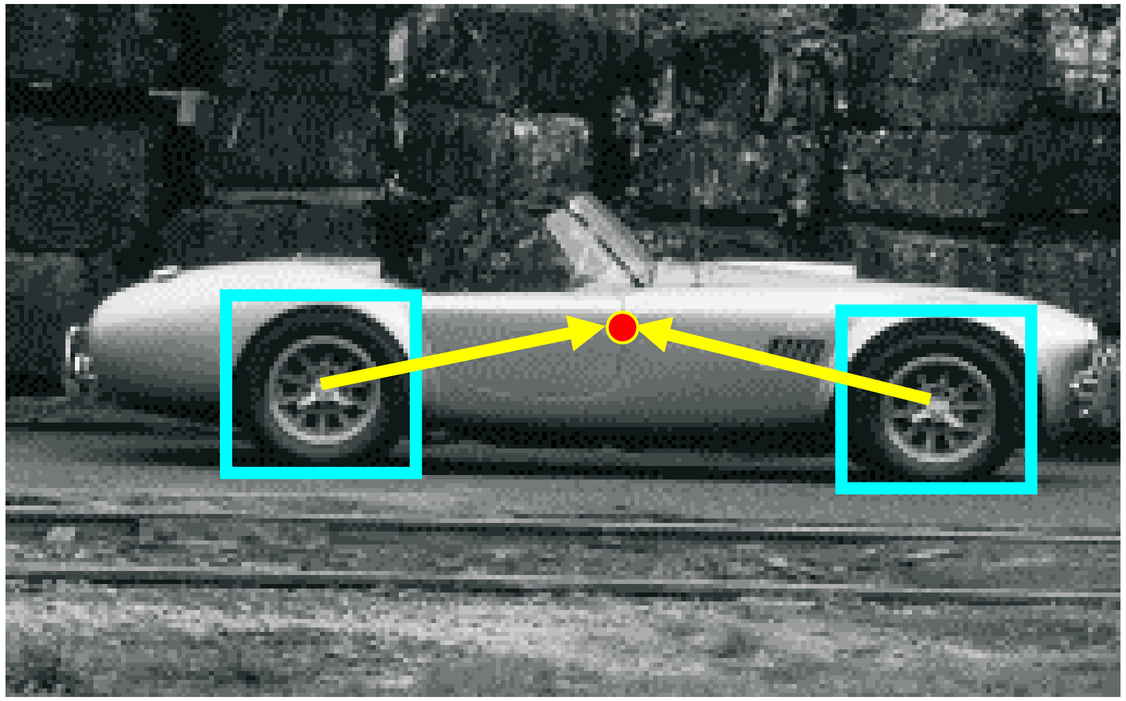


Novel image



Learn the distribution of object positions

Instead of indexing displacements by gradient orientation, index by matched local patterns.



training image

“visual codeword” with displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](), ECCV Workshop on Statistical Learning in Computer Vision 2004

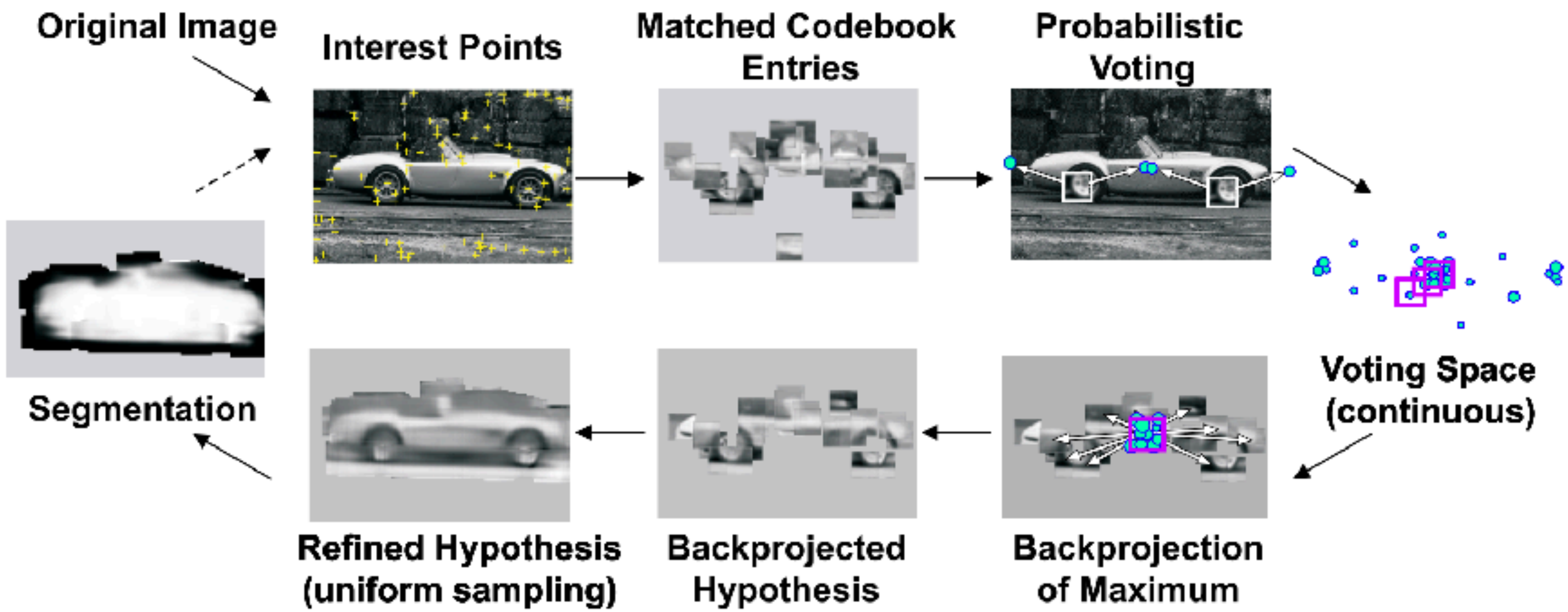
http://vision.cs.utexas.edu/376-spring2018/slides/lecture7-spring2018.pptx

Vote for objects

test image

B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](), ECCV Workshop on Statistical Learning in Computer Vision 2004

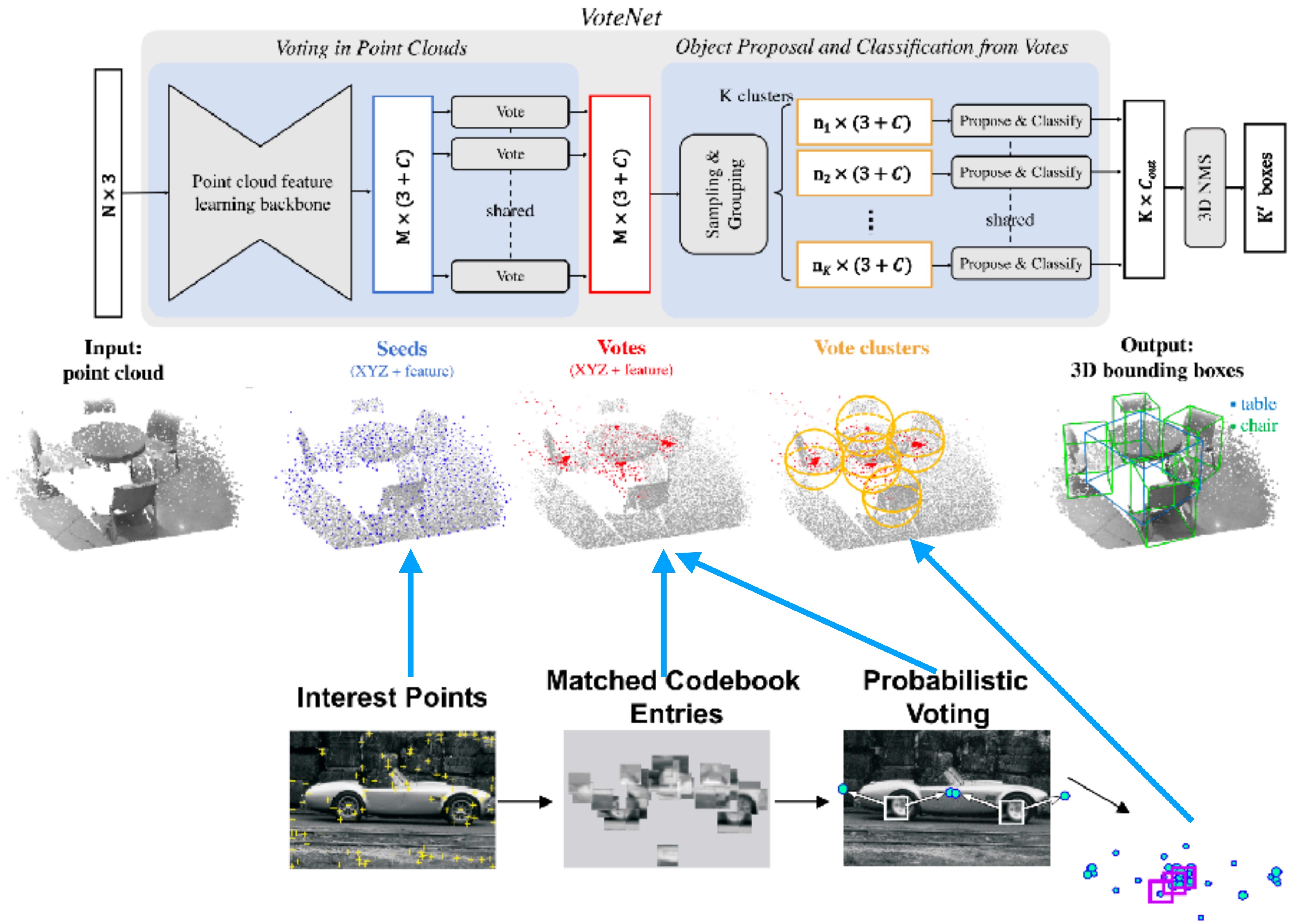
Implicit shape model



a bottom-up method

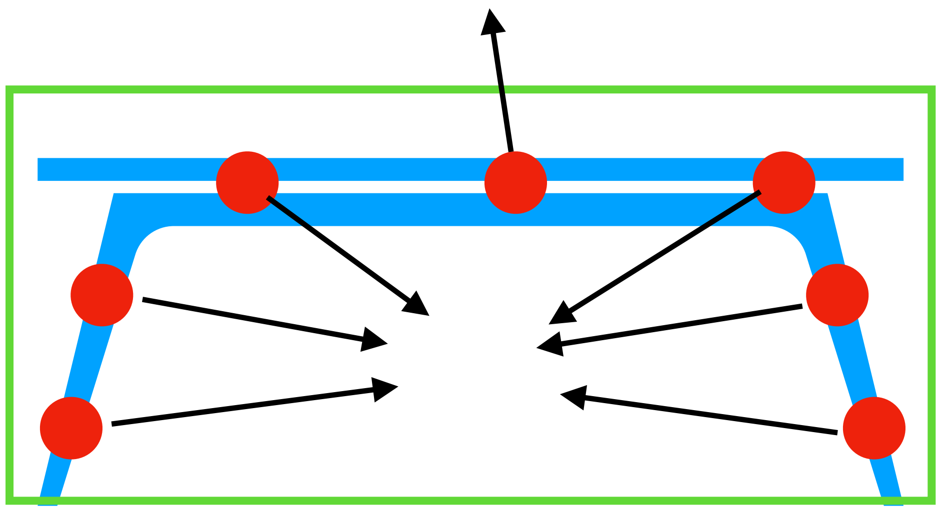
Leibe, Bastian, Ales Leonardis, and Bernt Schiele. "Combined object categorization and segmentation with an implicit shape model." *Workshop on statistical learning in computer vision, ECCV*. Vol. 2. No. 5. 2004.

VoteNet



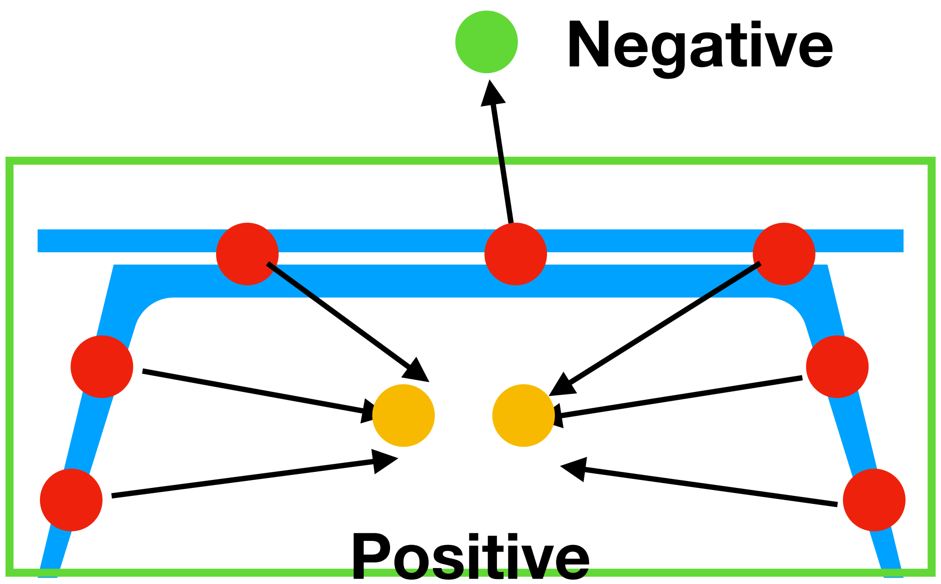
VoteNet
Voting in Point Clouds
Object Proposal and Classification from Votes
N × 3
Point cloud feature learning backbone
M × (3 + C)
Vote
Vote
shared
Vote
M × (3 + C)
K clusters
Sampling & Grouping
n1 × (3 + C)
n2 × (3 + C)
nK × (3 + C)
Propose & Classify
Propose & Classify
shared
Propose & Classify
K × Cout
3D NMS
K′ boxes
Input: point cloud
Seeds (XYZ + feature)
Votes (XYZ + feature)
Vote clusters
Output: 3D bounding boxes
table
chair
Interest Points
Matched Codebook Entries
Probabilistic Voting

Learning to vote



Learn the displacement to the bounding boxcenter

Cluster vote



Use farthest point sampling to get the cluster