PointNet and PointNet++

Date 2021-01-19

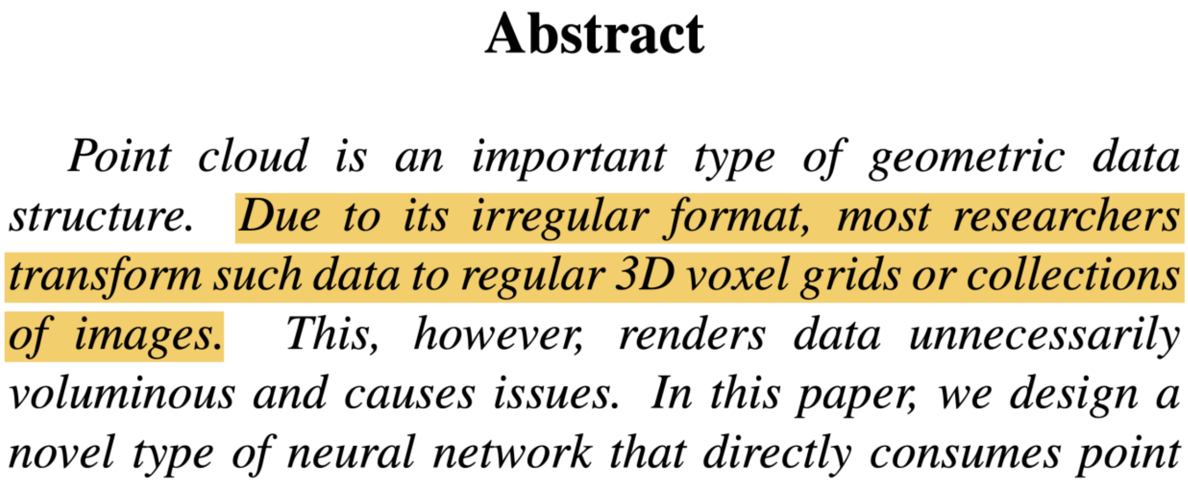
Presenter: Dylan Turpin

Instructor: Animesh Garg

UNIVERSITY OF TORONTO

Impact

2017 → Now



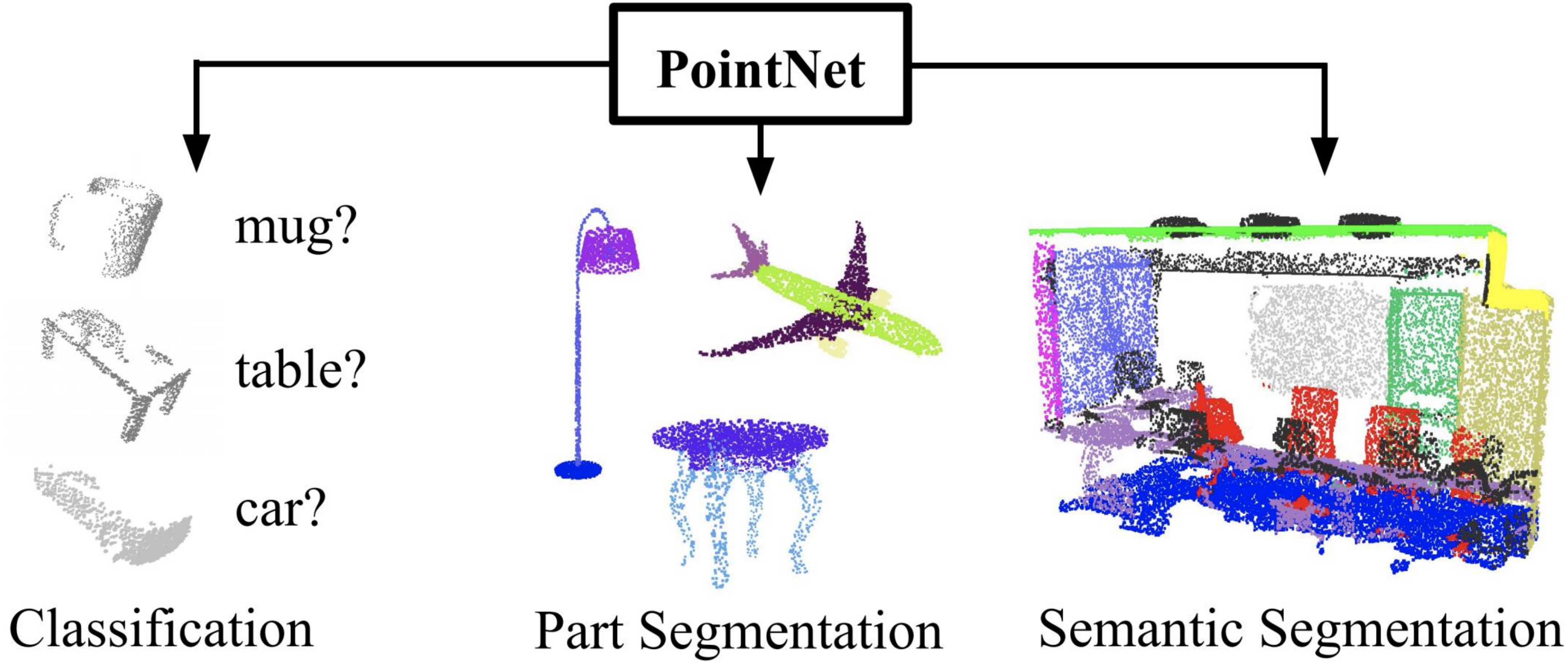
Abstract

Point cloud is an important type of geometric data structure. Due to its irregular format, most researchers transform such data to regular 3D voxel grids or collections of images. This, however, renders data unnecessarily voluminous and causes issues. In this paper, we design a novel type of neural network that directly consumes point

Point clouds are ubiquitous.

Often processed directly without converting to other data types.

Applications



Structure

- 3 properties of point sets
  - Unordered
  - "Interaction among points"
  - Invariance under transformations

→

- 3 architecture choices
  - Max pooling
  - Global-local feature concatenation
  - T-Nets

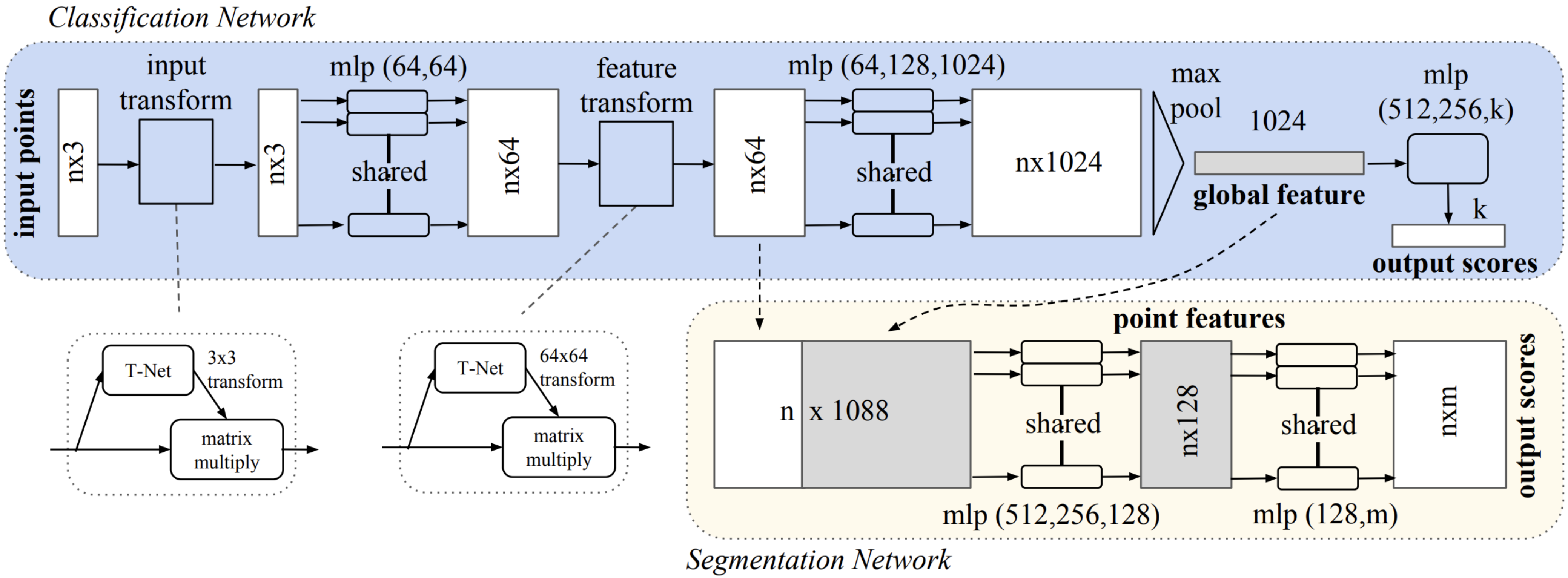
Structure

- 3 properties of point sets
  - Unordered
  - "Interaction among points"
  - Invariance under transformations

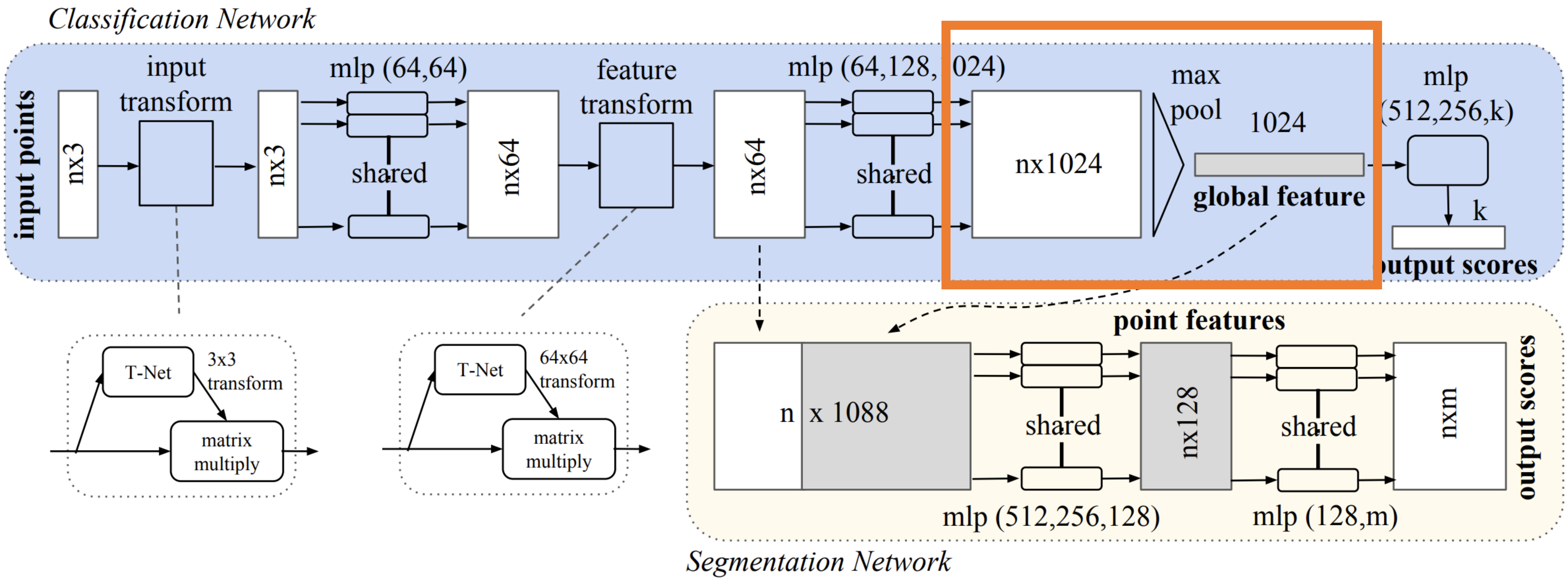
→

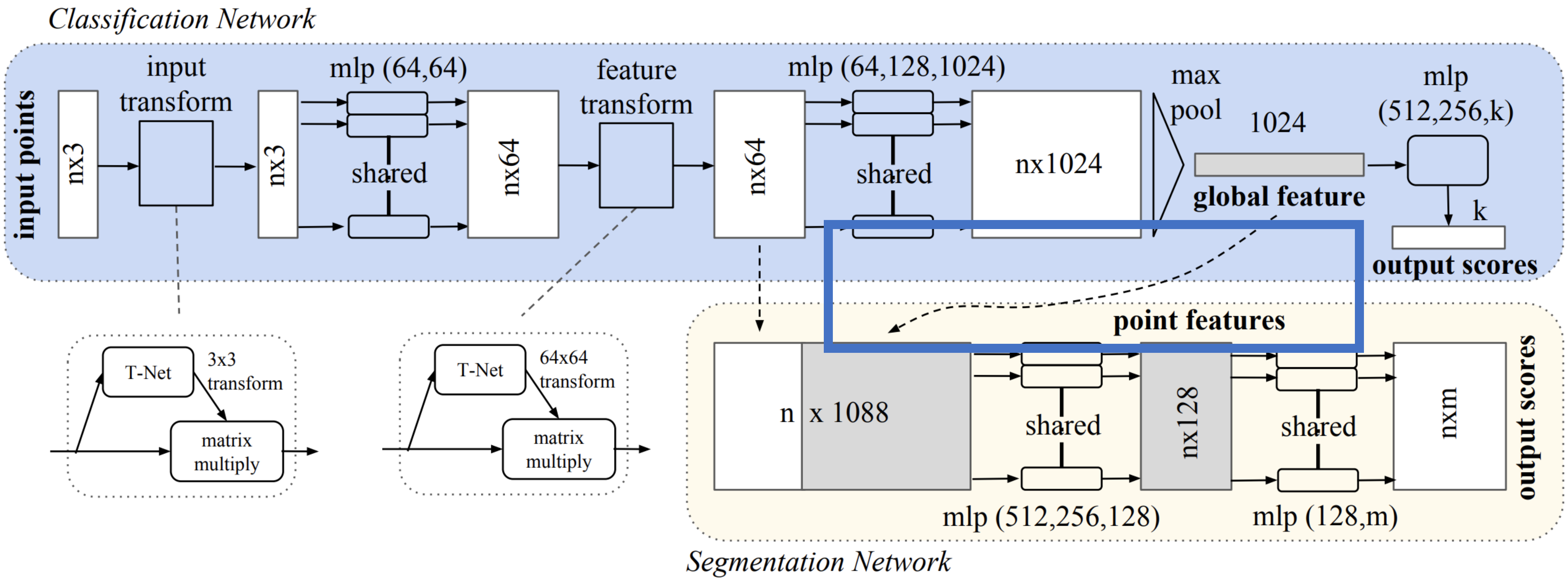
- 3 architecture choices
  - Max pooling
  - Global-local feature concatenation
  - T-Nets

I believe this one is inadequately addressed leading directly to the follow up PointNet++.

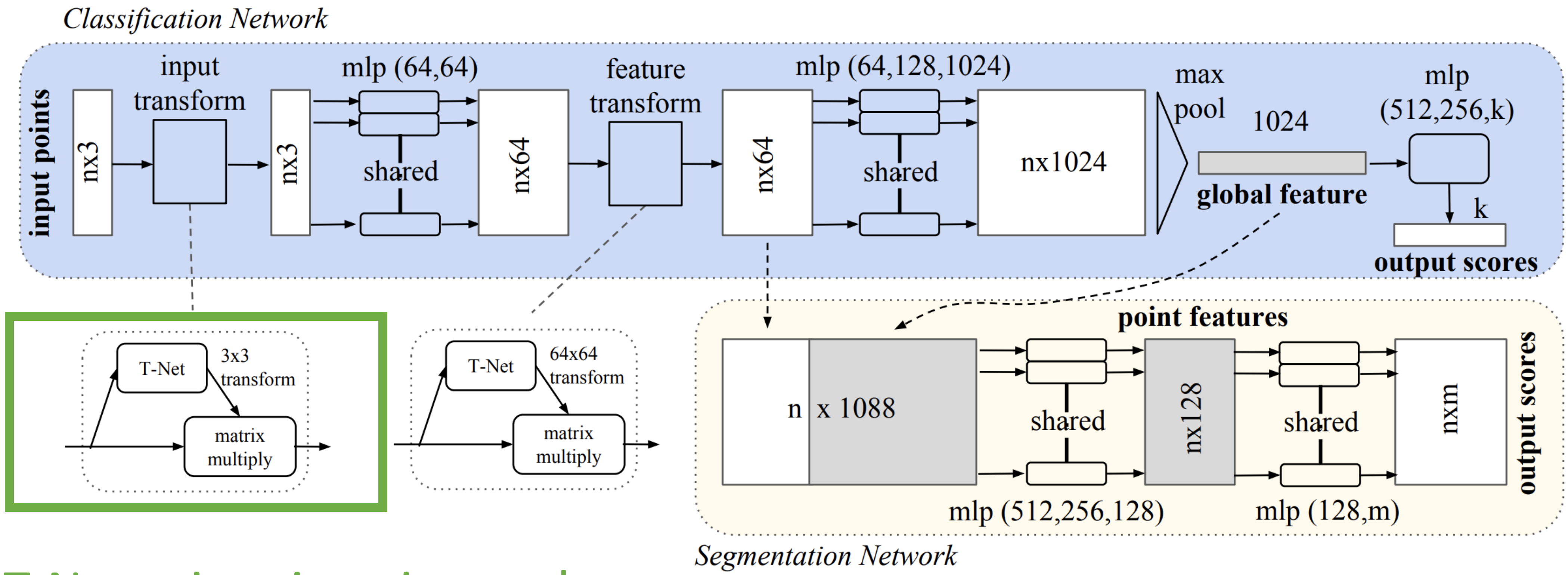

Classification Network
input points
nx3
input transform
nx3
mlp (64,64)
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
point features
n
x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

Max pooling gives order invariance





Concatenating global-to-local features combines local + global info

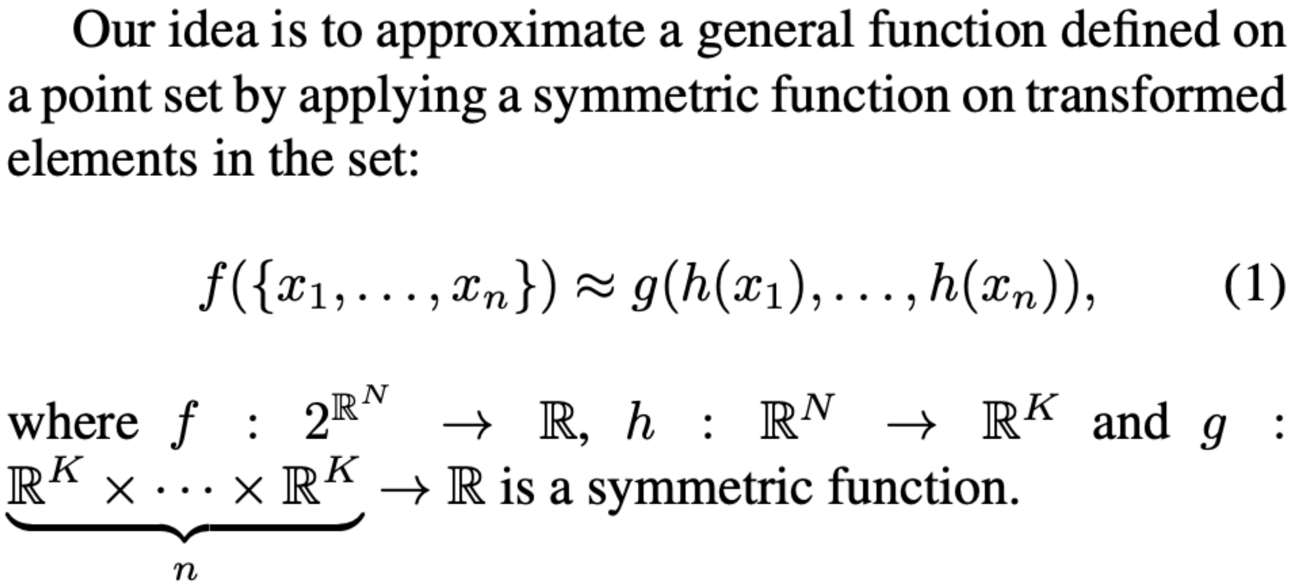


T-Nets gives invariances by transforming to canonical pose

Property #1: Unordered

- Point sets are unordered, so point set representations should be invariant to input ordering
- Three ways to achieve this
  - 1) Sort to a canonical order
  - 2) Use an RNN and augment training with random orders
  - 3) Use a symmetric function

Property #1: Unordered

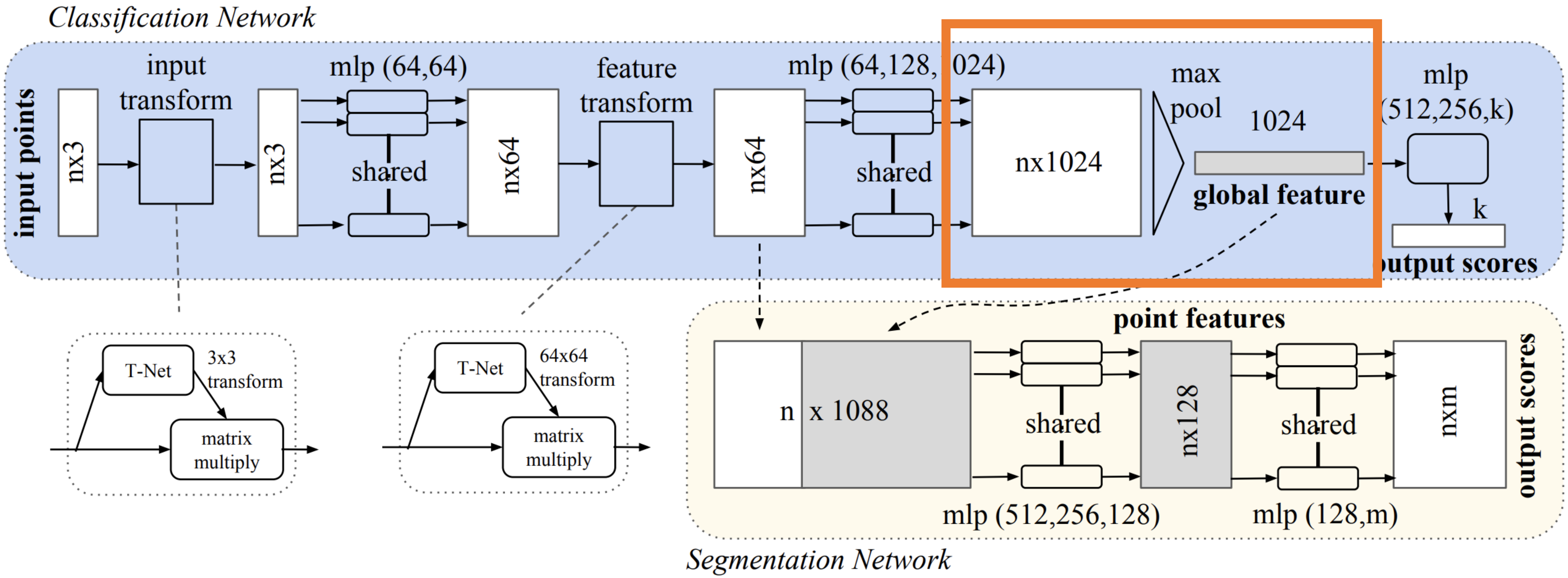


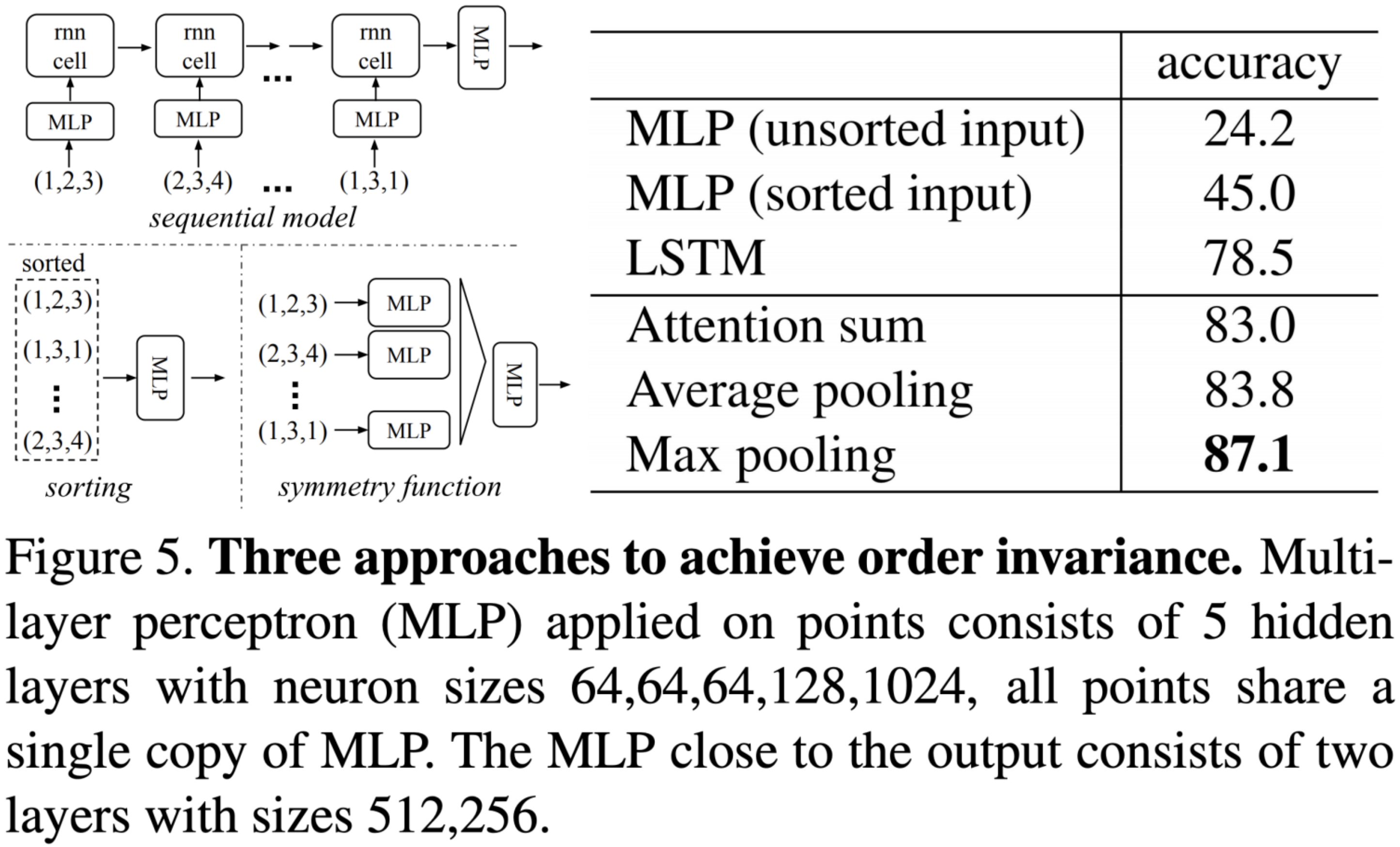
Our idea is to approximate a general function defined on a point set by applying a symmetric function on transformed elements in the set:

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n)), \qquad (1)$$

where $f : 2^{\mathbb{R}^N} \rightarrow \mathbb{R}$, $h : \mathbb{R}^N \rightarrow \mathbb{R}^K$ and $g : \underbrace{\mathbb{R}^K \times \cdots \times \mathbb{R}^K}_{n} \rightarrow \mathbb{R}$ is a symmetric function.

Max pooling gives order invariance





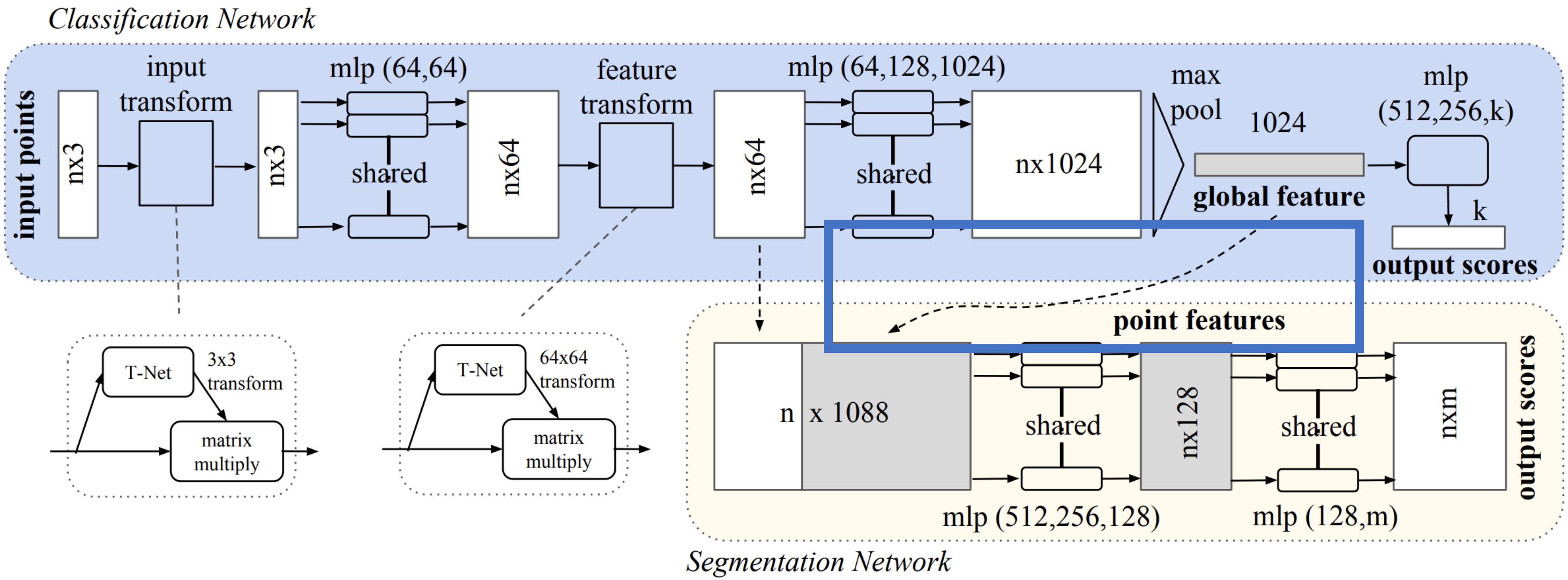
	accuracy
MLP (unsorted input)	24.2
MLP (sorted input)	45.0
LSTM	78.5
Attention sum	83.0
Average pooling	83.8
Max pooling	**87.1**

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.

Property #2: Interaction between points

- Points are in a metric space
- There are meaningful local neighborhoods of points
- Many local prediction tasks (e.g. normal prediction, segmentation) require summarizing information in a local neighborhood

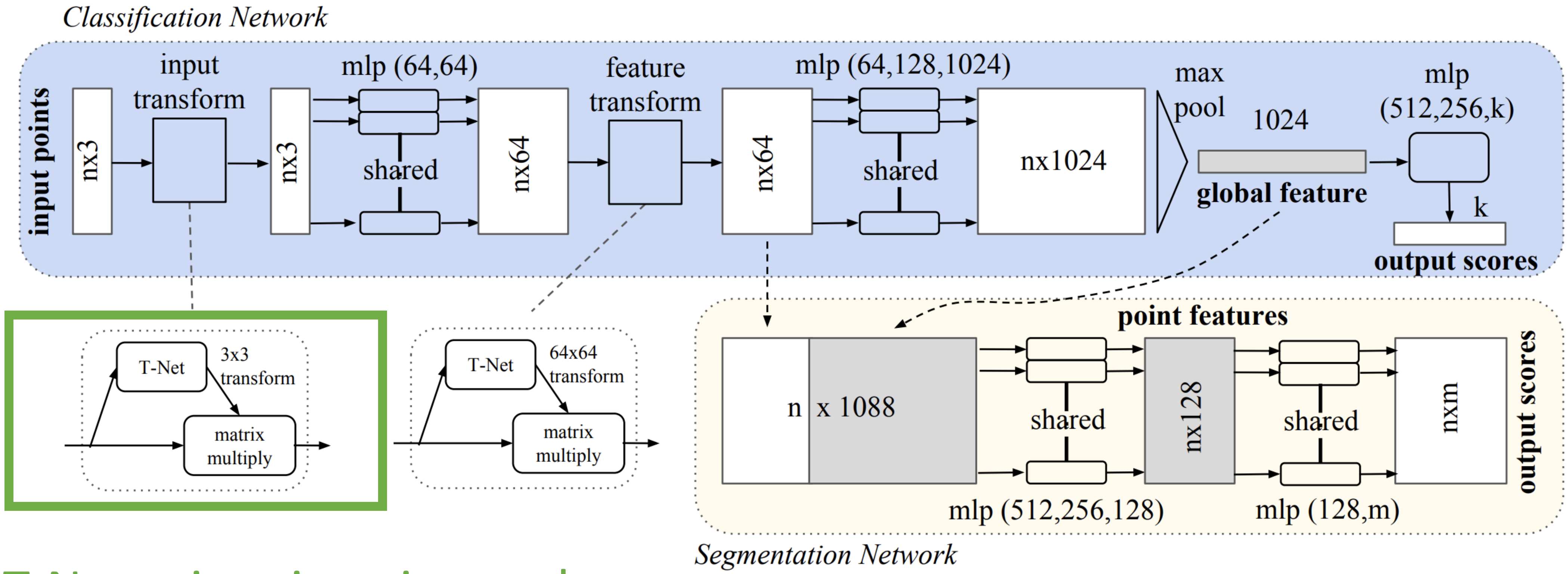
- How to get local features that take global context into account?
- Solution: Append global features to local features



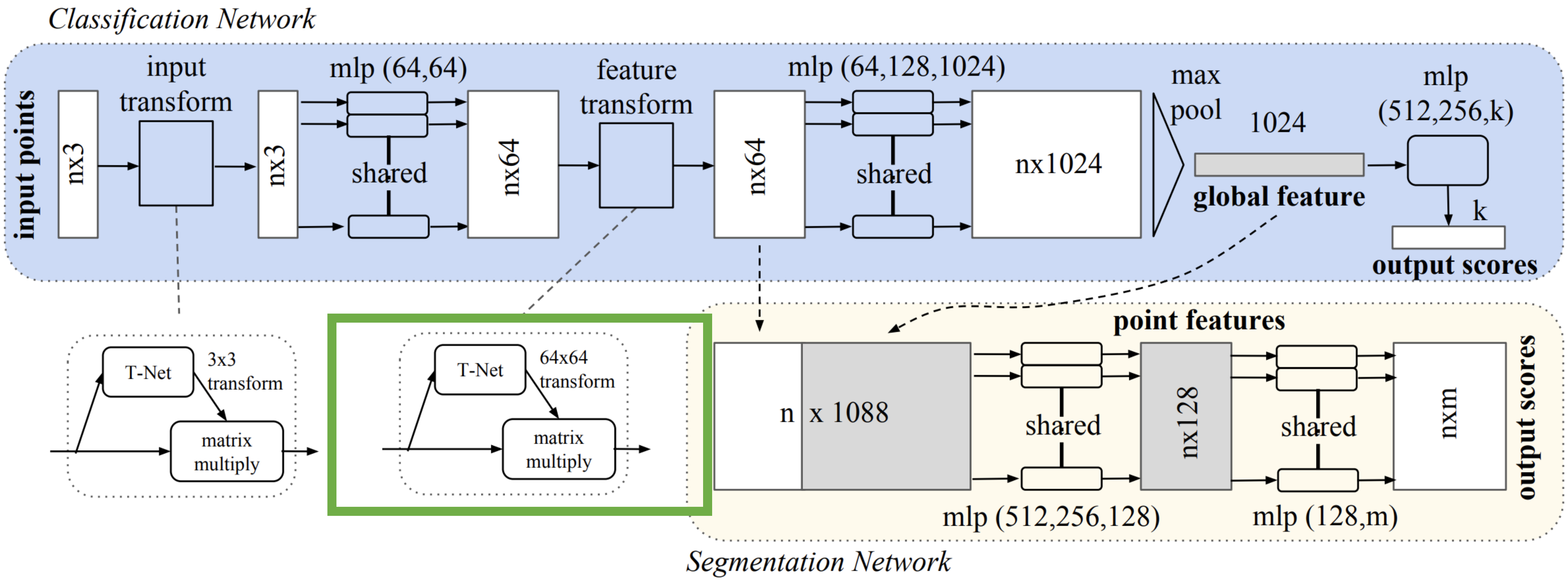
Concatenating global-to-local features combines local + global info

Property #3: Invariance under transformations

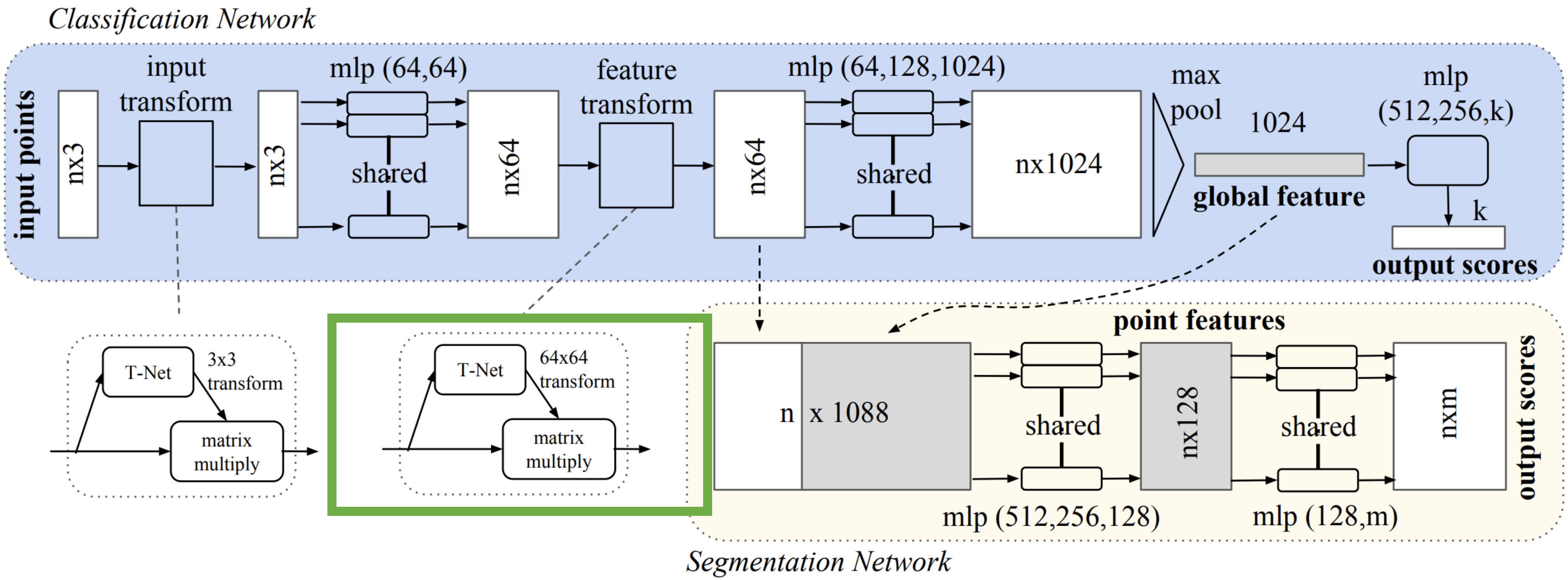
- Changing pose does not change object identity
- So representations should be invariant to
  e.g. rigid pose transformations (like translation, rotations)



T-Nets gives invariances by transforming to canonical pose

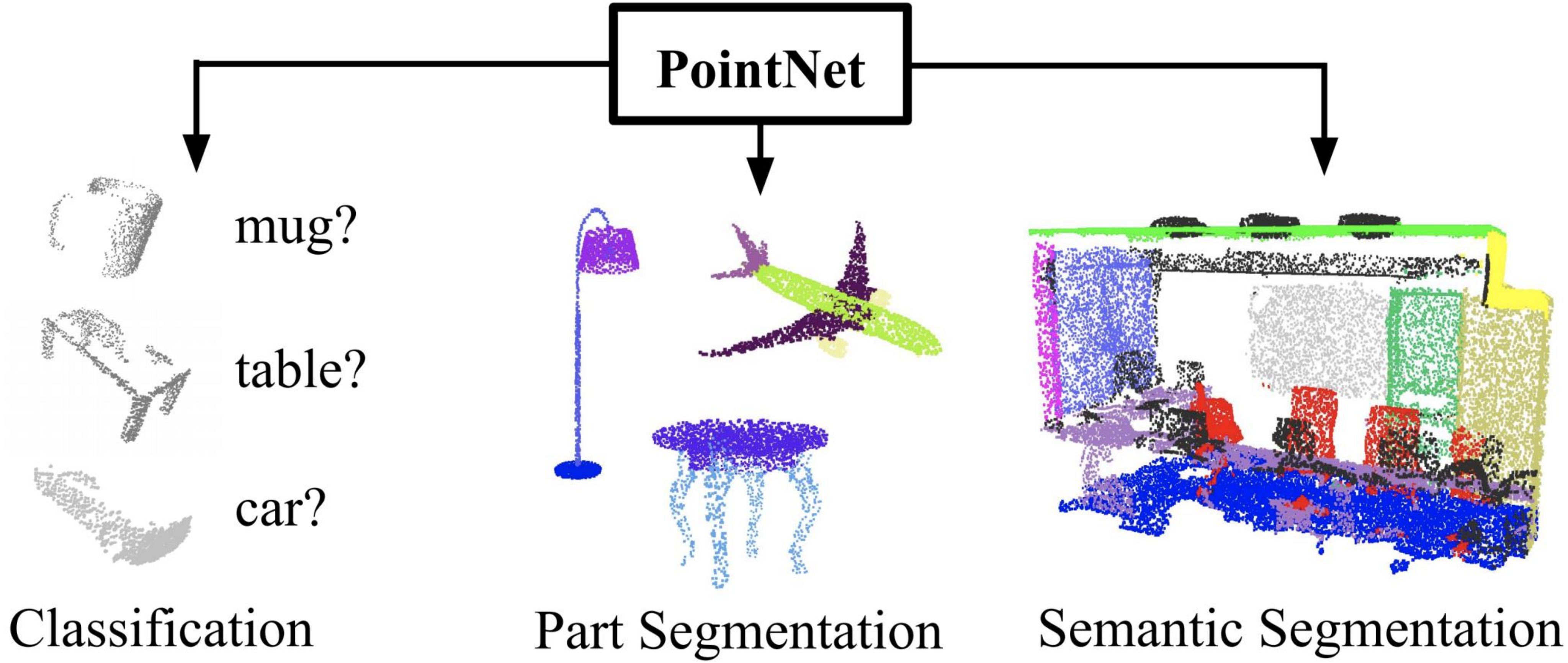


But what is this second one doing?

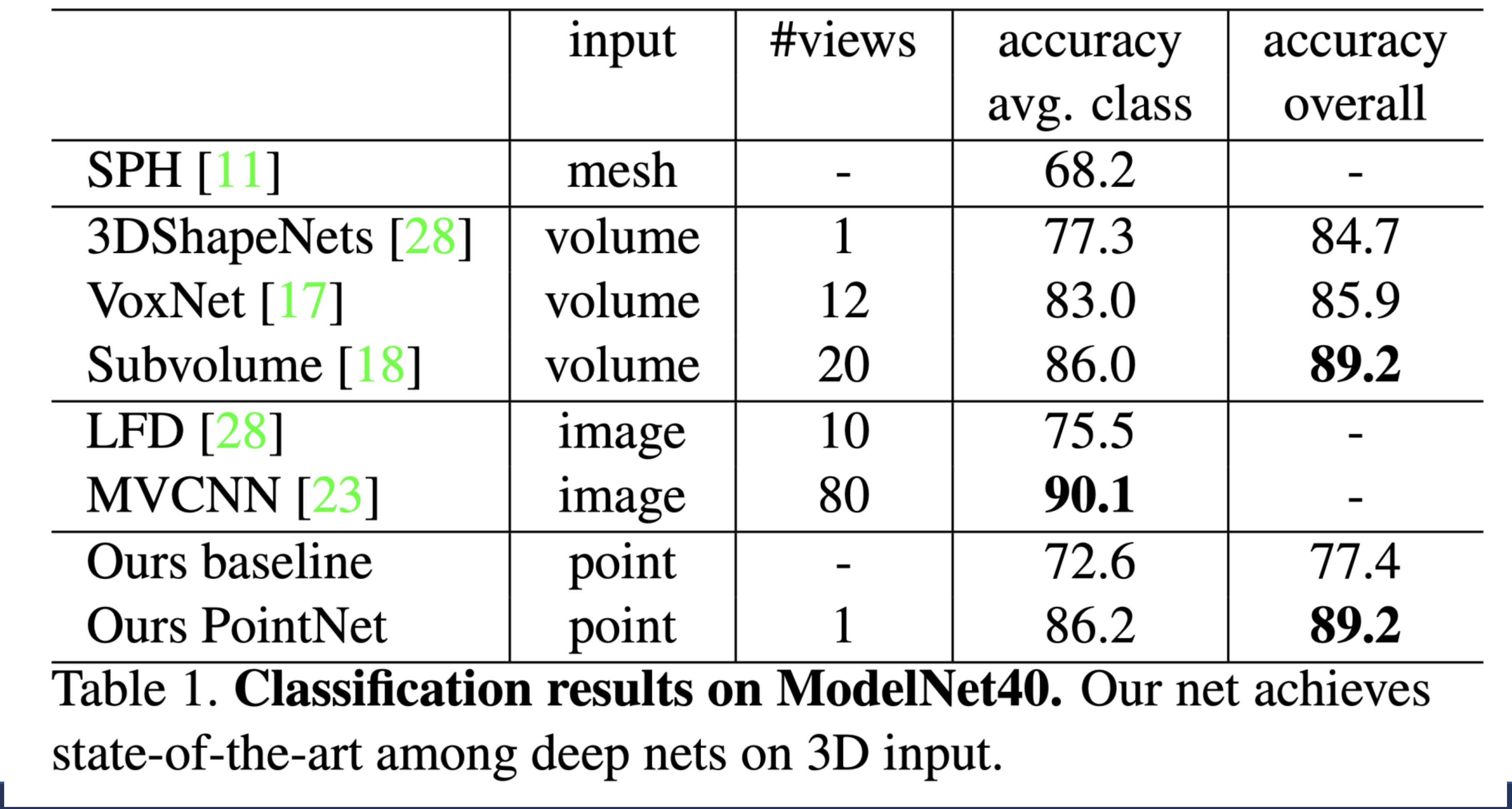


But what is this second one doing? "Feature alignment"

Results on 3 Applications



Classification



	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.

ShapeNet Segmentation

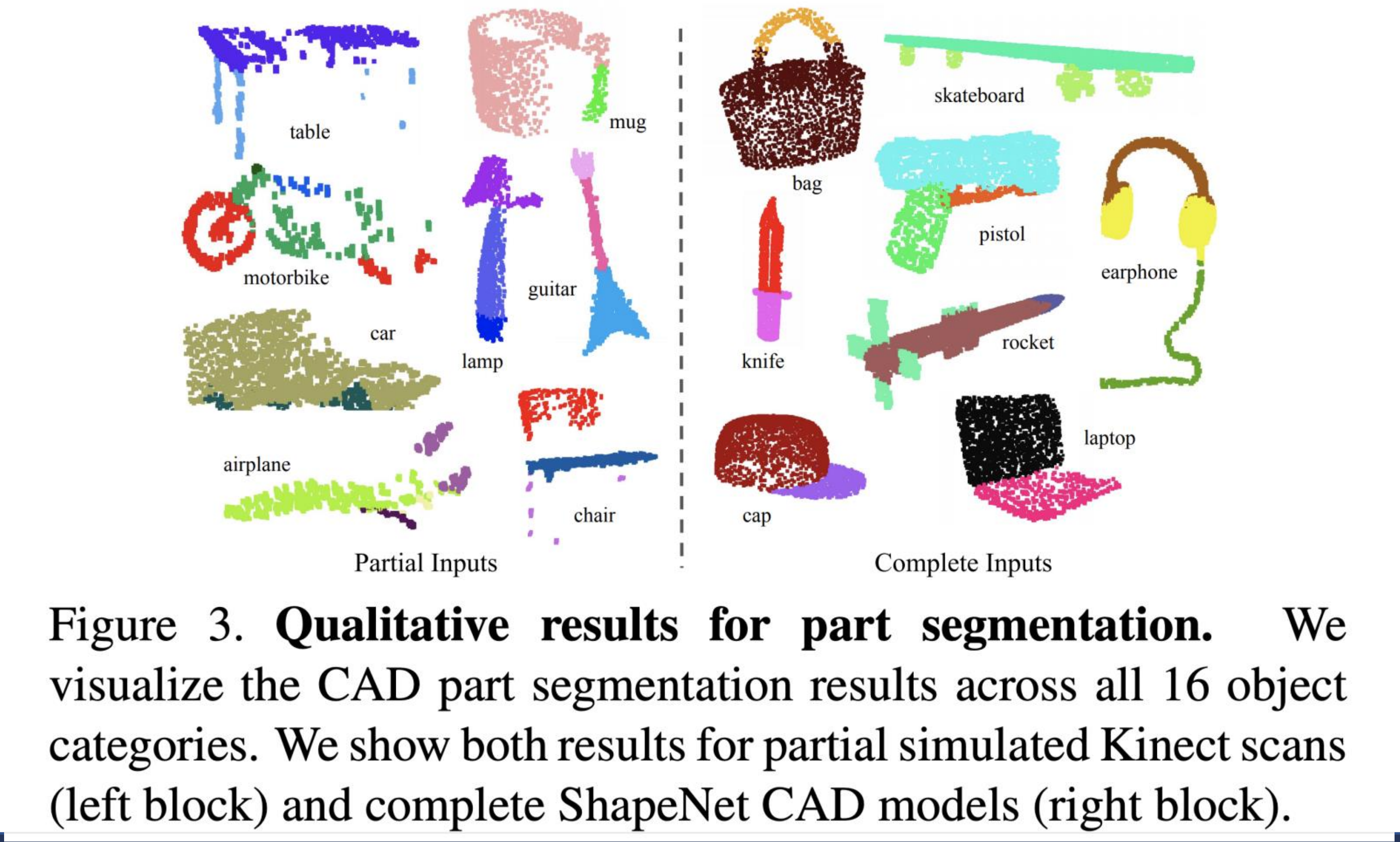
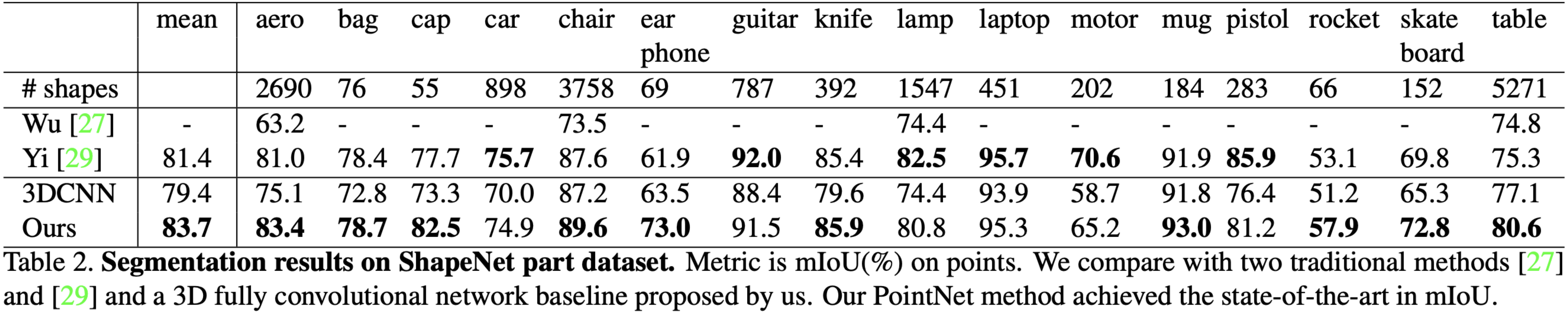


Figure 3. **Qualitative results for part segmentation.** We visualize the CAD part segmentation results across all 16 object categories. We show both results for partial simulated Kinect scans (left block) and complete ShapeNet CAD models (right block).

ShapeNet Segmentation



	mean	aero	bag	cap	car	chair	ear	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate	table
							phone									board	
# 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.

Scene Segmentation

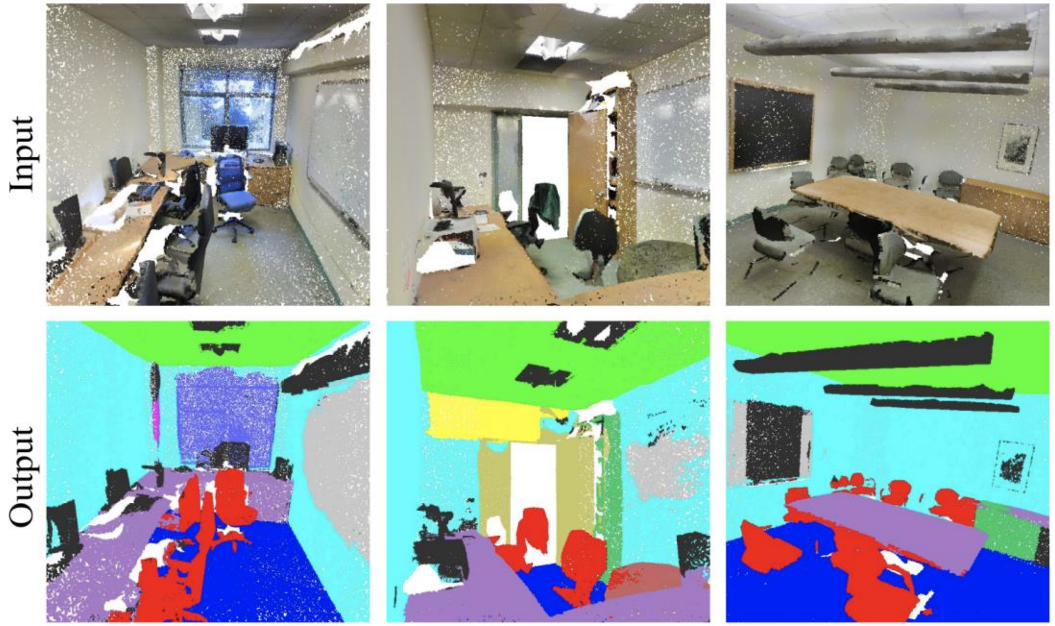
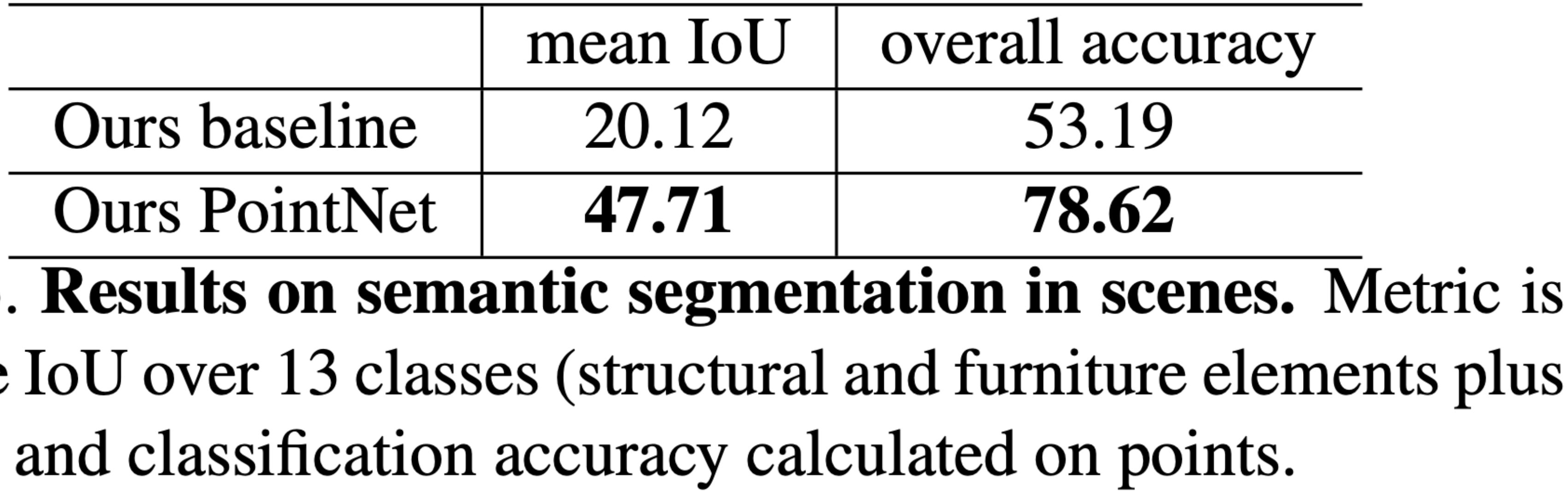


Figure 4. **Qualitative results for semantic segmentation.** Top row is input point cloud with color. Bottom row is output semantic segmentation result (on points) displayed in the same camera viewpoint as input.

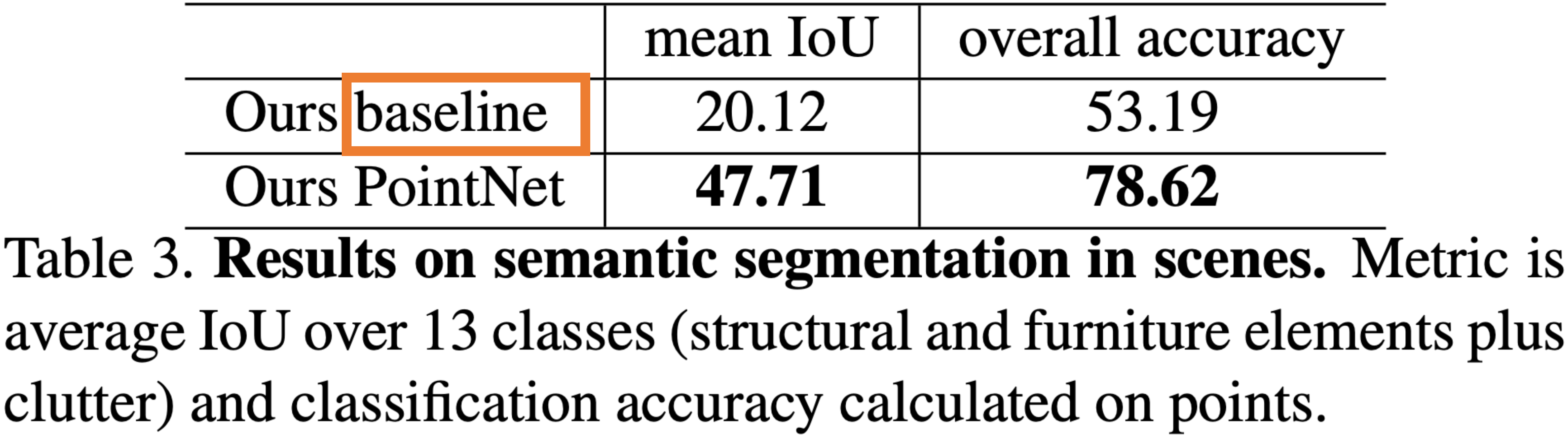
Scene Segmentation



	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.

Scene Segmentation



	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.

Hand-crafted point features, not sure how convincing this is

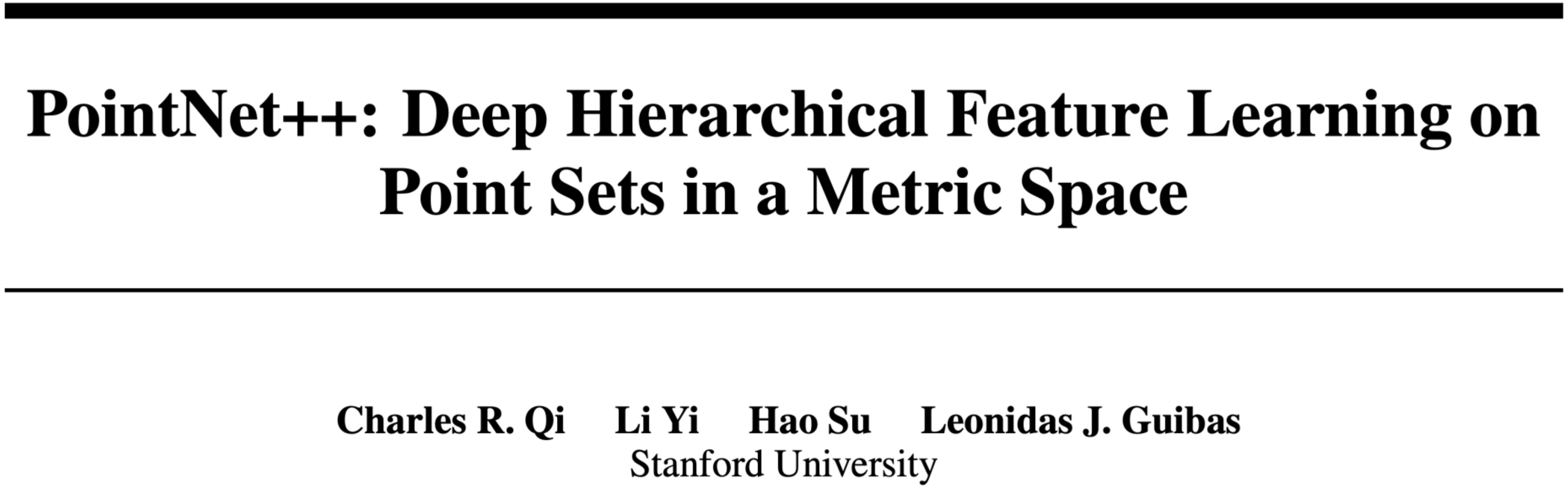
Contribution

- New architecture for point clouds with
  - Order invariance
  - Transformation invariance
  - Ability to combine local and global info
- Strong results on ShapeNet segmentation, scene segmentation and classification
- Empirical results supporting choice of order invariance mechanism, transformation invariance mechanism

What's missing?

- Hierarchy!
- By analogy to 2D image processing:
  - Imagine calculating per pixel features, then a single round of global pooling
  - Loses a lot of information!
  - Point clouds don't have the same info (ordering, adjacency),
    but they do have distances which define local neighborhoods that aren't being used here

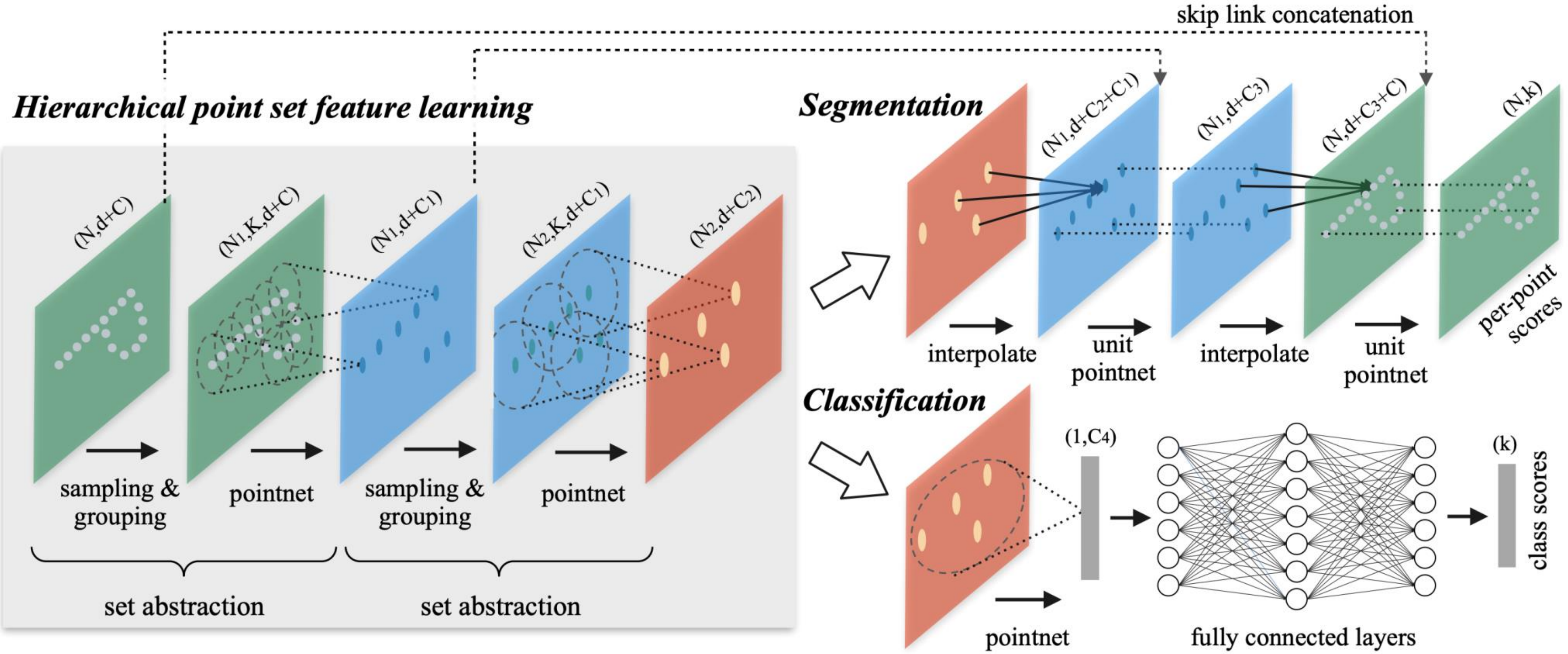
Enter: PointNet++



PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

Charles R. Qi Li Yi Hao Su Leonidas J. Guibas
Stanford University

Enter: PointNet++



Structure

Desired properties: → Architecture choices:

- 1) hierarchical feature learning
- 2) robustness to density changes
- 3) multi-scale information aggregation
  (not just local+global)

Architecture choices:

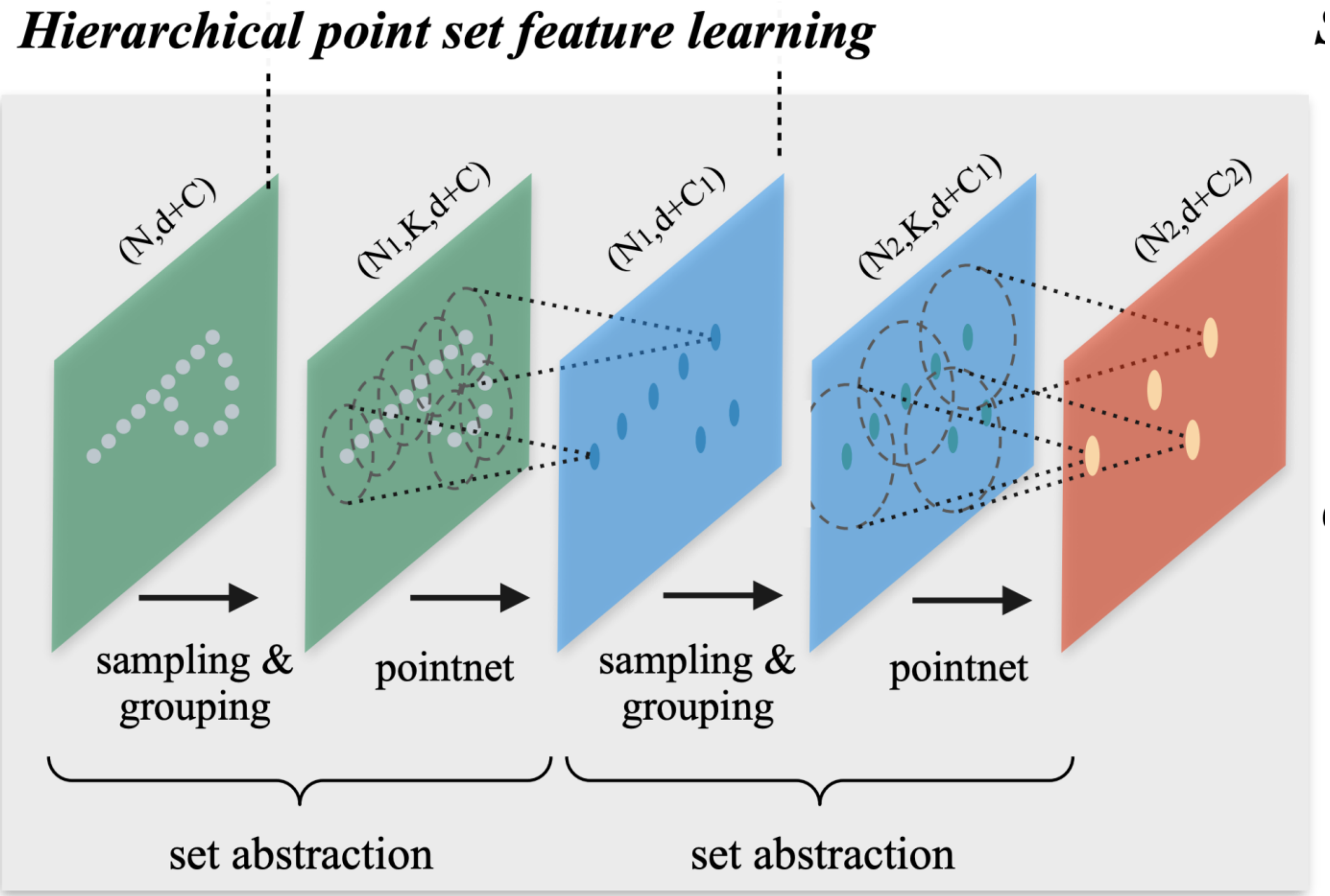
- 1) set abstraction layers
- 2) density-adaptive grouping
- 3) distance-based interpolation

Hierarchical feature learning

- Idea: Apply PointNet recursively to nested partitions of points
  - Need a way to partition points
  - Need stackable layers for recursion (output type = input type)

- Set abstraction layer
  - Sampling: Select centroids with iterative farthest point
  - Grouping: Select group of points for each centroid (KNN or ball-query)
  - PointNet: Run PointNet on each group

Hierarchical feature learning

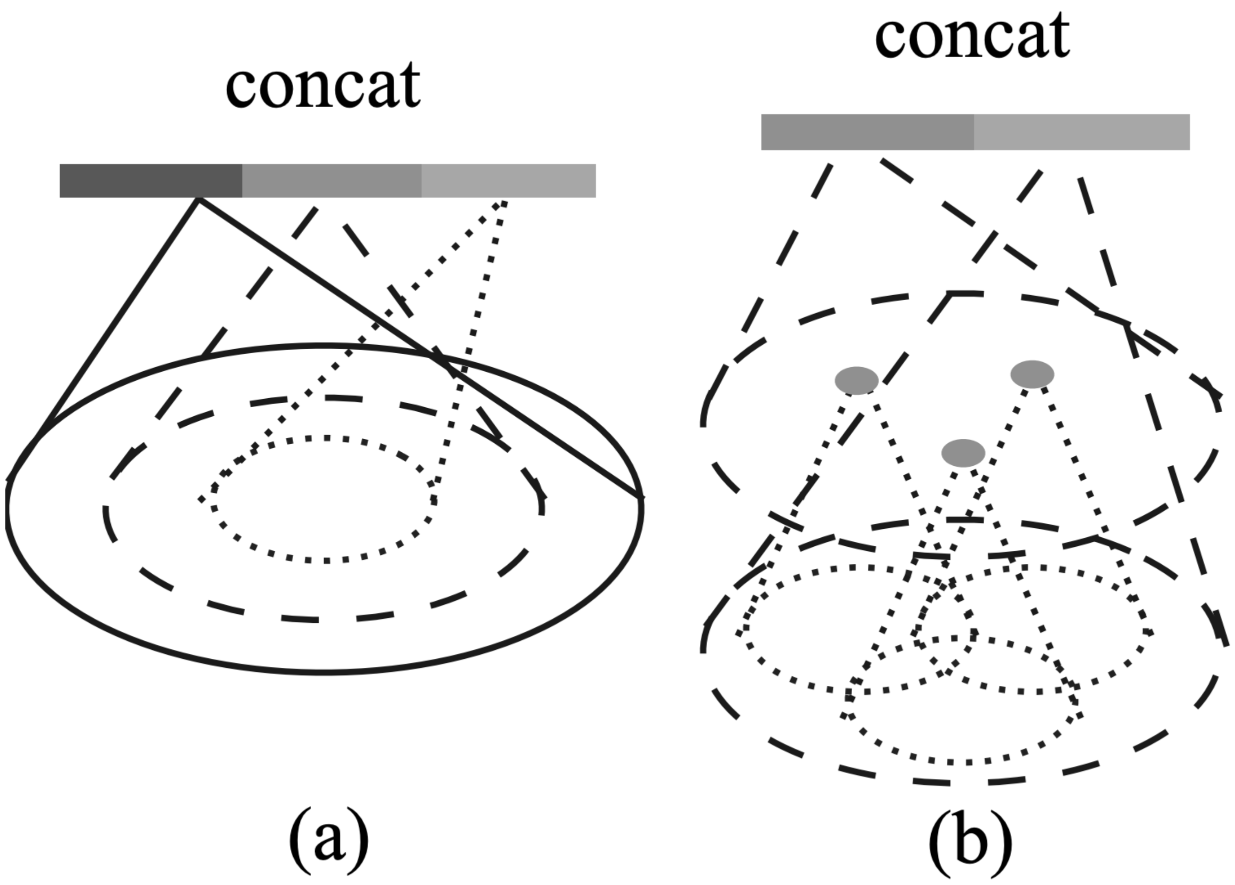


Robustness to density

- Idea: Adaptively group based on density
  - In high density areas, group tightly (to avoid losing detail)
  - In low density areas, group widely (or won't have sufficient info)

- Adaptive density-grouping
  - Sampling: Select centroids with iterative farthest point
  - Grouping: Select group of points for each centroid (KNN or ball-query)
  - PointNet: Run PointNet on each group

Robustness to density



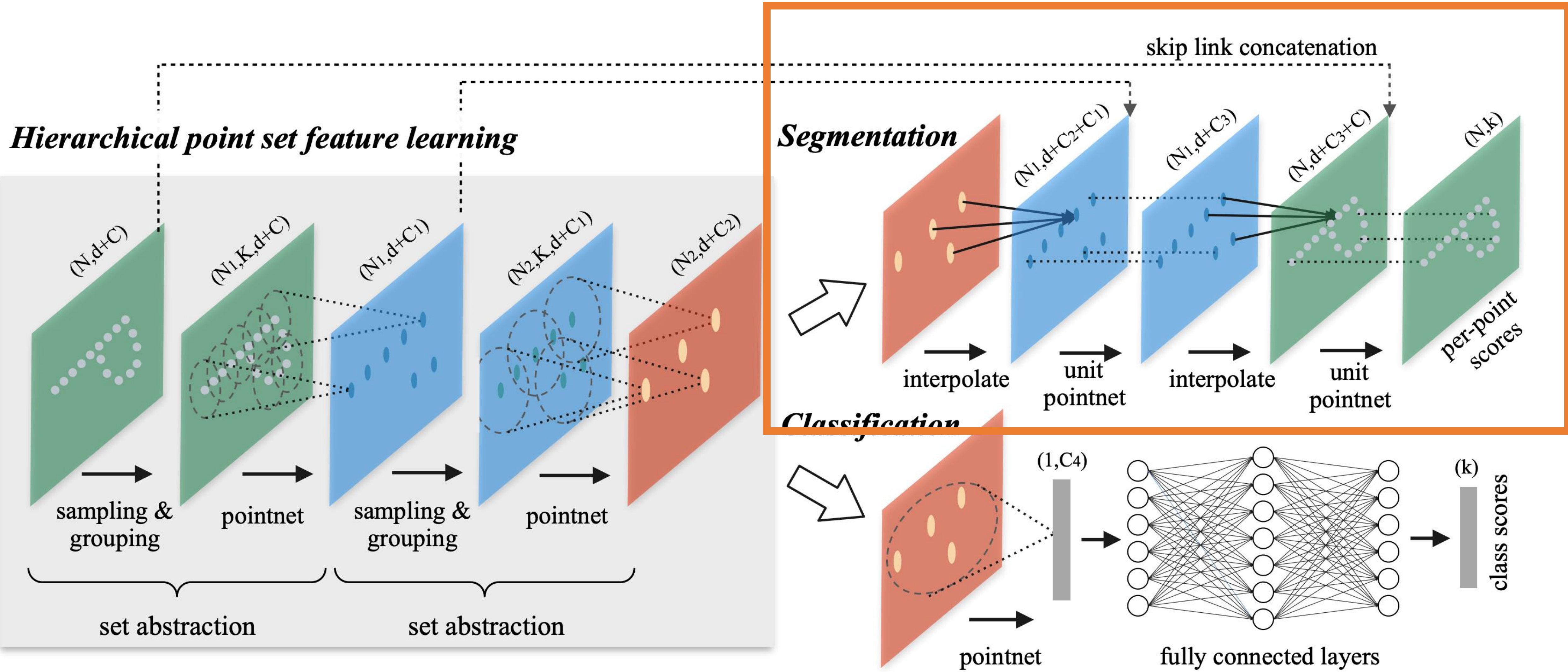
Multi-Scale Grouping Multi-Resolution Grouping

Multi-scale information aggregation

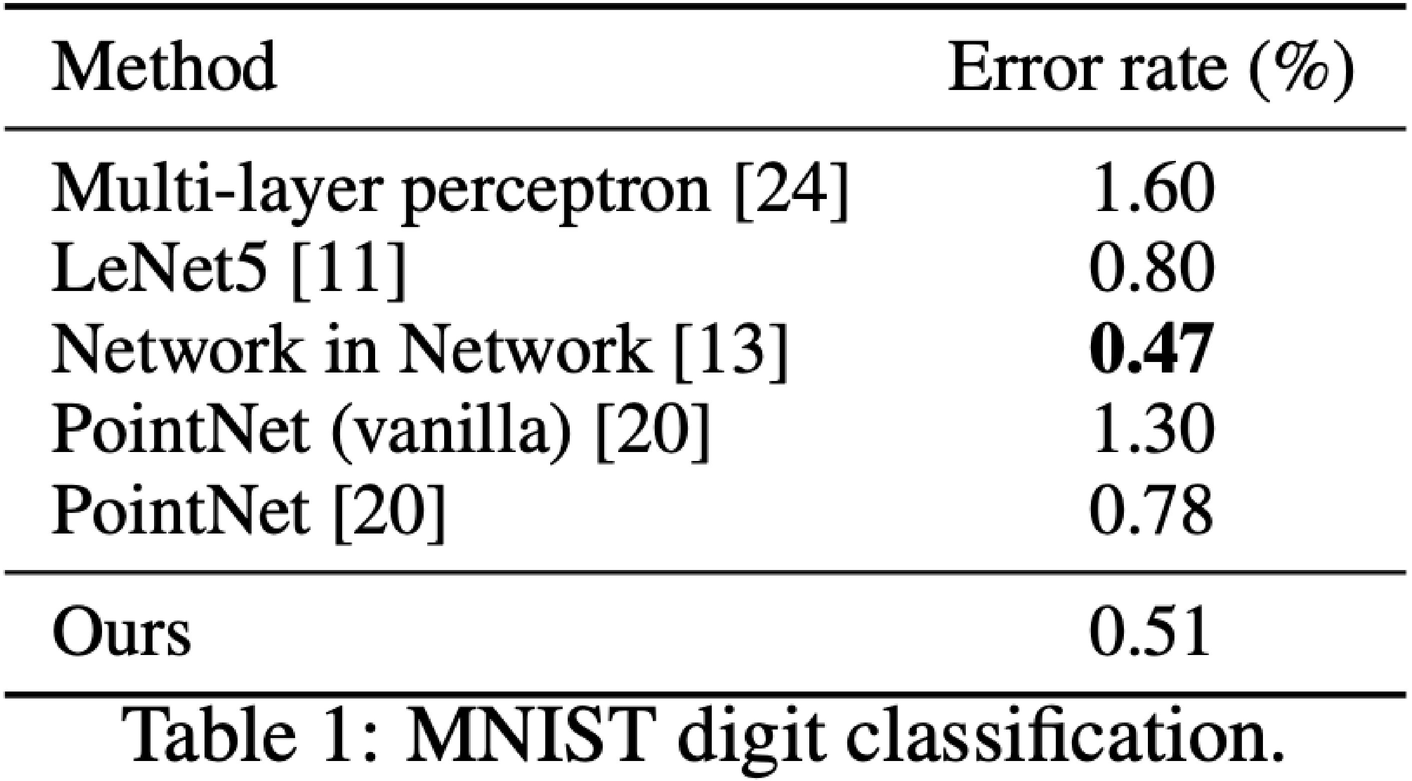
- For segmentation, normal prediction,
  need point features that incorporate global/neighborhood context

- Could just append local+global, but want to exploit hierarchy

- Idea: Distance-based interpolation
  - Interpolate higher layer features and append to points in lower layer


skip link concatenation
Hierarchical point set feature learning
$(N,d+C)$
$(N_1,K,d+C)$
$(N_1,d+C_1)$
$(N_2,K,d+C_1)$
$(N_2,d+C_2)$
sampling & grouping
pointnet
sampling & grouping
pointnet
set abstraction
set abstraction
Segmentation
$(N_1,d+C_2+C_1)$
$(N_1,d+C_3)$
$(N,d+C_3+C)$
(N,k)
interpolate
unit pointnet
interpolate
unit pointnet
per-point scores
Classification
$(1,C_4)$
pointnet
fully connected layers
(k)
class scores

Results: MNIST classification

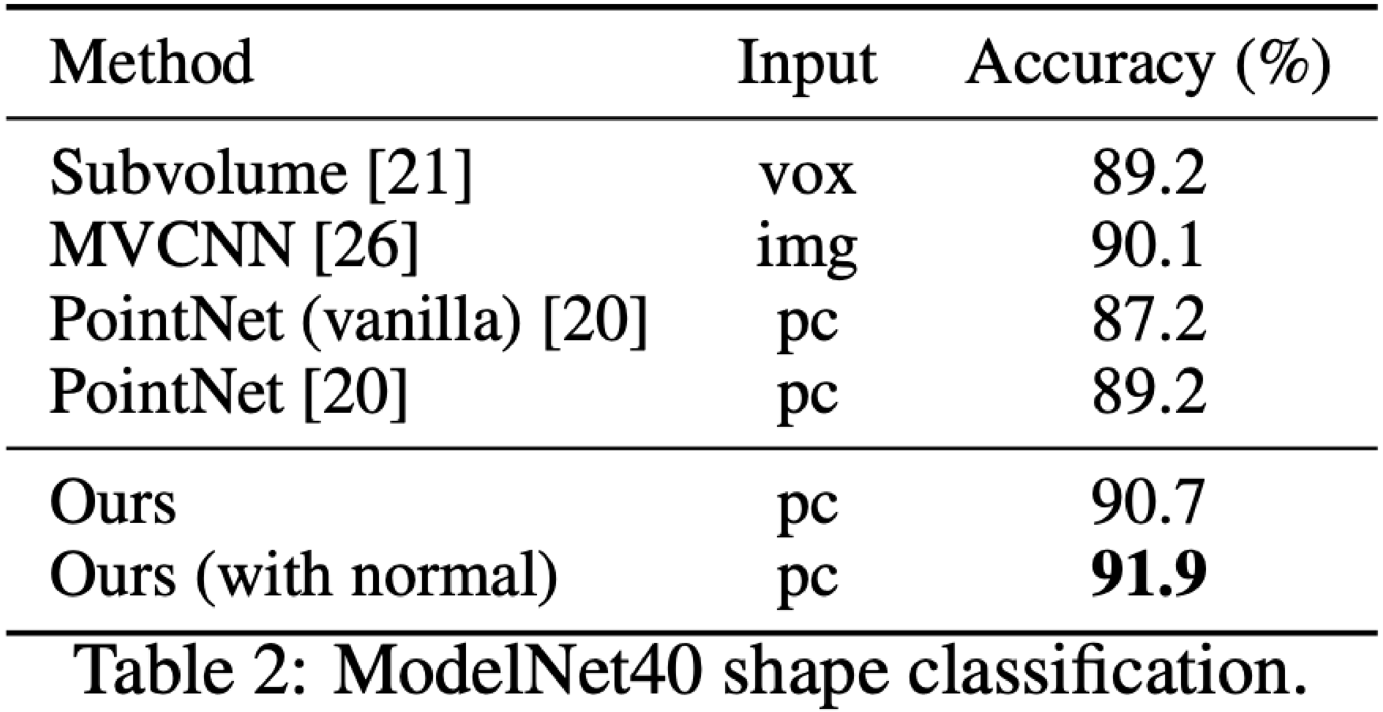


Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

Table 1: MNIST digit classification.

Note: comparing methods which take (2D) point sets
to methods that take image inputs. Used 512 2D points.

Results: ModelNet40 classification

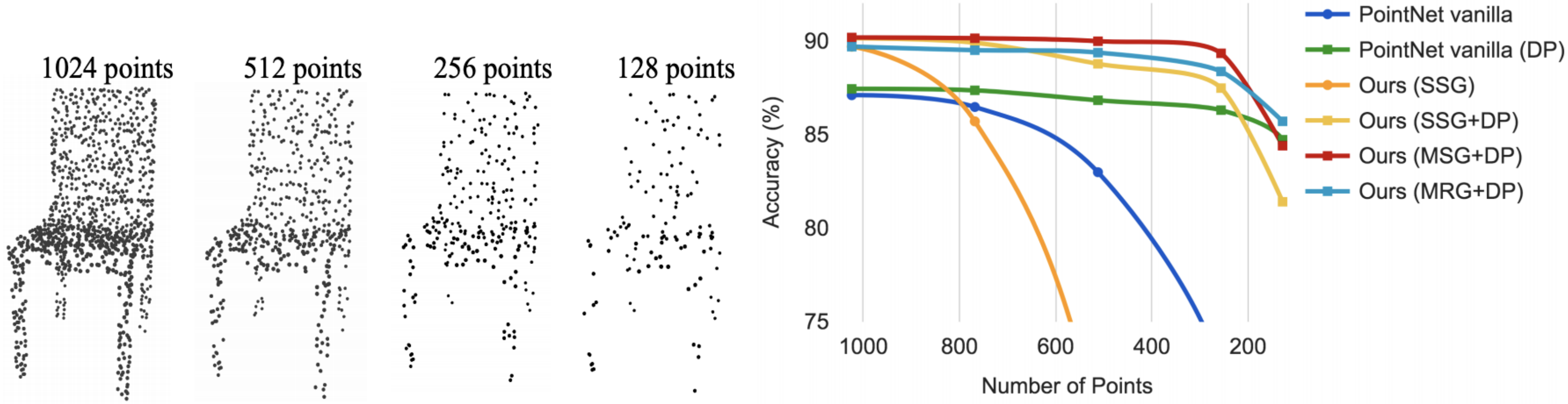


Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Table 2: ModelNet40 shape classification.

Note: "Ours (with normal)" takes normal at each point as input, also uses additional points (5000 instead of 1024)

Results: Robustness to density



"DP" = random input dropout during training.
"SSG" = single scale grouping, "MSG" = multi scale grouping,
"MRG" = multi resolution grouping

Contribution

- Proposed a new Hierarchical PointNet architecture
- Beats vanilla PointNet by a large margin in 2D and 3D tests
- Matches or beats "mature CNN" architectures
- Greatly improved robustness to density with adaptive grouping

Comparison

	PointNet	PointNet++
Information aggregation?	Single Max Pool layer	Hierarchical "set abstraction layers"
Density adaptive?	N	Y
Adding global info to point features?	Appends global to local	Distance-based interpolation