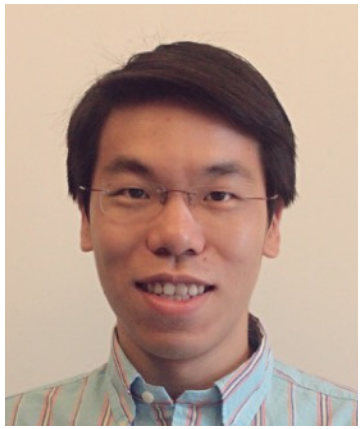
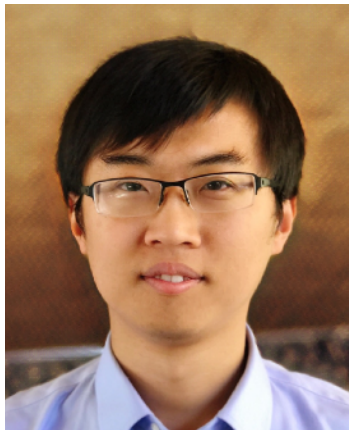
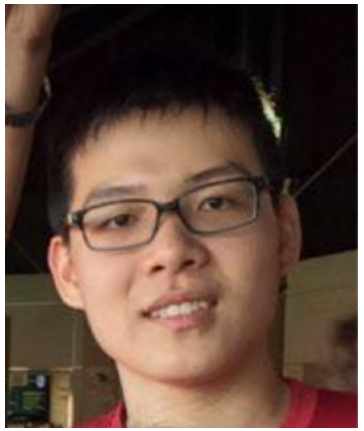
PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

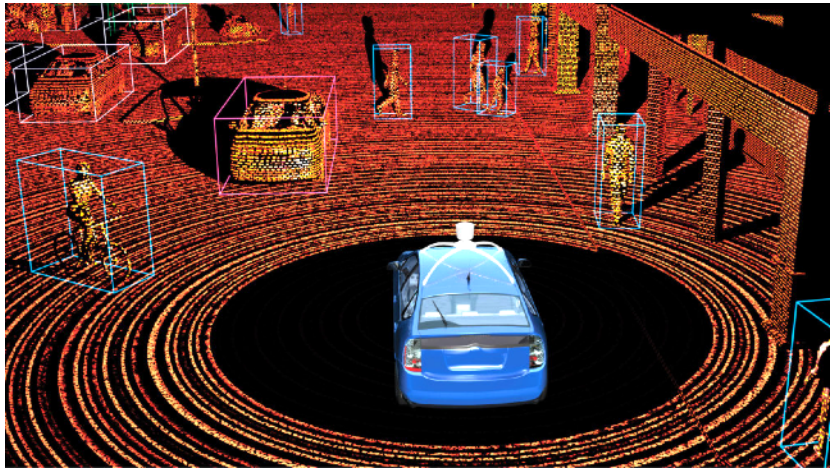





Charles R. Qi*
Hao Su*
Kaichun Mo
Leonidas J. Guibas

Stanford University

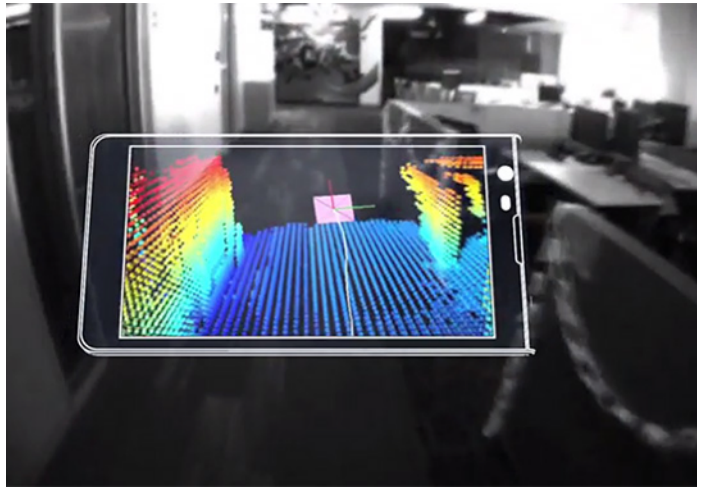
Big Data + Deep Representation Learning

Robot Perception



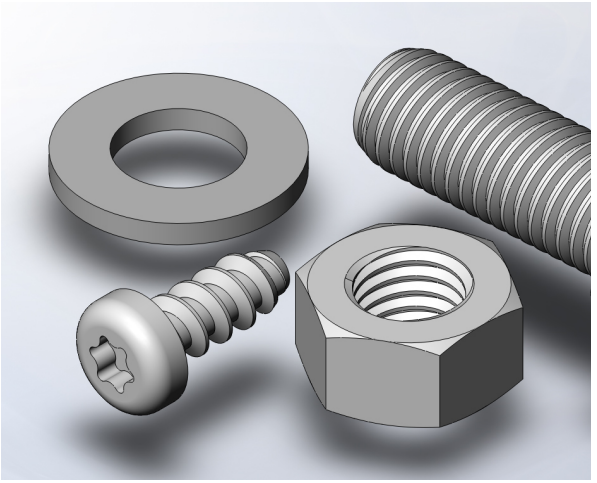
source: Scott J Grunewald

Augmented Reality



source: Google Tango

Shape Design

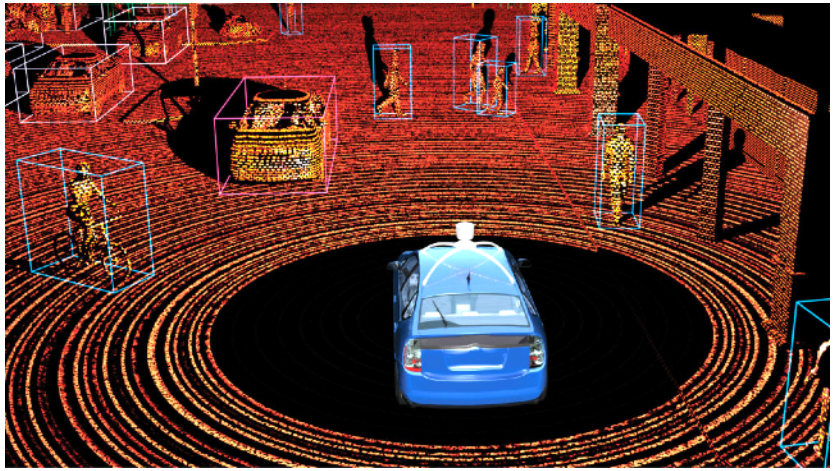


source: solidsolutions

Emerging 3D Applications

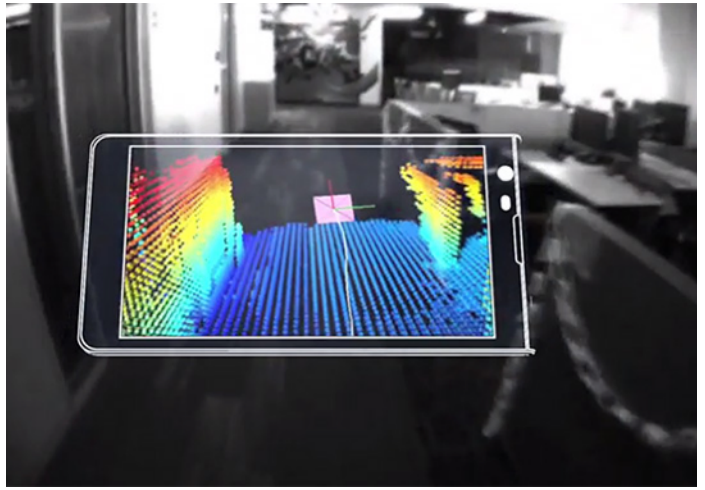
Big Data + Deep Representation Learning

Robot Perception



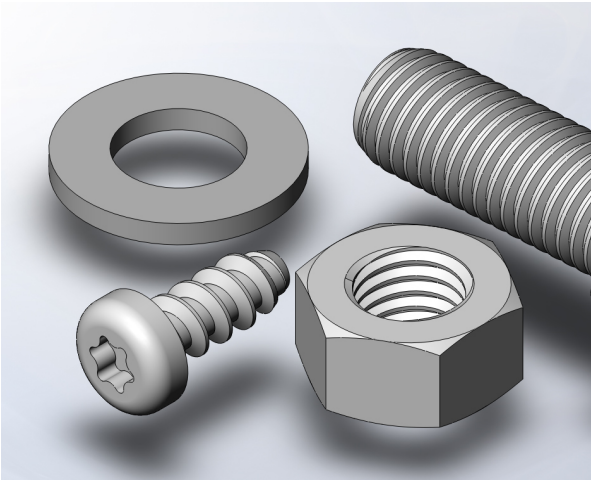
source: Scott J Grunewald

Augmented Reality



source: Google Tango

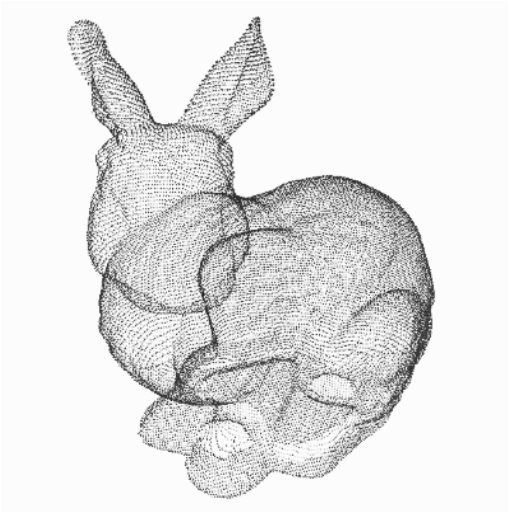
Shape Design



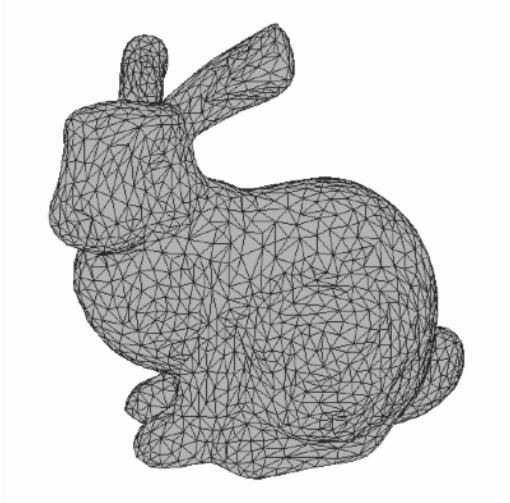
source: solidsolutions

Need for 3D Deep Learning!

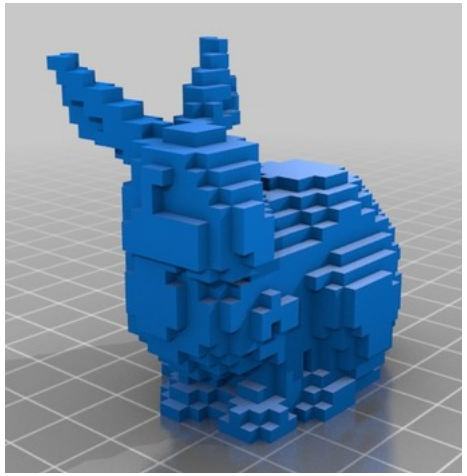
3D Representations



Point Cloud



Mesh



Volumetric

Projected View
RGB(D)

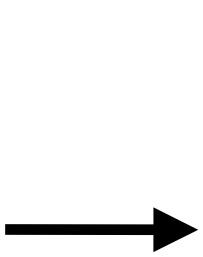
...

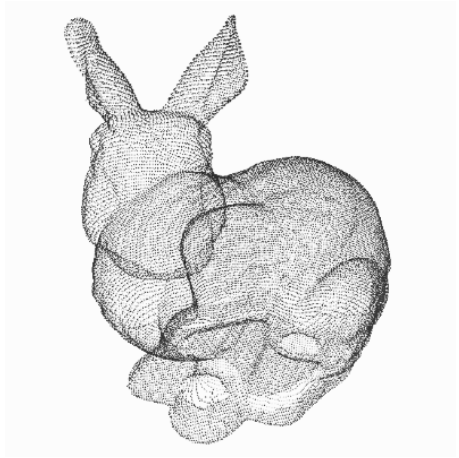
3D Representation: Point Cloud

Point cloud is close to raw sensor data

LiDAR

Depth Sensor





Point Cloud

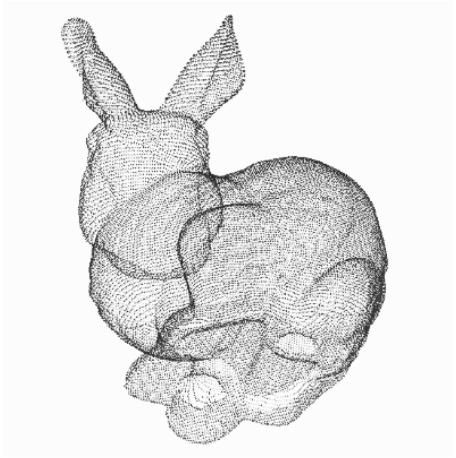
3D Representation: Point Cloud

Point cloud is close to raw sensor data

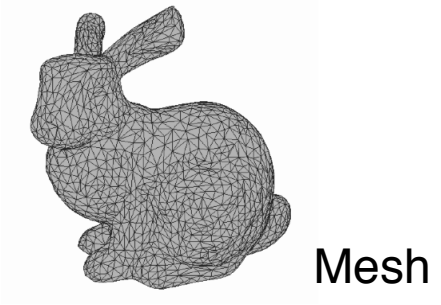
Point cloud is canonical

LiDAR

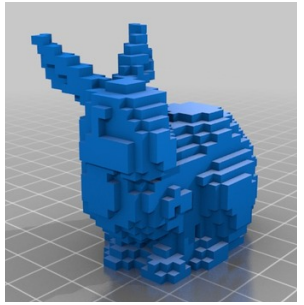
Depth Sensor



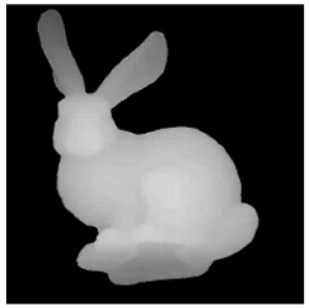
Point Cloud



Mesh



Volumetric



Depth Map

Previous Works

Most existing point cloud features are **handcrafted** towards specific tasks

Feature Name	Supports Texture / Color	Local / Global / Regional	Best Use Case
PFH	No	L	
FPFH	No	L	2.5D Scans (Pseudo single position range images)
VFH	No	G	Object detection with basic pose estimation
CVFH	No	R	Object detection with basic pose estimation, detection of partial objects
RIFT	Yes	L	Real world 3D-Scans with no mirror effects. RIFT is vulnerable against flipping.

Source: https://github.com/PointCloudLibrary/pcl/wiki/Overview-and-Comparison-of-Features

Previous Works

Point cloud is **converted to other representations** before it's fed to a deep neural network

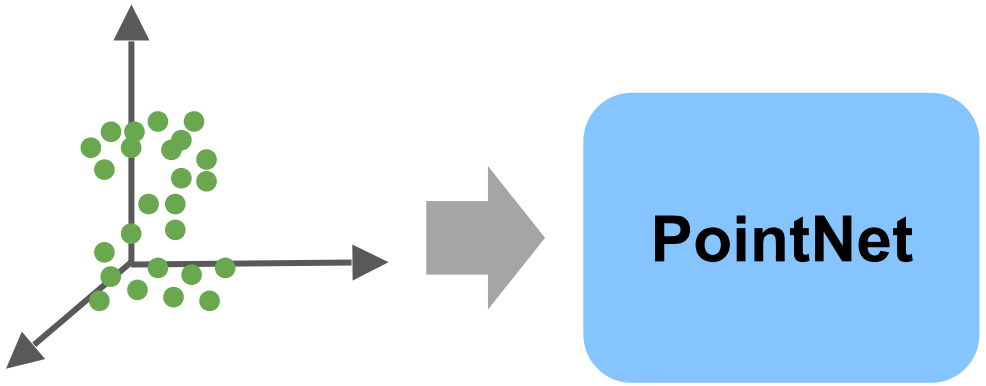
Conversion	Deep Net
Voxelization	3D CNN
Projection/Rendering	2D CNN
Feature extraction	Fully Connected

Research Question:

Can we achieve effective **feature learning directly** on point clouds?

Our Work: PointNet

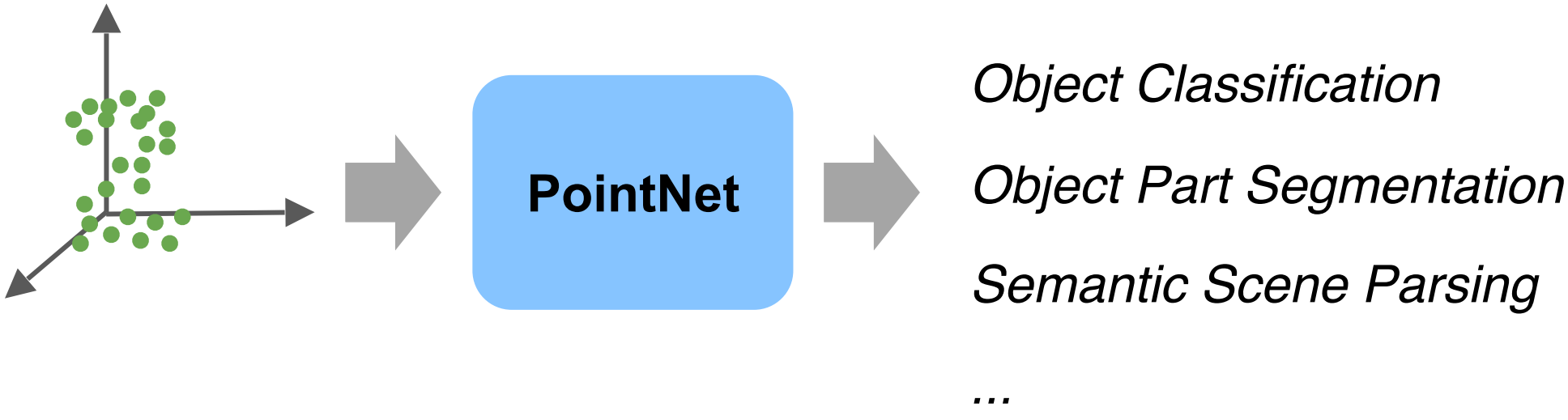
End-to-end learning for **scattered, unordered** point data



Our Work: PointNet

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

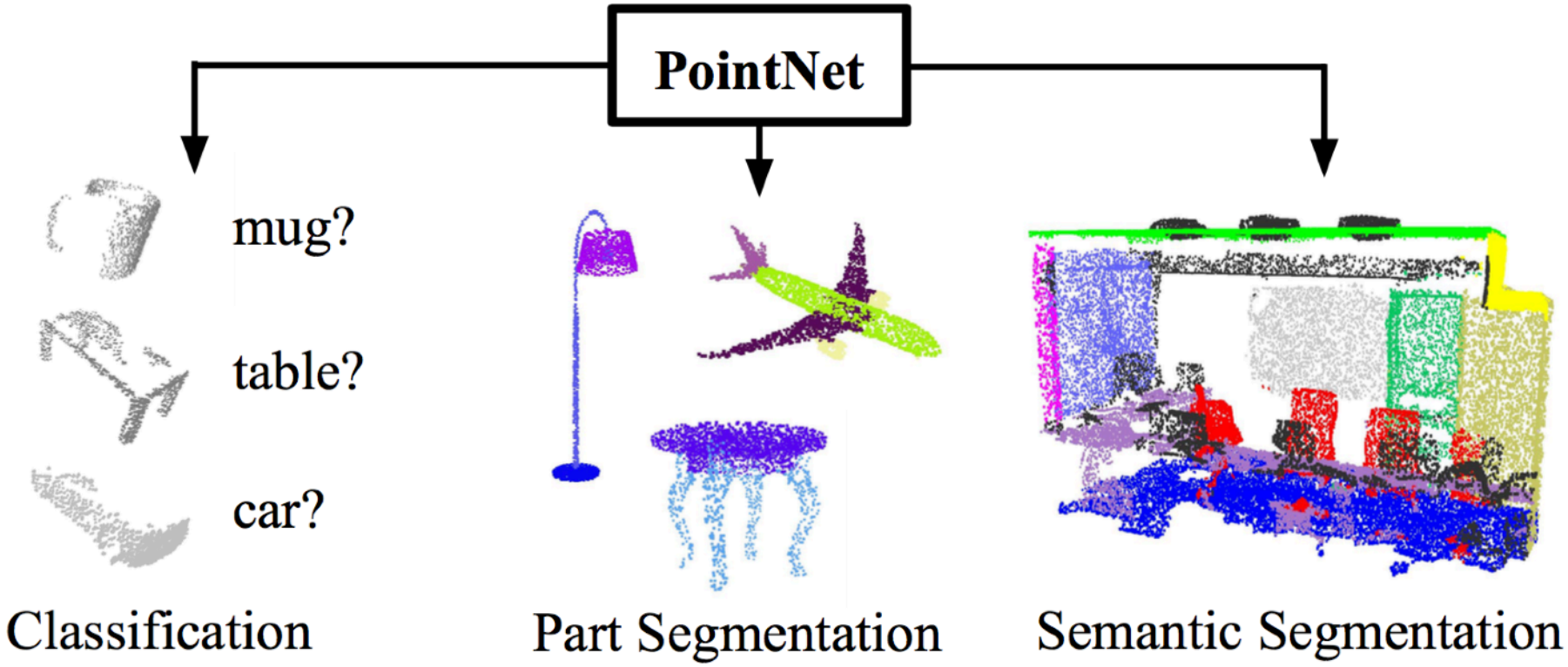
Unified framework for various tasks



Our Work: PointNet

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

Unified framework for various tasks



Challenges

Unordered point set as input

Model needs to be invariant to N! permutations.

Invariance under geometric transformations

Point cloud rotations should not alter classification results.

Challenges

Unordered point set as input

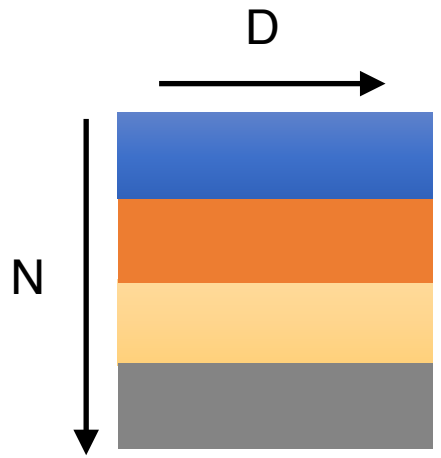
Model needs to be invariant to N! permutations.

Invariance under geometric transformations

Point cloud rotations should not alter classification results.

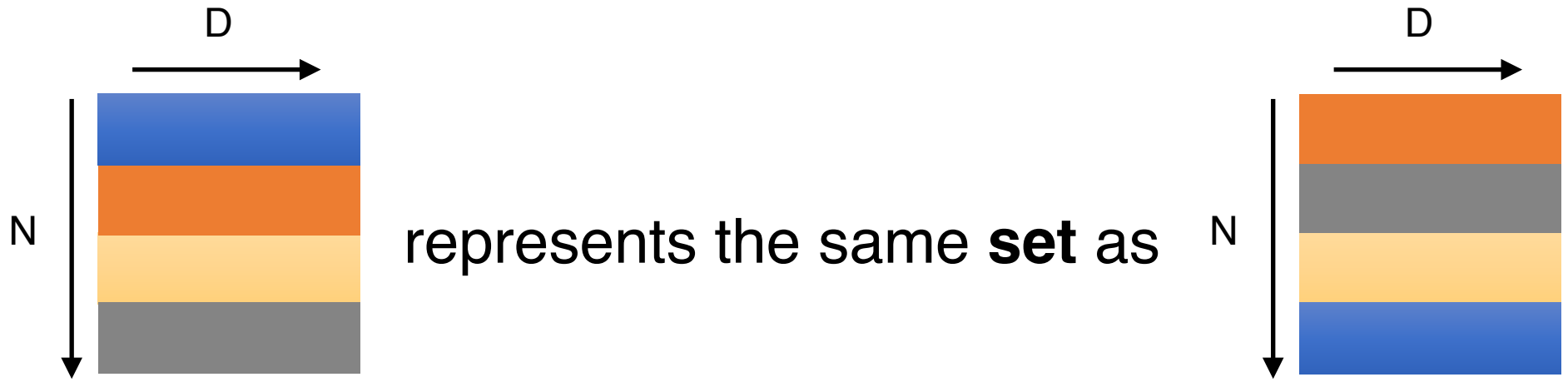
Unordered Input

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



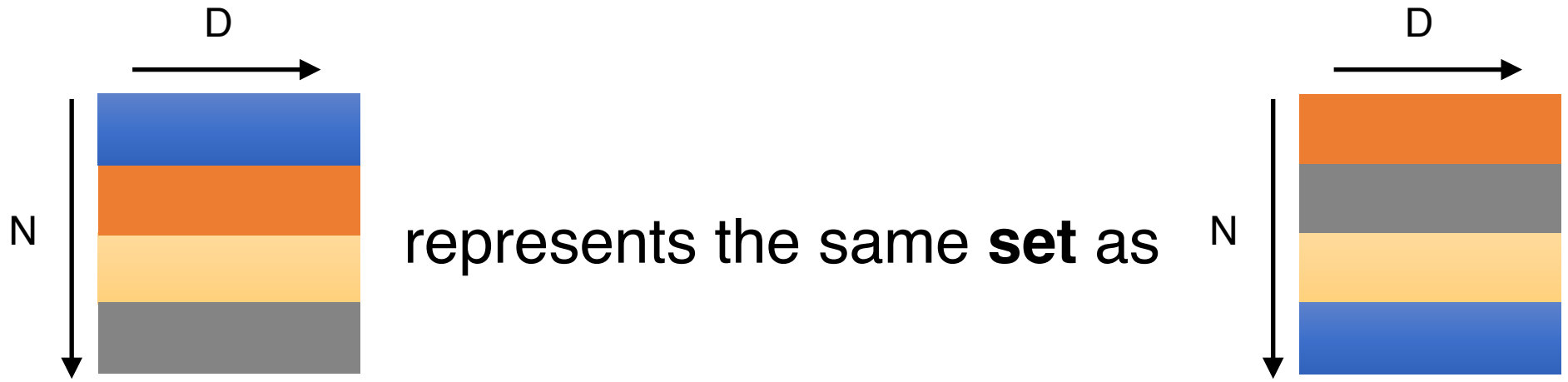
Unordered Input

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



Unordered Input

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



Model needs to be invariant to N! permutations

Permutation Invariance: Symmetric Function

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Permutation Invariance: Symmetric Function

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \ldots, x_n) = \max\{x_1, x_2, \ldots, x_n\}$$

$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

…

Permutation Invariance: Symmetric Function

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

$$f(x_1, x_2, \ldots, x_n) = \max\{x_1, x_2, \ldots, x_n\}$$

$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

…

How can we construct a family of symmetric functions by neural networks?

Permutation Invariance: Symmetric Function

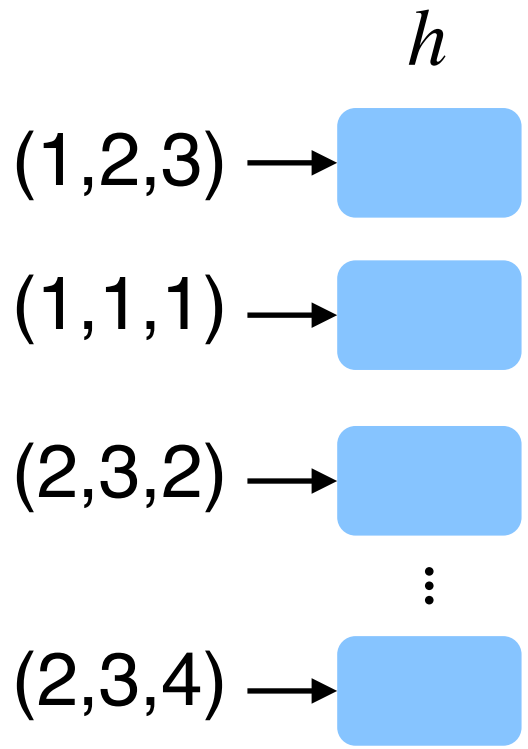
Observe:

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

Permutation Invariance: Symmetric Function

Observe:

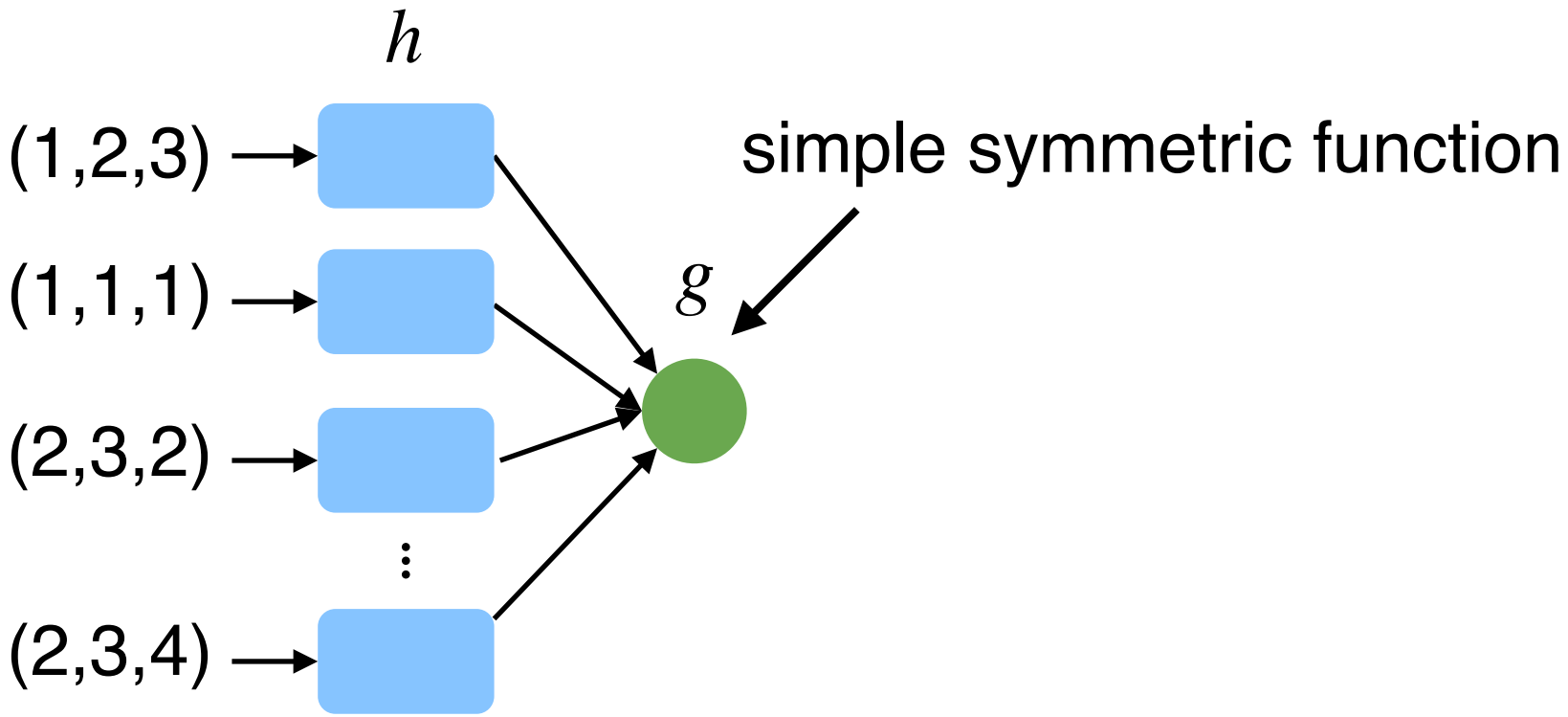
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Permutation Invariance: Symmetric Function

Observe:

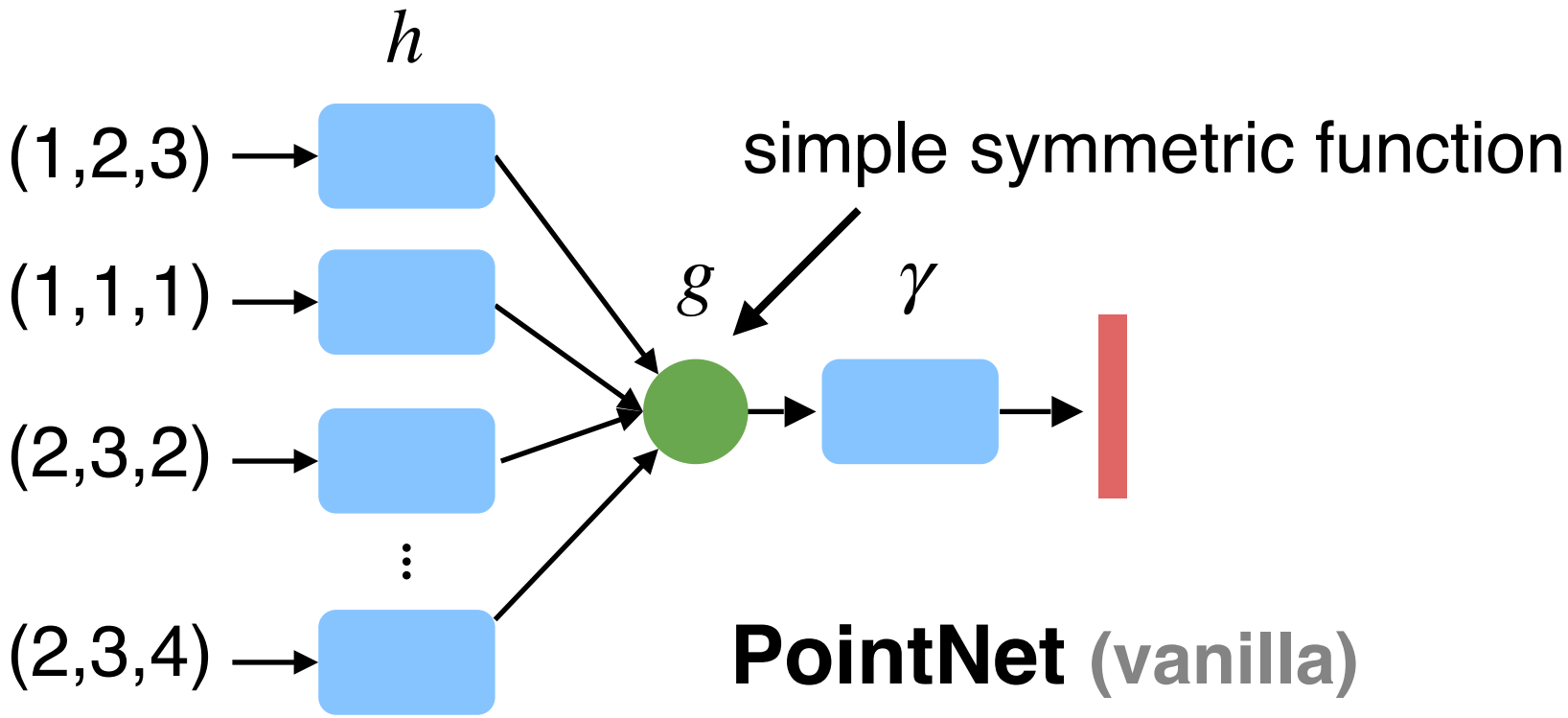
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Permutation Invariance: Symmetric Function

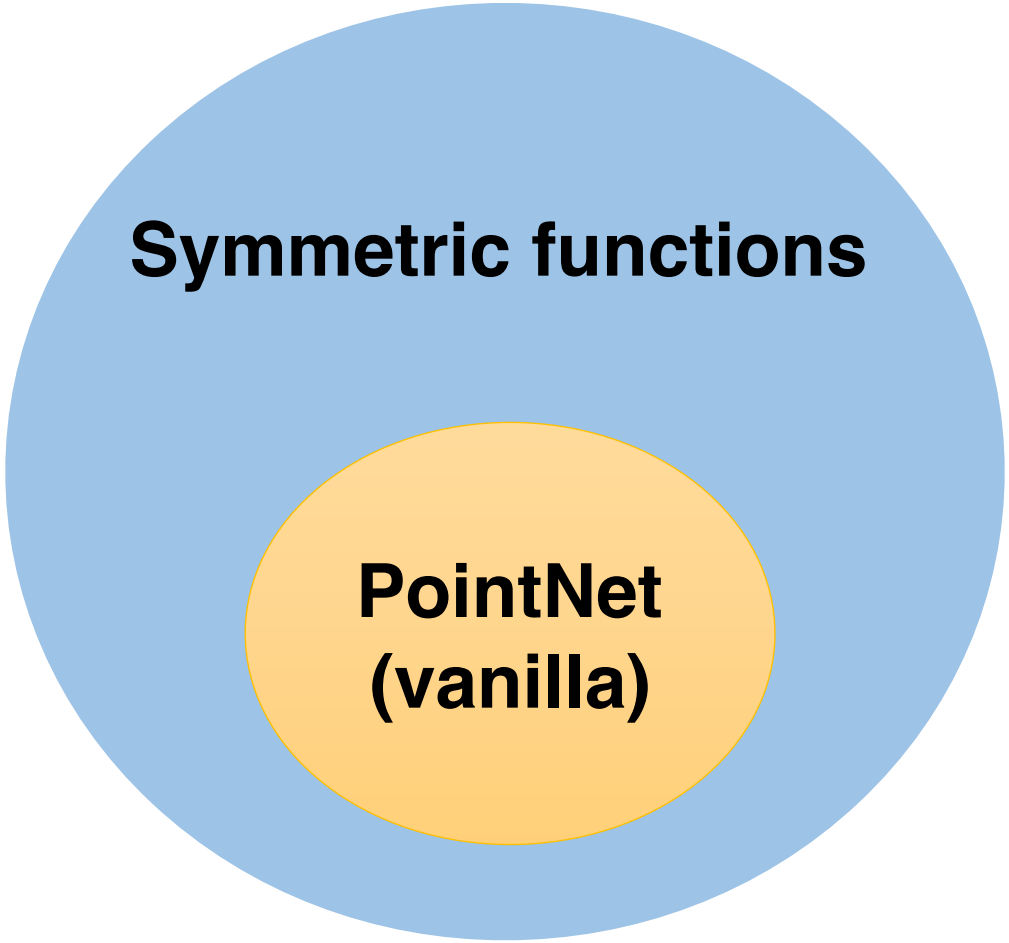
Observe:

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



Permutation Invariance: Symmetric Function

What symmetric functions can be constructed by PointNet?



Universal Set Function Approximator

Theorem:

A Hausdorff continuous symmetric function $f : 2^{\mathcal{X}} \to \mathbb{R}$ can be arbitrarily approximated by PointNet.

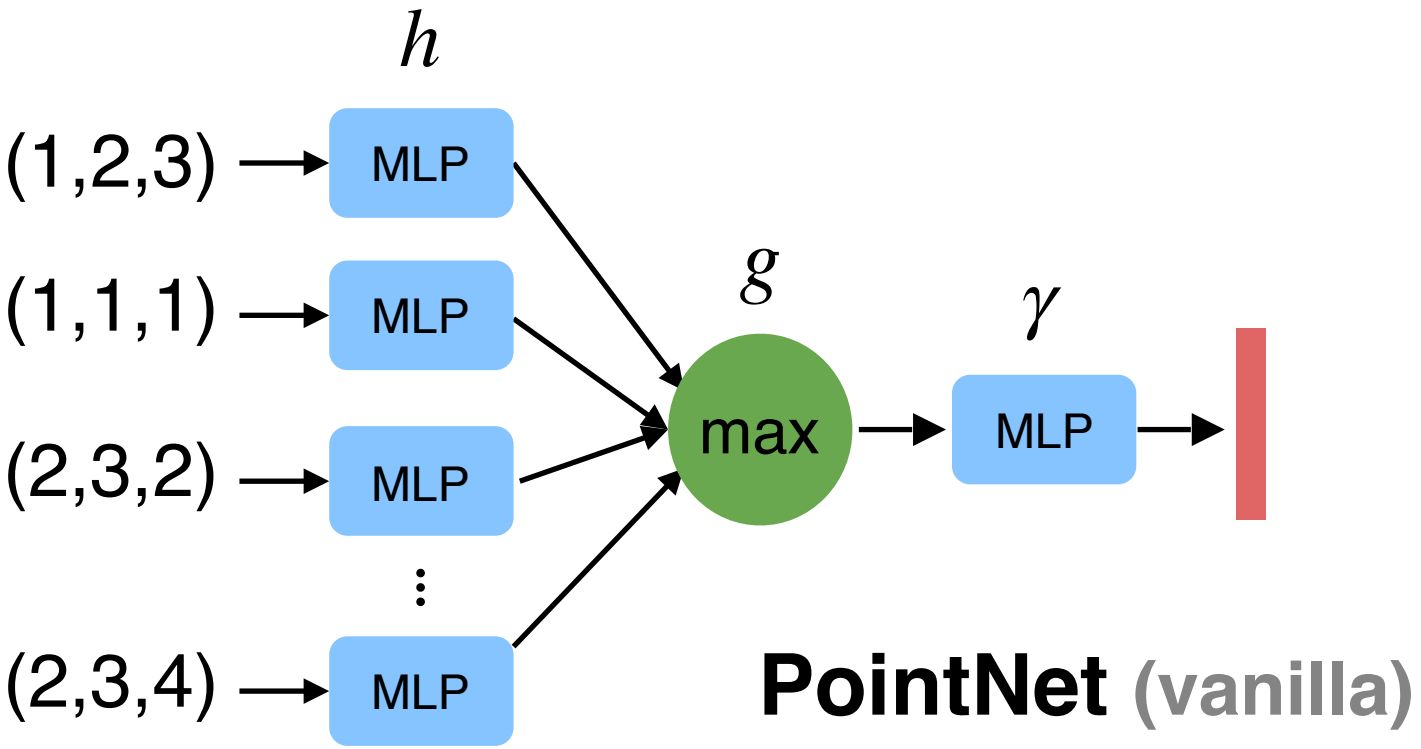
$$\left| f(S) - \gamma \left(\underset{x_i \in S}{\mathrm{MAX}} \{ h(x_i) \} \right) \right| < \epsilon$$

$S \subseteq \mathbb{R}^d$

PointNet (vanilla)

Basic PointNet Architecture

Empirically, we use **multi-layer perceptron (MLP)** and **max pooling**:



Challenges

Unordered point set as input

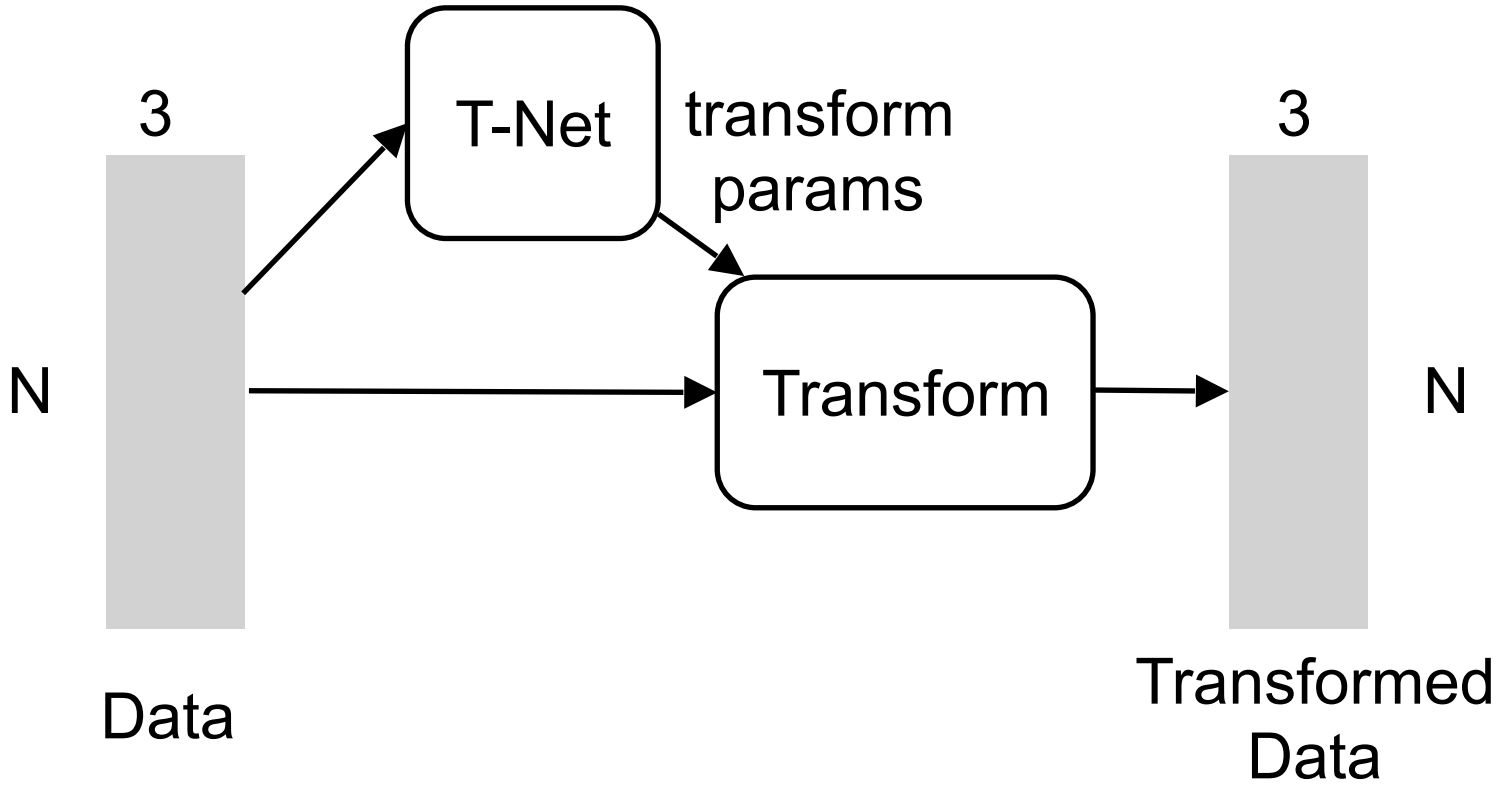
Model needs to be invariant to N! permutations.

Invariance under geometric transformations

Point cloud rotations should not alter classification results.

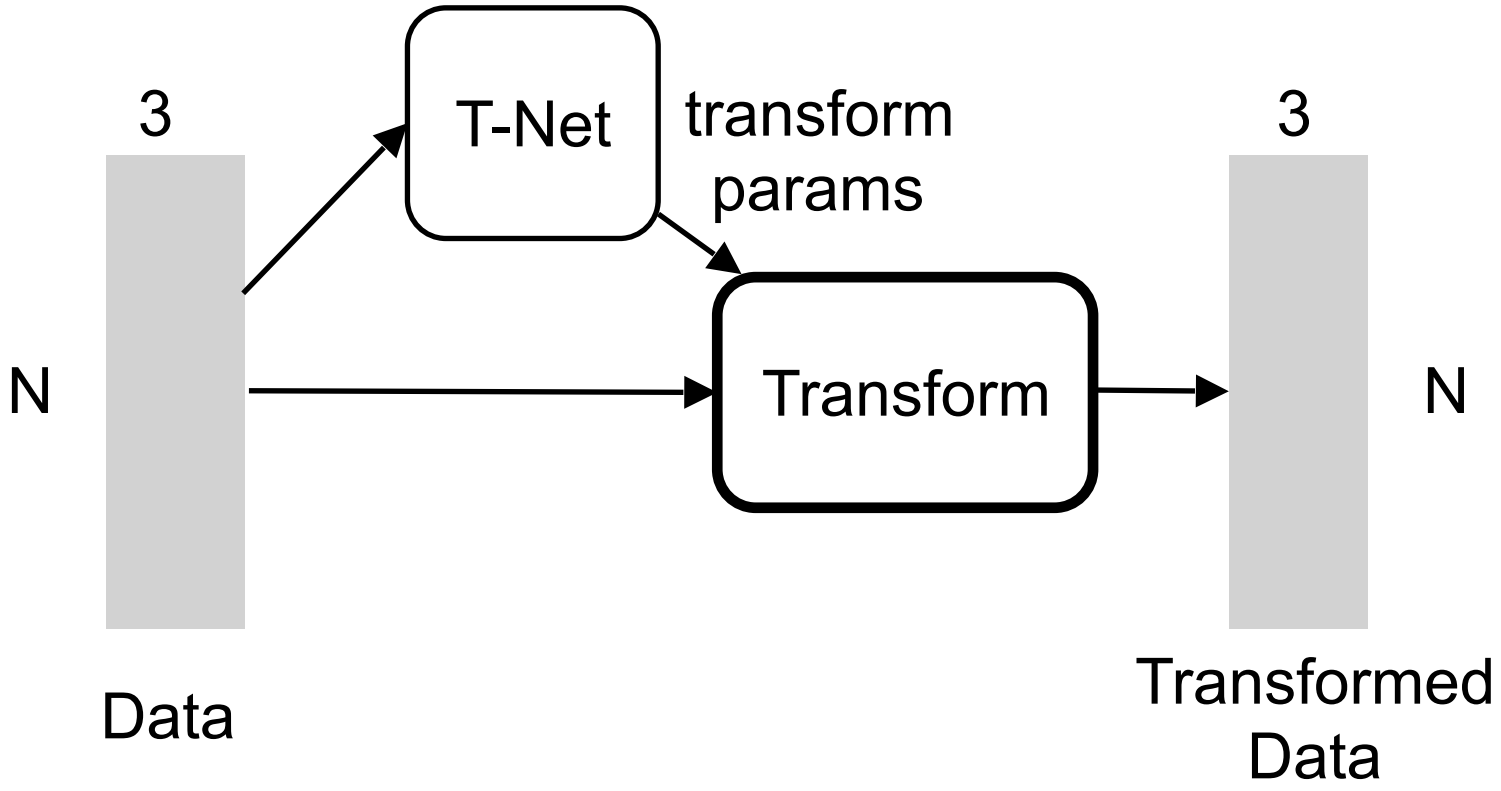
Input Alignment by Transformer Network

Idea: Data dependent transformation for automatic alignment



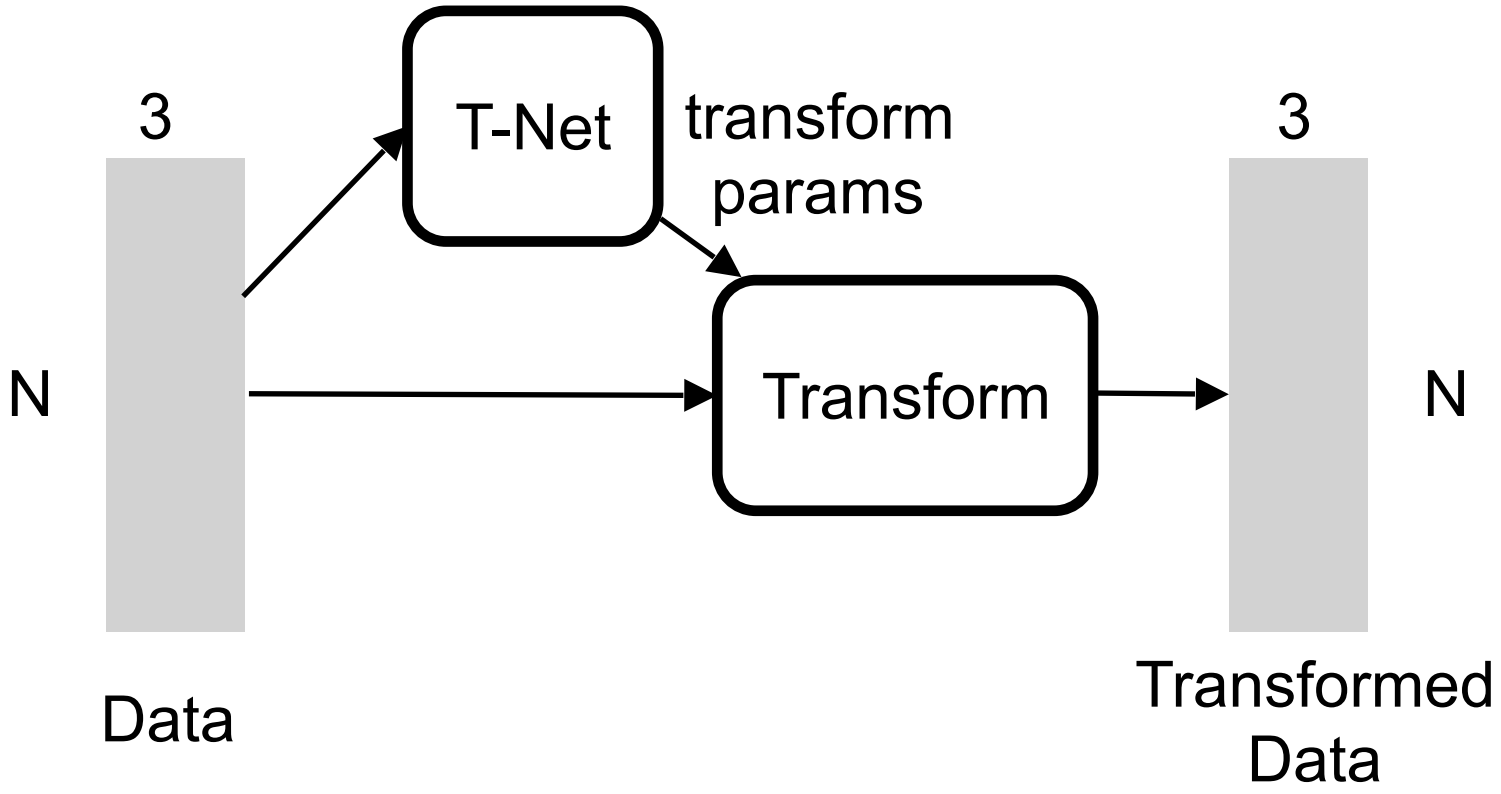
Input Alignment by Transformer Network

Idea: Data dependent transformation for automatic alignment



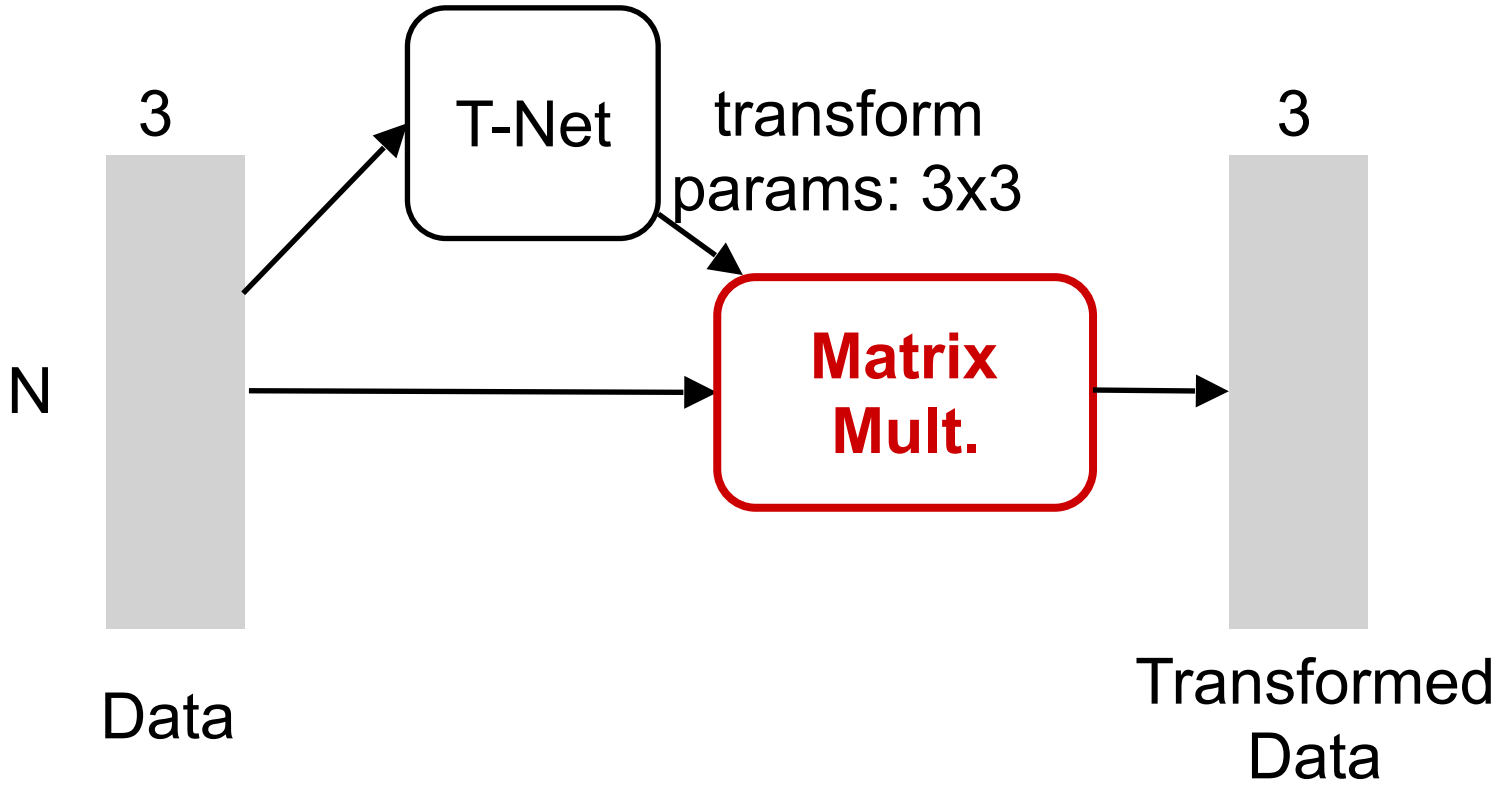
Input Alignment by Transformer Network

Idea: Data dependent transformation for automatic alignment

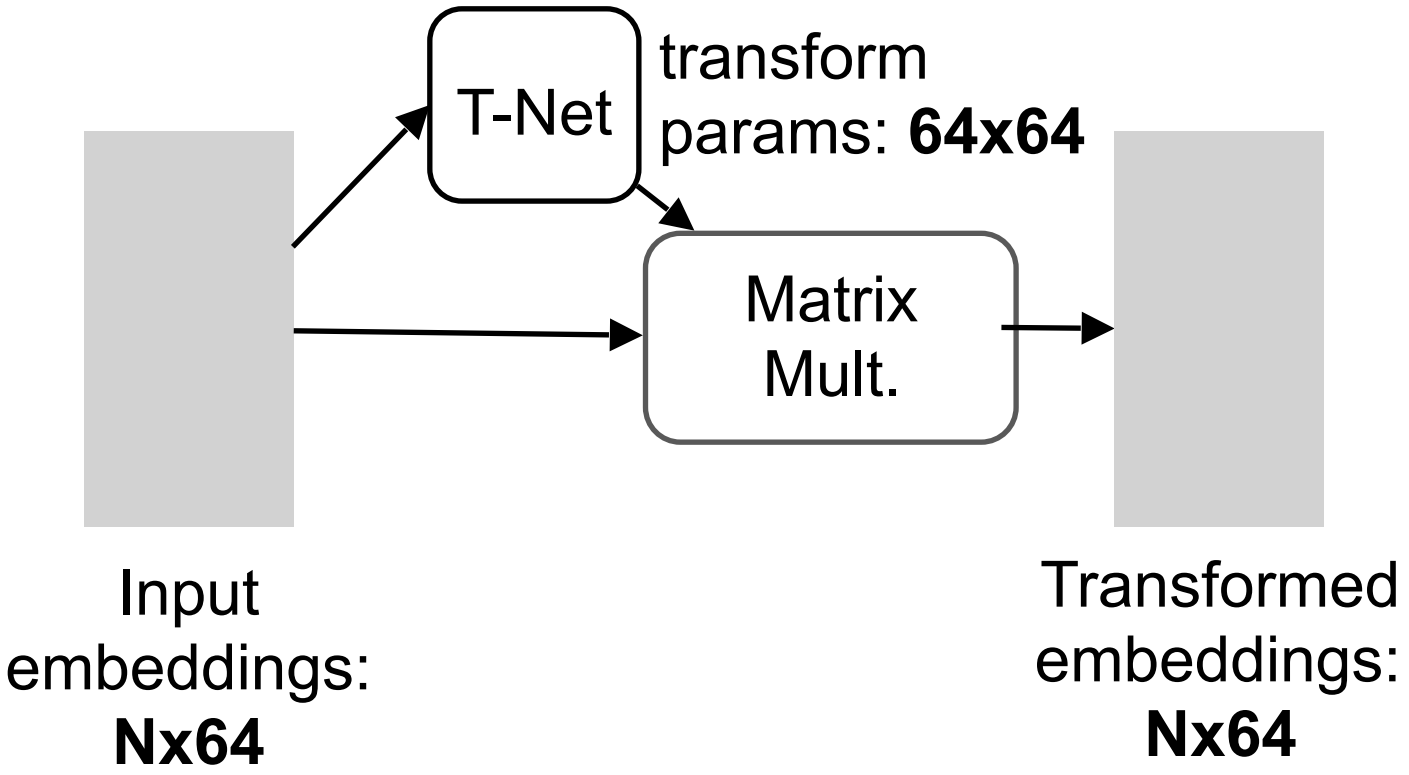


Input Alignment by Transformer Network

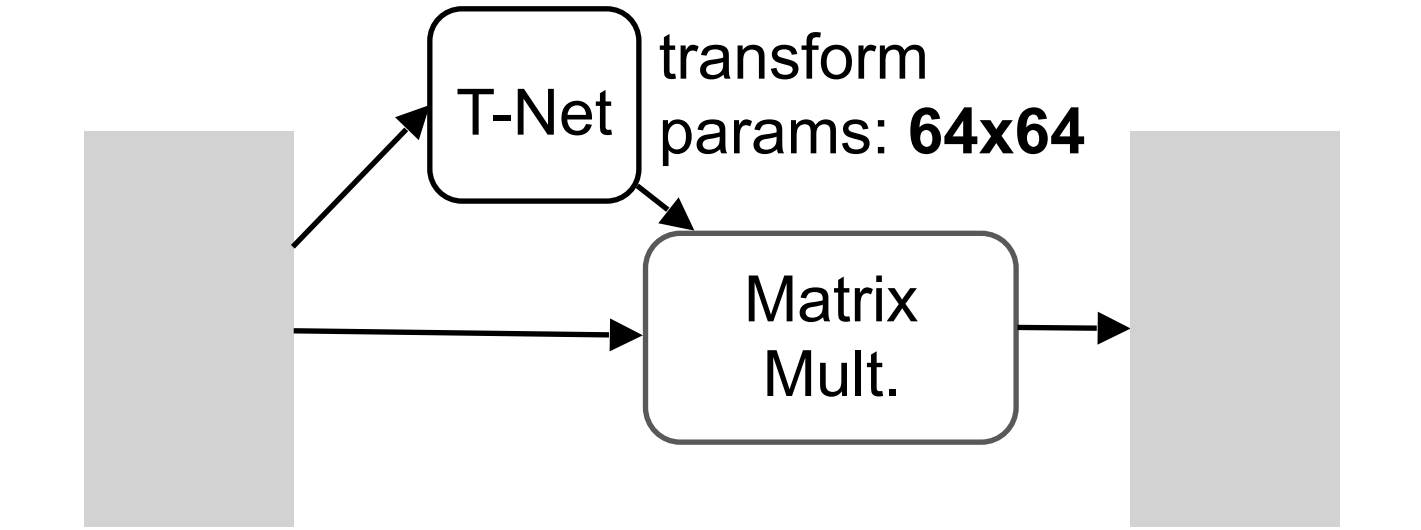
The transformation is just matrix multiplication!



Embedding Space Alignment



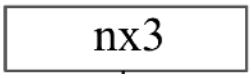
Embedding Space Alignment



Regularization:

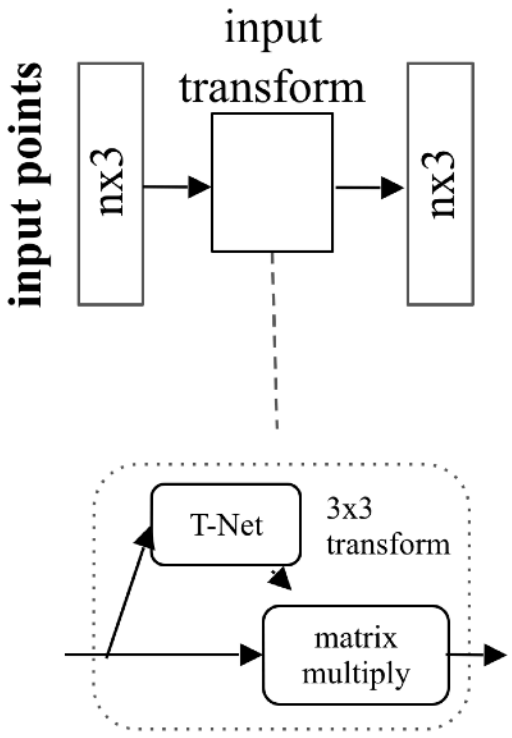
Transform matrix A 64x64 close to orthogonal:

$$L_{reg} = \|I - AA^T\|_F^2$$

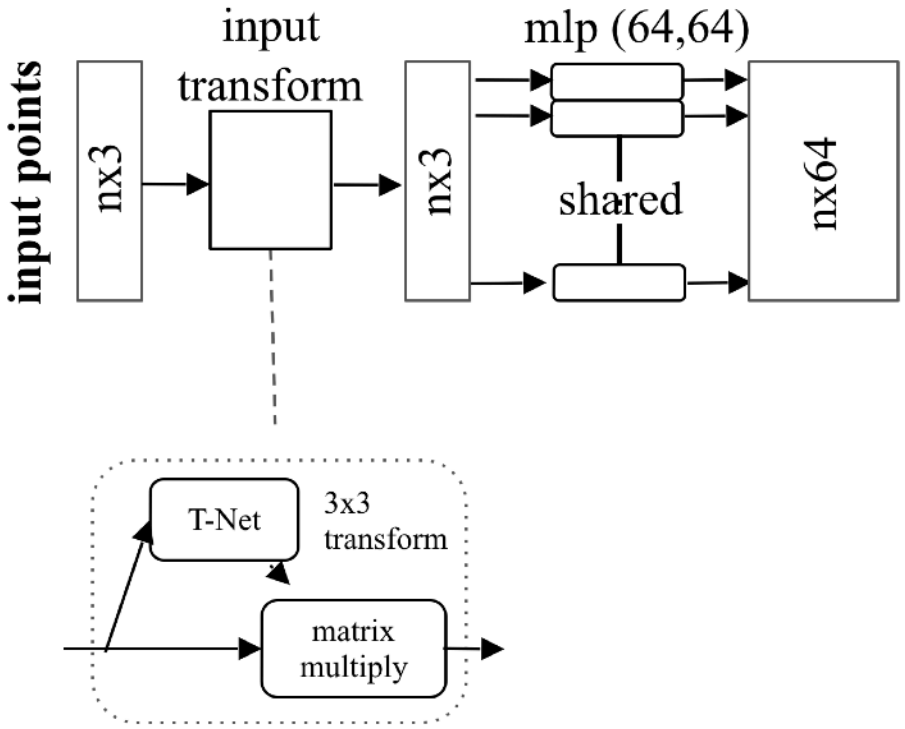


PointNet Classification Network

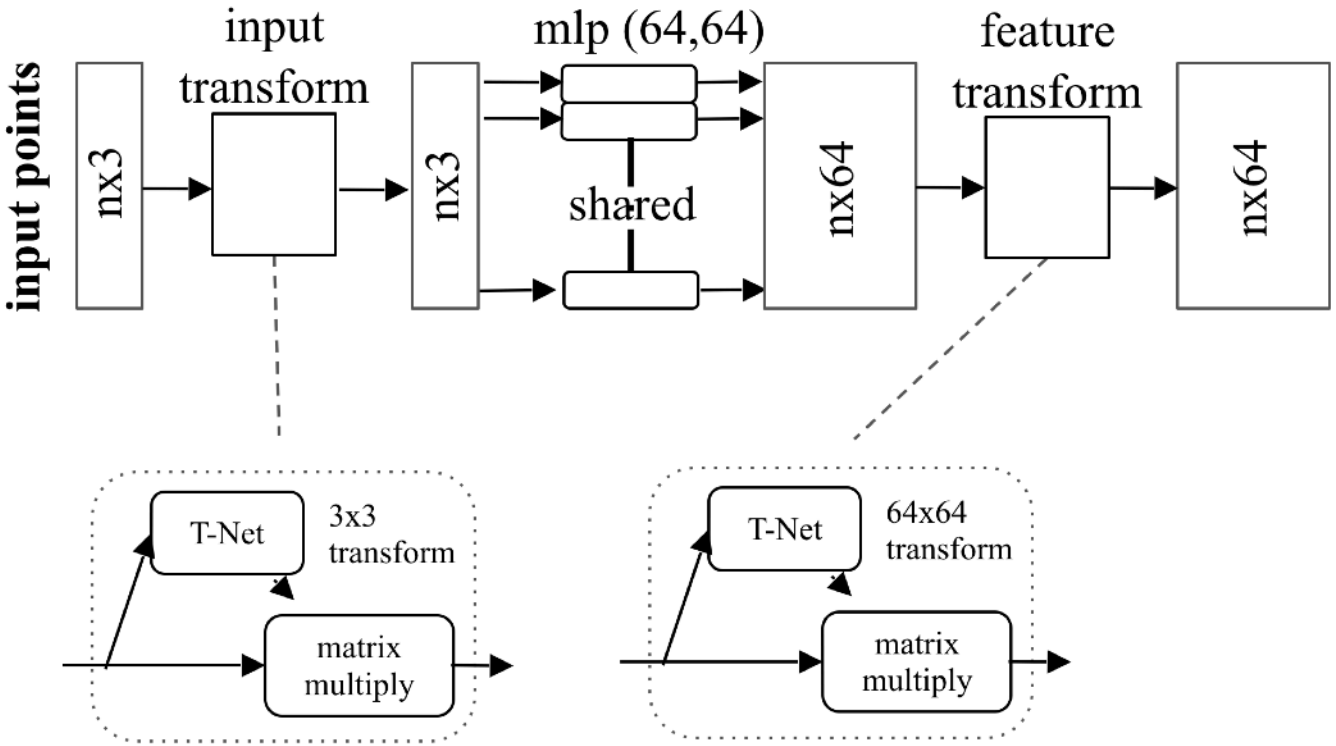
PointNet Classification Network



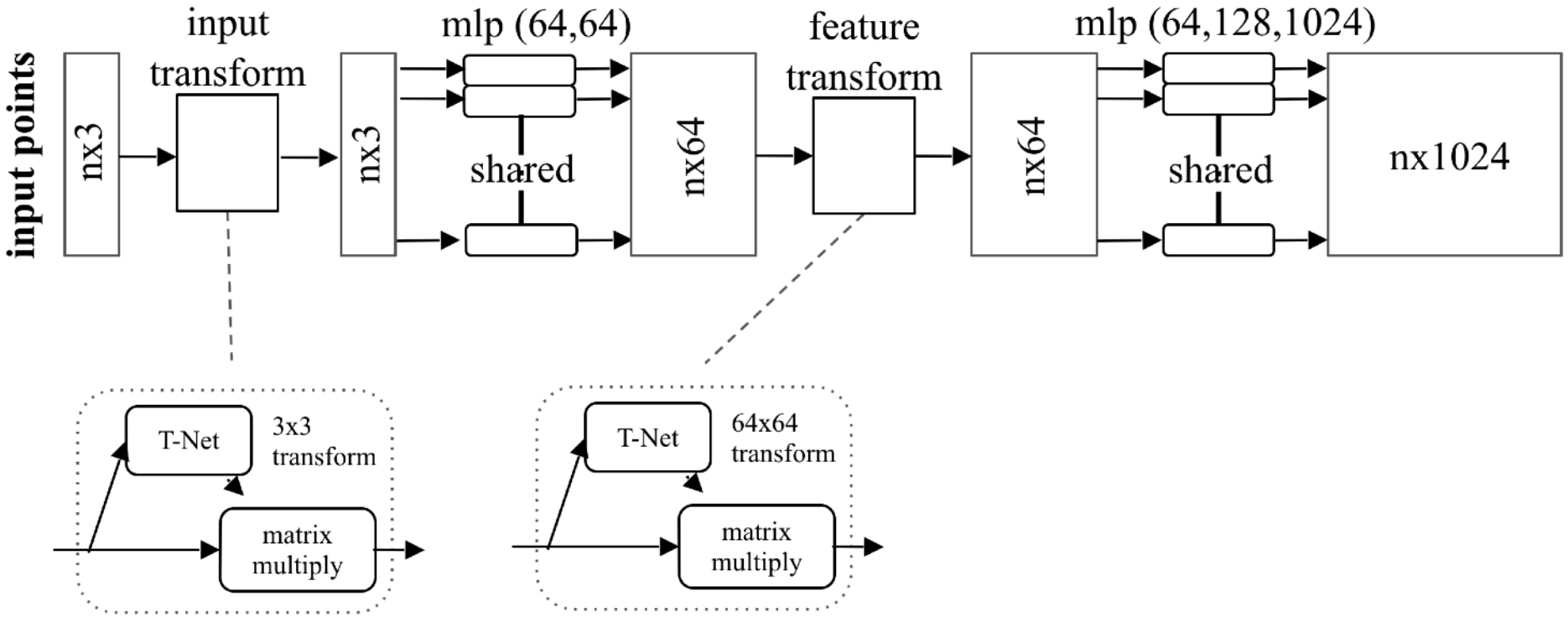
PointNet Classification Network



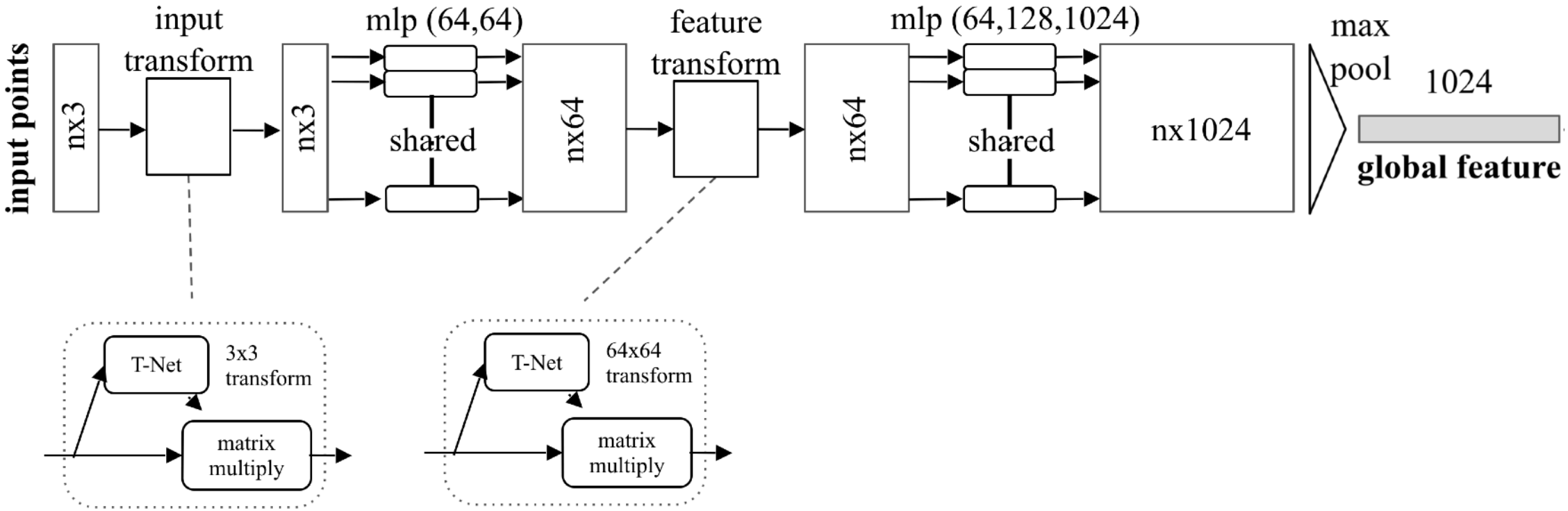
PointNet Classification Network



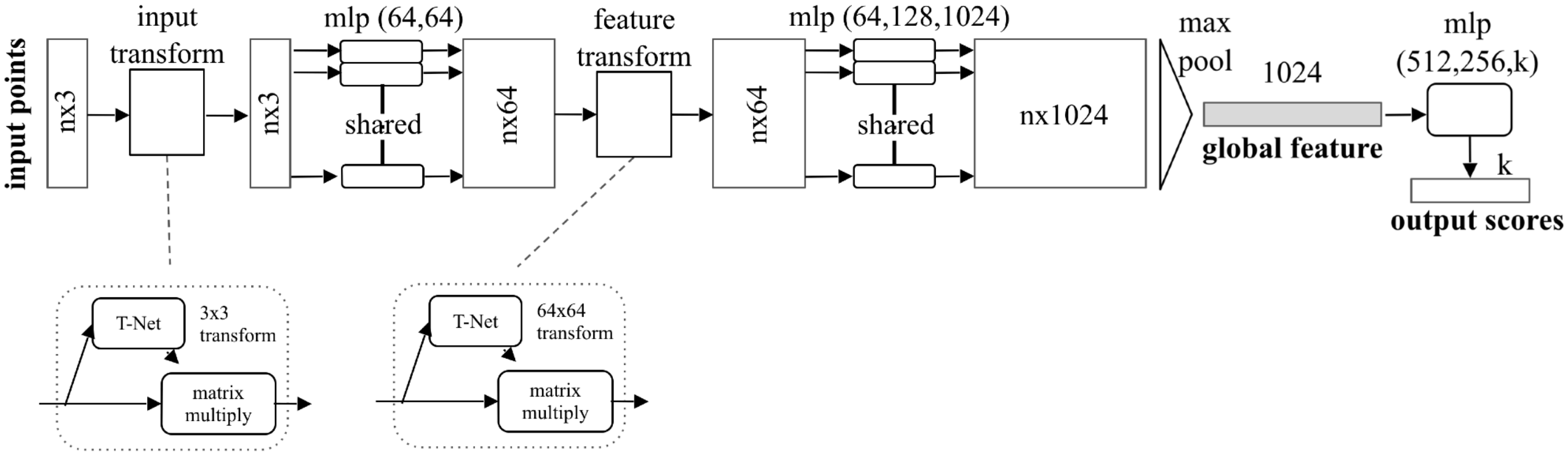
PointNet Classification Network



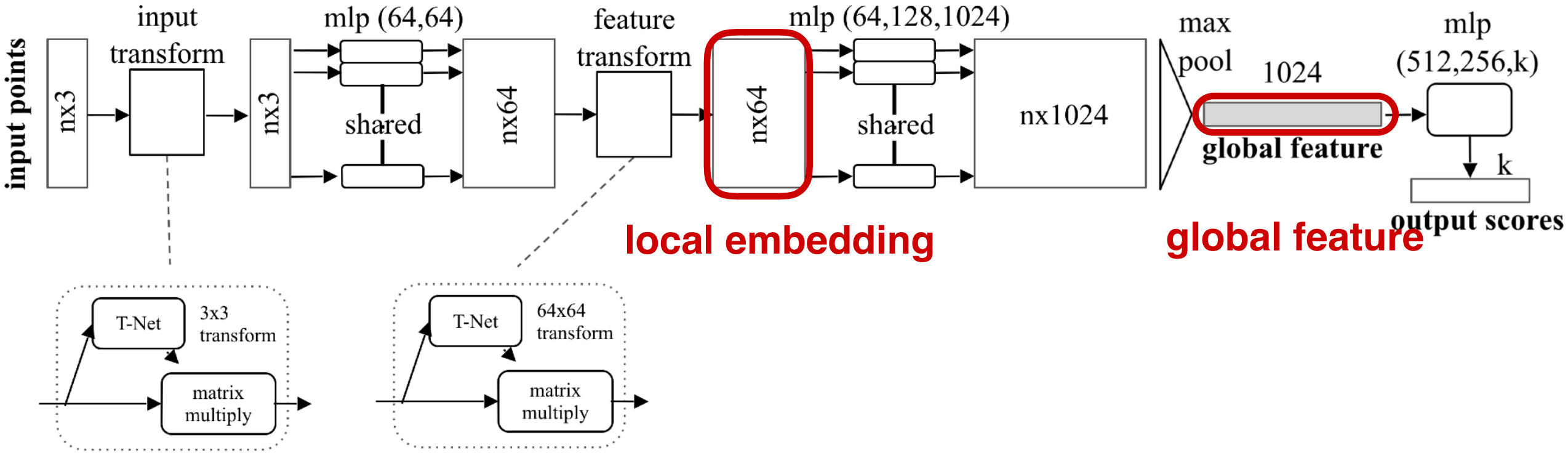
PointNet Classification Network



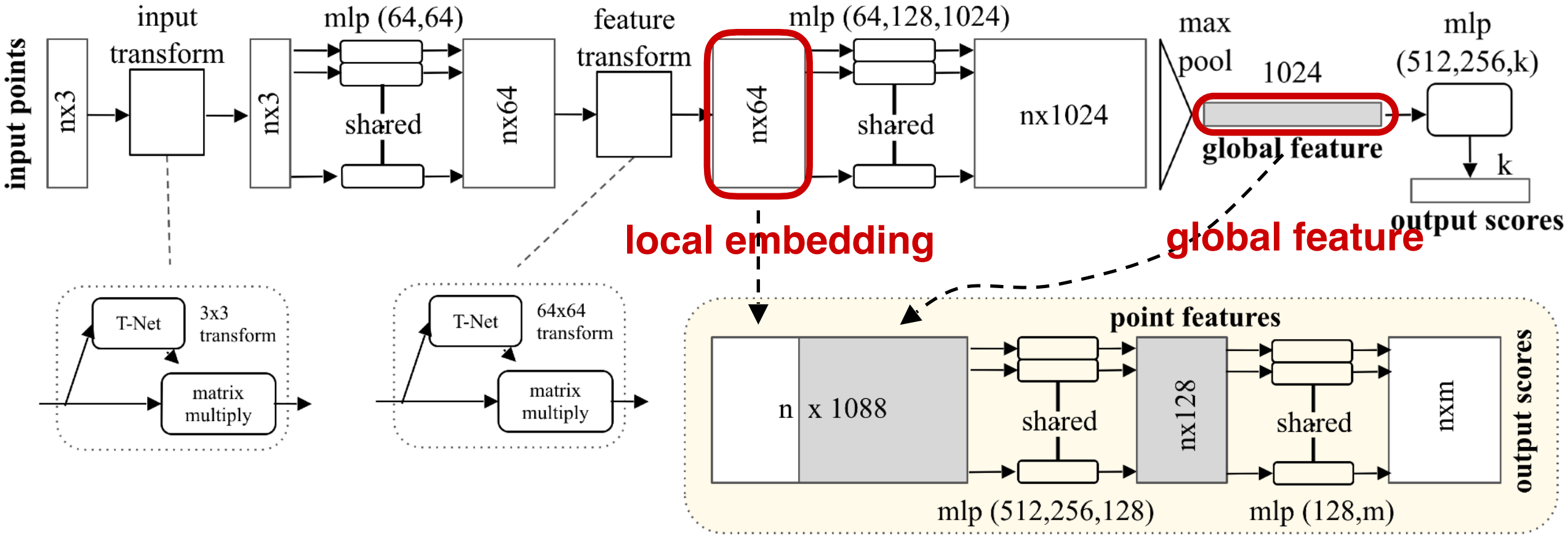
PointNet Classification Network



Extension to PointNet Segmentation Network



Extension to PointNet Segmentation Network



Results

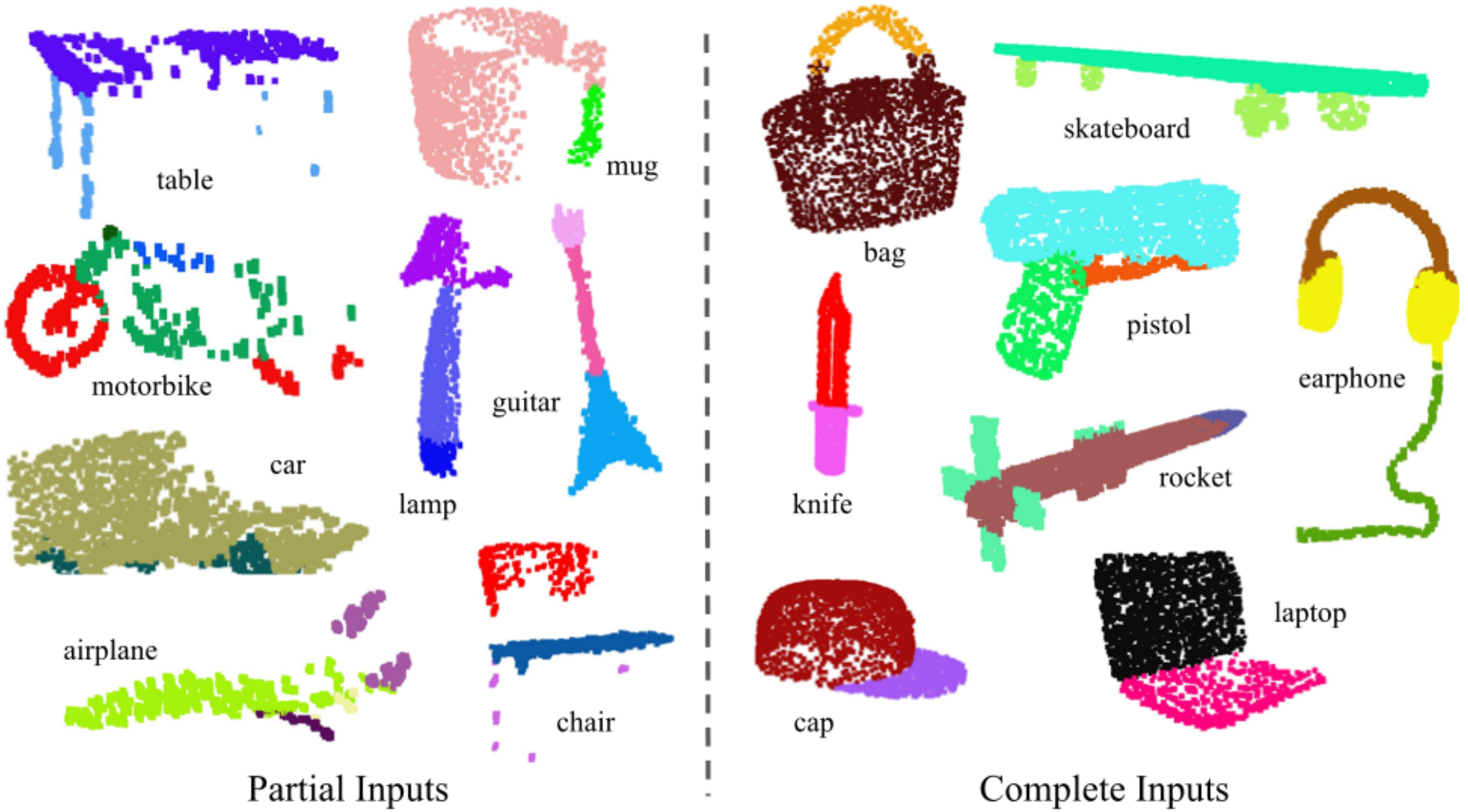
Results on Object Classification

	input	#views	accuracy avg. class	accuracy overall
SPH [12]	mesh	-	68.2	
3DShapeNets [29]	volume	1	77.3	84.7
VoxNet [18]	volume	12	83.0	85.9
Subvolume [19]	volume	20	86.0	**89.2**
LFD [29]	image	10	75.5	-
MVCNN [24]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

3D CNNs

dataset: ModelNet40; metric: 40-class classification accuracy (%)

Results on Object Part Segmentation



Results on Object Part Segmentation

	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 [28]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [30]	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**

dataset: ShapeNetPart; metric: mean IoU (%)

Results on Semantic Scene Parsing

dataset: Stanford 2D-3D-S (Matterport scans)

Robustness to Data Corruption

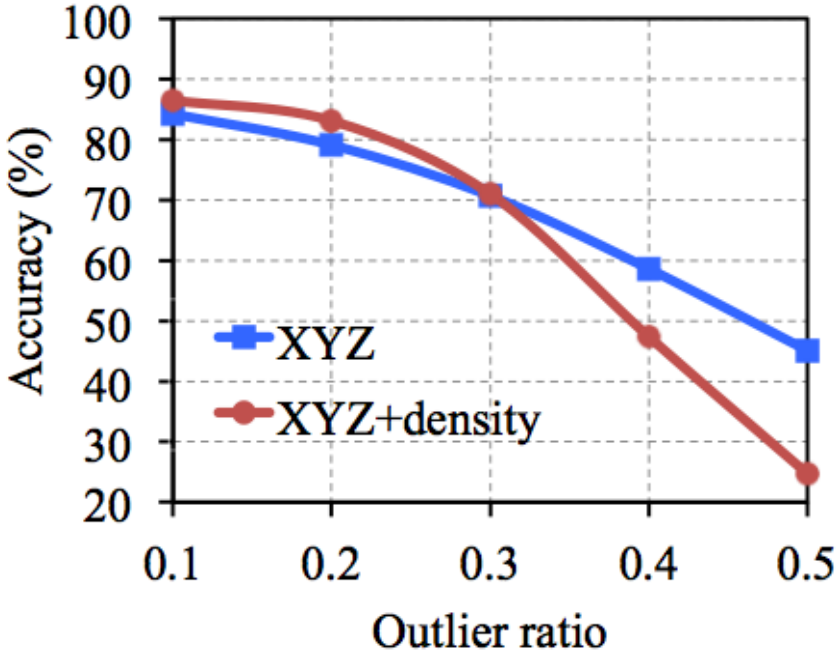
dataset: ModelNet40; metric: 40-class classification accuracy (%)

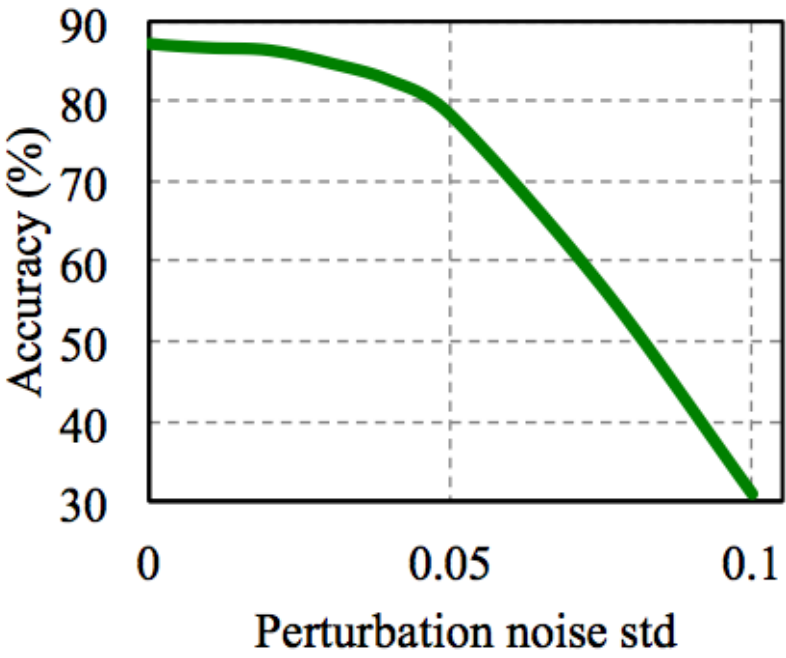
Robustness to Data Corruption

Less than 2% accuracy drop with 50% missing data

dataset: ModelNet40; metric: 40-class classification accuracy (%)

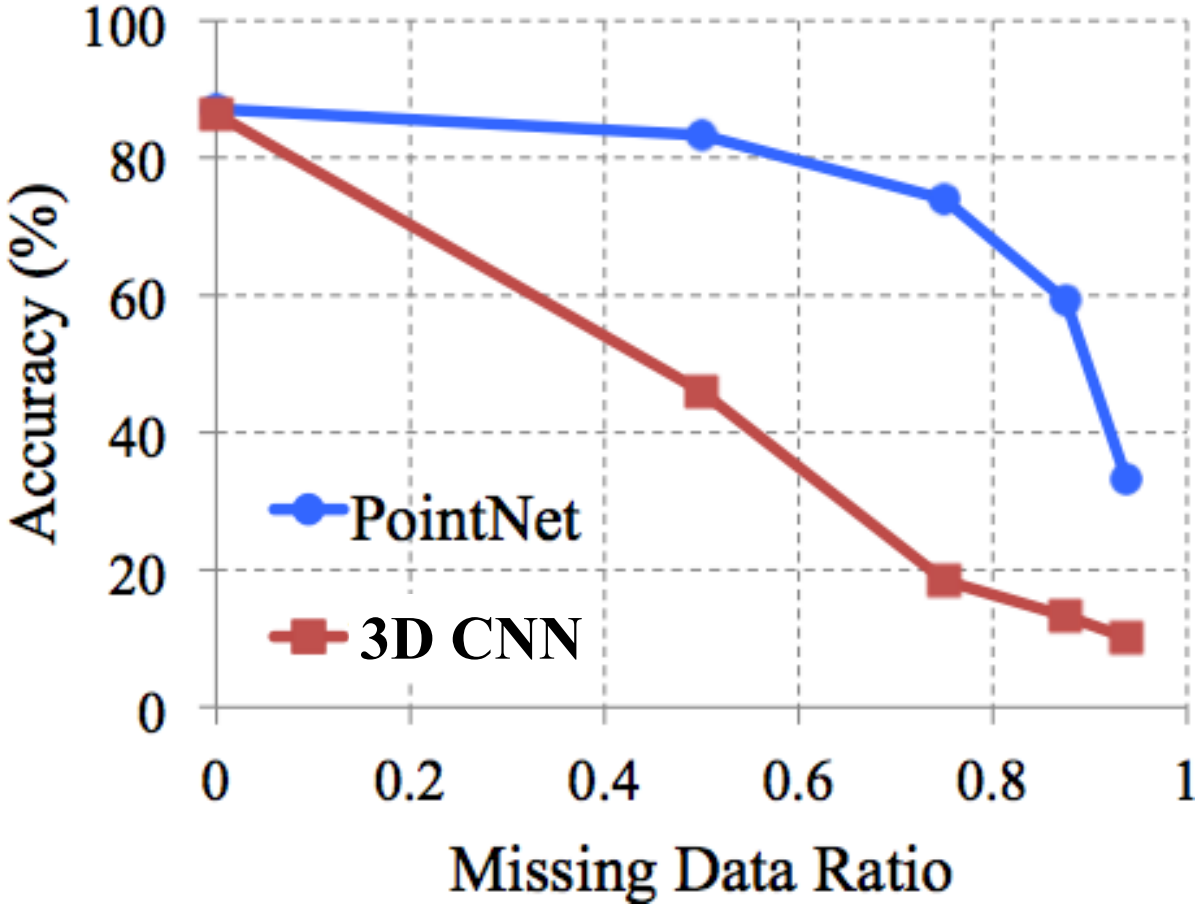
Robustness to Data Corruption





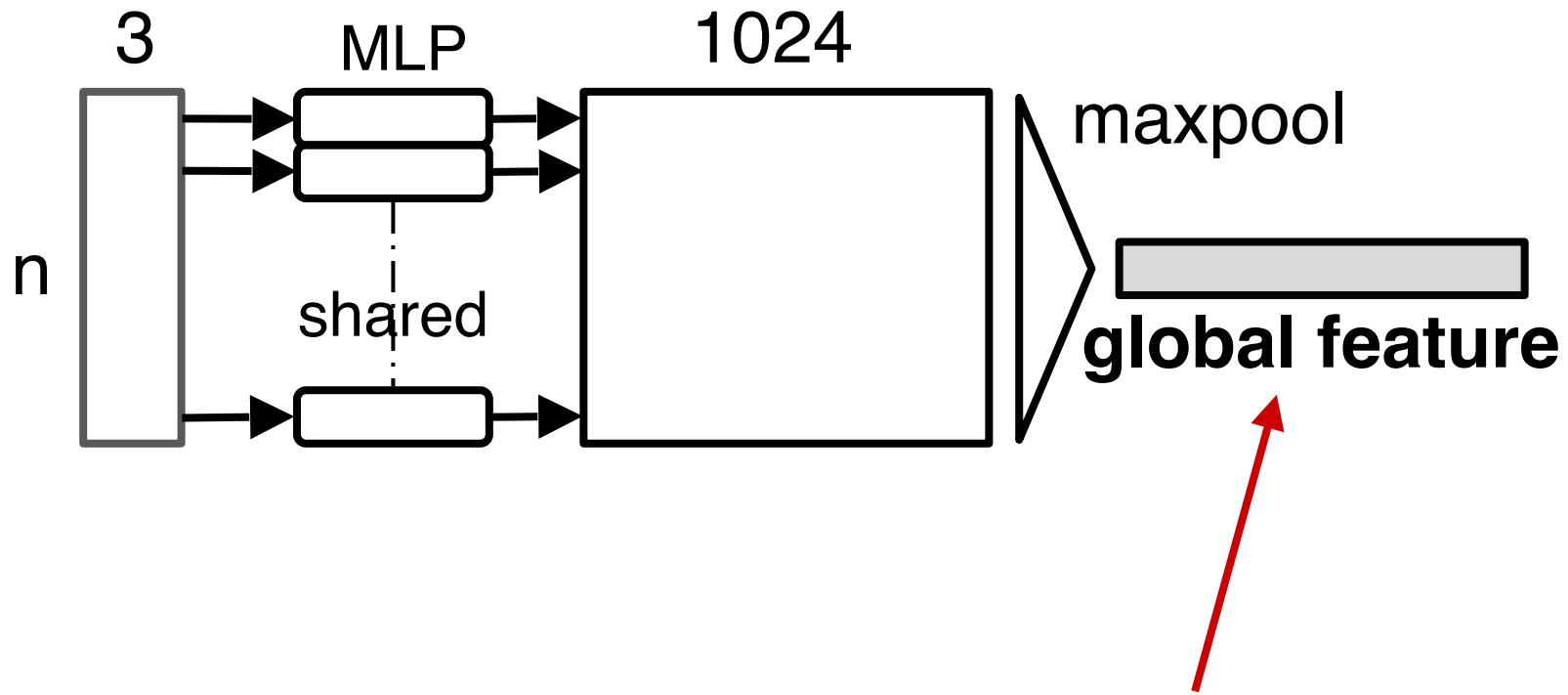
dataset: ModelNet40; metric: 40-class classification accuracy (%)

Robustness to Data Corruption



Why is PointNet so robust to missing data?

Visualizing Global Point Cloud Features

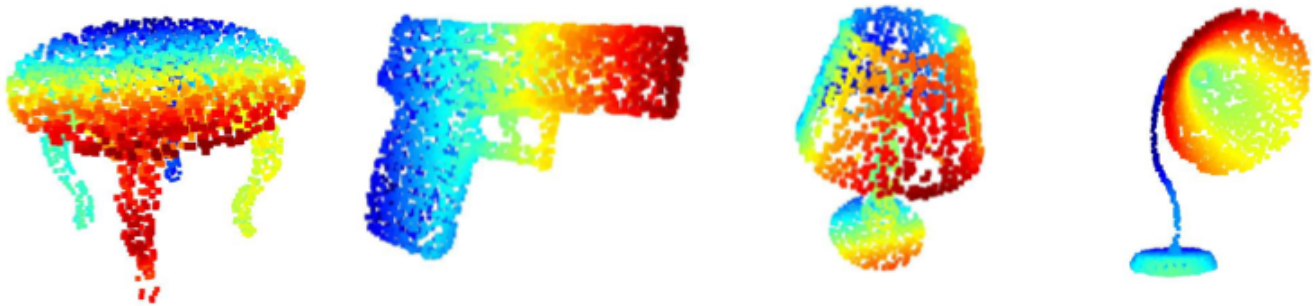


Which input points are contributing to the global feature?

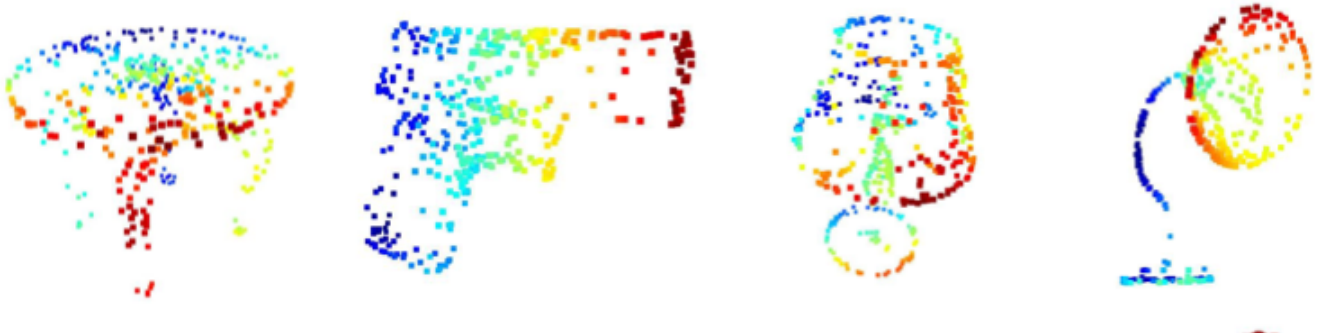
(critical points)

Visualizing Global Point Cloud Features

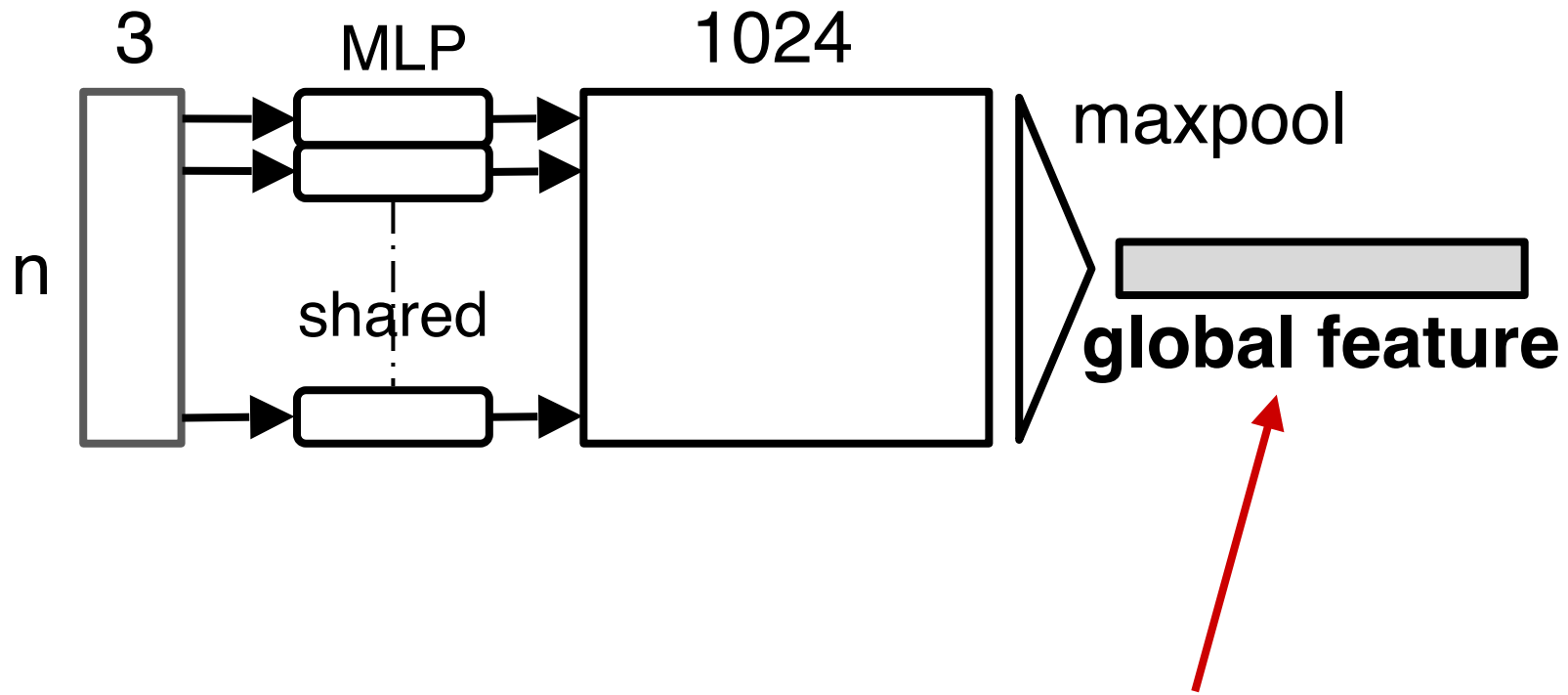
Original Shape:



Critical Point Sets:



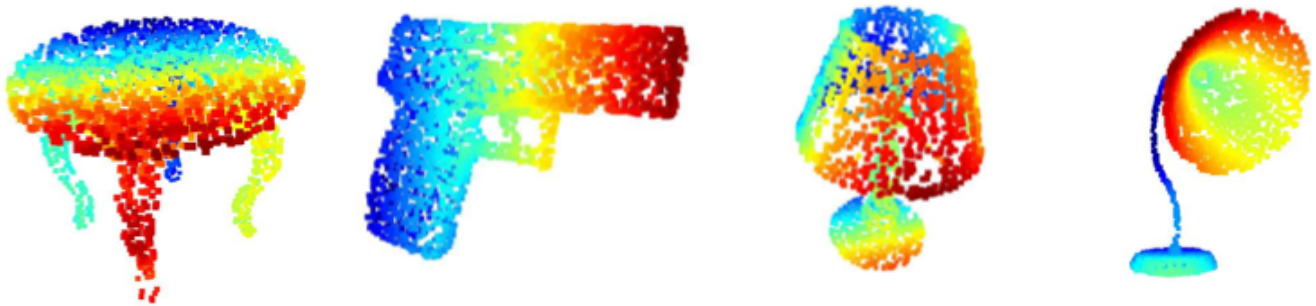
Visualizing Global Point Cloud Features



Which points won't affect the global feature?

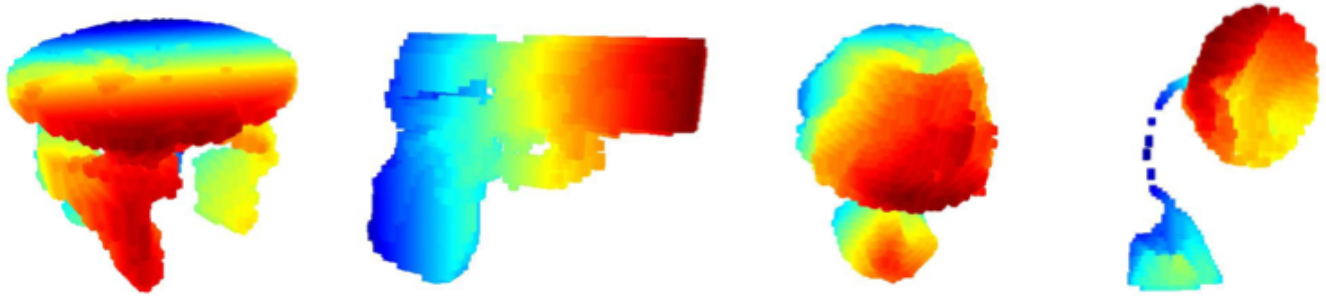
Visualizing Global Point Cloud Features

Original Shape:



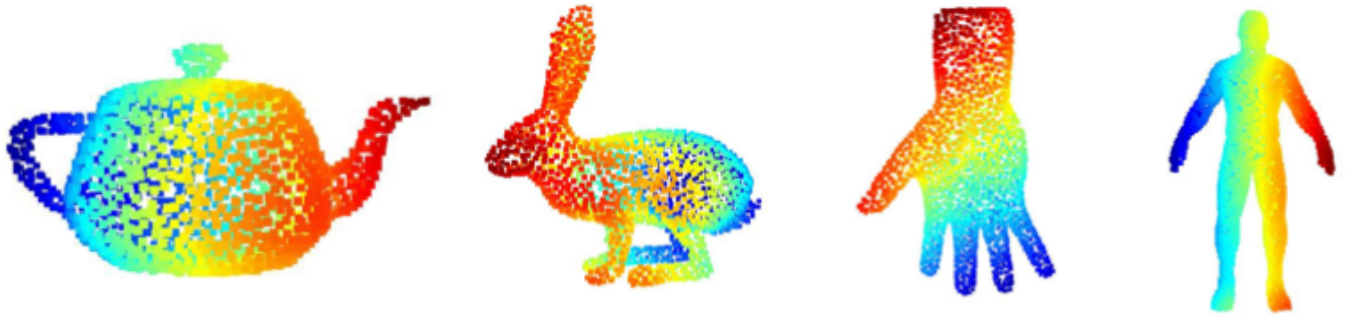
Critical Point Set:

Upper bound set:

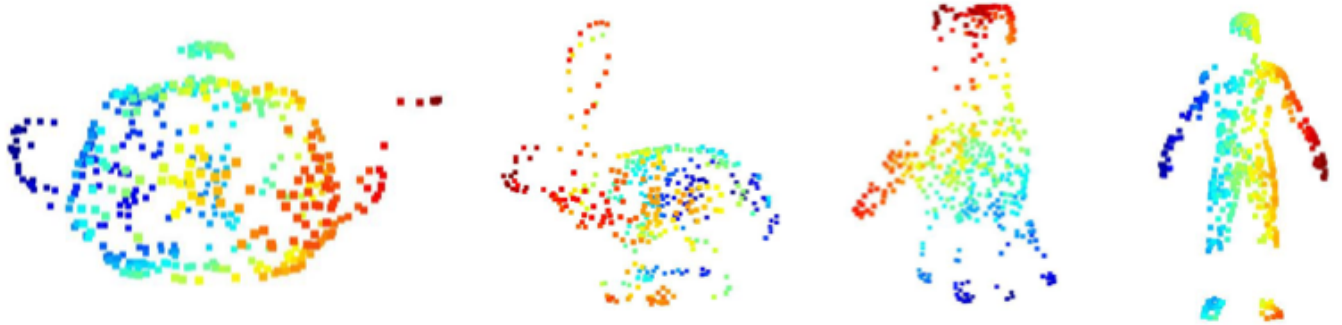


Visualizing Global Point Cloud Features (OOS)

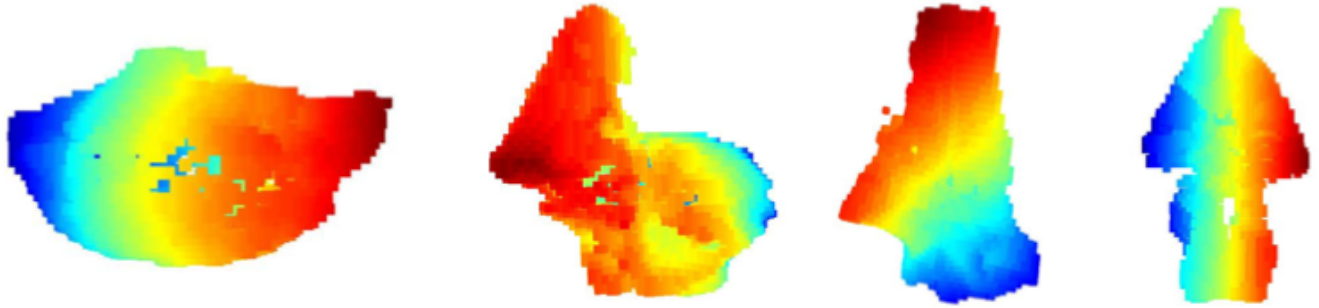
Original Shape:



Critical Point Set:

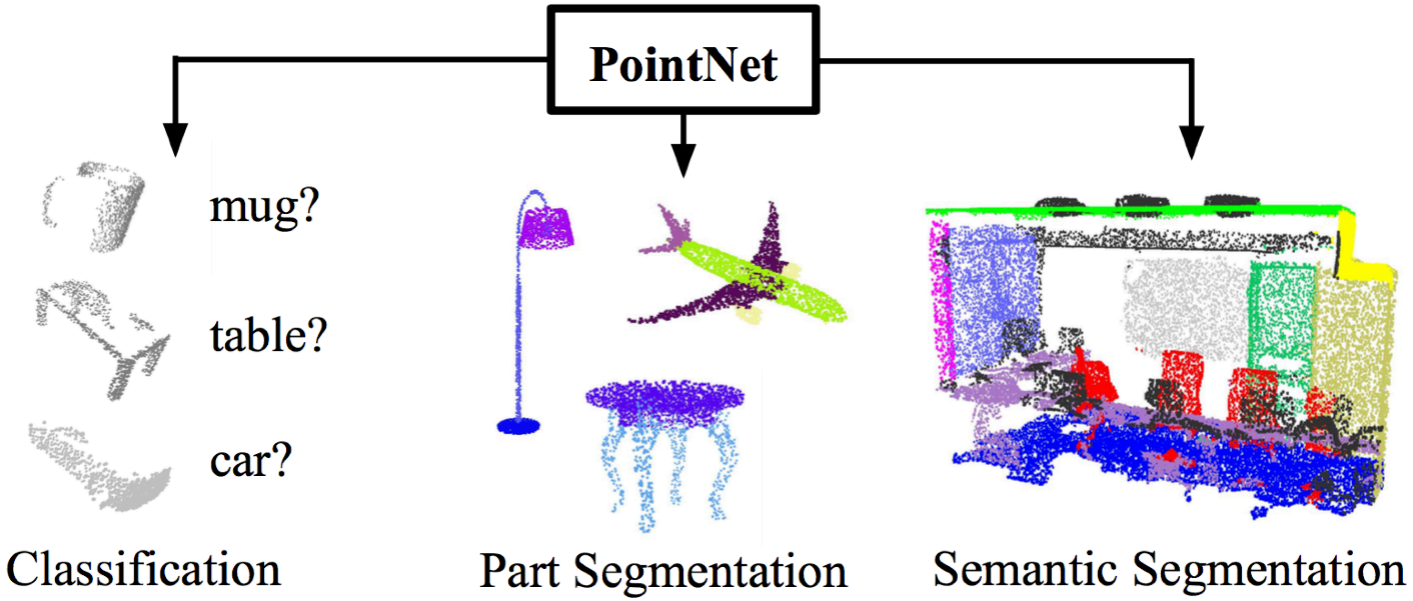


Upper bound Set:



Conclusion

- PointNet is a novel deep neural network that directly consumes point cloud.
- A unified approach to various 3D recognition tasks.
- Rich theoretical analysis and experimental results.



Code & Data Available!
http://stanford.edu/~rqi/pointnet

See you at Poster 9!

Thank you!


STANFORD
BIO-X


NSF


Office of Naval Research
USN
ONR
Science & Technology


NVIDIA


SAMSUNG


Google


Adobe

THE END

Speed and Model Size

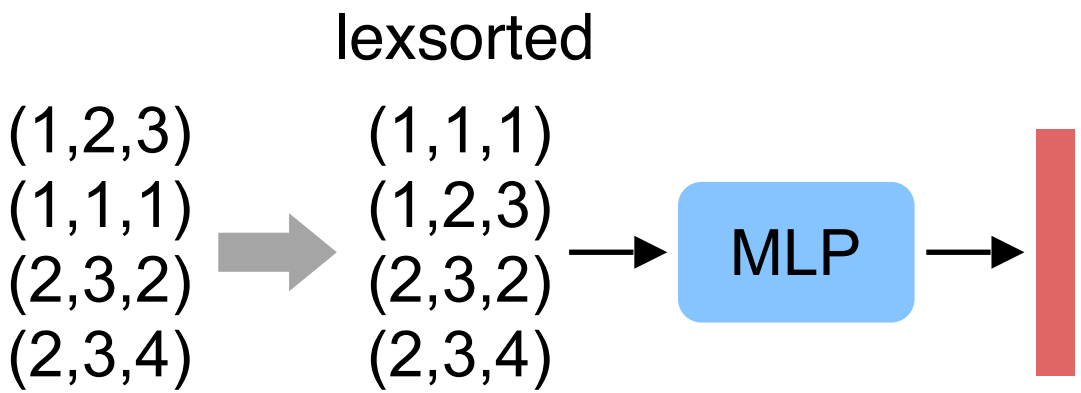
	#params	FLOPs/sample
PointNet (vanilla)	0.8M	148M
PointNet	3.5M	440M
Subvolume [16]	16.6M	3633M
MVCNN [20]	60.0M	62057M

Inference time 11.6ms, 25.3ms GTX1080, batch size 8

Permutation Invariance: How about Sorting?

“Sort” the points before feeding them into a network.

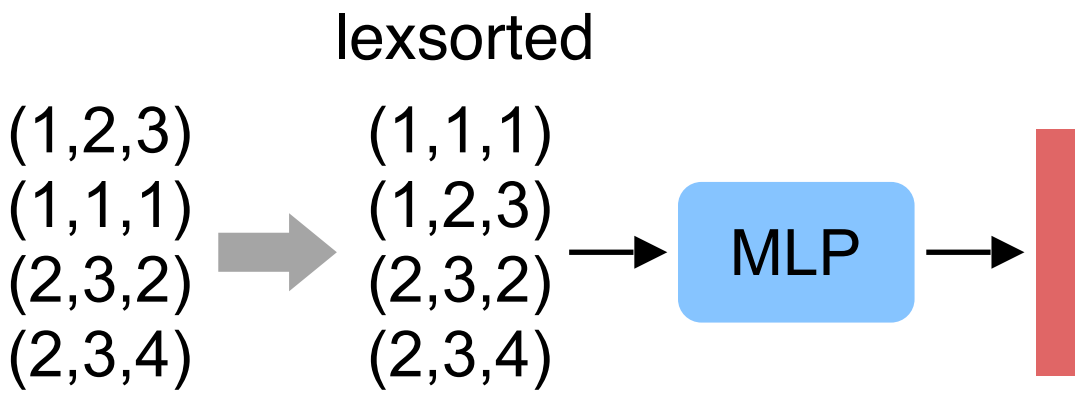
Unfortunately, there is no canonical order in high dim space.



Permutation Invariance: How about Sorting?

"Sort" the points before feeding them into a network.

Unfortunately, there is no canonical order in high dim space.



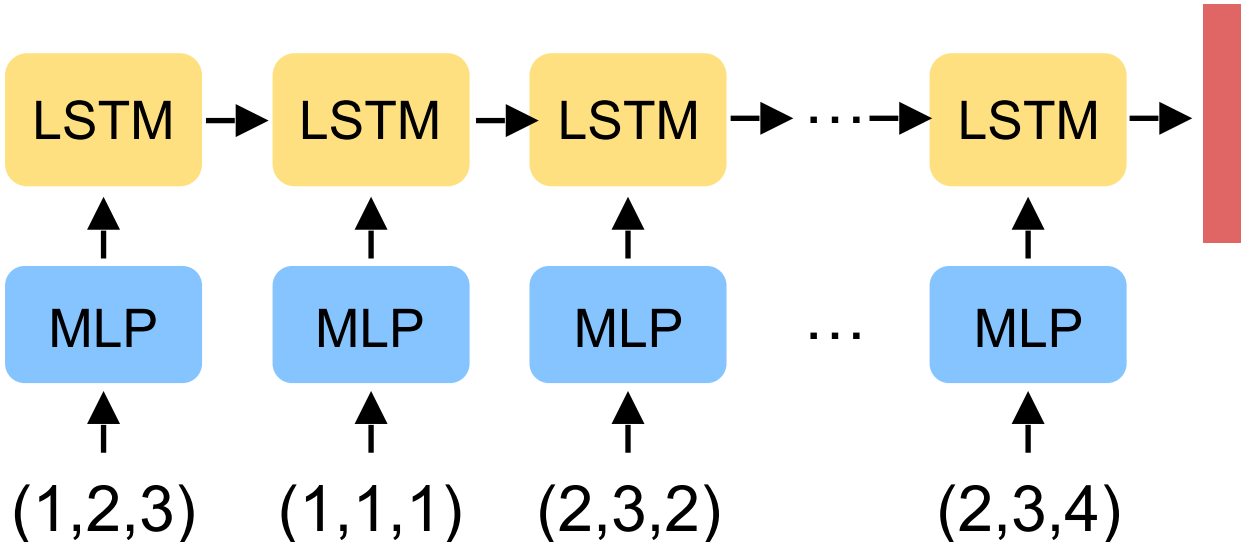
Multi-Layer Perceptron
(ModelNet shape classification)

	Accuracy
Unordered Input	12%
Lexsorted Input	40%
PointNet (vanilla)	**87%**

Permutation Invariance: How about RNNs?

Train RNN with permutation augmentation.

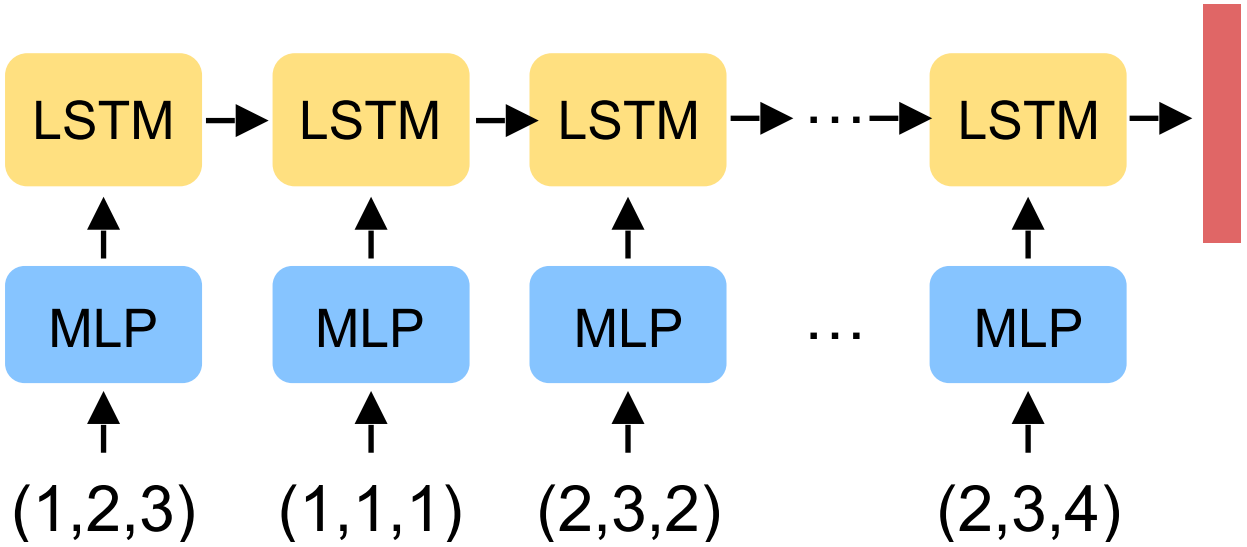
However, RNN forgets and order matters.



Permutation Invariance: How about RNNs?

Train RNN with permutation augmentation.

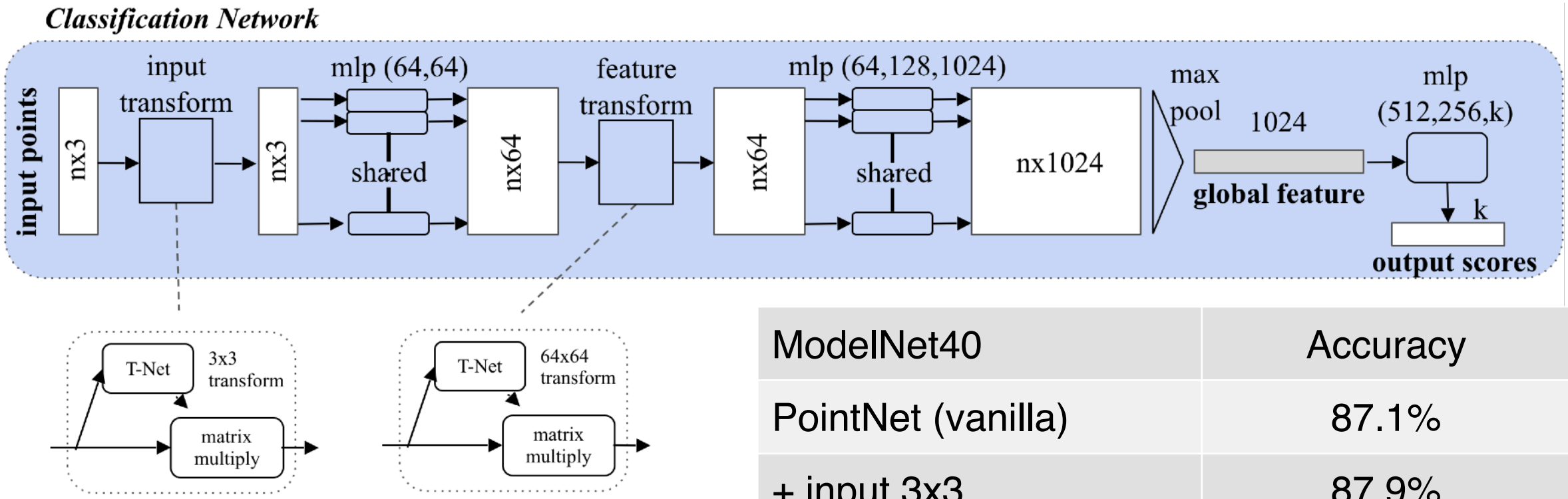
However, RNN forgets and order matters.



LSTM Network
(ModelNet shape classification)

	Accuracy
LSTM	75%
PointNet (vanilla)	**87%**

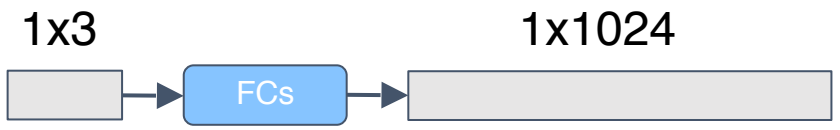
PointNet Classification Network



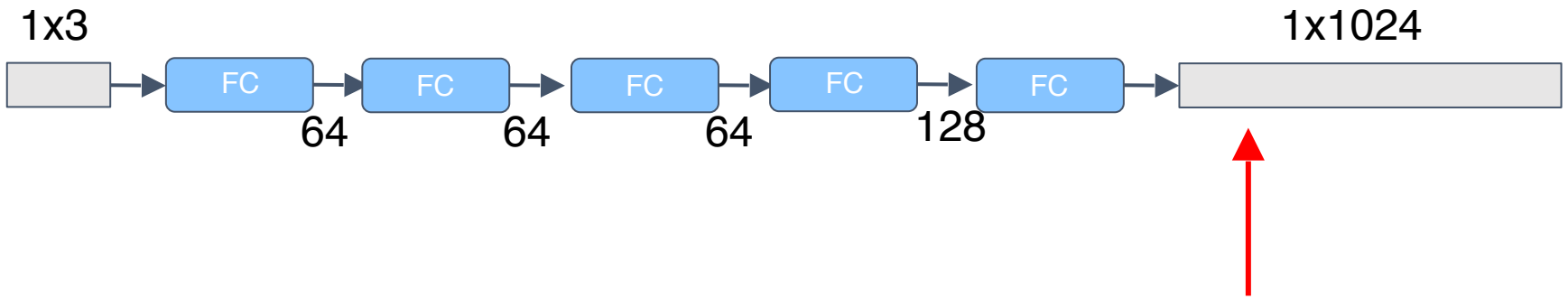
ModelNet40	Accuracy
PointNet (vanilla)	87.1%
+ input 3x3	87.9%
+ feature 64x64	86.9%
+ feature 64x64 + reg	87.4%
+ both	**89.2%**

Visualizing Point Functions

Compact View:



Expanded View:



Which input point will activate neuron X?

Find the top-K points in a dense volumetric grid that activates neuron X.

Visualizing Point Functions