DeepRob

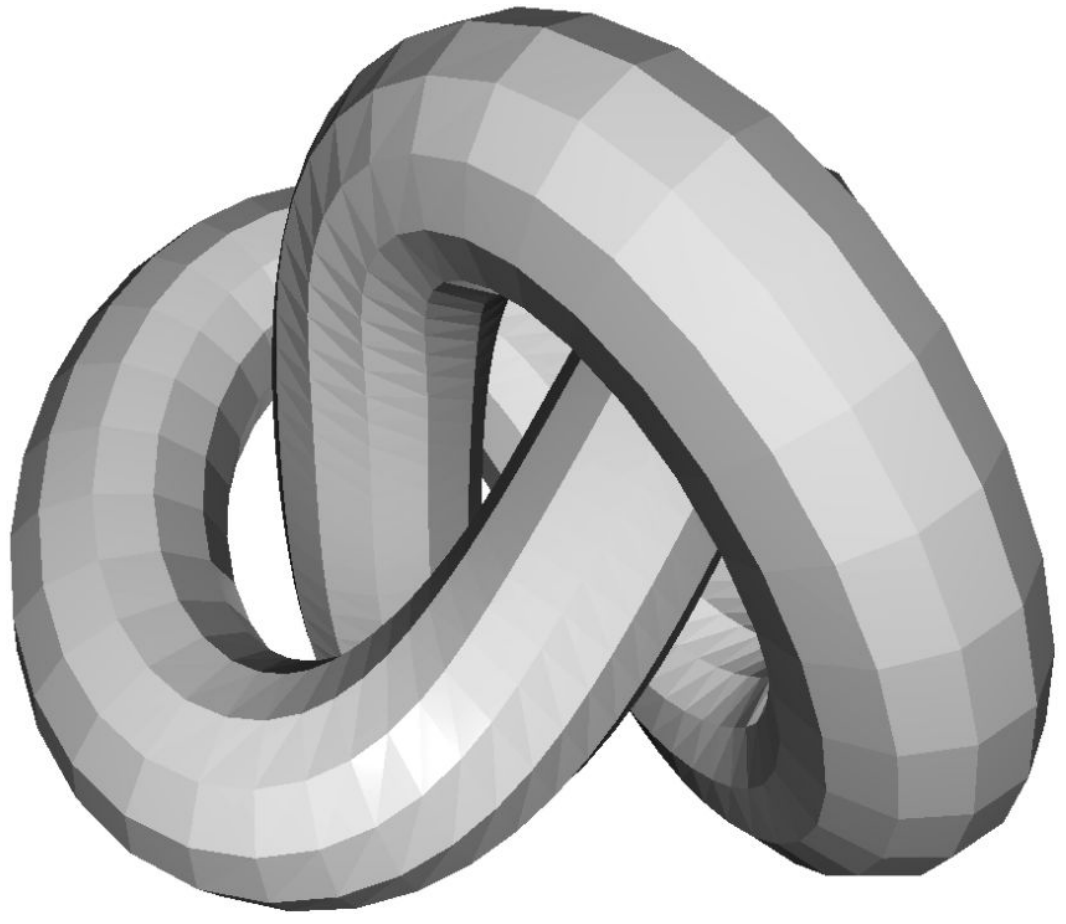
[Group 3] Lecture 2

by Nikil Krishnakumar, Nanditha Naik

Pointnet and 3D Networks for Manipulation

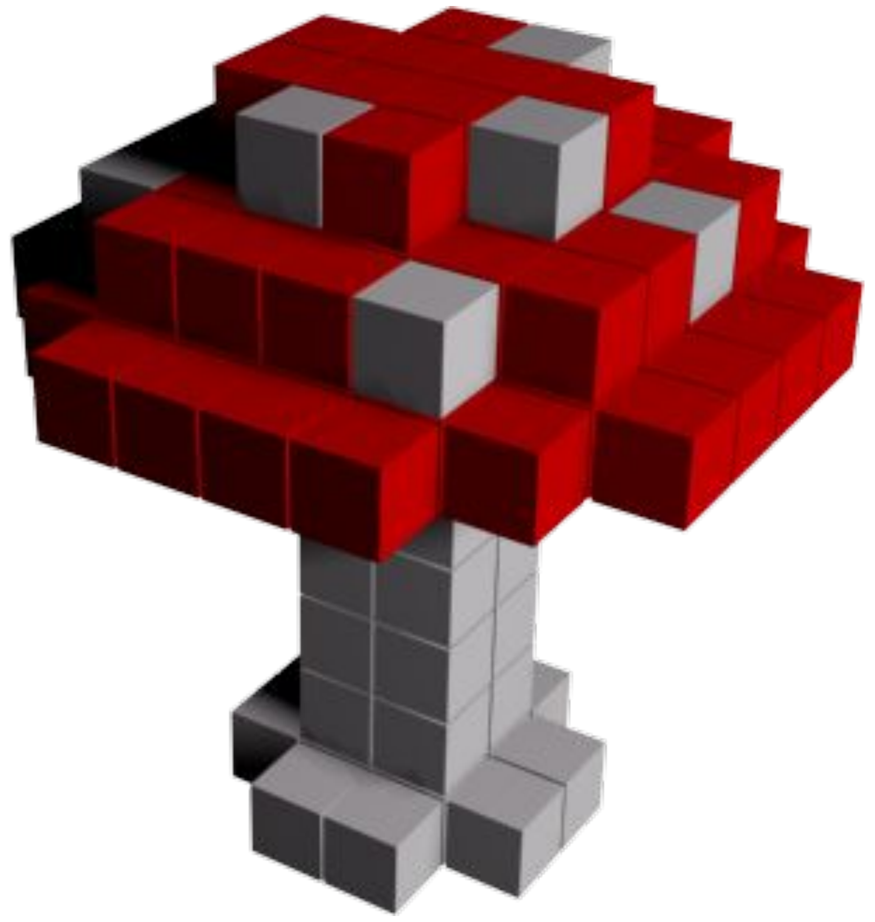
University of Minnesota

Data structures for 3D Representations



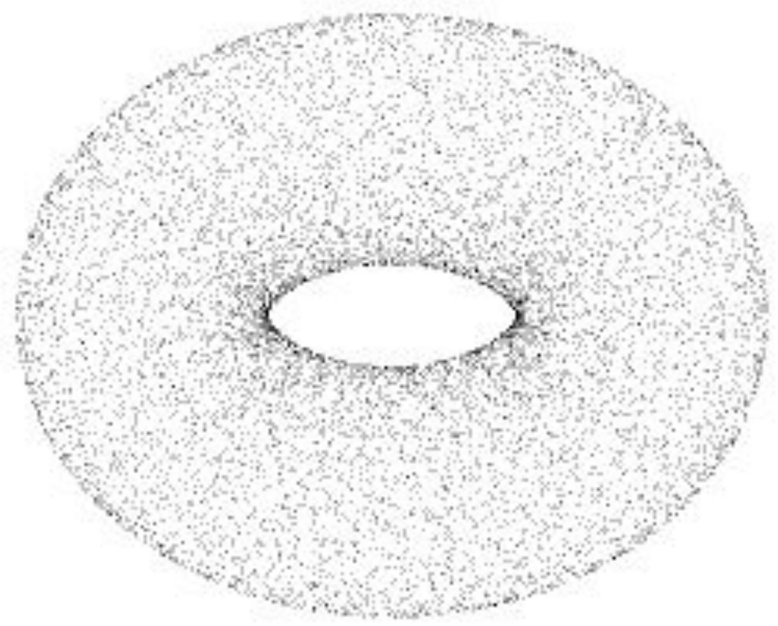
https://open3d.org/html/tutorial/Basic/mesh.html

Meshes



https://blog.spatial.com/the-main-benefits-and-disadvantages-of-voxel-modeling

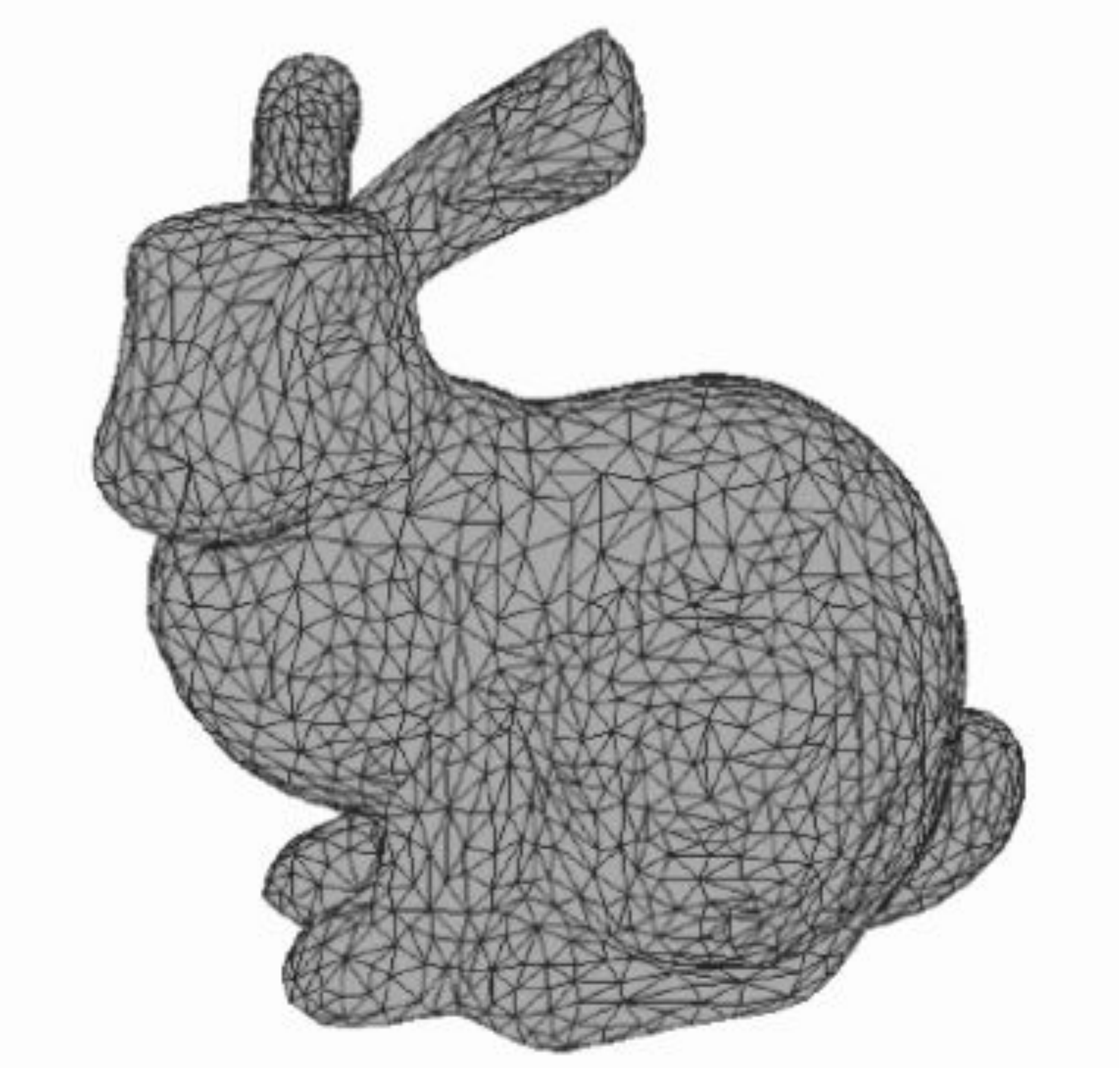
Voxel



https://en.wikipedia.org/wiki/Point_cloud#/media/File:Point_cloud_torus.gif

Point cloud

Meshes



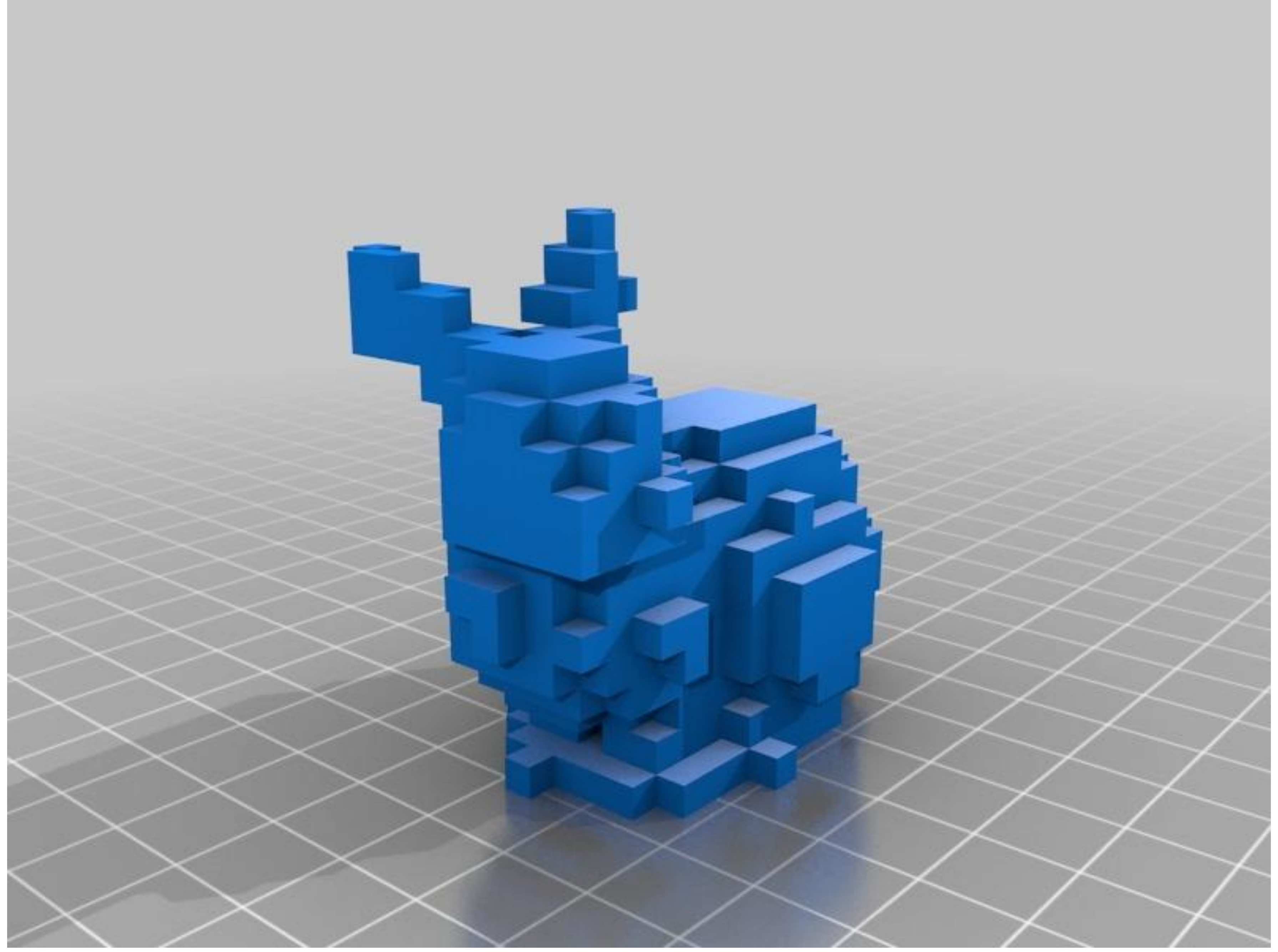
https://graphics.stanford.edu/data/3Dscanrep/

Mesh

Consist of interconnected vertices, edges, and polygonal faces (often triangles or quads) that shape 3D surfaces.

Useful for 3D object detection but high resolution meshes slow and complicate training

Voxels



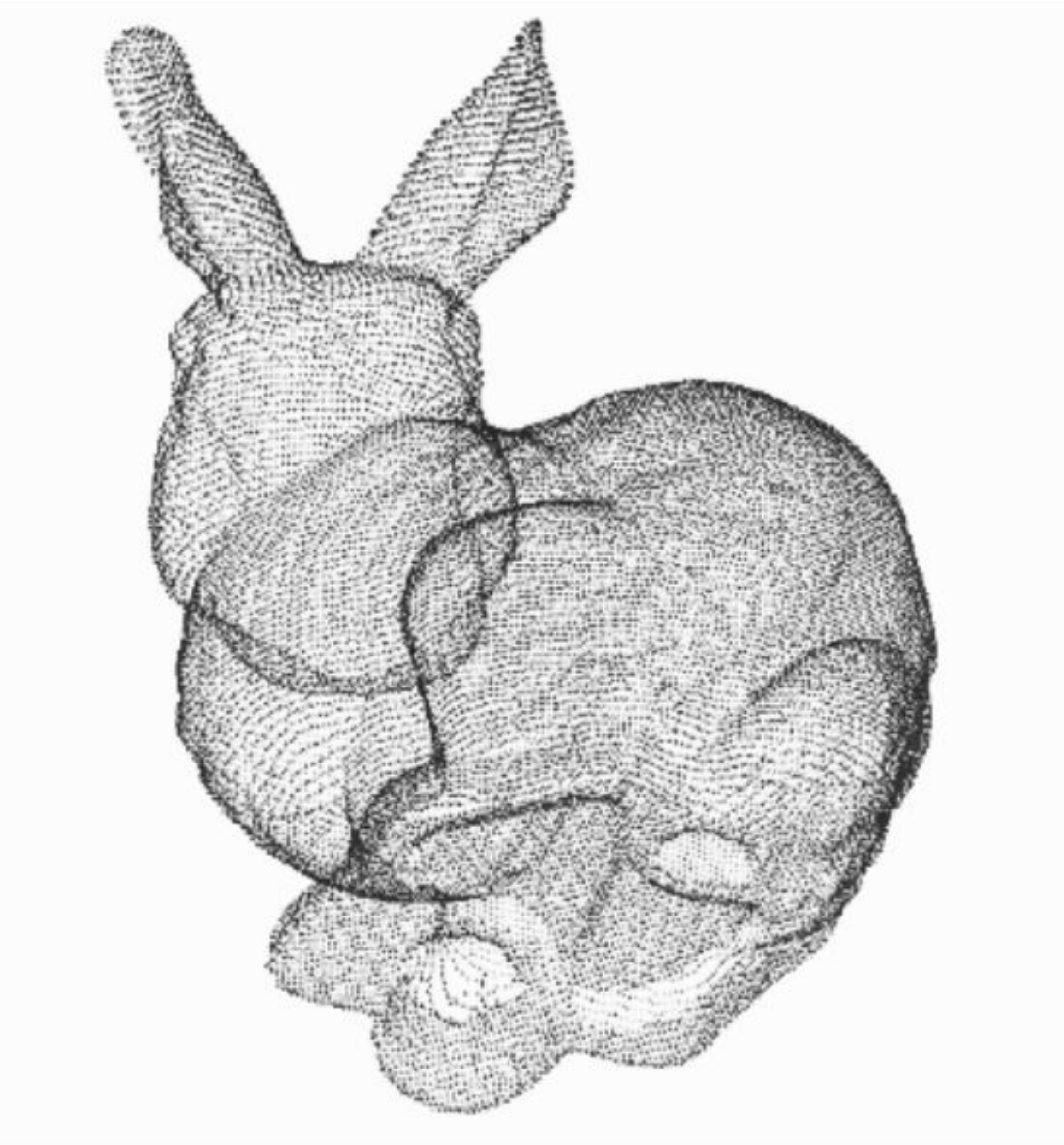
https://graphics.stanford.edu/data/3Dscanrep/

Voxel

Voxels are the three-dimensional equivalents of pixels, represented as cubic elements that occupy space in a 3D grid.

Sparse grids with many empty voxels lead to storage and processing inefficiencies, limiting scalability for large 3D scenes.

Pointcloud



https://graphics.stanford.edu/data/3Dscanrep/

Point cloud

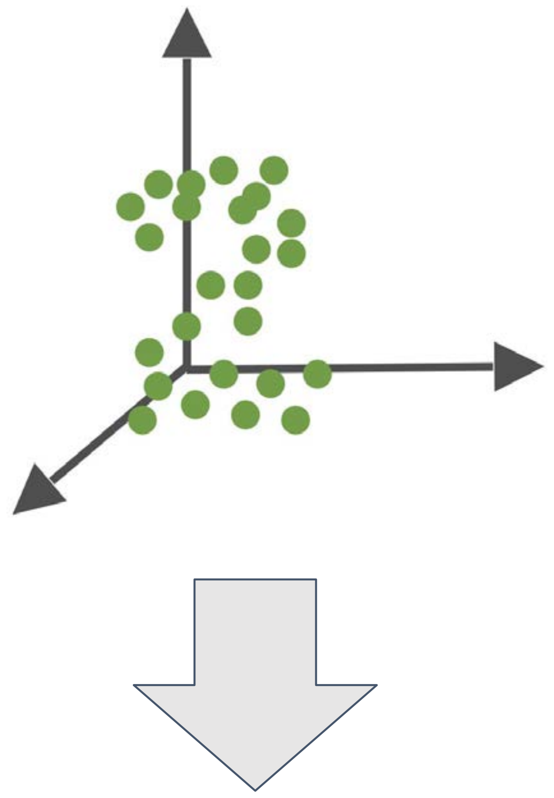
Point + cloud = Pointcloud

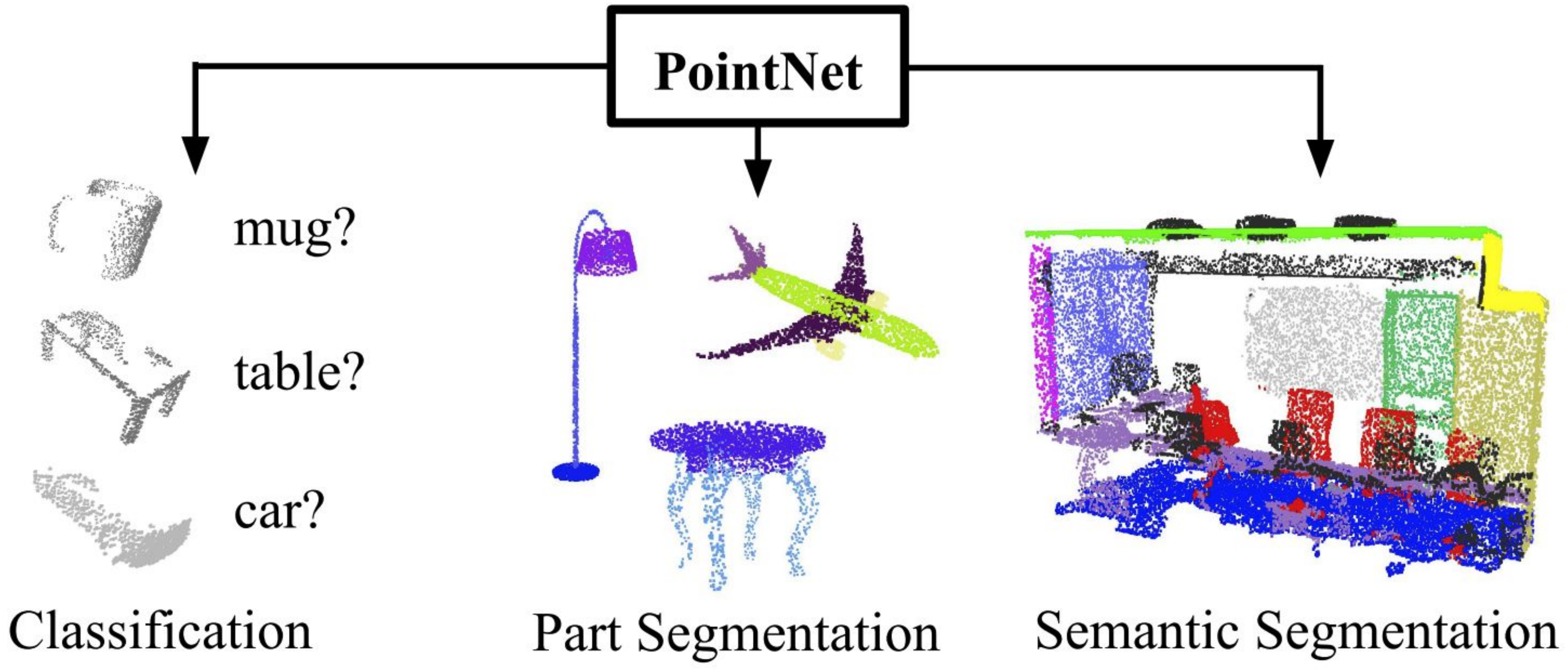
Measurement unit that is represented using x, y, and z coordinates.

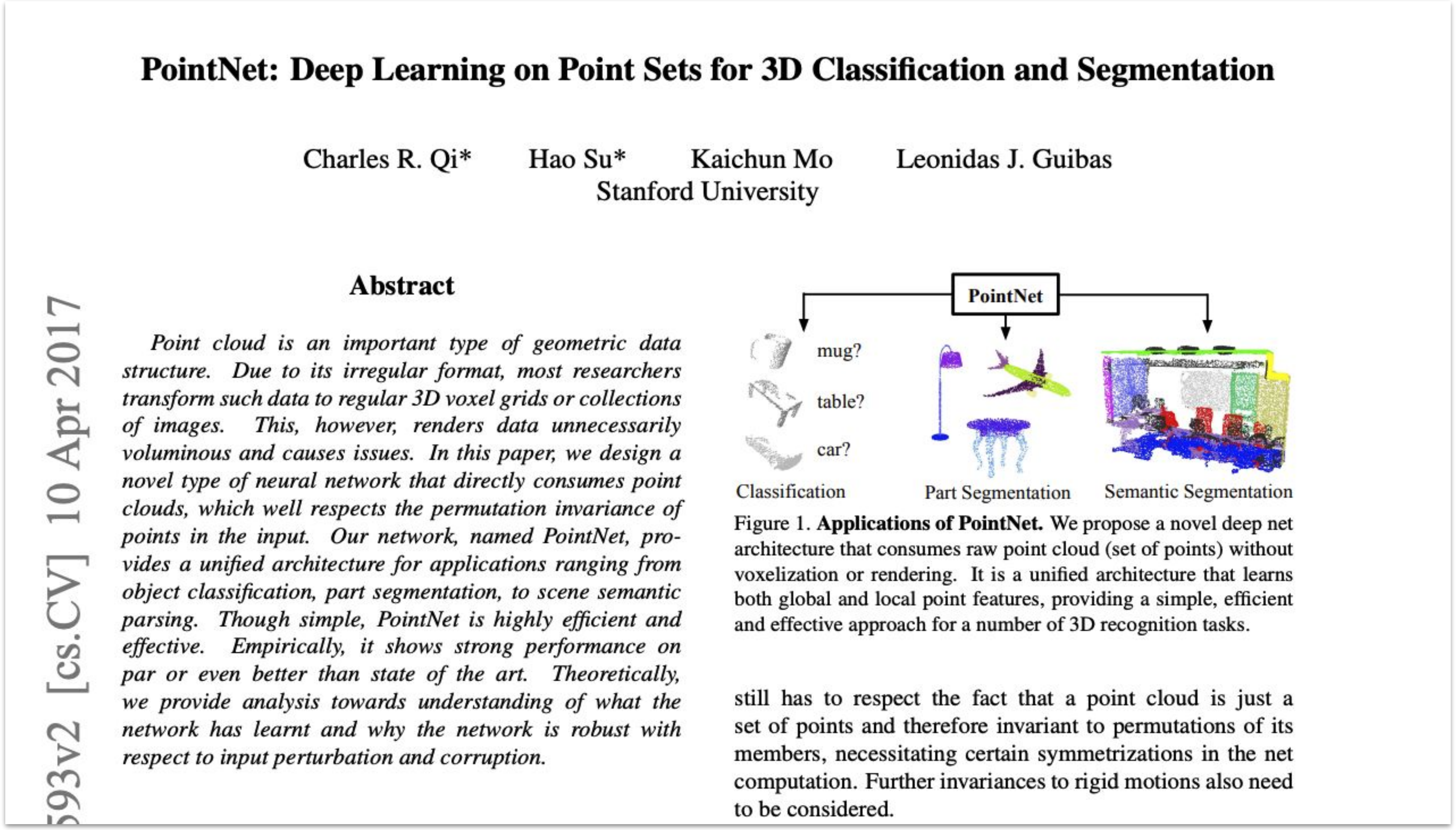
A set of points in a space that represent some 3D shape or object

PointNet

PointNet







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

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

593v2 [cs.CV] 10 Apr 2017

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 clouds, which well respects the permutation invariance of points in the input. Our network, named PointNet, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though simple, PointNet is highly efficient and effective. Empirically, it shows strong performance on par or even better than state of the art. Theoretically, we provide analysis towards understanding of what the network has learnt and why the network is robust with respect to input perturbation and corruption.

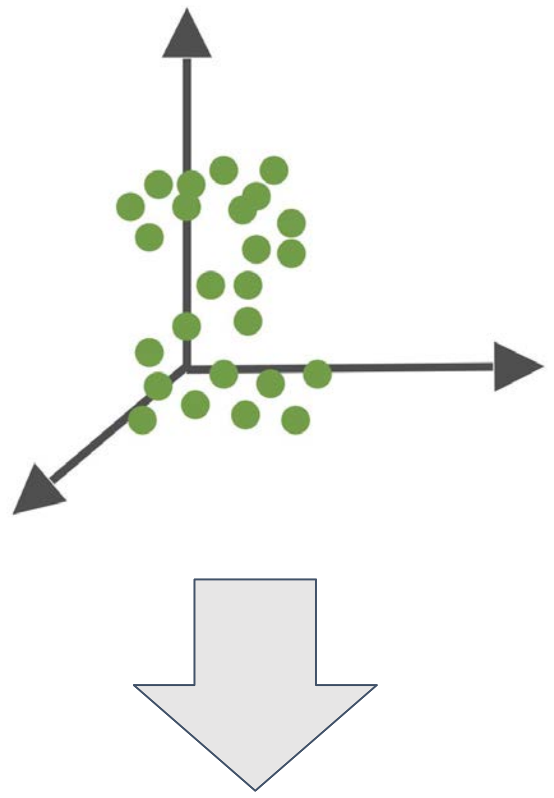
PointNet
mug?
table?
car?
Classification Part Segmentation Semantic Segmentation

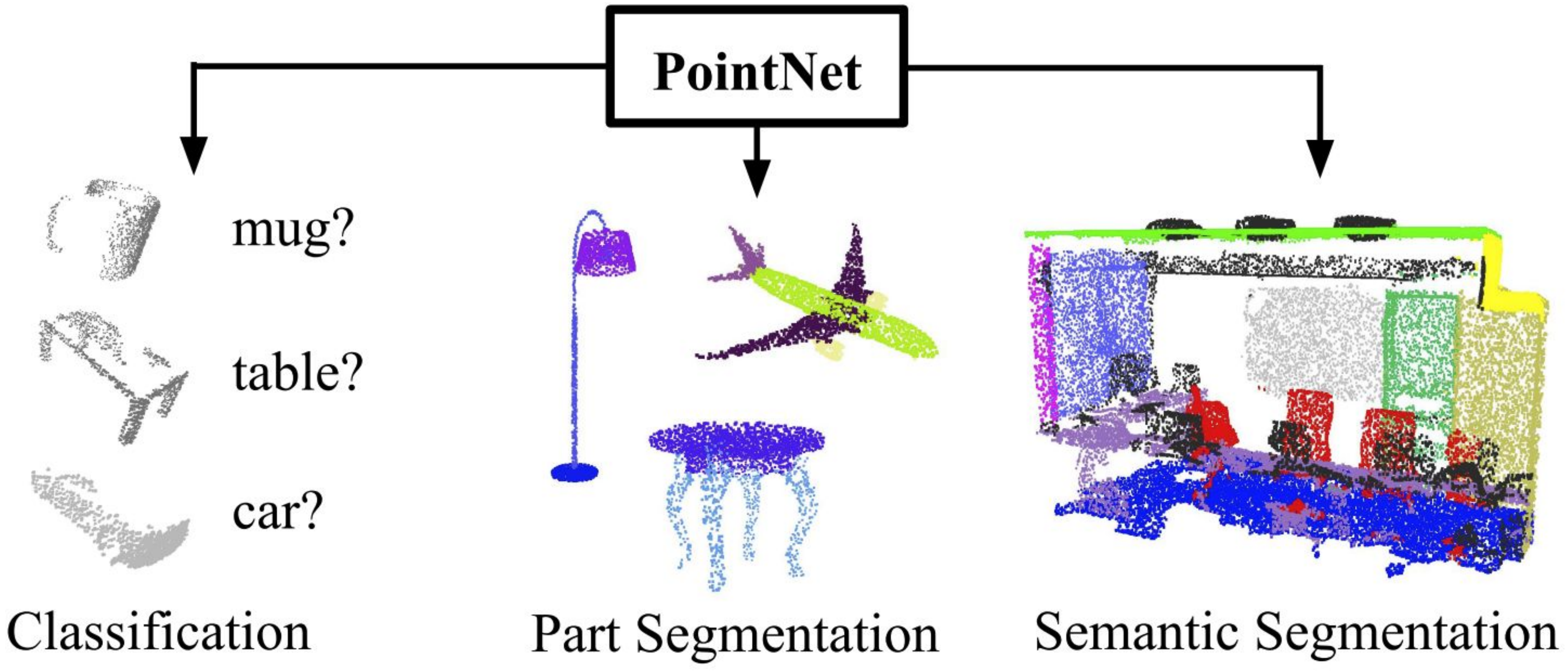
Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

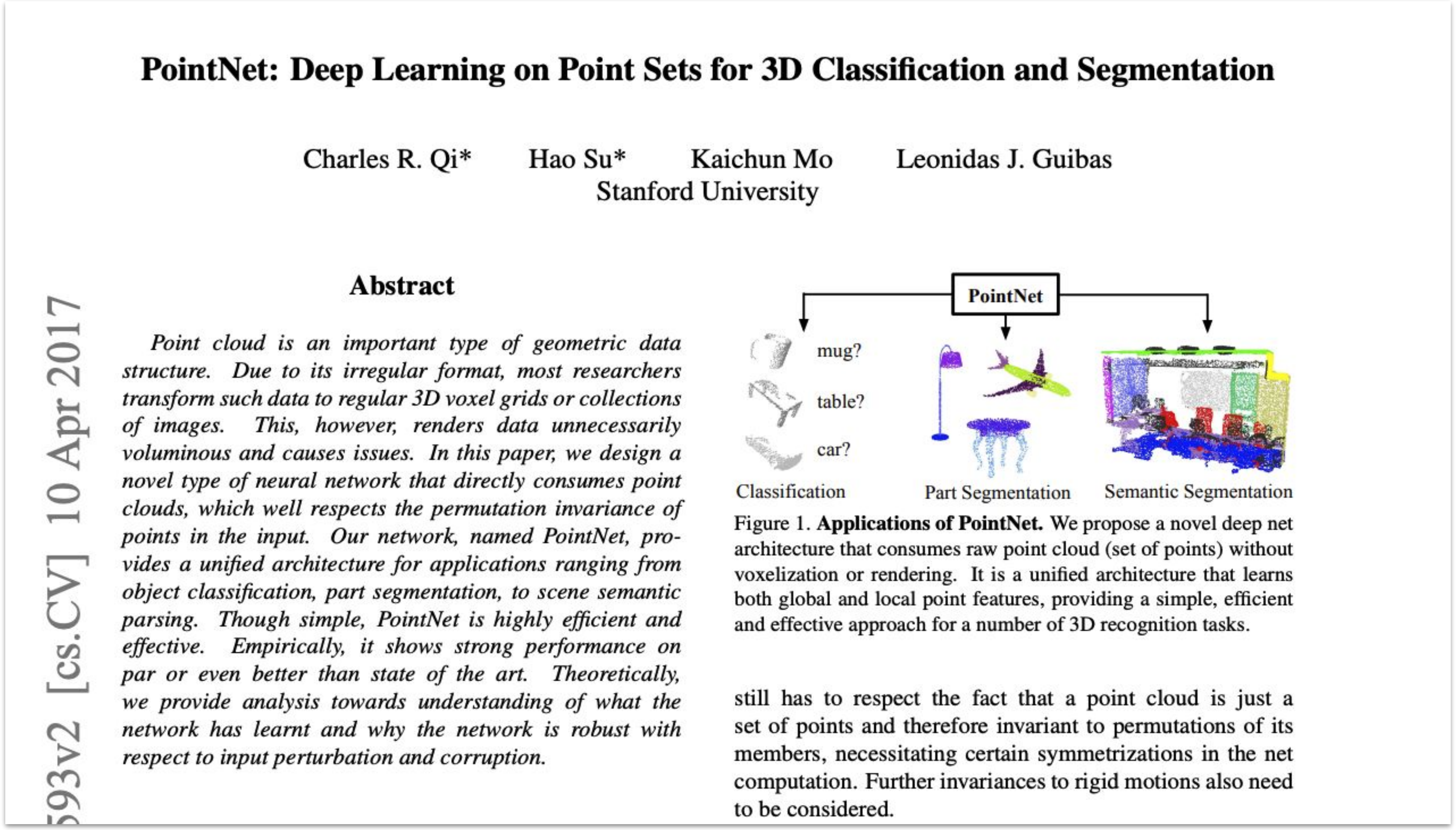
still has to respect the fact that a point cloud is just a set of points and therefore invariant to permutations of its members, necessitating certain symmetrizations in the net computation. Further invariances to rigid motions also need to be considered.

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.

PointNet







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

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

arXiv:1612.00593v2 [cs.CV] 10 Apr 2017

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 clouds, which well respects the permutation invariance of points in the input. Our network, named PointNet, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though simple, PointNet is highly efficient and effective. Empirically, it shows strong performance on par or even better than state of the art. Theoretically, we provide analysis towards understanding of what the network has learnt and why the network is robust with respect to input perturbation and corruption.

PointNet
mug?
table?
car?
Classification Part Segmentation Semantic Segmentation

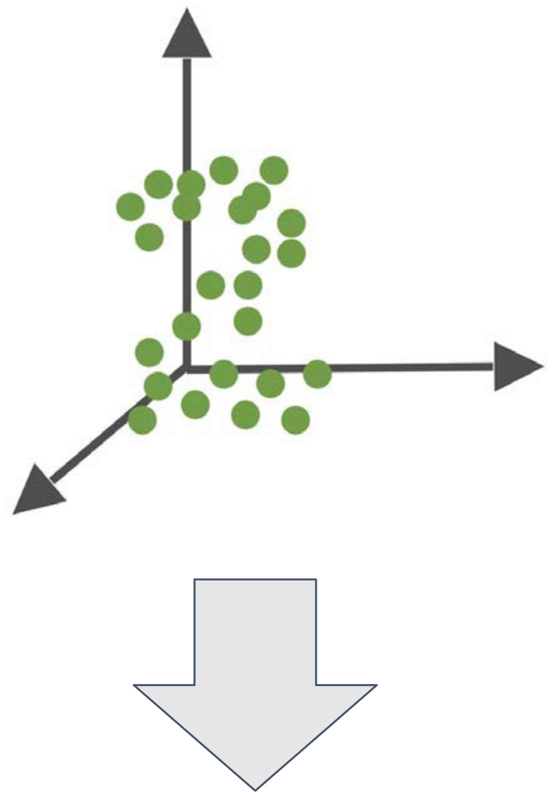
Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

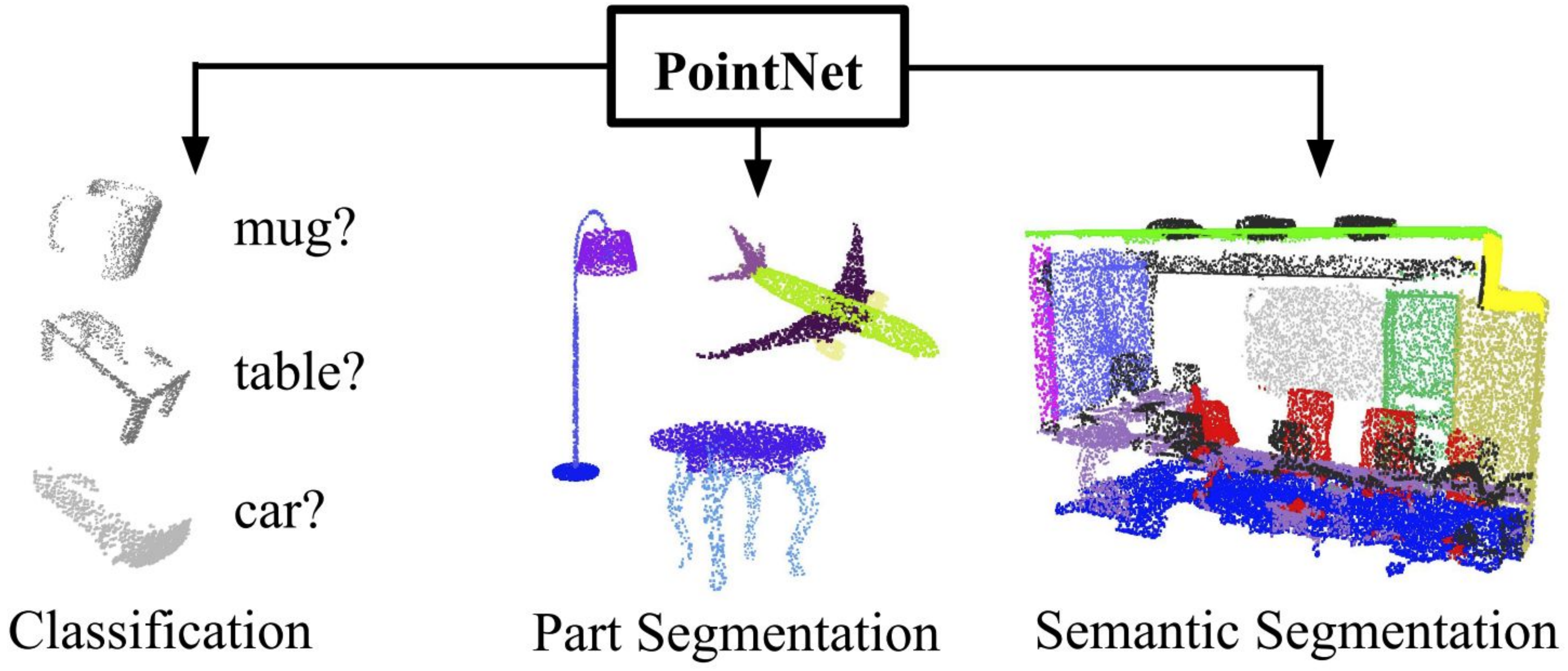
still has to respect the fact that a point cloud is just a set of points and therefore invariant to permutations of its members, necessitating certain symmetrizations in the net computation. Further invariances to rigid motions also need to be considered.

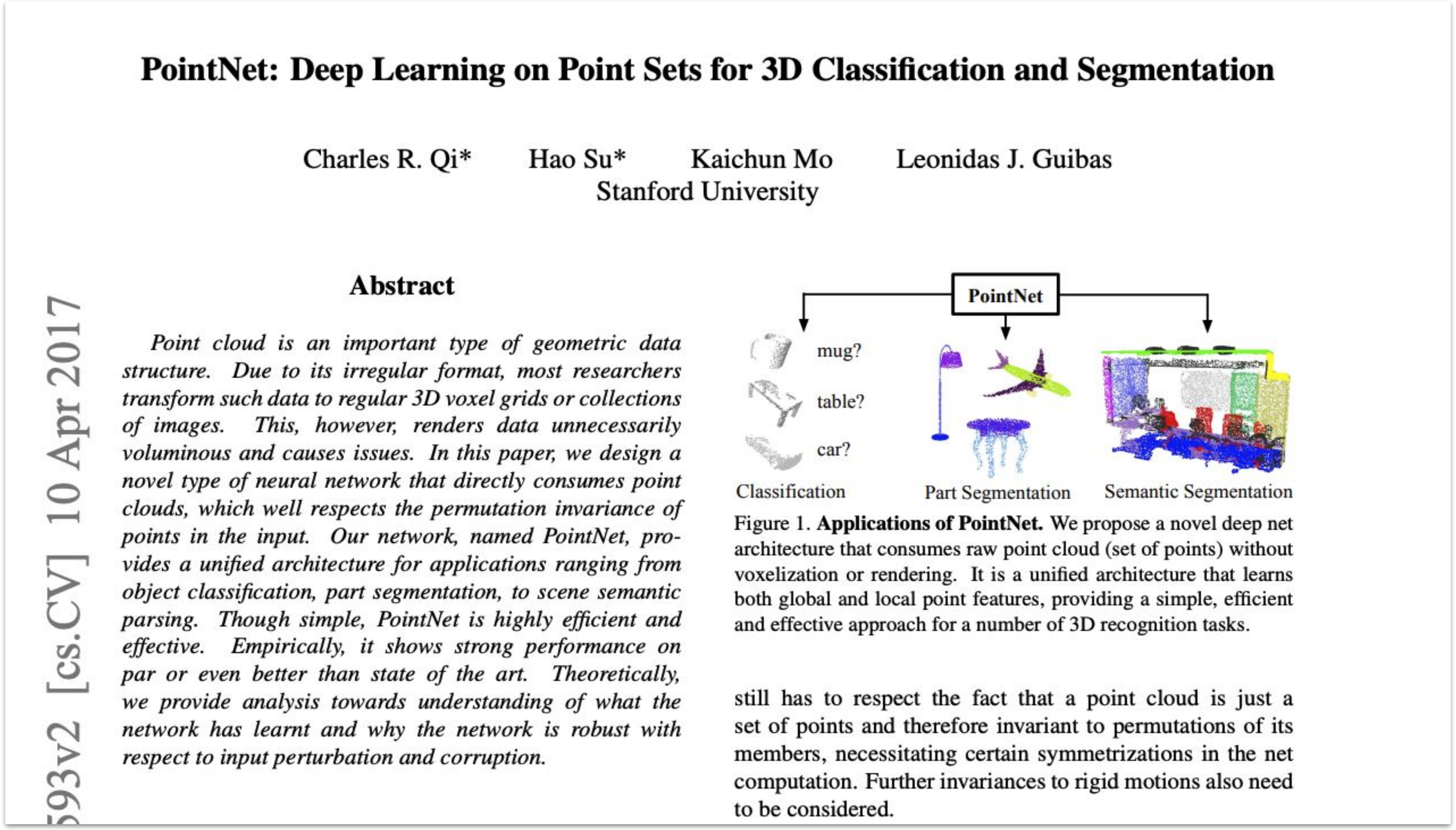
1. End-to-end learning for scattered and unordered point data

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.

PointNet







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

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

593v2 [cs.CV] 10 Apr 2017

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 clouds, which well respects the permutation invariance of points in the input. Our network, named PointNet, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though simple, PointNet is highly efficient and effective. Empirically, it shows strong performance on par or even better than state of the art. Theoretically, we provide analysis towards understanding of what the network has learnt and why the network is robust with respect to input perturbation and corruption.

Figure 1. **Applications of PointNet.** We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

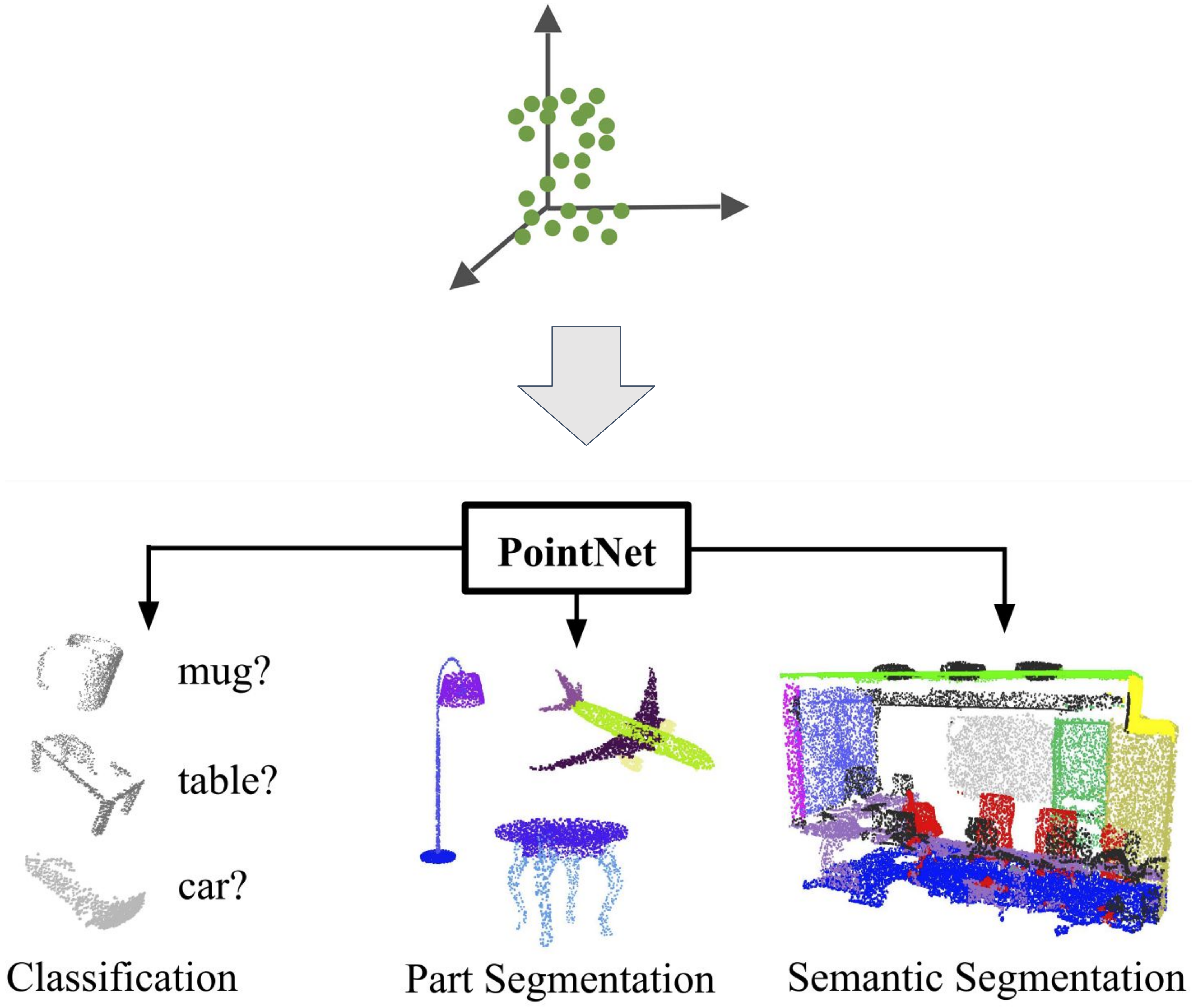
still has to respect the fact that a point cloud is just a set of points and therefore invariant to permutations of its members, necessitating certain symmetrizations in the net computation. Further invariances to rigid motions also need to be considered.

1. End-to-end learning for scattered and unordered point data

2. Unified framework for various tasks

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.

PointNet



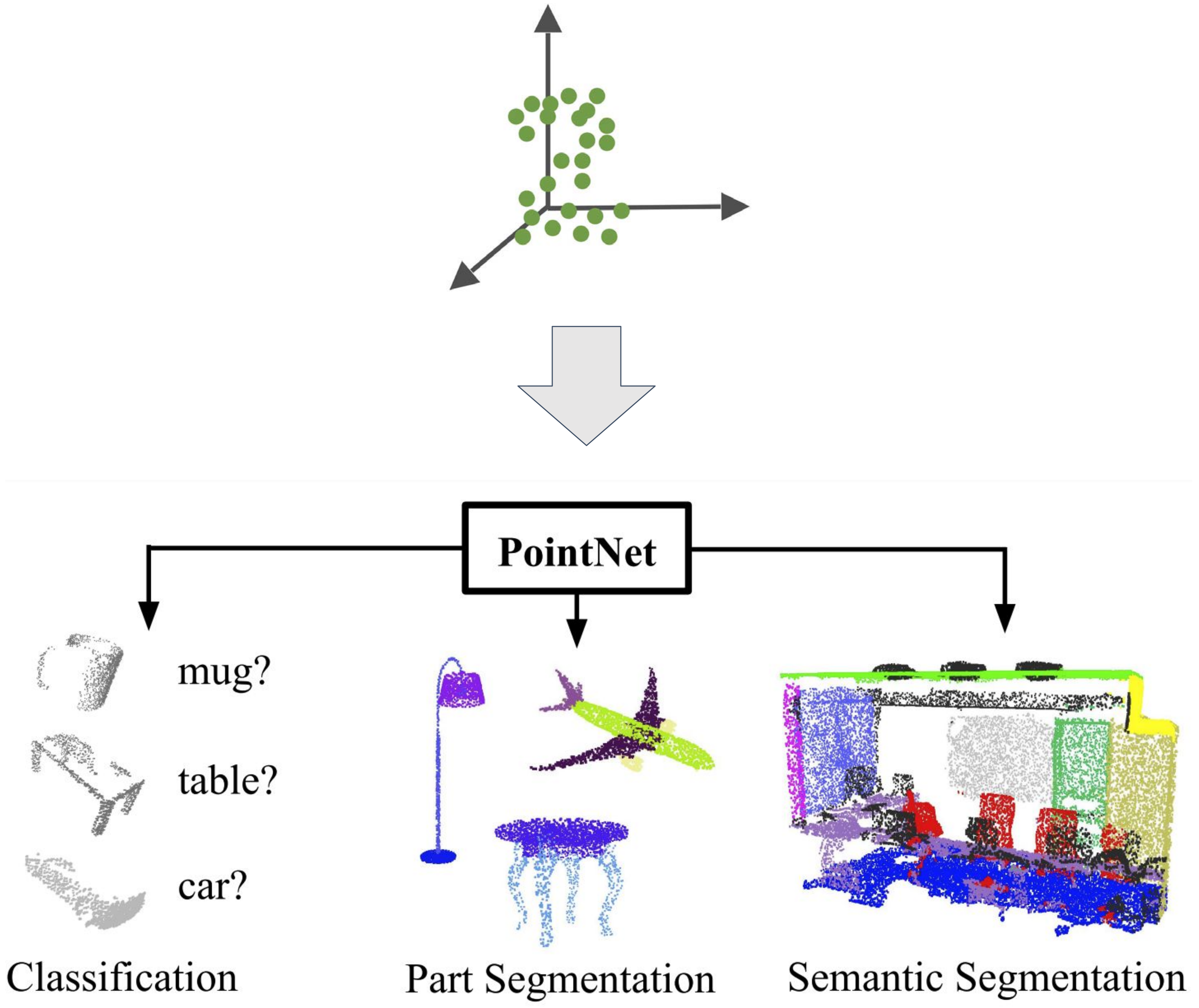
Properties of point sets in R^n

1. Unordered:

Consume N 3D point to be **invariant** to **N!** of input set in data feeding

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.

PointNet



Properties of point sets in R^n

1. Unordered:

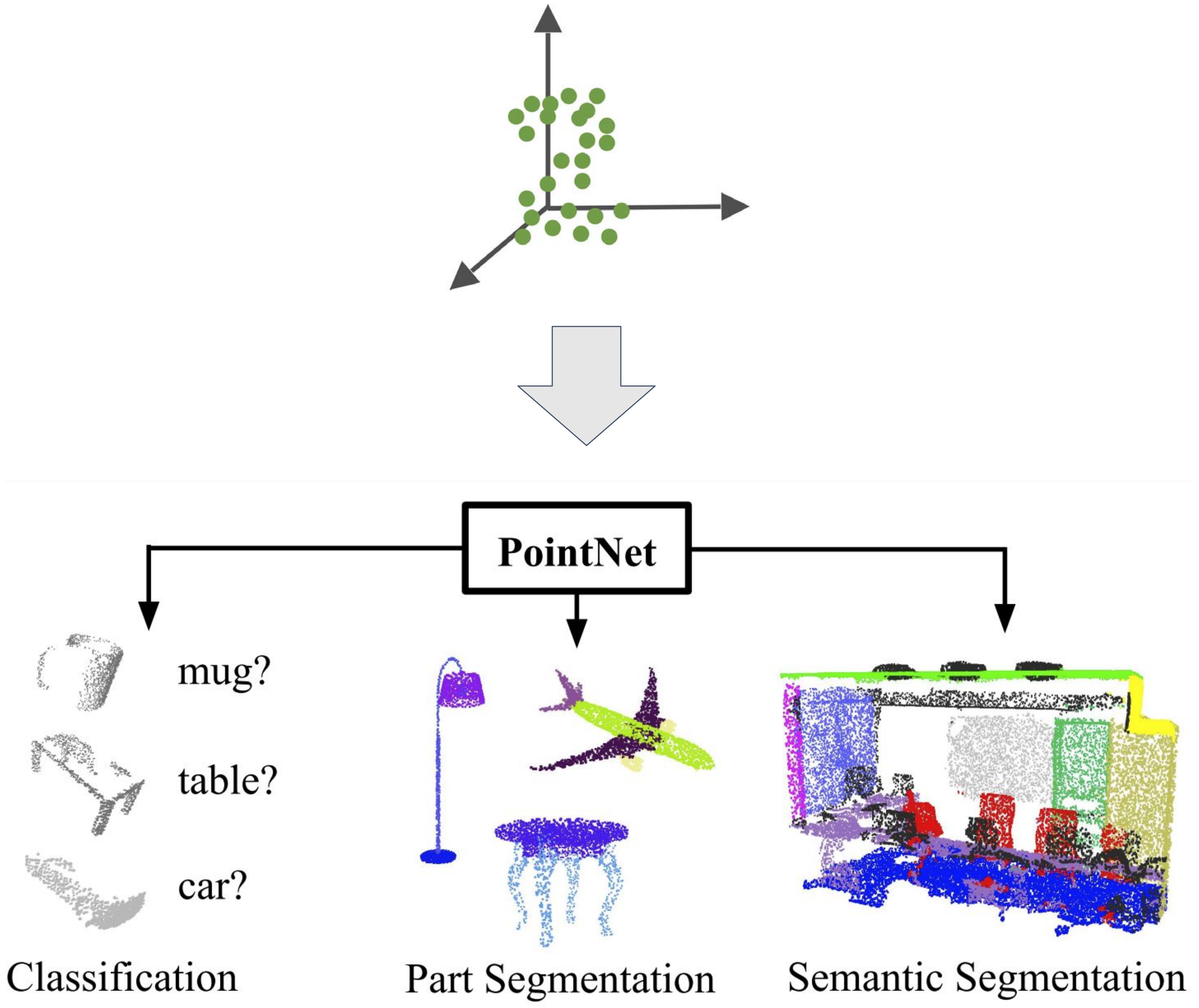
Consume N 3D point to be **invariant** to **N!** of input set in data feeding

2. Interaction among points:

Points are not **isolated** i.e the neighboring points provide meaningful information like **local structures** and **combinatorial interactions**

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.

PointNet



Properties of point sets in R^n

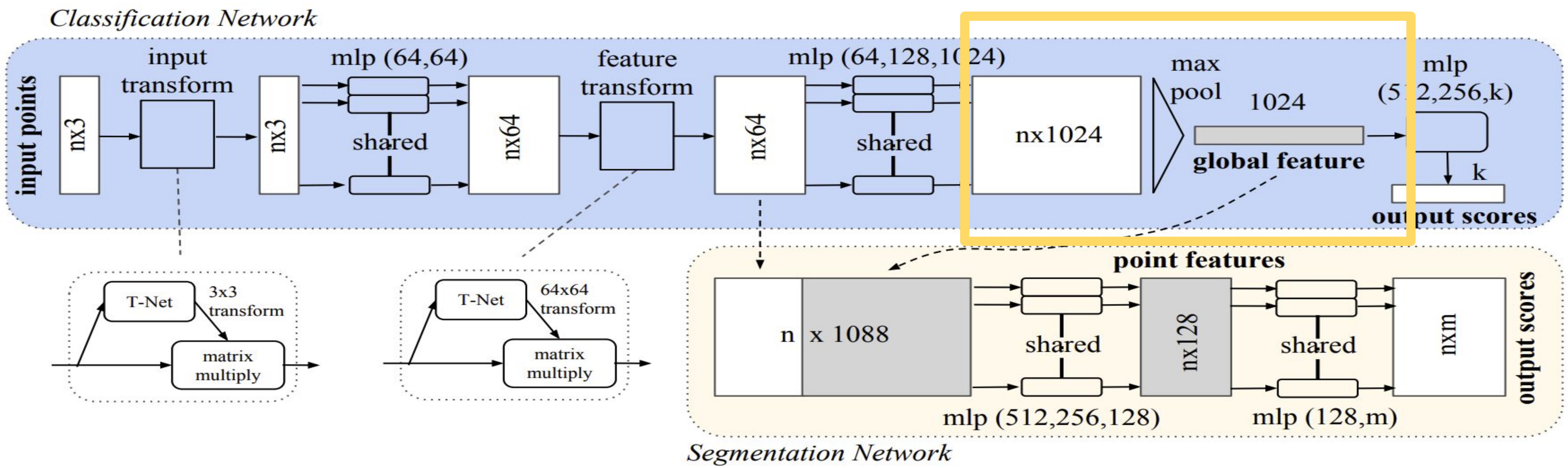
1. Unordered:
   Consume N 3D point to be **invariant** to **N!** of input set in data feeding

2. Interaction among points:
   Points are not **isolated** i.e the neighboring points provide meaningful information like **local structures** and **combinatorial interactions**

3. Invariant to transformations:
   The geometric representations learned by the network are **invariant** to **transformations**

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.

Pointnet Architecture

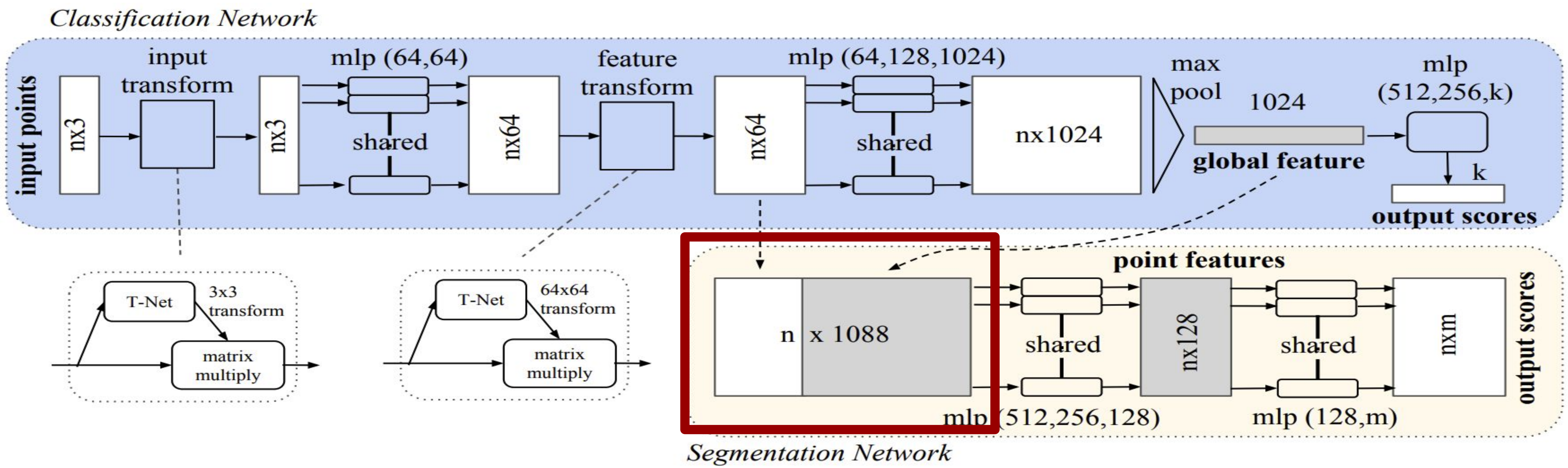


It has three key modules:

1. Max pooling: Gives order to invariance

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.

Pointnet Architecture

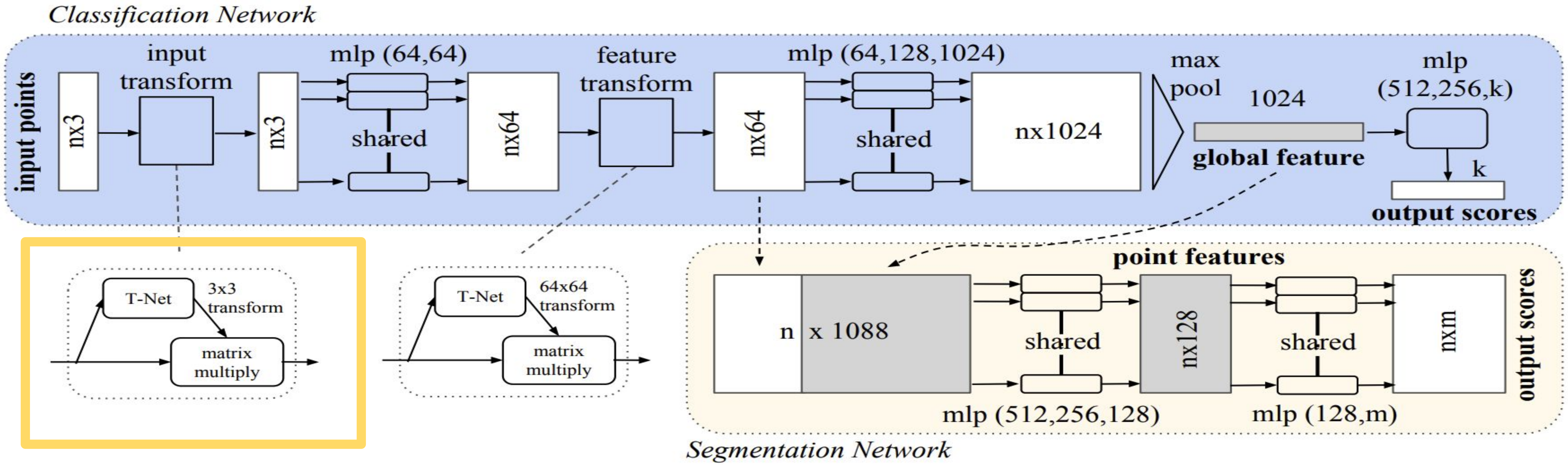


It has three key modules:

1. Max pooling: Gives order to invariance
2. Local and Global features combination: This modification allows network to predict per-point quantities based on local and global geometry

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.

Pointnet Architecture

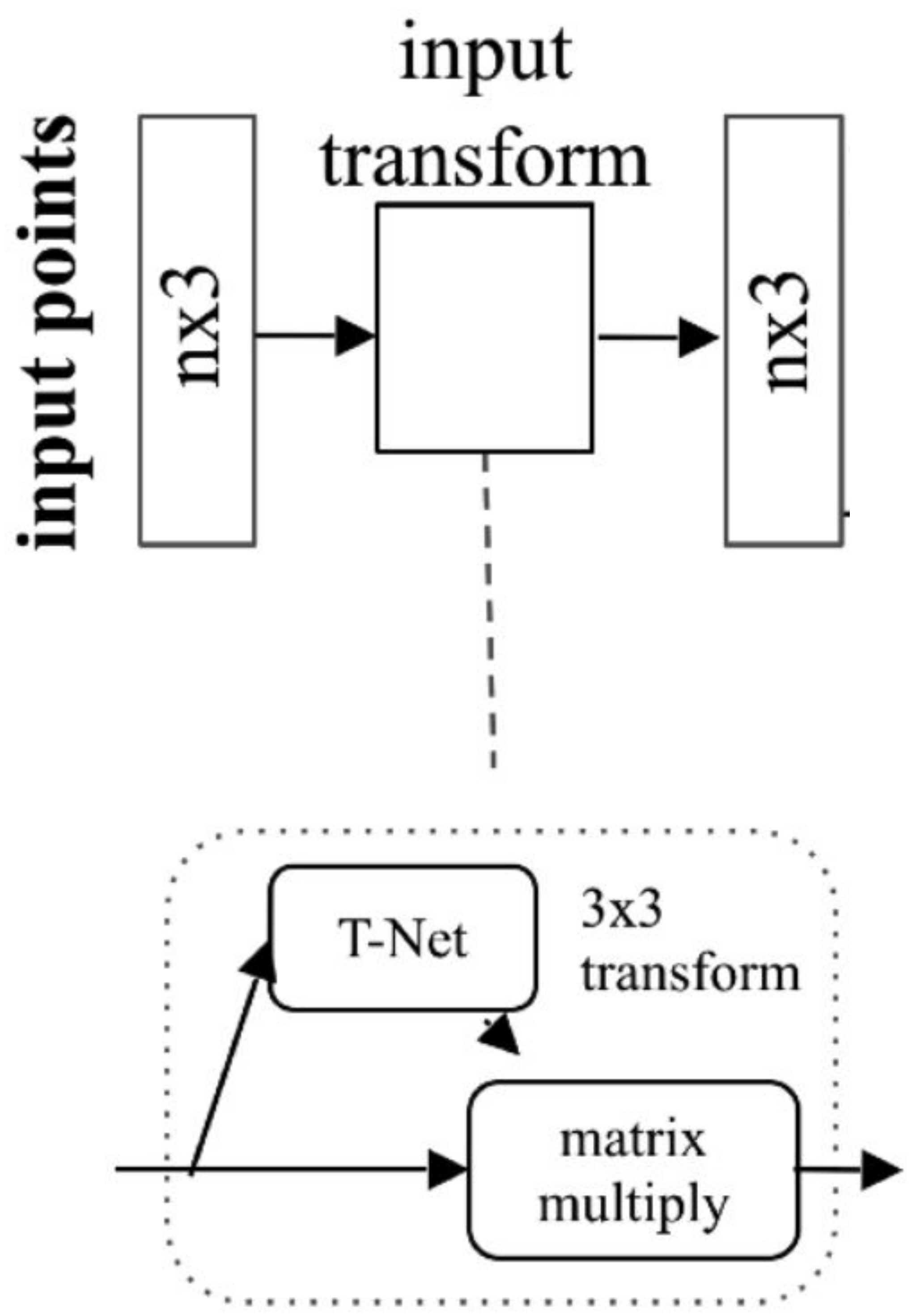


It has three key modules:

1. Max pooling: Gives order to invariance
2. Local and Global features combination: This modification allows network to predict per-point quantities based on local and global geometry
3. Joint alignment Network (T-Net): Gives invariances by transforming to canonical pose

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.

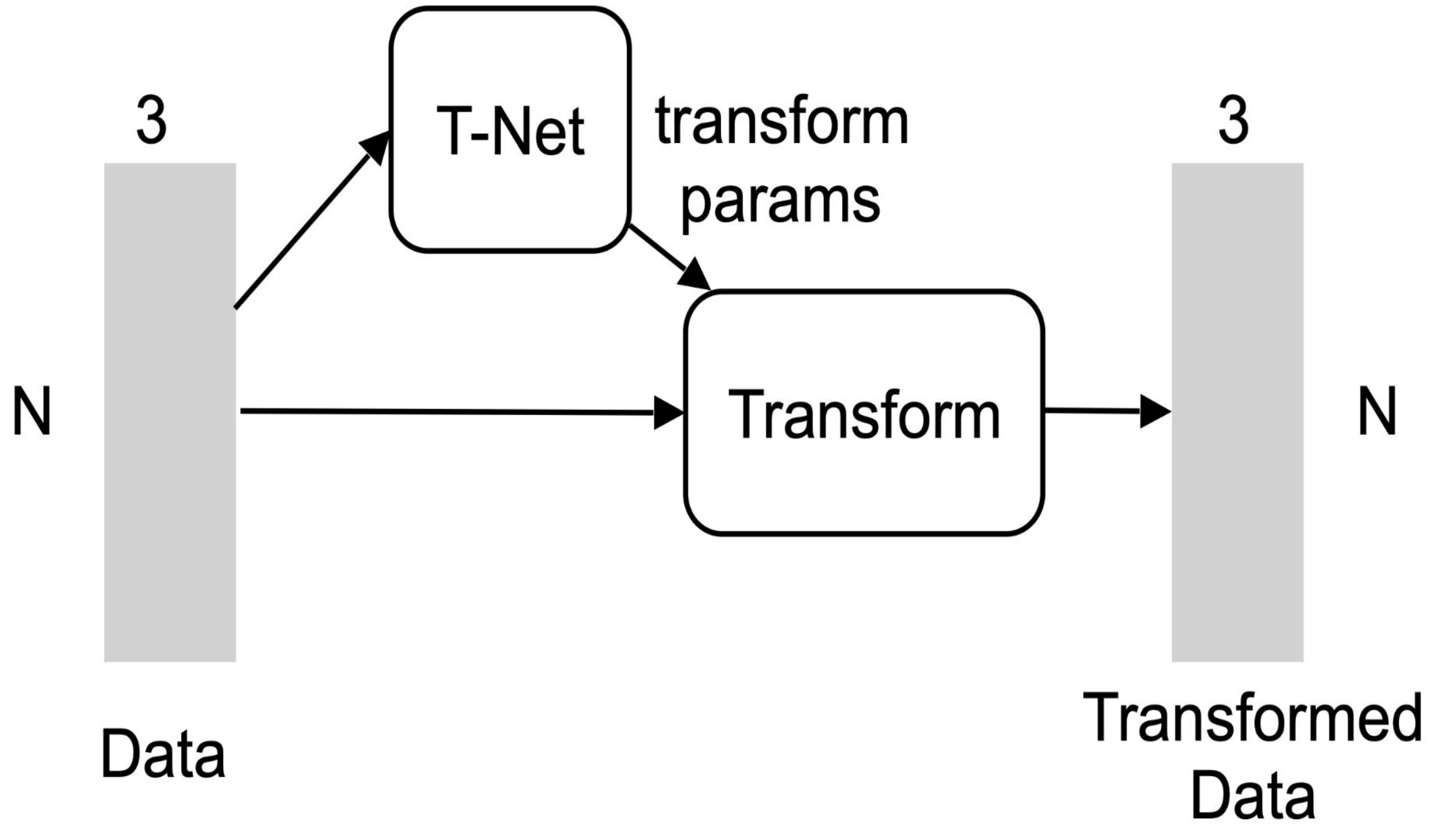
Classification Network of PointNet



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.

Classification Network of PointNet

Input Alignment by T-Network:

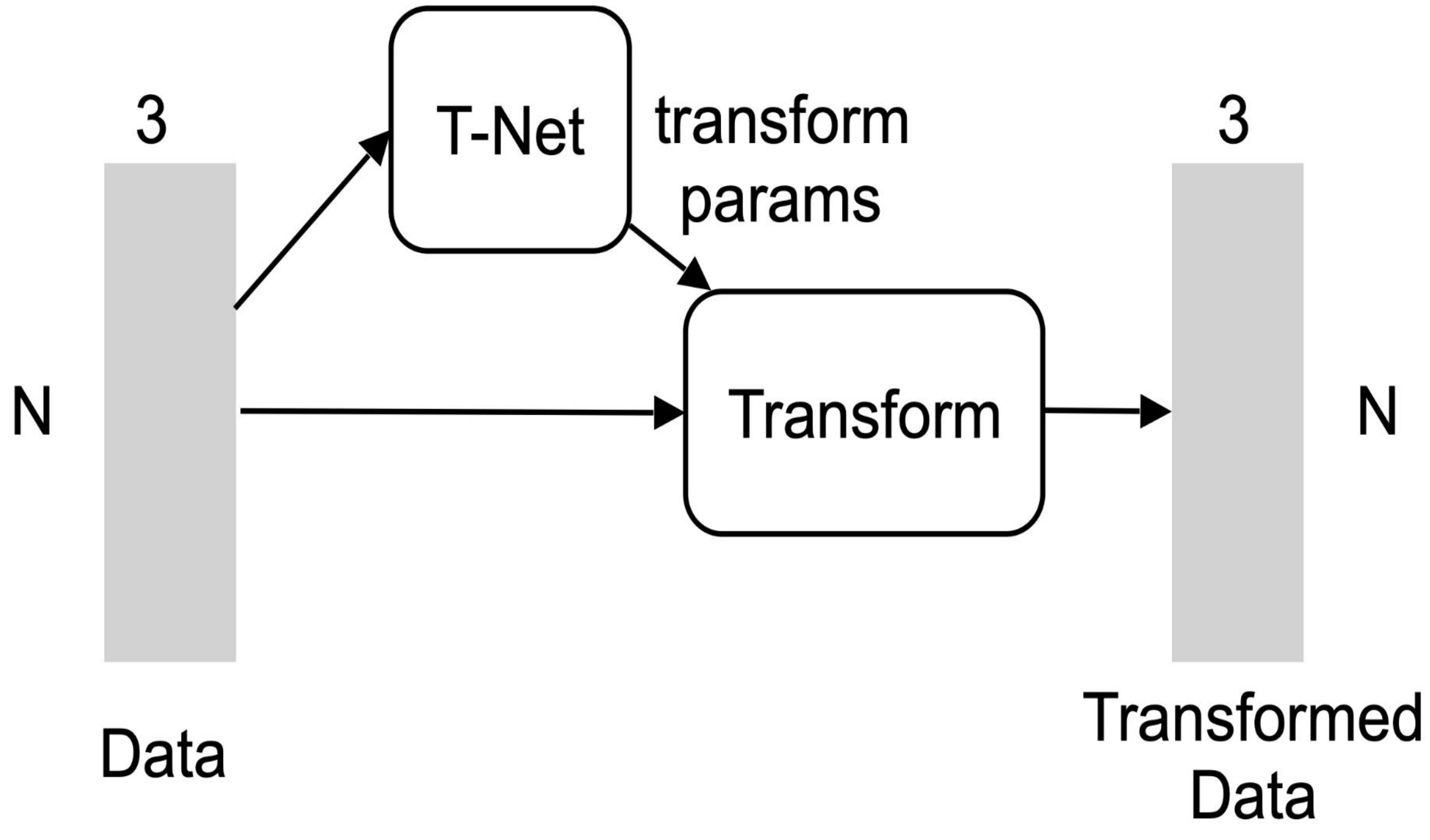


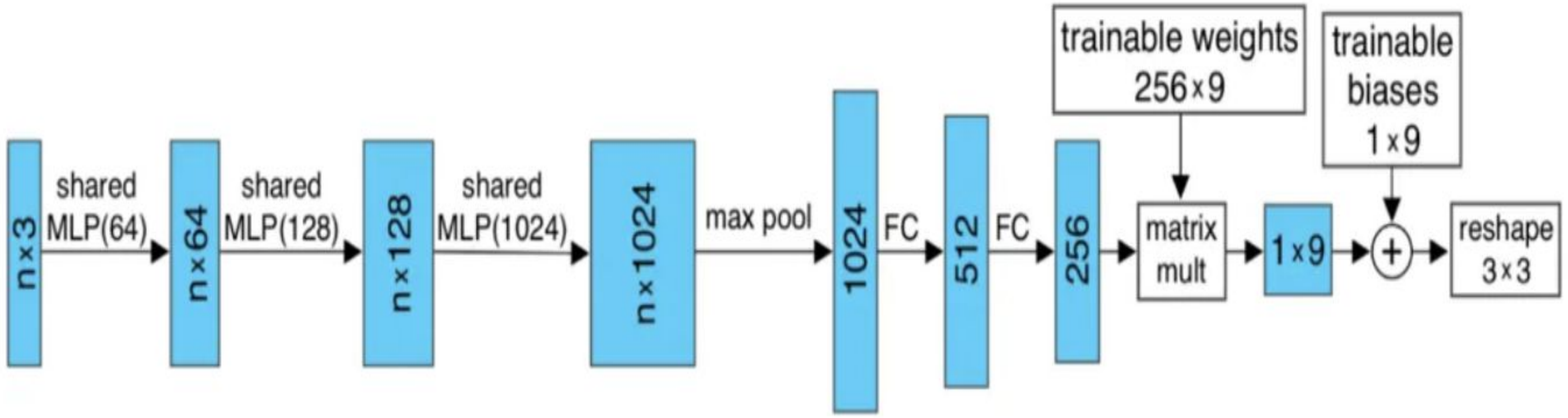
Data dependent transformation for automatic alignment

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.

Classification Network of PointNet

Input Alignment by T-Network:



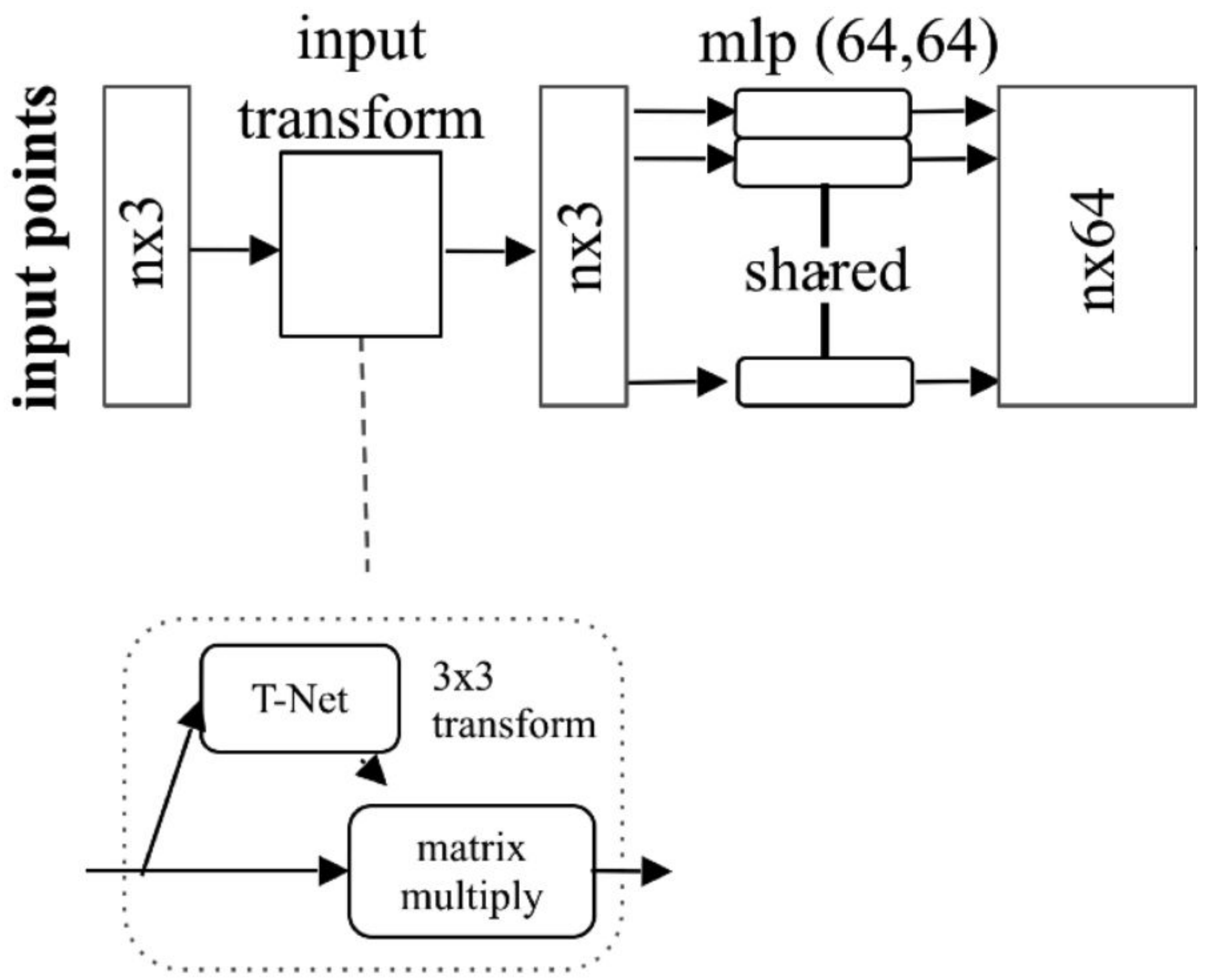


T-Net Architecture

Data dependent transformation for automatic alignment

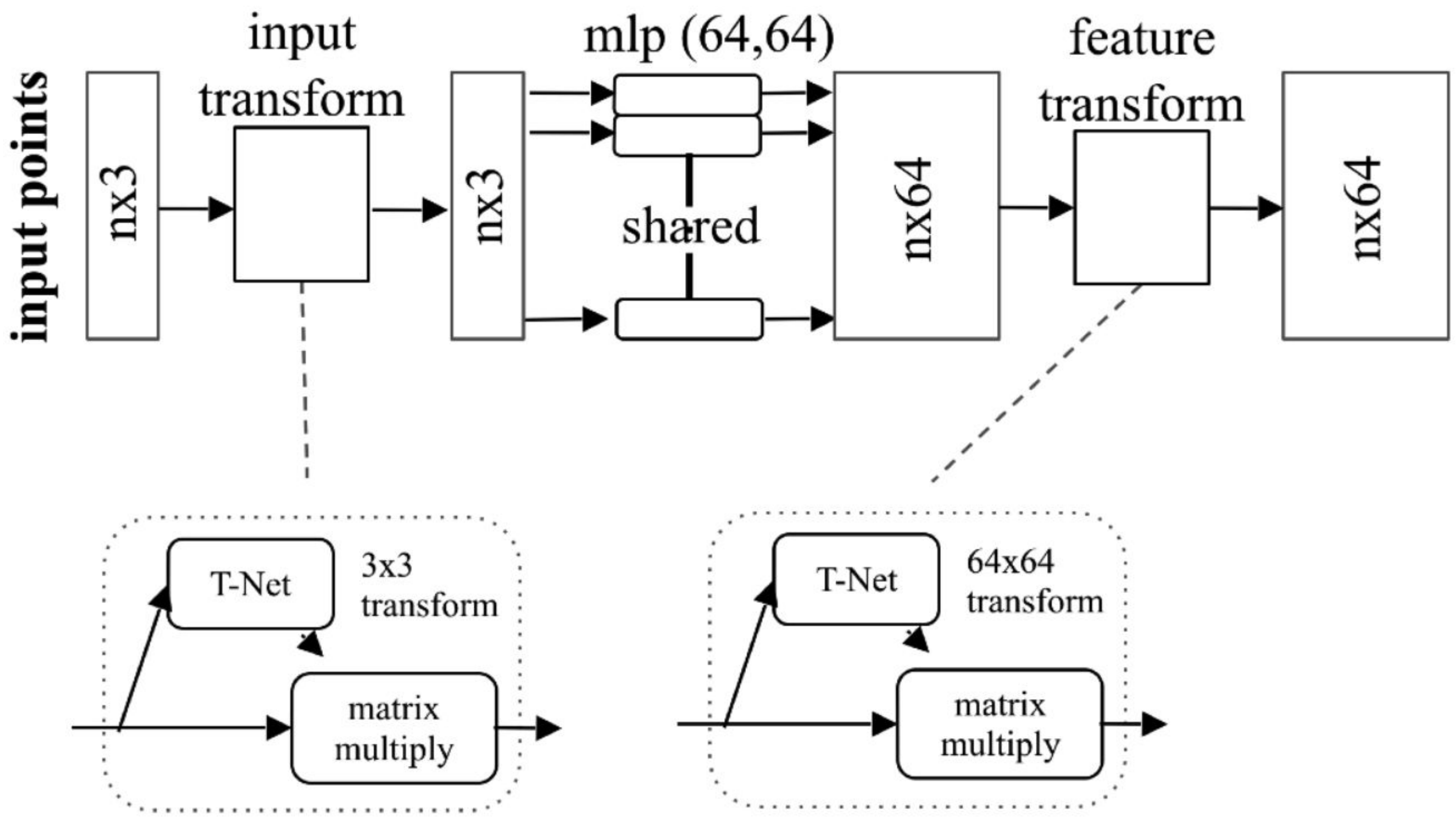
Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Classification Network of 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.

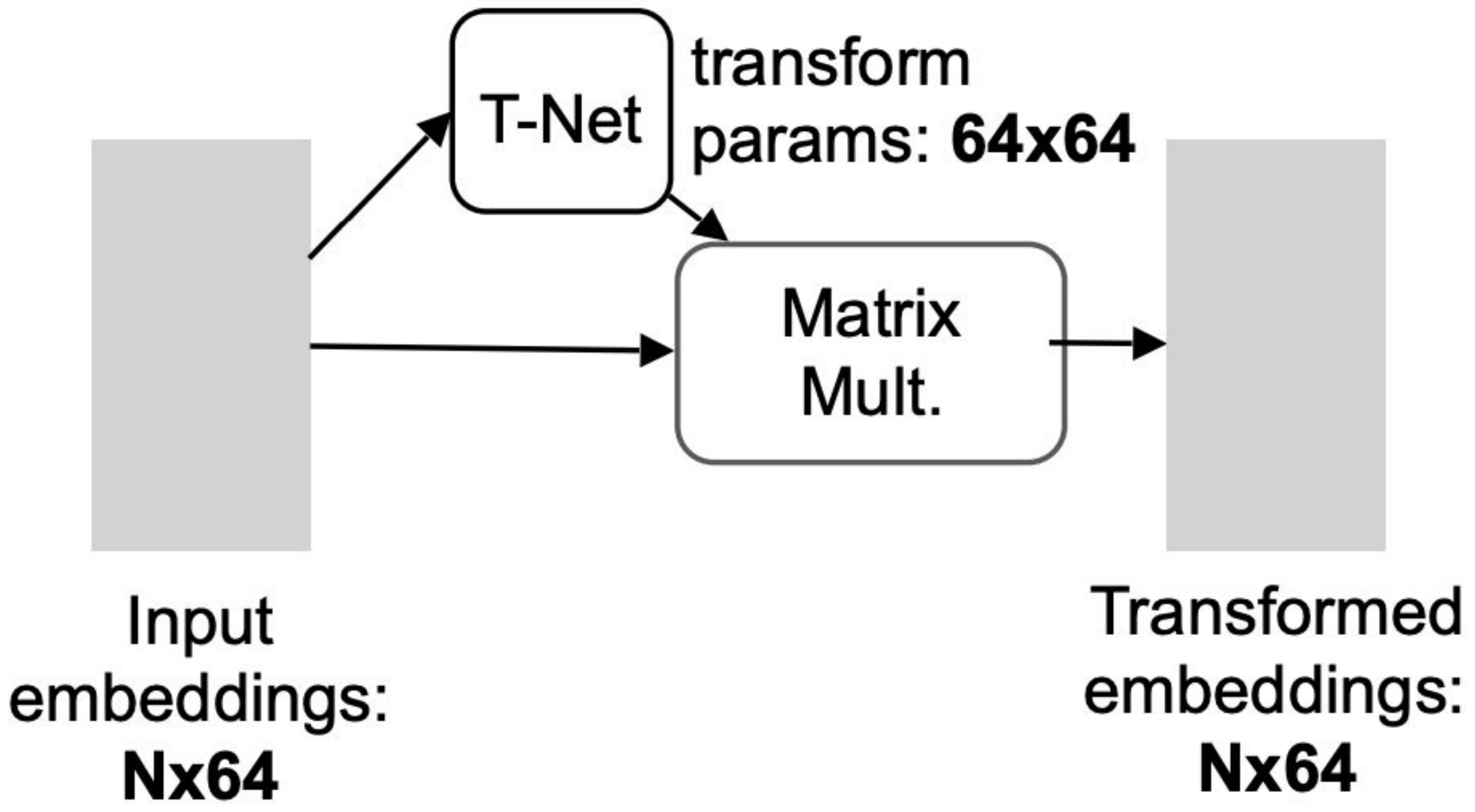
Classification Network of PointNet



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.

Classification Network of PointNet

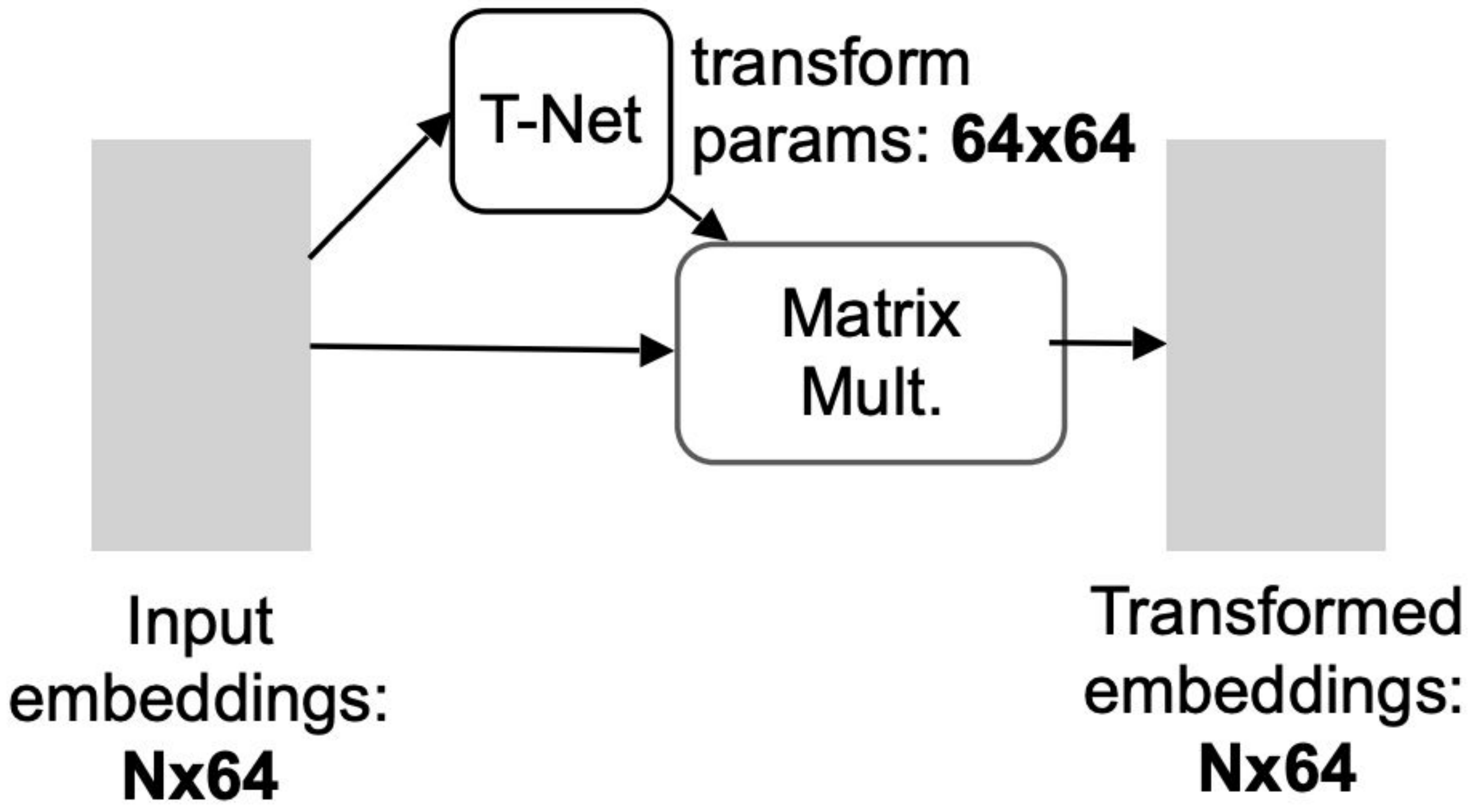
Embedded Space Alignment by T-Network:



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.

Classification Network of PointNet

Embedded Space Alignment by T-Network:



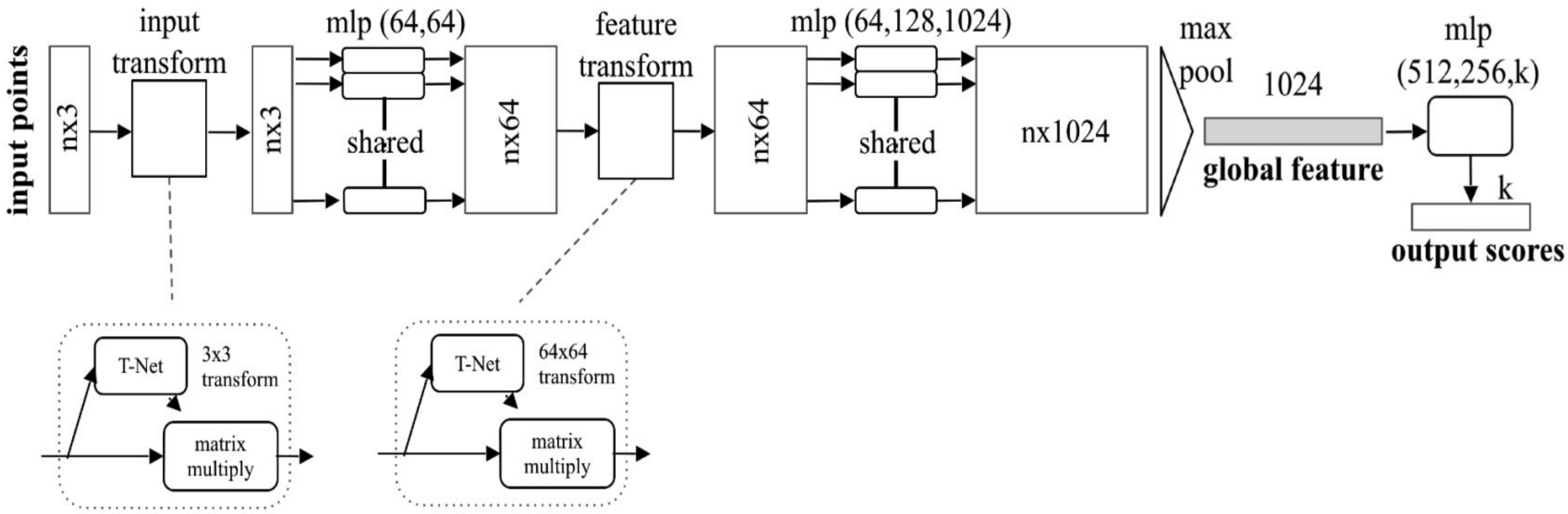
Regularization:

Transform matrix A 64x64 close to orthogonal:

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

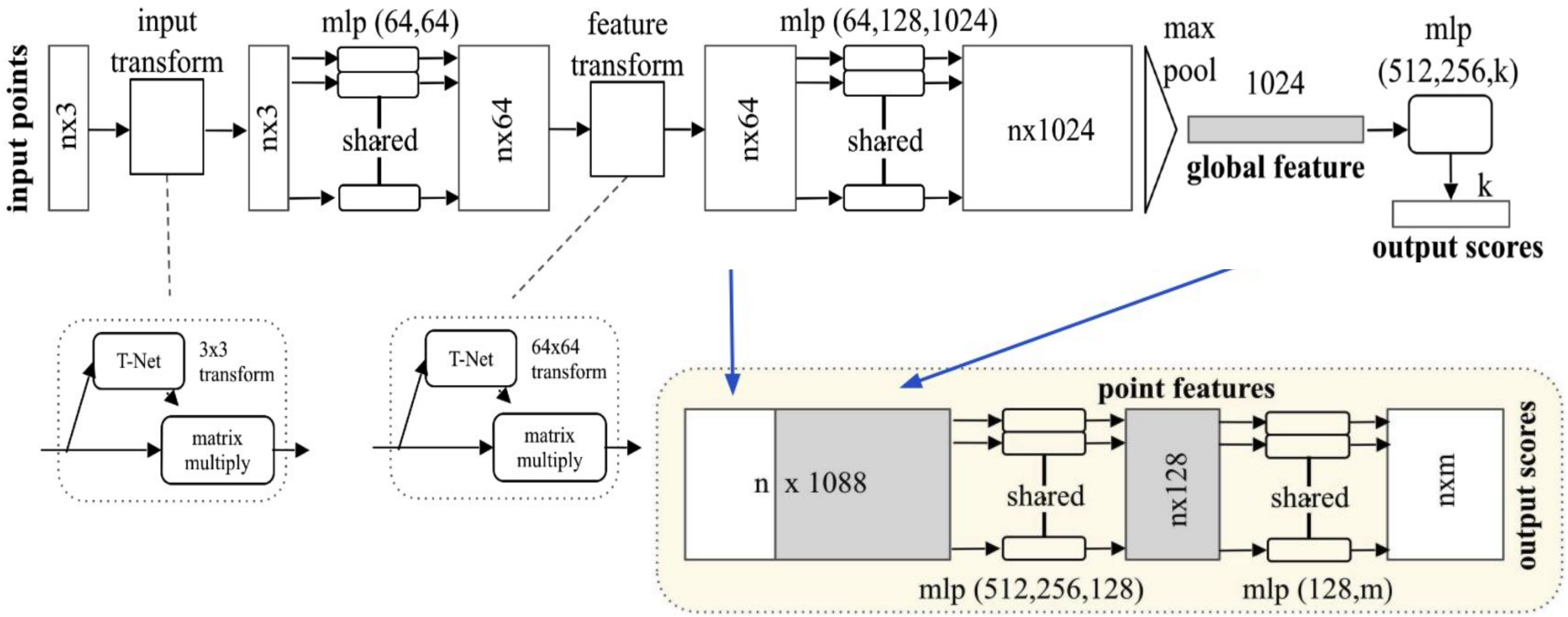
Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Classification Network of 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.

Extension of PointNet Network for Segmentation



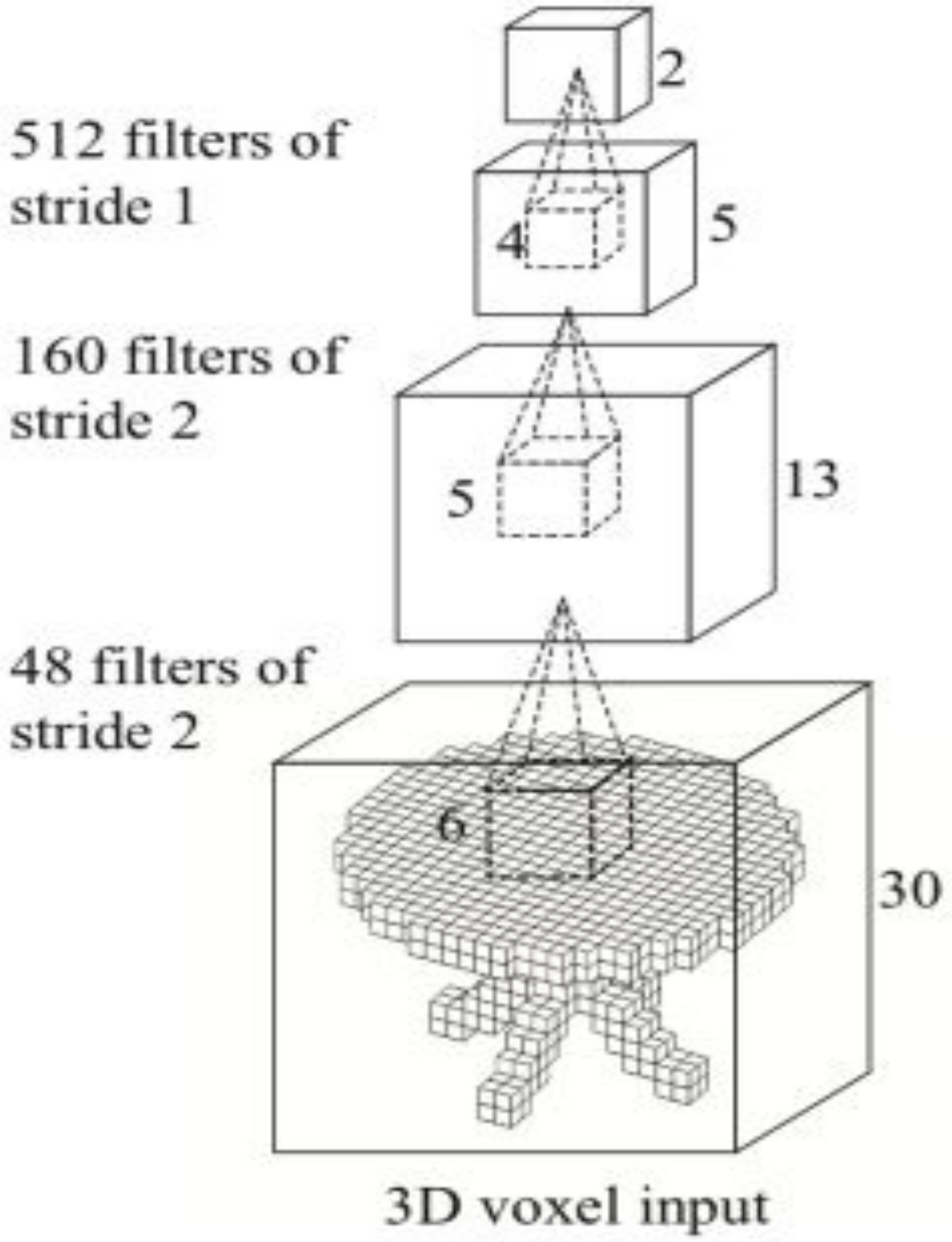
Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

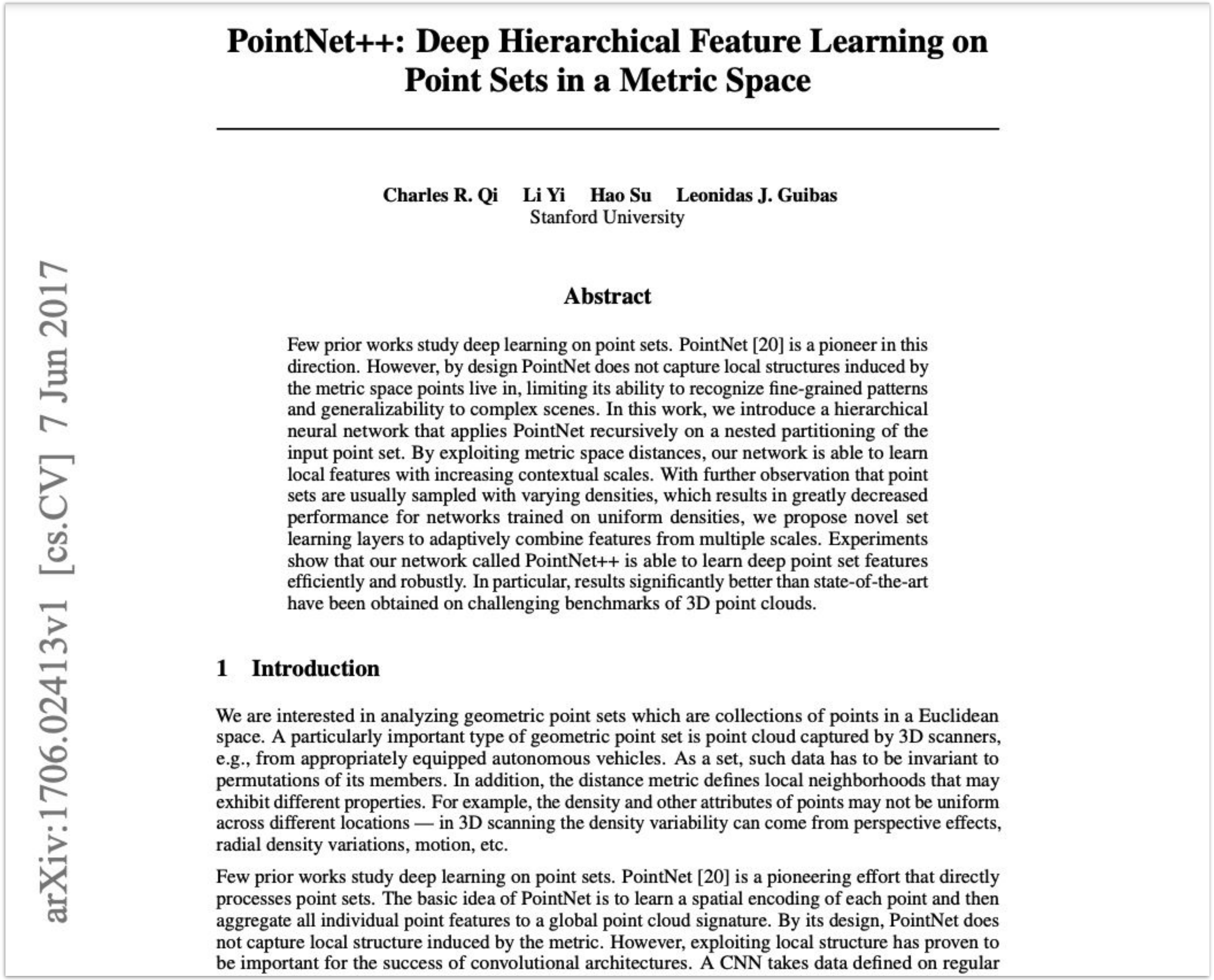
What's missing in PointNet?

Pointnet ++

What's missing in PointNet?

1. Hierarchical feature learning in multiple levels of abstraction





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

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

arXiv:1706.02413v1 [cs.CV] 7 Jun 2017

Abstract

Few prior works study deep learning on point sets. PointNet [20] is a pioneer in this direction. However, by design PointNet does not capture local structures induced by the metric space points live in, limiting its ability to recognize fine-grained patterns and generalizability to complex scenes. In this work, we introduce a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. By exploiting metric space distances, our network is able to learn local features with increasing contextual scales. With further observation that point sets are usually sampled with varying densities, which results in greatly decreased performance for networks trained on uniform densities, we propose novel set learning layers to adaptively combine features from multiple scales. Experiments show that our network called PointNet++ is able to learn deep point set features efficiently and robustly. In particular, results significantly better than state-of-the-art have been obtained on challenging benchmarks of 3D point clouds.

1 Introduction

We are interested in analyzing geometric point sets which are collections of points in a Euclidean space. A particularly important type of geometric point set is point cloud captured by 3D scanners, e.g., from appropriately equipped autonomous vehicles. As a set, such data has to be invariant to permutations of its members. In addition, the distance metric defines local neighborhoods that may exhibit different properties. For example, the density and other attributes of points may not be uniform across different locations — in 3D scanning the density variability can come from perspective effects, radial density variations, motion, etc.

Few prior works study deep learning on point sets. PointNet [20] is a pioneering effort that directly processes point sets. The basic idea of PointNet is to learn a spatial encoding of each point and then aggregate all individual point features to a global point cloud signature. By its design, PointNet does not capture local structure induced by the metric. However, exploiting local structure has proven to be important for the success of convolutional architectures. A CNN takes data defined on regular

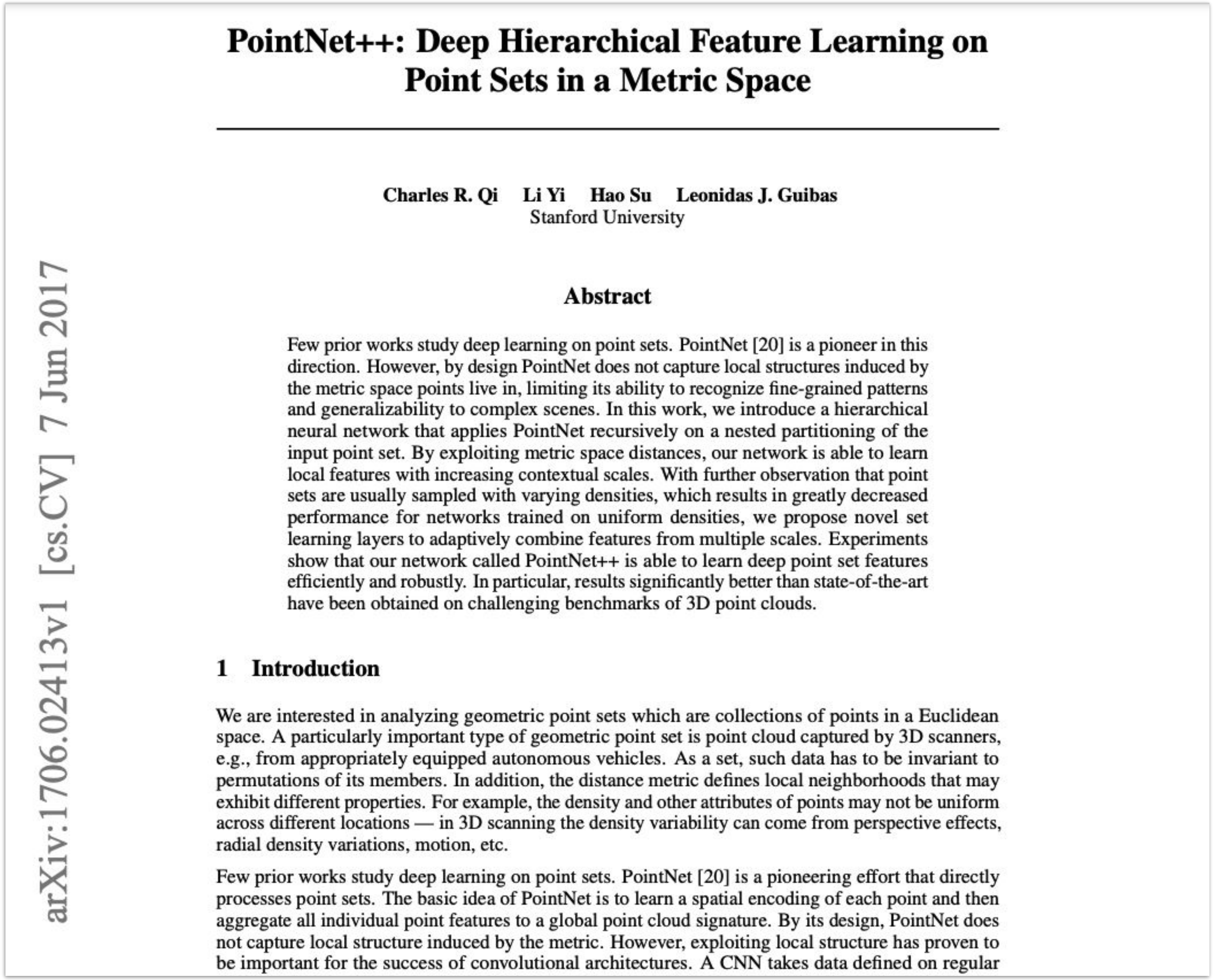
Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Pointnet ++

What’s missing in PointNet?

1. Hierarchical feature learning in multiple levels of abstraction

2. Robust feature learning for Non-Uniform Sampling Density



arXiv:1706.02413v1 [cs.CV] 7 Jun 2017

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

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

Abstract

Few prior works study deep learning on point sets. PointNet [20] is a pioneer in this direction. However, by design PointNet does not capture local structures induced by the metric space points live in, limiting its ability to recognize fine-grained patterns and generalizability to complex scenes. In this work, we introduce a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. By exploiting metric space distances, our network is able to learn local features with increasing contextual scales. With further observation that point sets are usually sampled with varying densities, which results in greatly decreased performance for networks trained on uniform densities, we propose novel set learning layers to adaptively combine features from multiple scales. Experiments show that our network called PointNet++ is able to learn deep point set features efficiently and robustly. In particular, results significantly better than state-of-the-art have been obtained on challenging benchmarks of 3D point clouds.

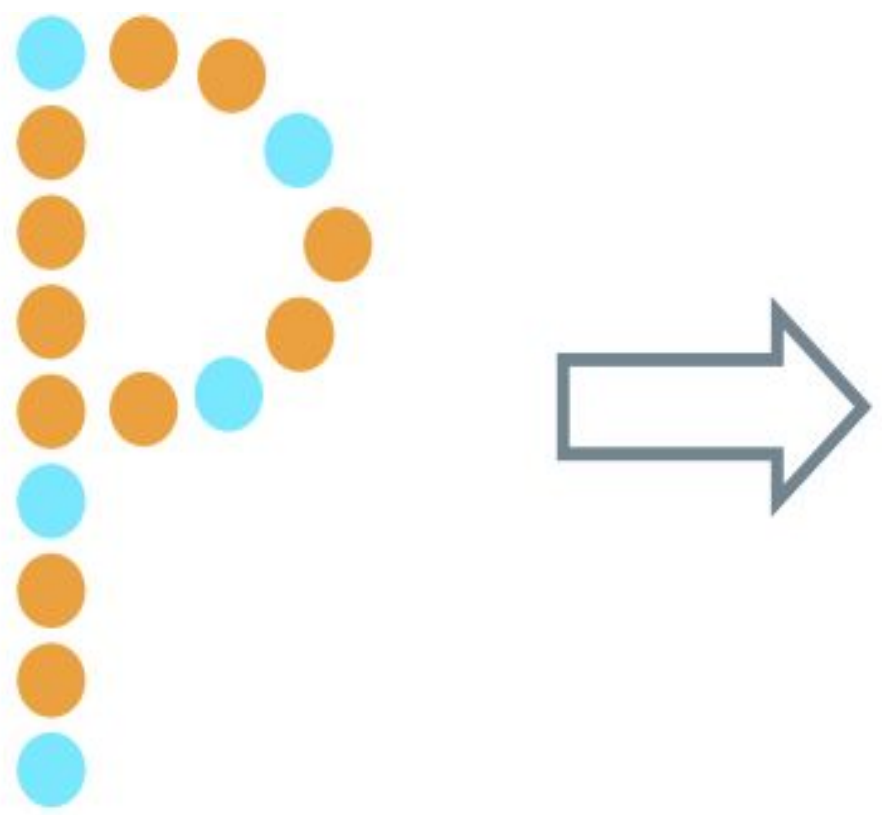
1 Introduction

We are interested in analyzing geometric point sets which are collections of points in a Euclidean space. A particularly important type of geometric point set is point cloud captured by 3D scanners, e.g., from appropriately equipped autonomous vehicles. As a set, such data has to be invariant to permutations of its members. In addition, the distance metric defines local neighborhoods that may exhibit different properties. For example, the density and other attributes of points may not be uniform across different locations — in 3D scanning the density variability can come from perspective effects, radial density variations, motion, etc.

Few prior works study deep learning on point sets. PointNet [20] is a pioneering effort that directly processes point sets. The basic idea of PointNet is to learn a spatial encoding of each point and then aggregate all individual point features to a global point cloud signature. By its design, PointNet does not capture local structure induced by the metric. However, exploiting local structure has proven to be important for the success of convolutional architectures. A CNN takes data defined on regular

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Hierarchical Point Set Feature learning



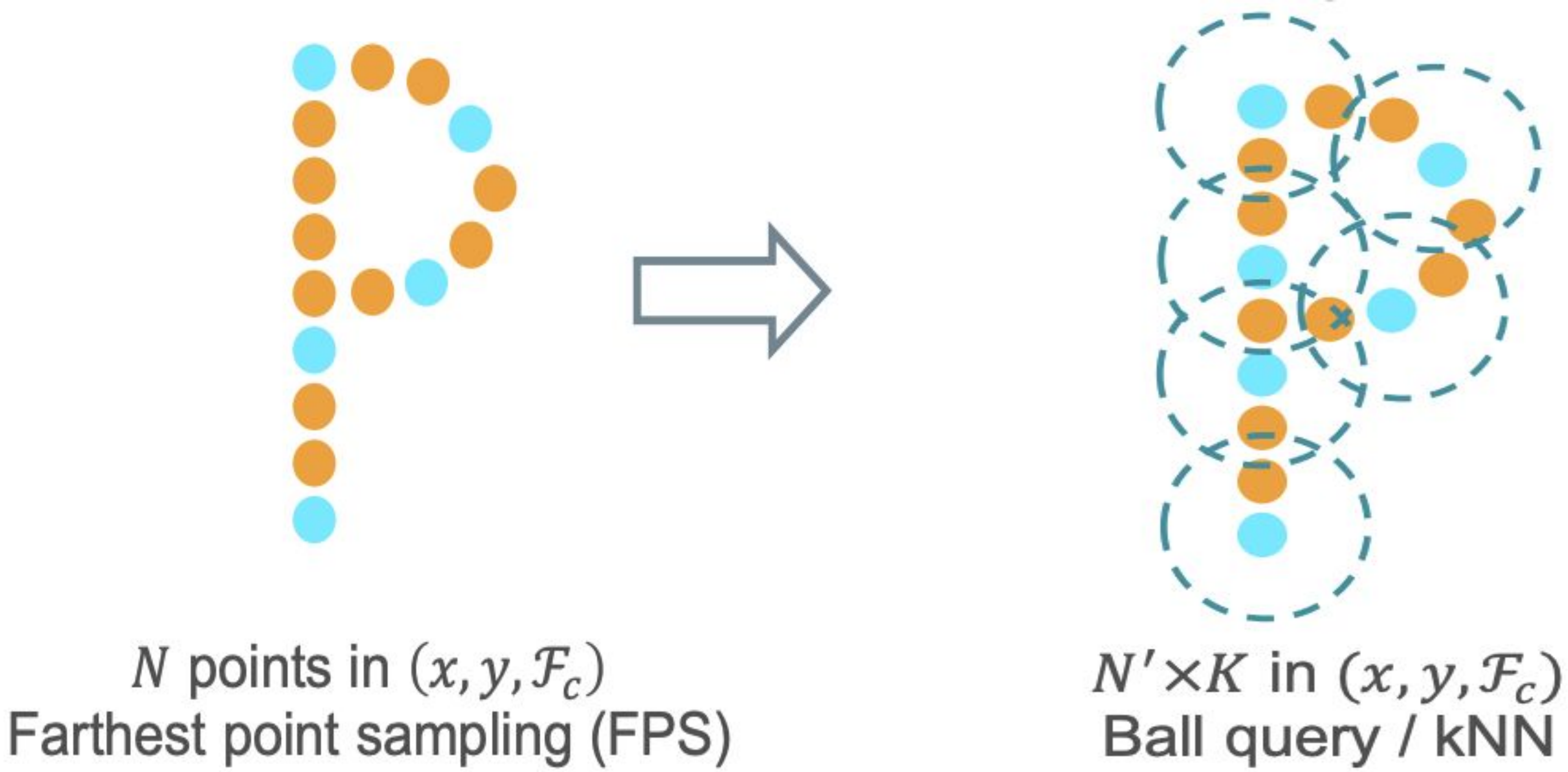
N points in $(x, y, \mathcal{F}_c)$
Farthest point sampling (FPS)

Layers of a set abstraction level:

1. Sampling Layer: Iterative Farthest Point Sampling (FPS)

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Hierarchical point set Feature learning

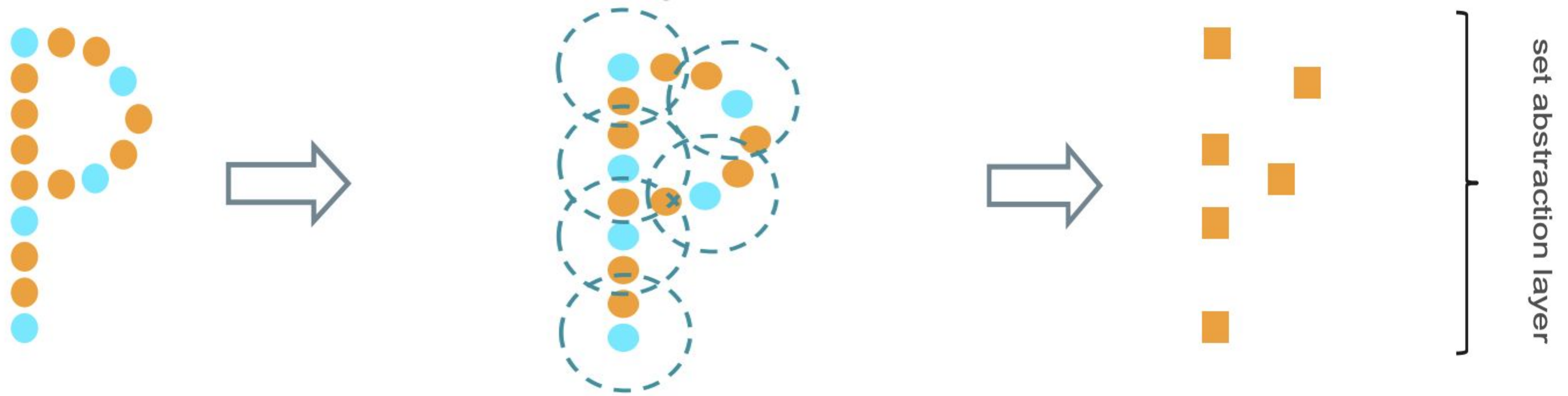


Layers of a set abstraction level:

1. **Sampling Layer:** Iterative Farthest Point Sampling (FPS)
2. **Grouping Layer:** Select points for each neighborhood centroid

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Hierarchical point set Feature learning



N points in $(x, y, \mathcal{F}_c)$
Farthest point sampling (FPS)

$N' \times K$ in $(x, y, \mathcal{F}_c)$
Ball query / kNN

N' points in $(x, y, \mathcal{F}_{c'})$
PointNet

Layers of a set abstraction level:

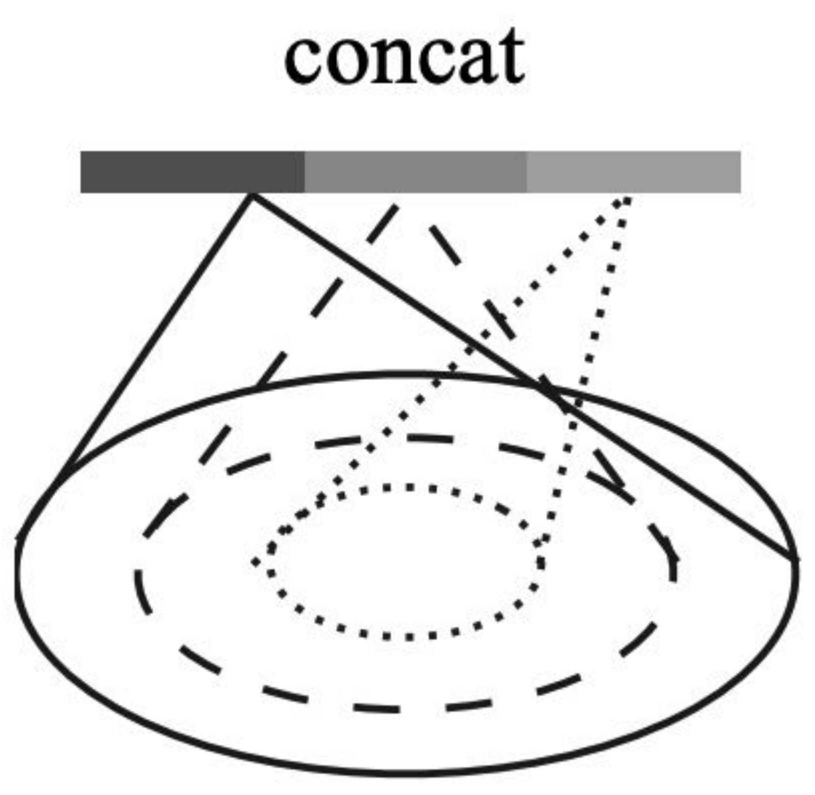
1. **Sampling Layer:** Iterative Farthest Point Sampling (FPS)
2. **Grouping Layer:** Select points for each neighborhood centroid
3. **PointNet Layer:** Applies a small PointNet to a given set of points for feature extraction

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Non-uniform Sampling Density

Proposed two types of density layers:

1. Multi-scale grouping (MSG):
   a. Applies grouping layers with different scales
   b. Random input dropout



Multi-Scale Grouping

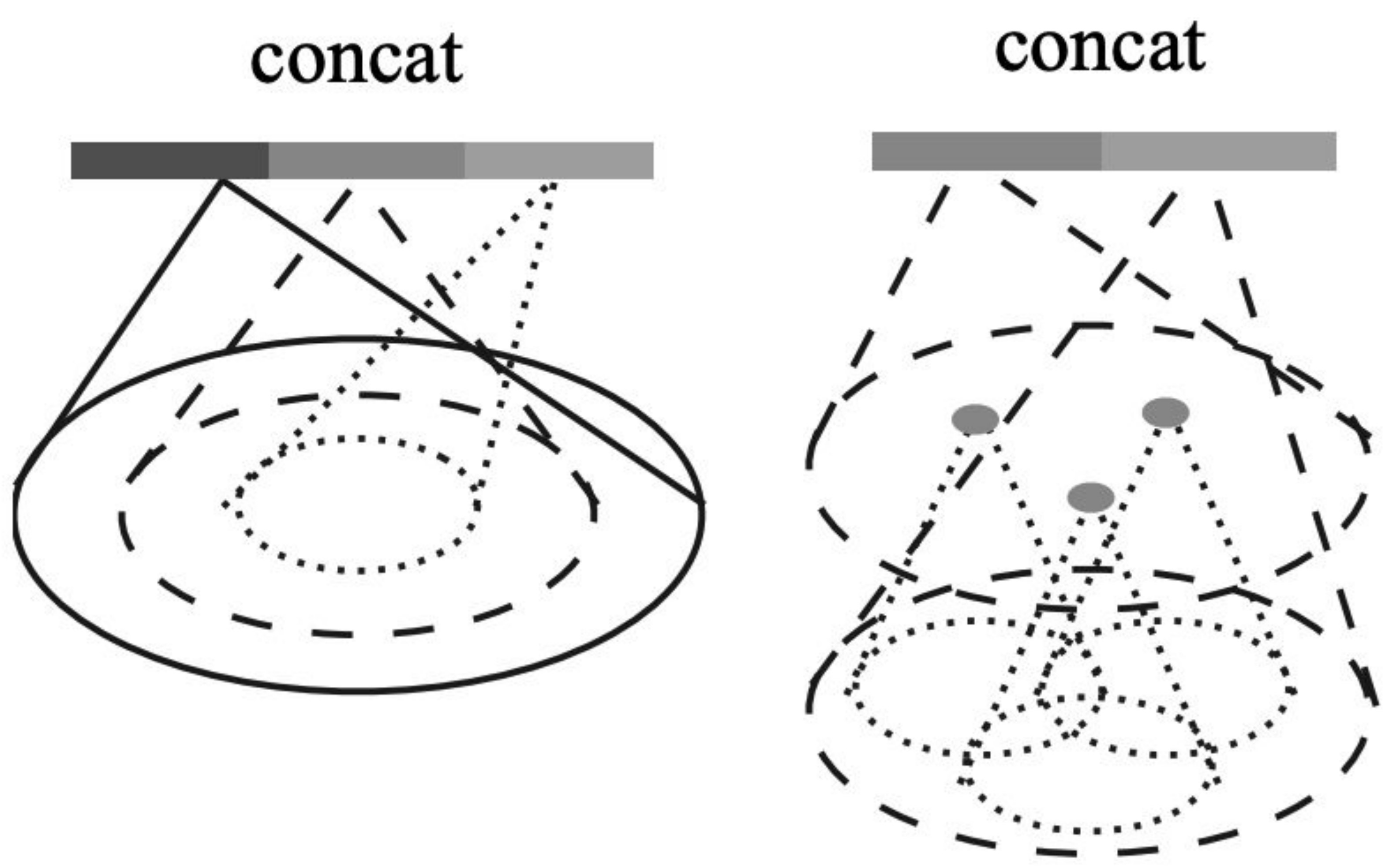
Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Non-uniform Sampling Density

Proposed two types of density layers:

1. Multi-scale grouping (MSG):
   a. Applies grouping layers with different scales
   b. Random input dropout

2. Multi-resolution grouping (MRG):
   a. Summarizes features from lower level and process all raw points in the local region

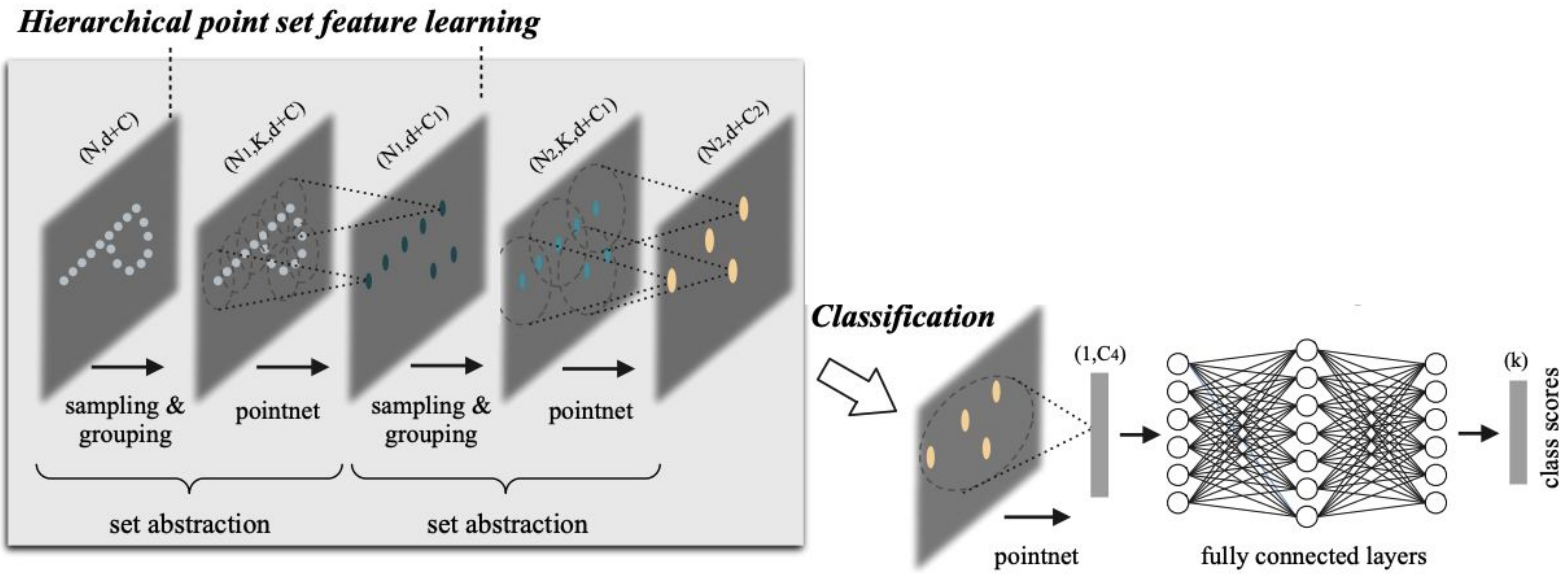


Multi-Scale Grouping

Multi-Resolution Grouping

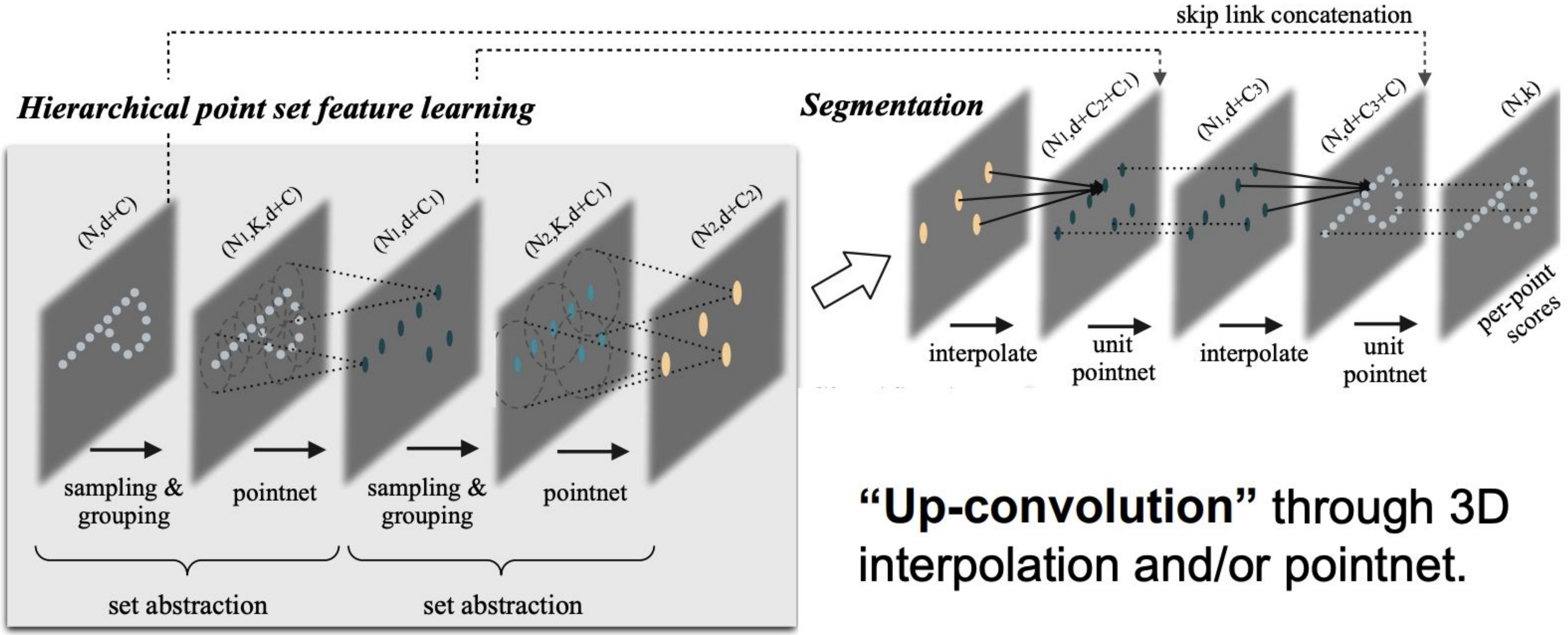
Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Pointnet ++ for Classification



Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

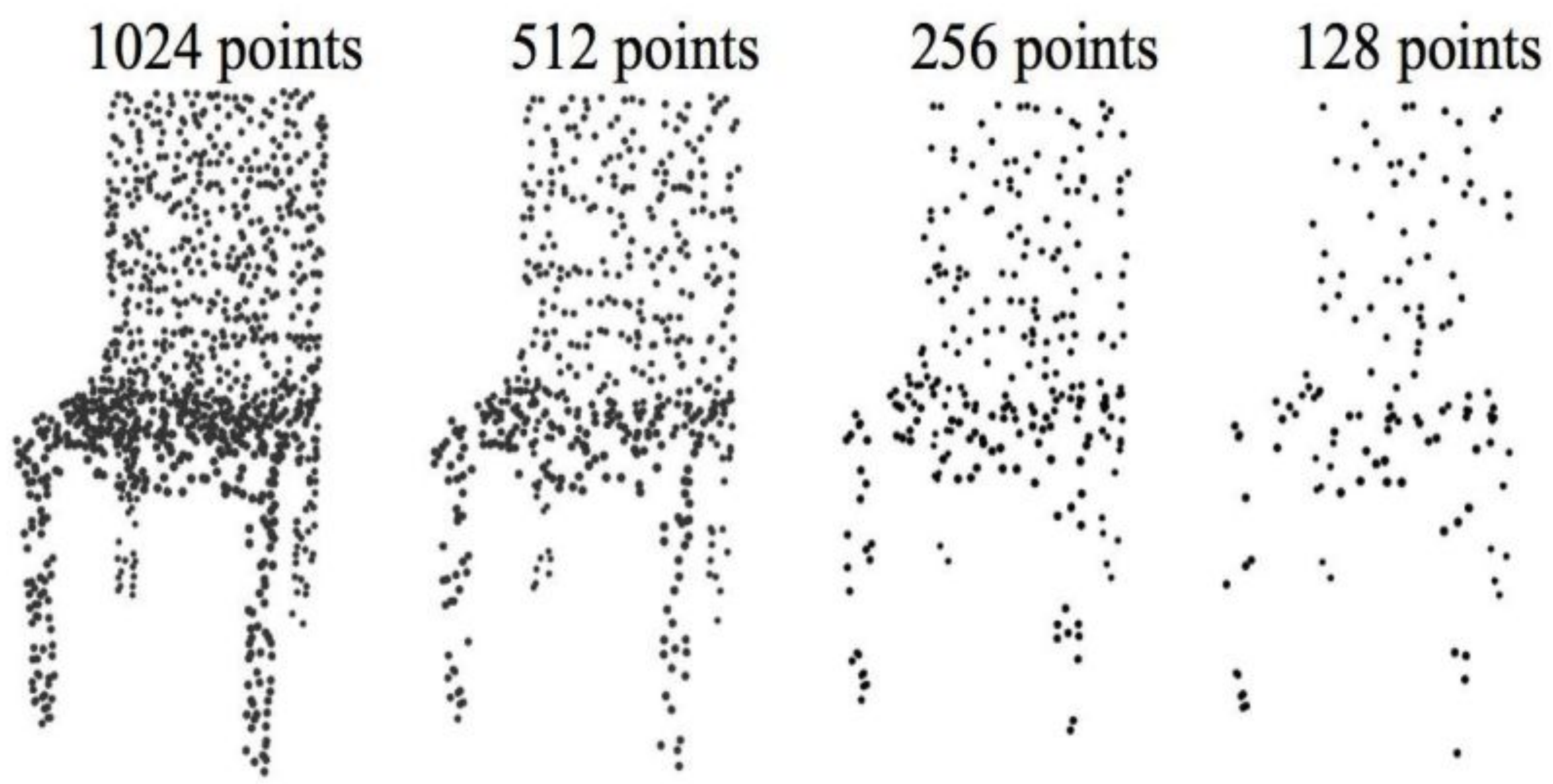
PointNet++ for Segmentation



"Up-convolution" through 3D interpolation and/or pointnet.

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Pointnet vs Pointnet++



Examples of a point cloud (in this case, a chair) represented with different numbers of points:

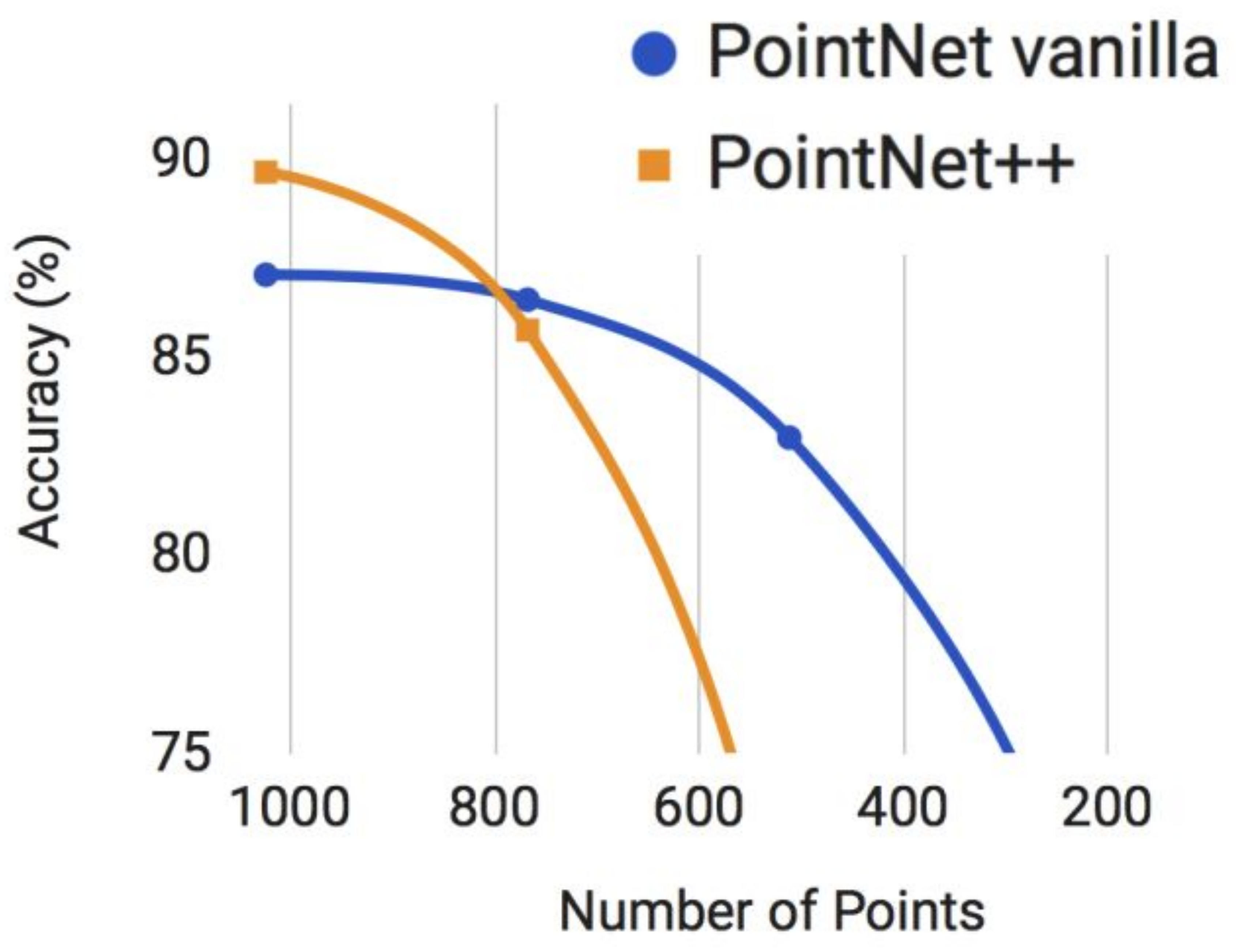
- **1024 points**: High-resolution point cloud with a dense distribution of points.
- **512 points**: Reduced resolution; still captures most of the shape details.
- **256 points**: Lower resolution; fewer points, but the basic structure is still discernible.
- **128 points**: Very sparse; only a rough outline of the chair is visible.

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Pointnet vs Pointnet++

PointNet :

- PointNet's accuracy drops significantly as point count decreases.
- This decline shows PointNet's sensitivity to lower-resolution point clouds.
- Without explicit local feature capture, PointNet struggles with shape recognition in low-density data.
- Accuracy falls sharply when points drop below ~800.

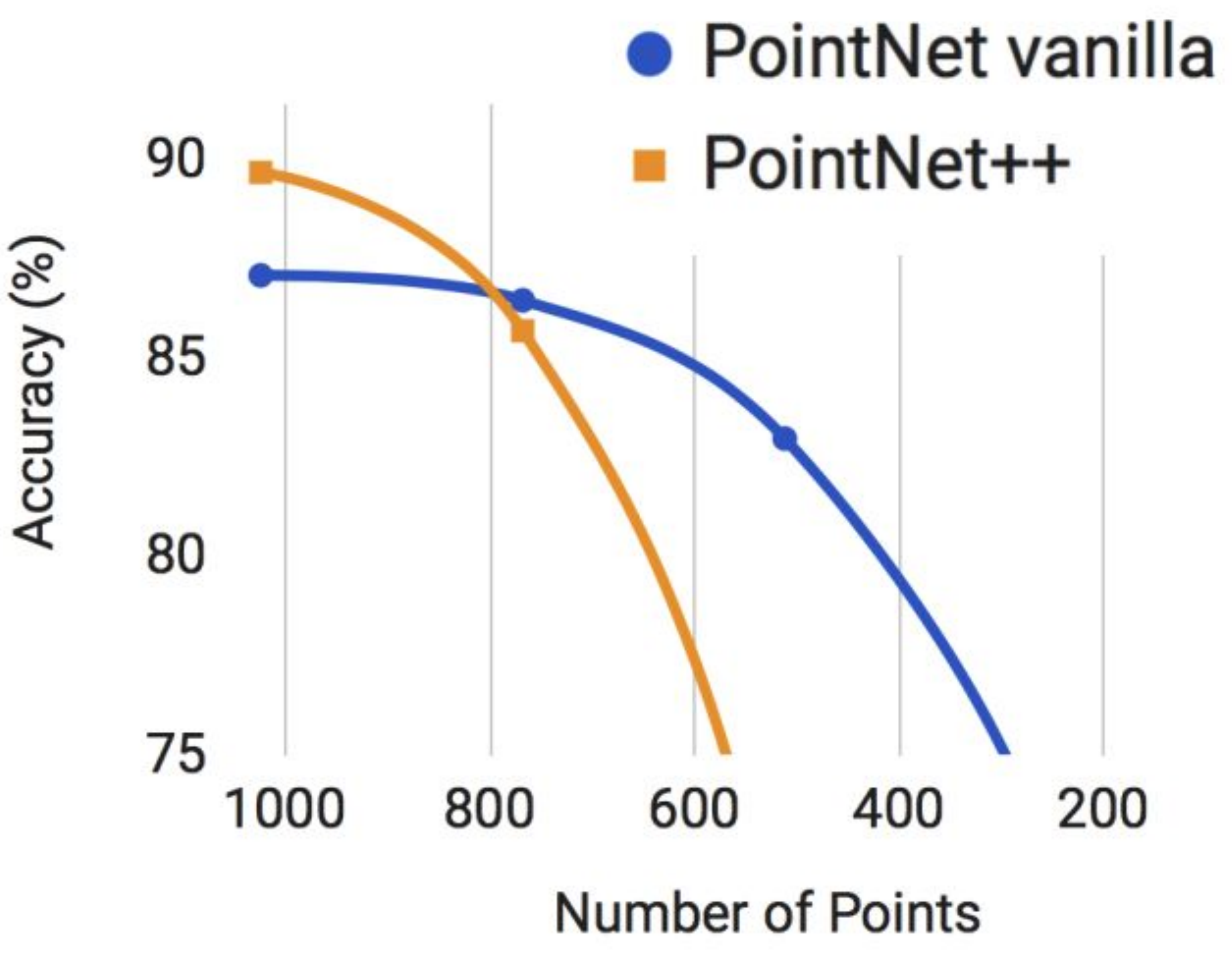


Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Pointnet vs Pointnet++

PointNet++ :

- PointNet's accuracy drops significantly as point count decreases.
- This decline shows PointNet's sensitivity to lower-resolution point clouds.
- Without explicit local feature capture, PointNet struggles with shape recognition in low-density data.
- Accuracy falls sharply when points drop below ~800.



Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Pointnet vs Pointnet++

When to Use PointNet :

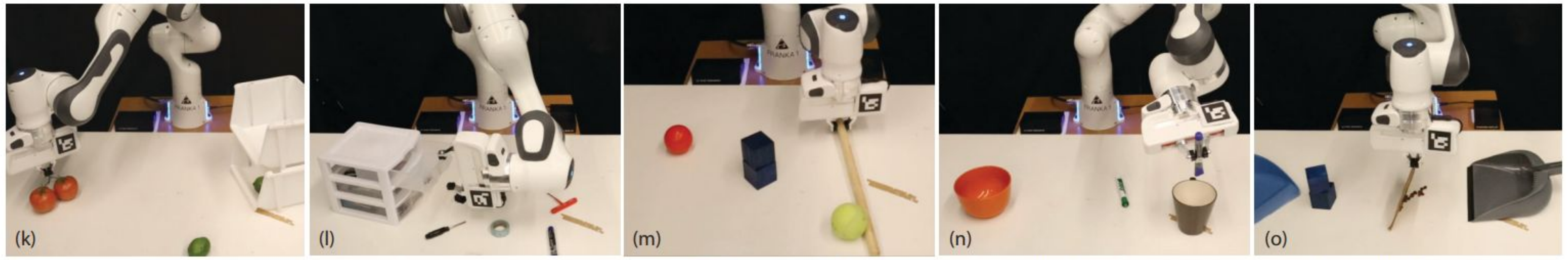
- Simple Objects or Environments
- Densely-Sampled or High-Resolution Point Clouds
- Denser, Lightweight Applications

When to Use PointNet++:

- Complex Objects or Environments
- Sparse or Non-Uniformly Sampled Point Clouds
- Applications Requiring Local Detail and Contextual Hierarchy

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

Let us now discuss about 3D data based imitation learning for manipulation



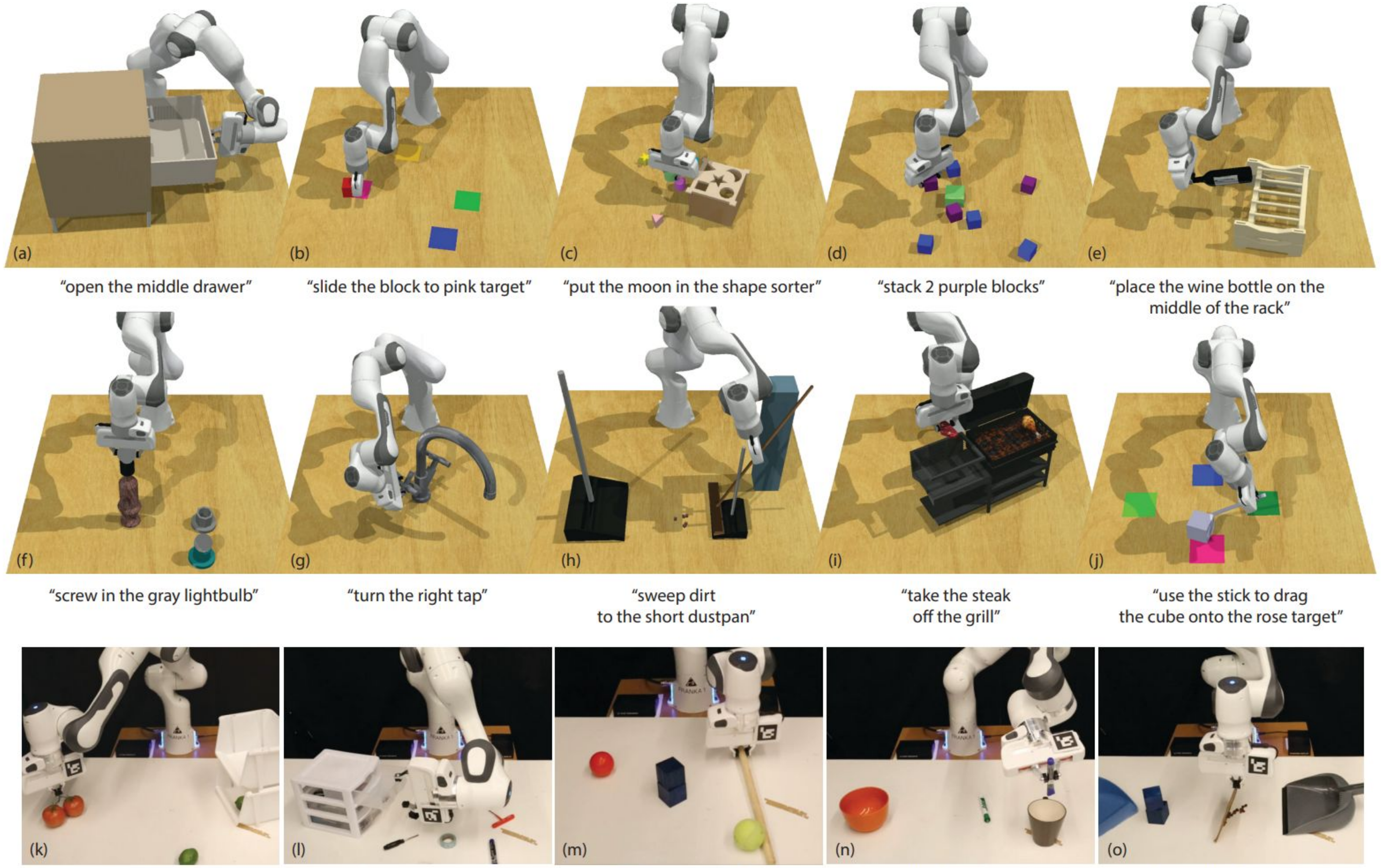
"put the tomatoes in the top bin" "put the tape in the top drawer" "hit the green ball with the stick" "place the blue whiteboard marker in the mug" "sweep the beans into the gray dustpan"

Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation

Transformer for 3D

Mohit Shridhar[1], Lucas Manuelli[2], Dieter Fox[1, 2]
[1]University of Washington, [2]NVIDIA

Multi Task



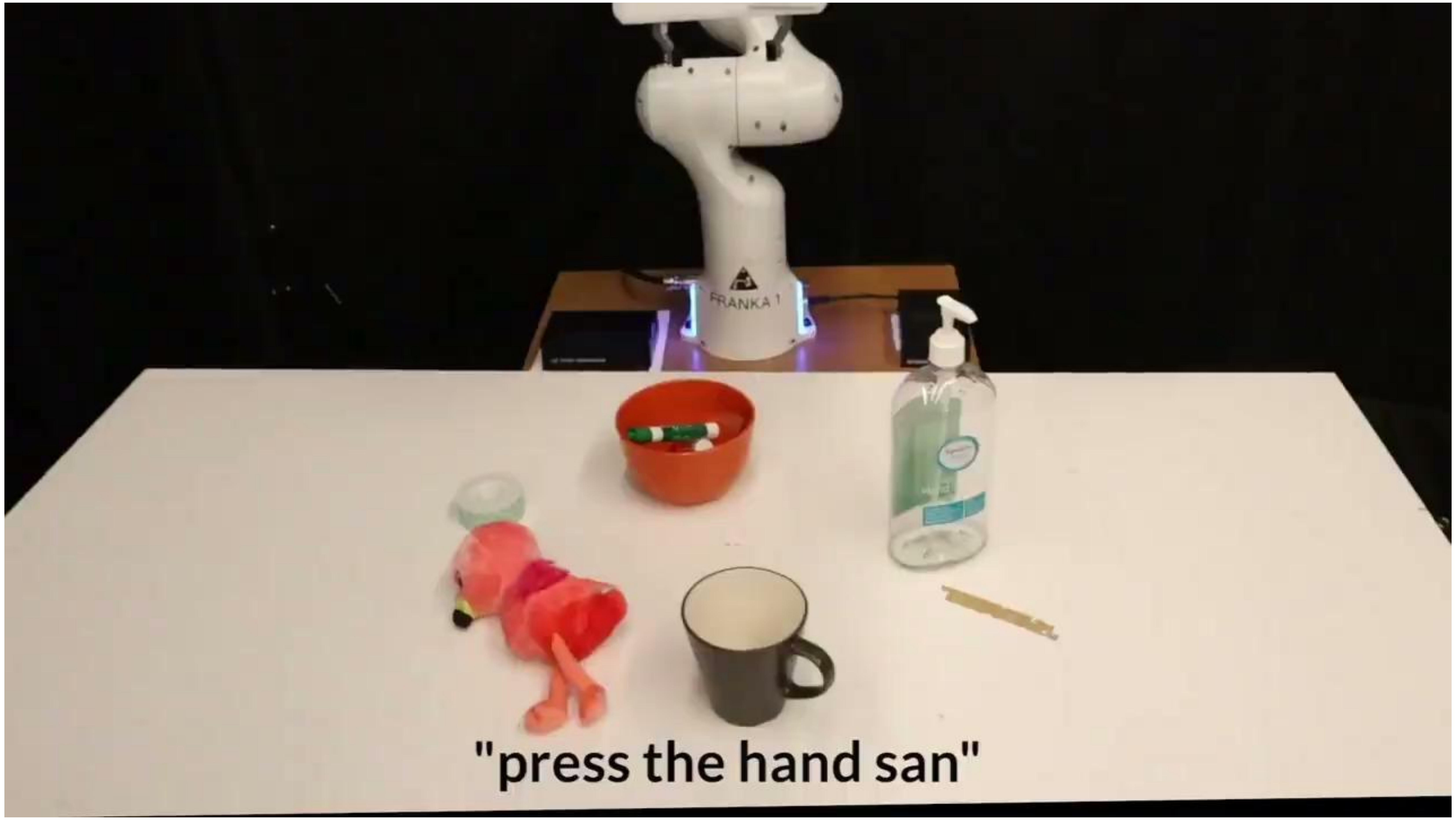
"open the middle drawer" "slide the block to pink target" "put the moon in the shape sorter" "stack 2 purple blocks" "place the wine bottle on the middle of the rack"

"screw in the gray lightbulb" "turn the right tap" "sweep dirt to the short dustpan" "take the steak off the grill" "use the stick to drag the cube onto the rose target"

"put the tomatoes in the top bin" "put the tape in the top drawer" "hit the green ball with the stick" "place the blue whiteboard marker in the mug" "sweep the beans into the gray dustpan"

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

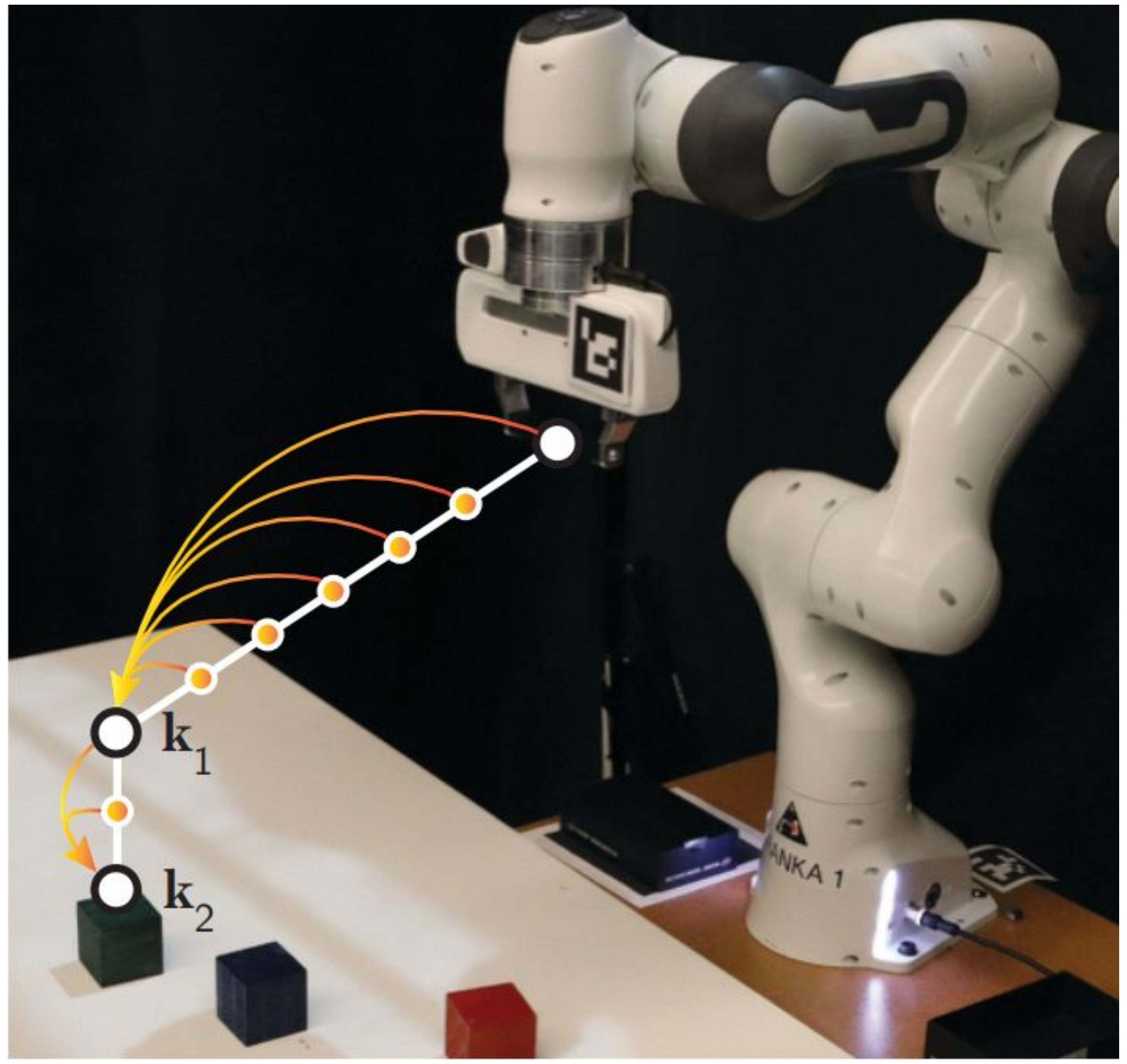
Perceiver-Actor

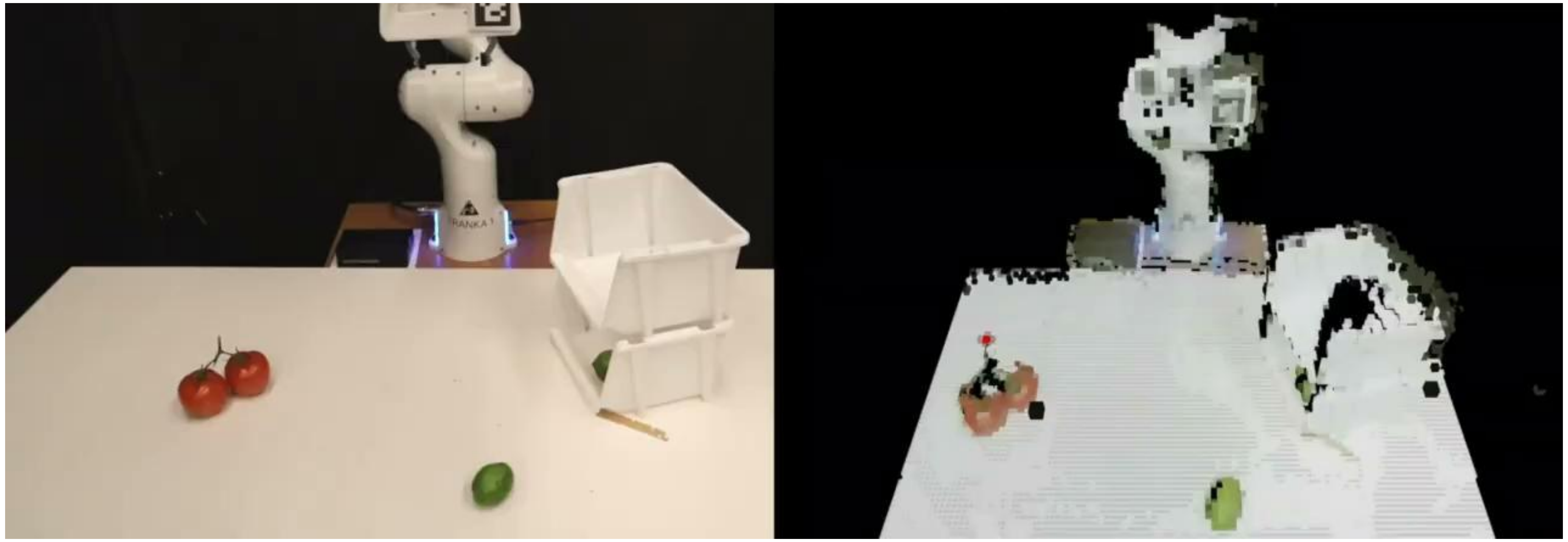


Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Perceiver-Actor

Multi-task 6-DoF manipulation agent
End-to-end few-shot imitation learning
Input: RGB-D Voxels & Language Goal
Output: Discretized 6-DoF action
+ open/close





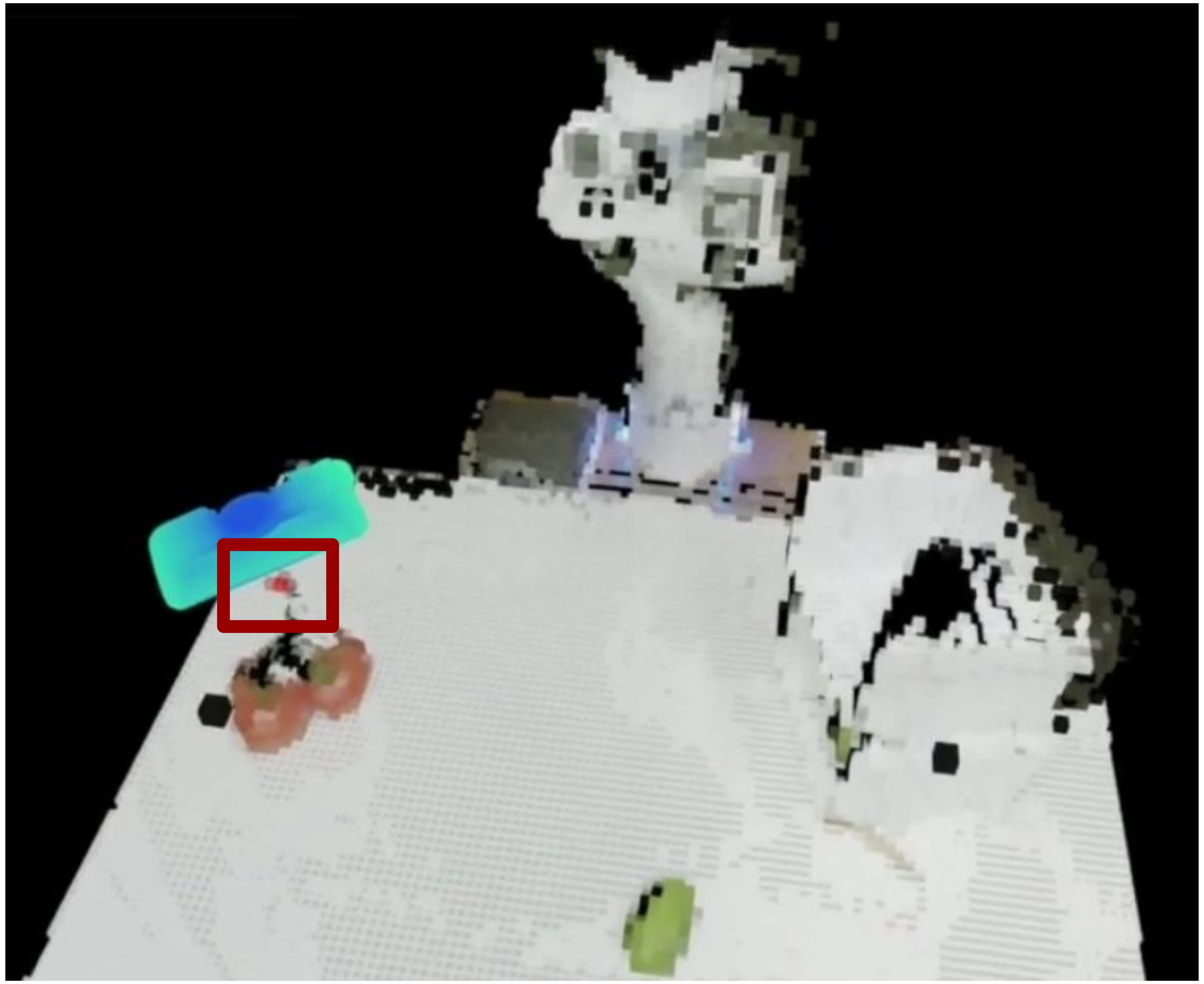
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Perceiver-Actor

Observation space almost equivalent to Action space
detects actions, not objects.

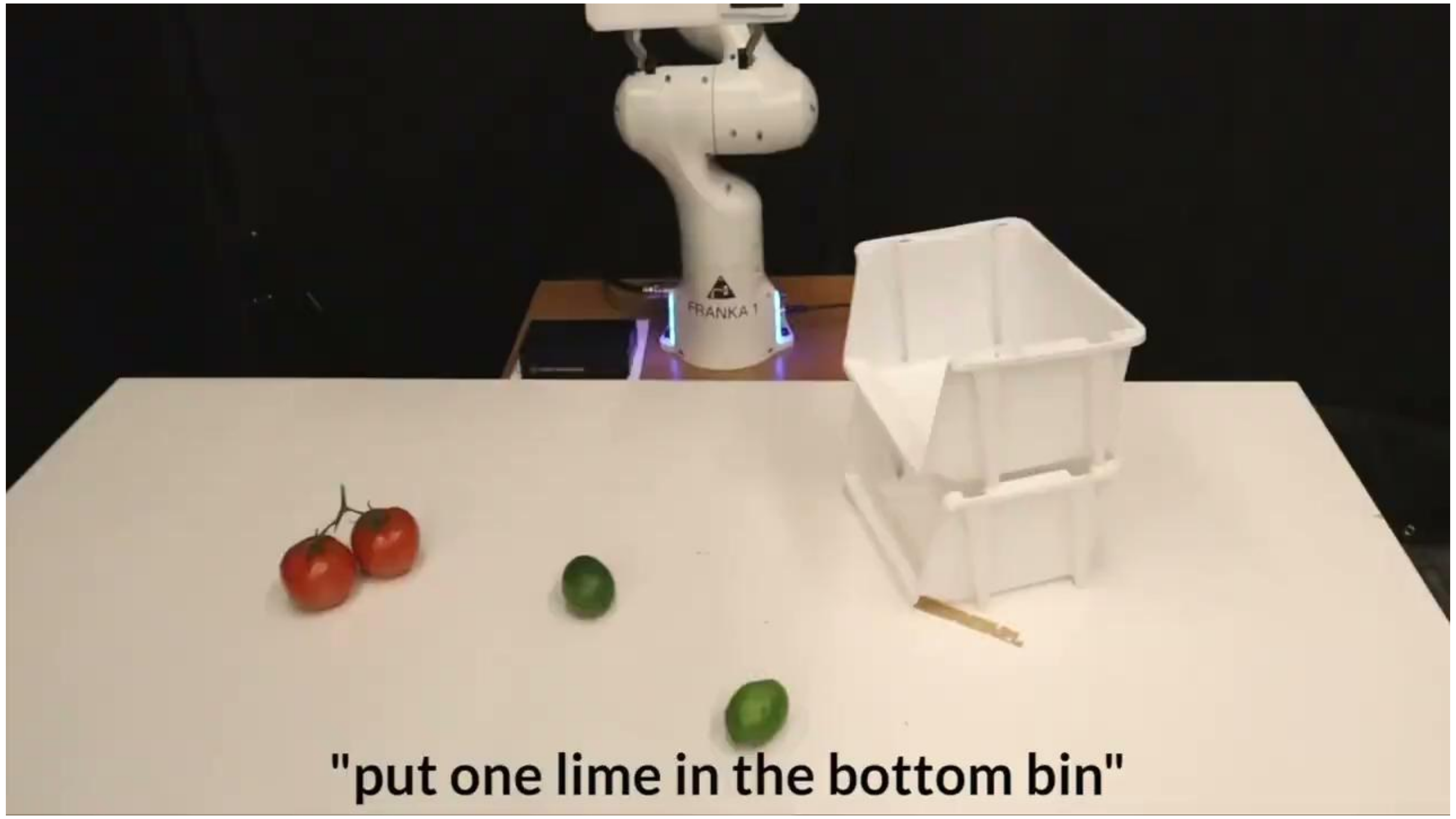
Predicts the Keyframe which is projected and highlighted in red contains end effector translation

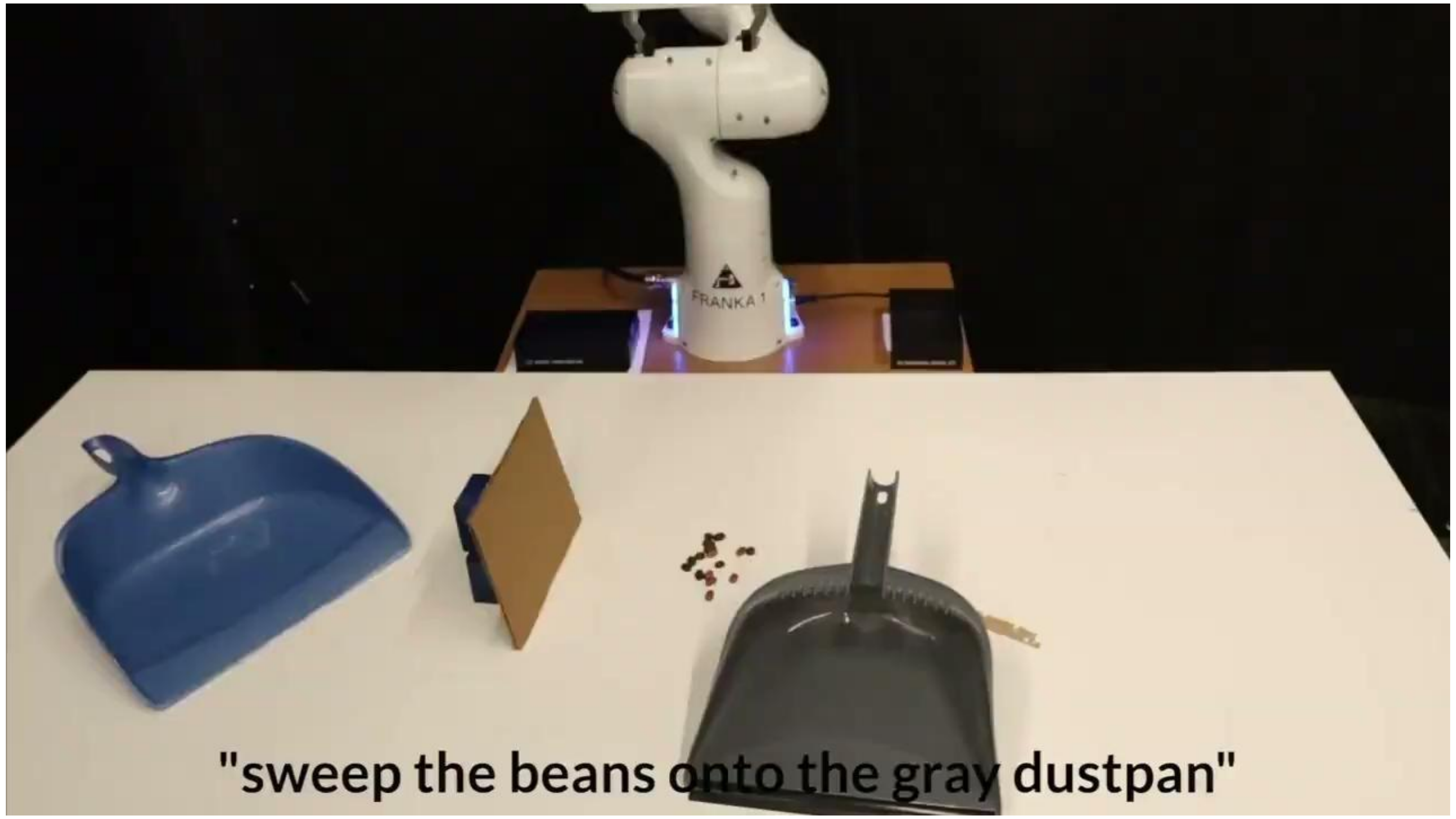
The green with blue projection signifies rotation, gripper state and collision



Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Why to predict action instead of detecting object state?





Difficult to estimate object state for tomato stem and scattered beans

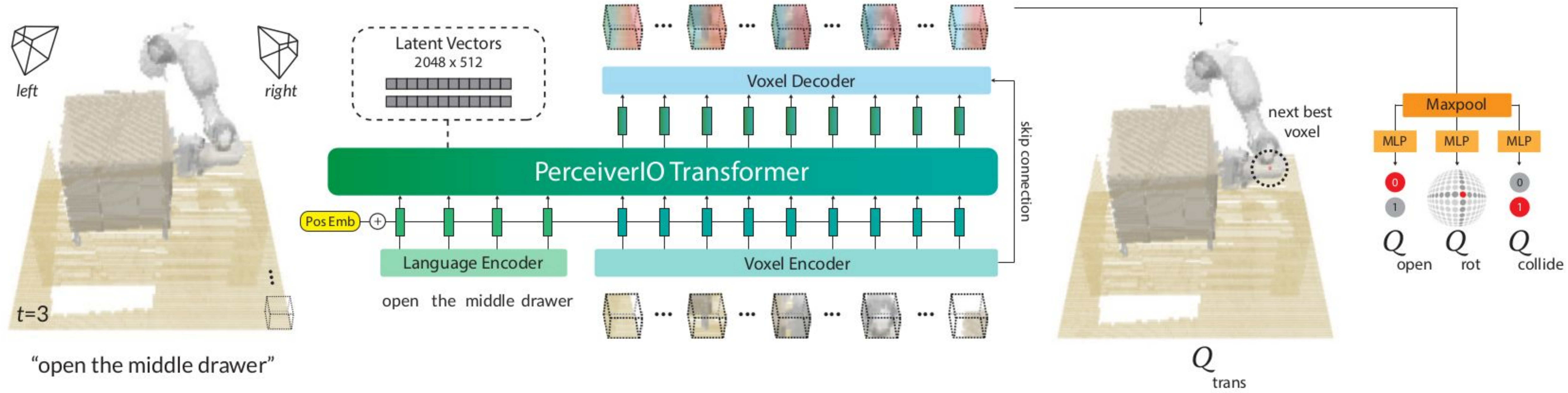
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Inference time

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

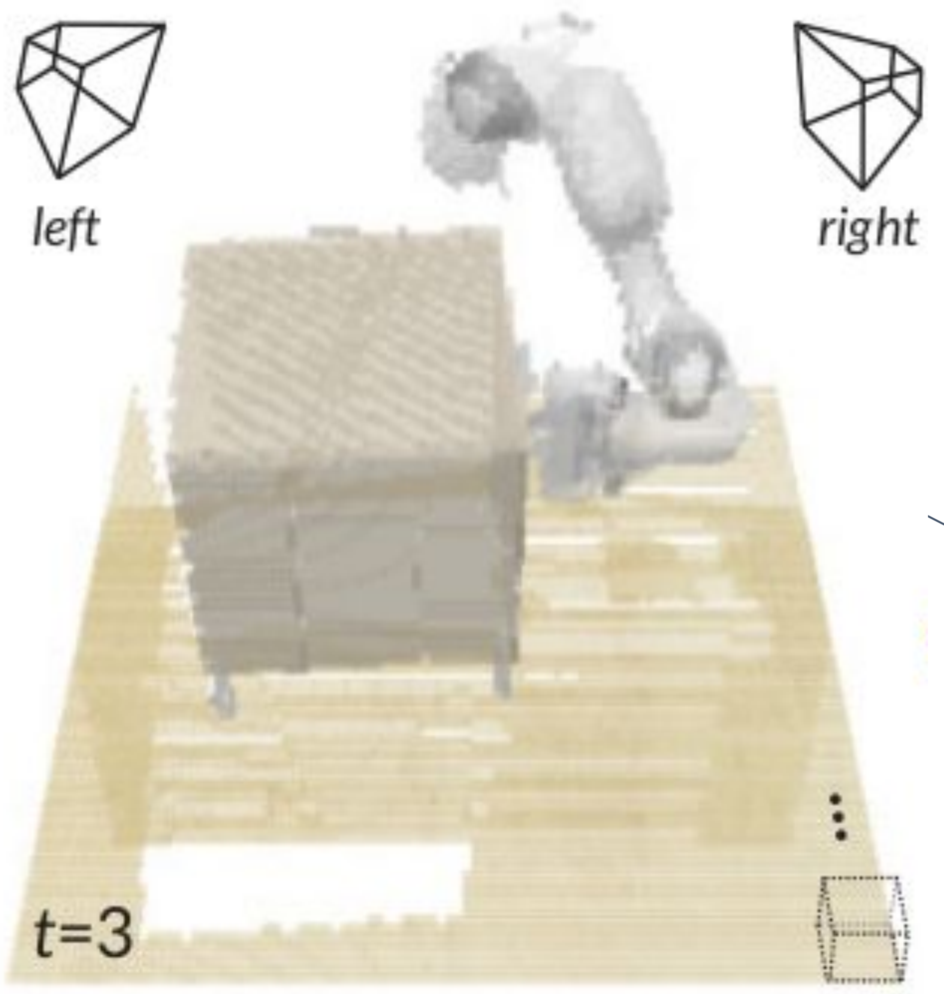
How does it work?

Perceiver-Actor Architecture



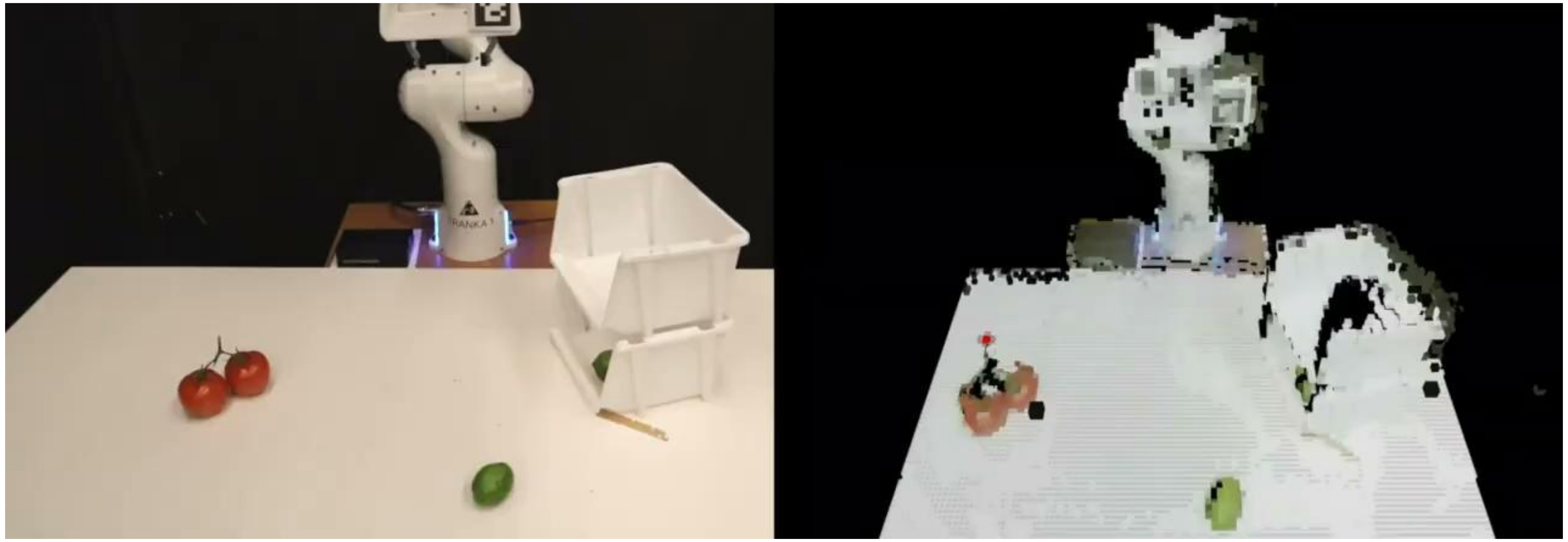
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

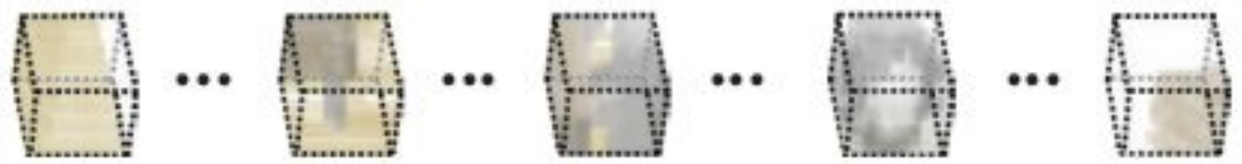
Perceiver-Actor



"open the middle drawer"

voxelized reconstruction of the scene





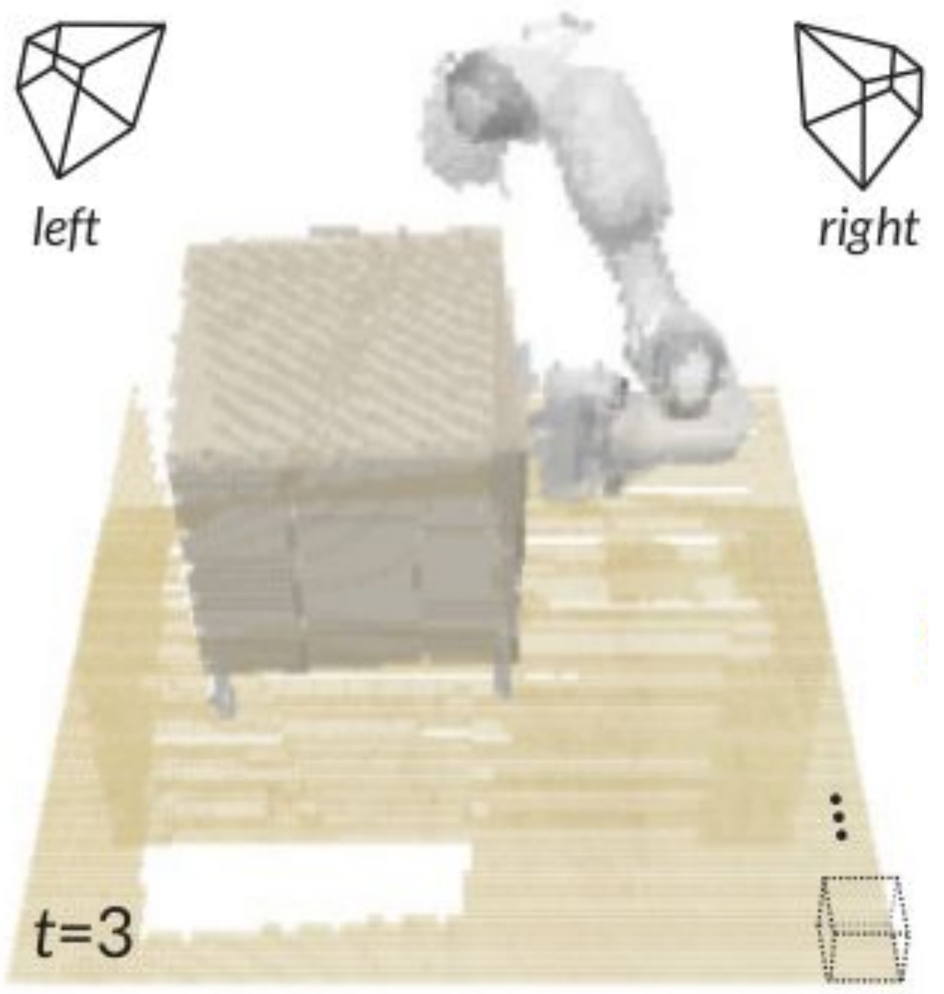
5x5x5 patches with 100x100x100 grid = 8000 patches

100x100x100x10

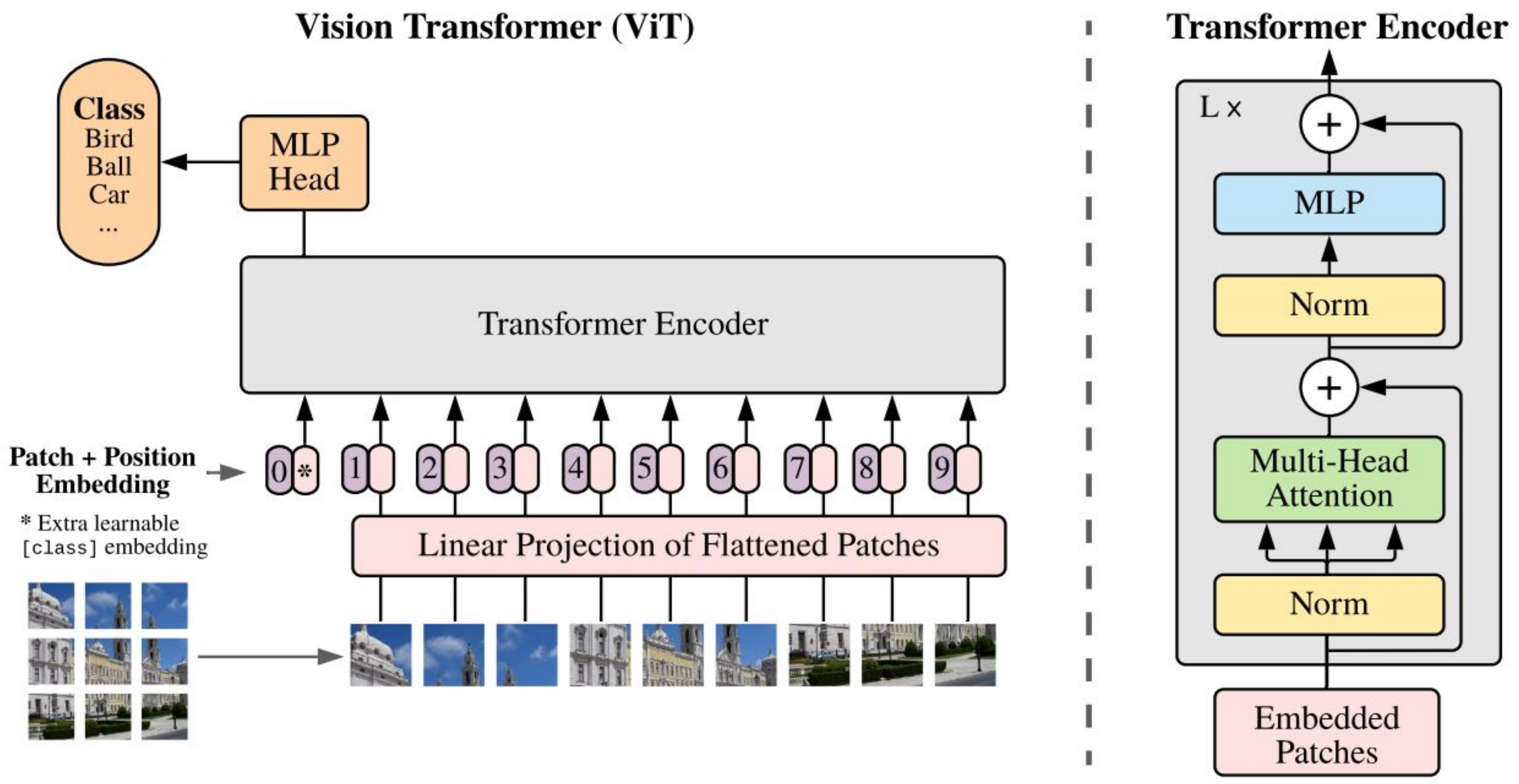
3 RGB, 3 point, 1 occupancy,and 3 position index values

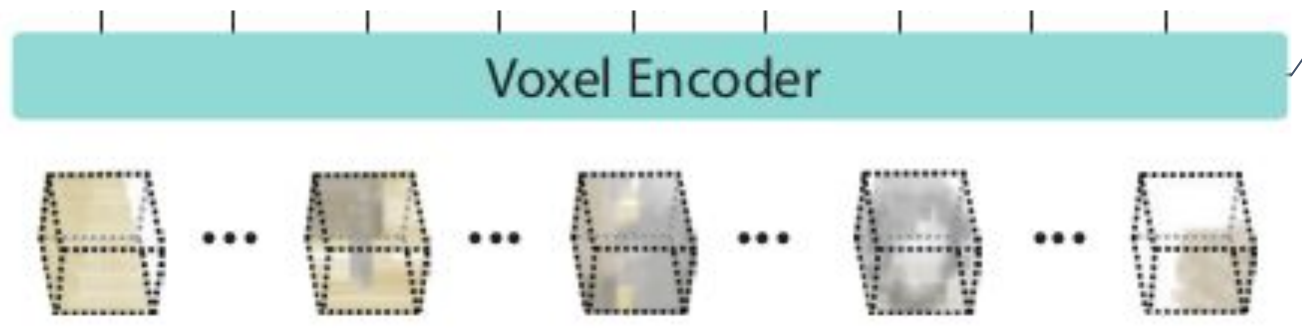
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Perceiver-Actor



"open the middle drawer"





3D convolution layer with a kernel-size 1x1

100x100x100x10 — 100x100x100x64

3D convolution layer with a kernel-size and stride of 5

100x100x100x64 — 20x20x20x64

Flattened into a sequence of voxel encodings

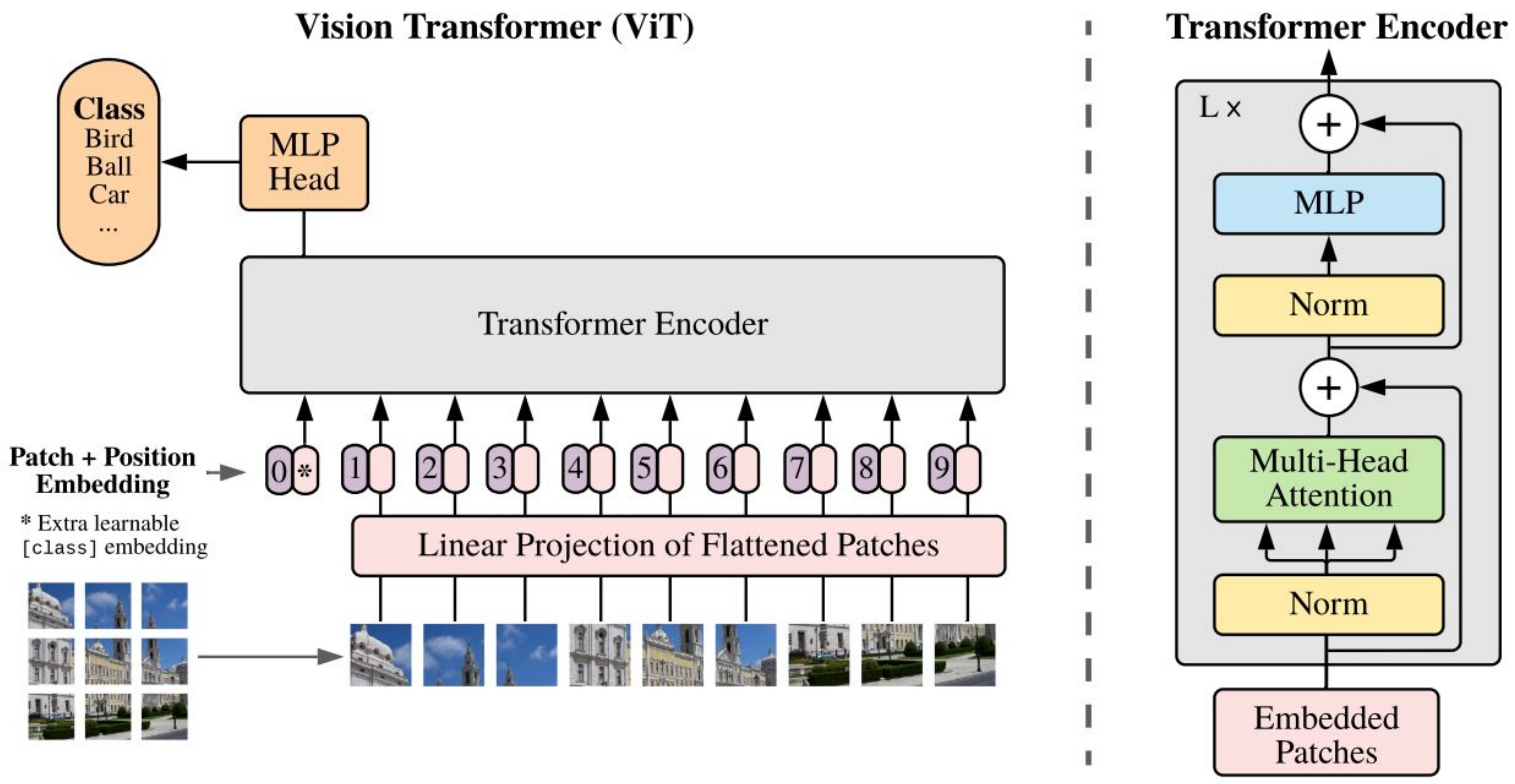
8000x64

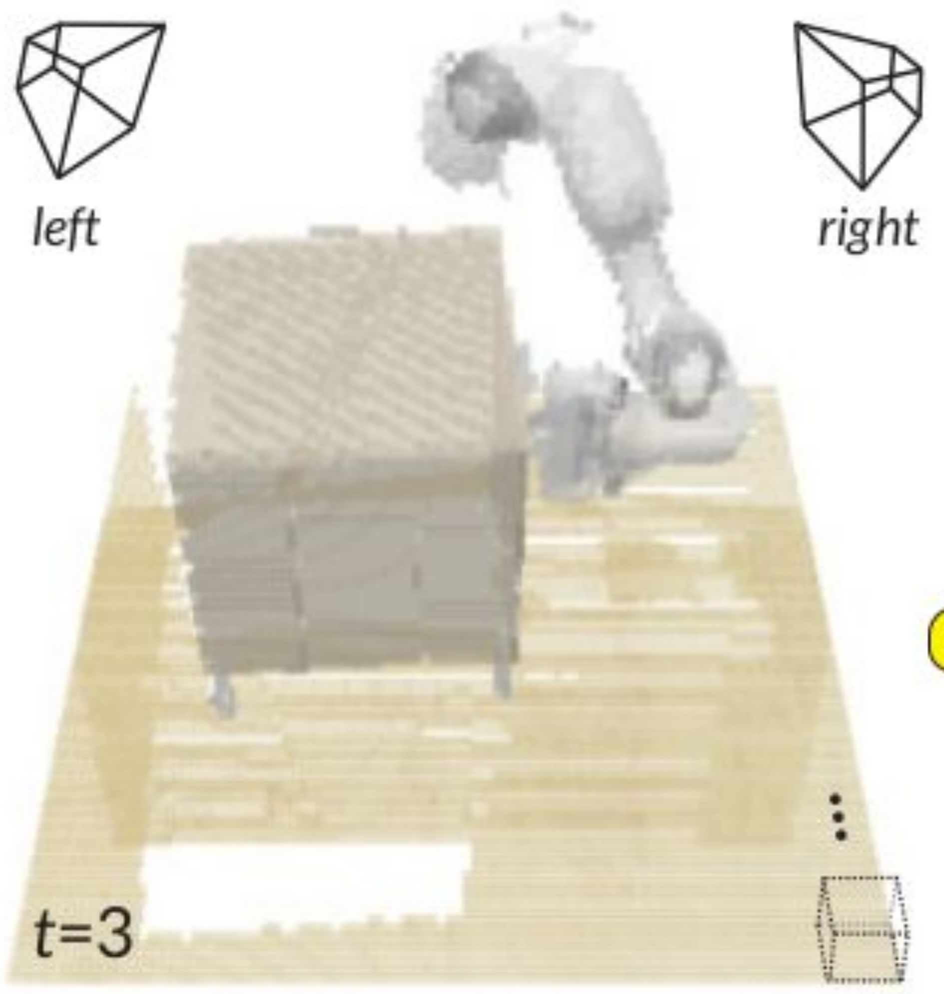
proprioception data

8000x128

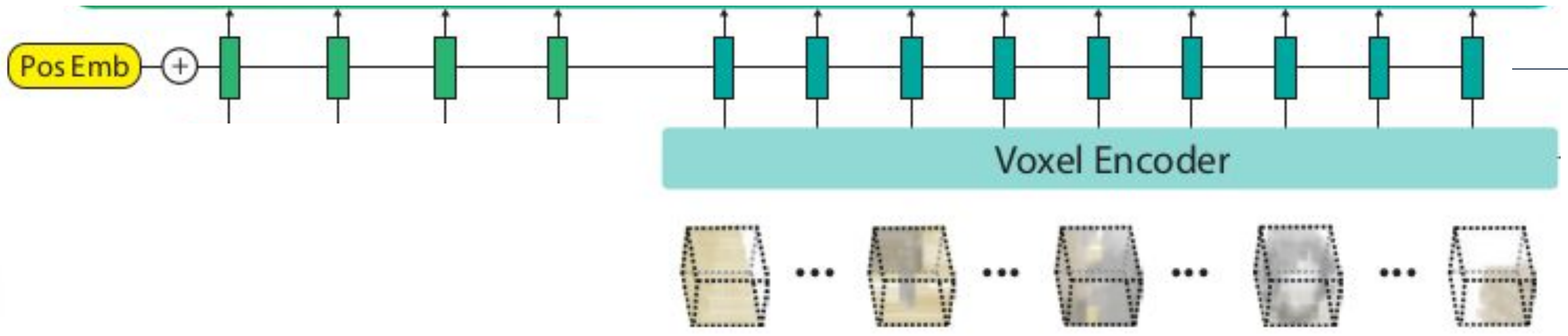
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Perceiver-Actor





"open the middle drawer"



Learned Positional Encoding

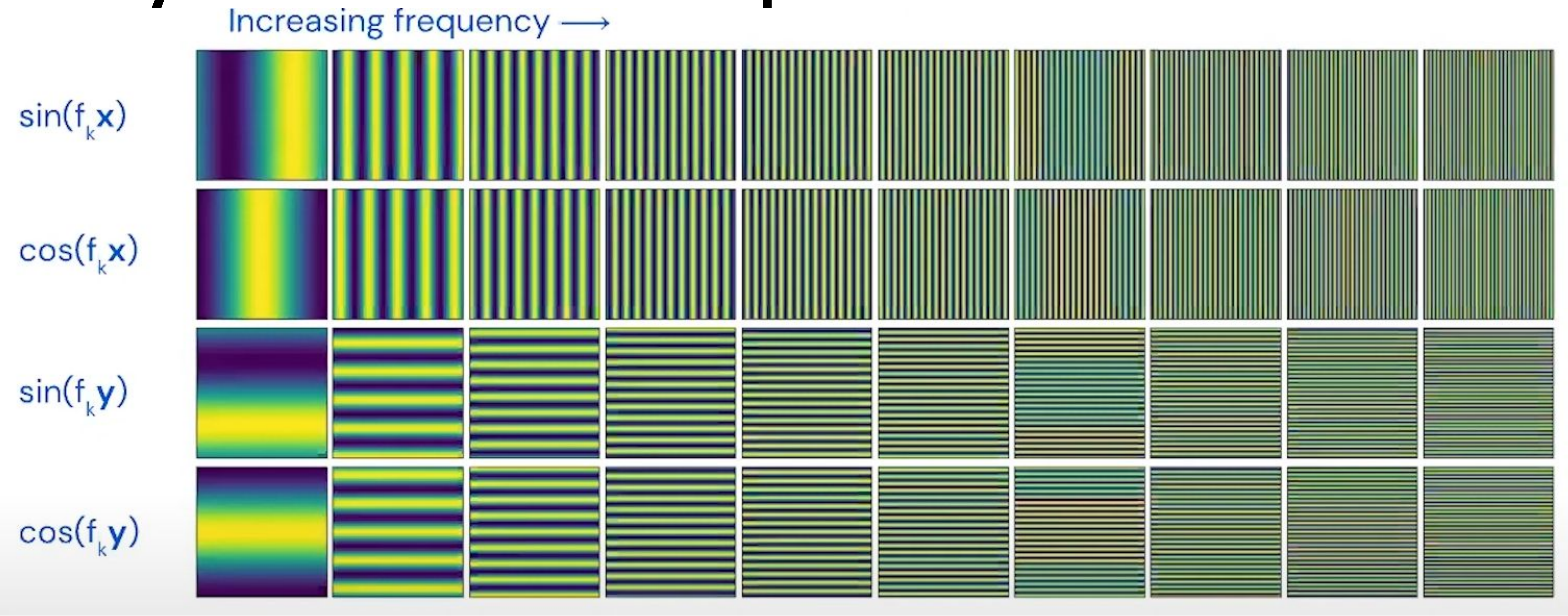
Model initializes each position with a trainable vector

During training, these vectors are adjusted to capture spatial or sequential dependencies

8000x128 — 8000x512

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

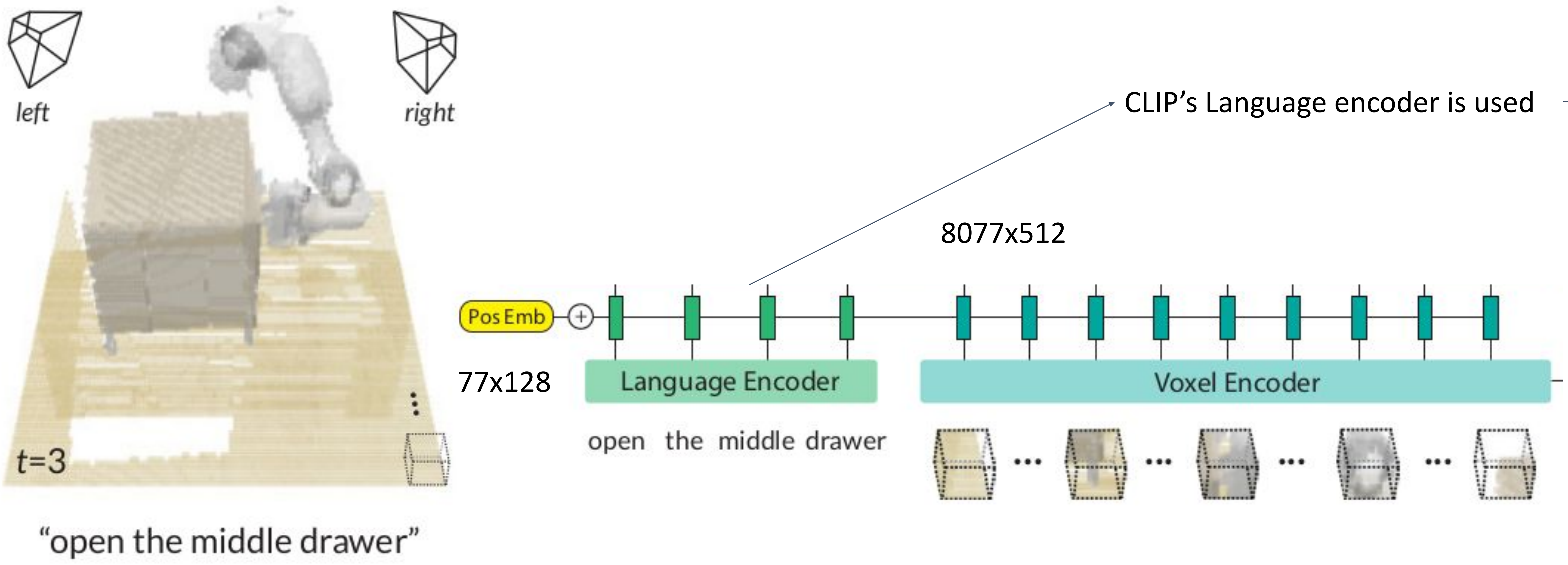
Why use learned positional encoding?



1) The original PerceiverIO transformer paper considers images as input:
The positional encodings are constructed using sine and cosine (2D fourier transforms)functions of different frequencies
2) Perceiver-Actor author says that it lead to worse performance for voxels so chose to use learned positional encoding

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., ... & Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, *33*, 7537-7547.

Perceiver-Actor



8077x512

77x128

8000x128

CLIP’s Language encoder is used → Contrastive Language-Image Pretraining

↓

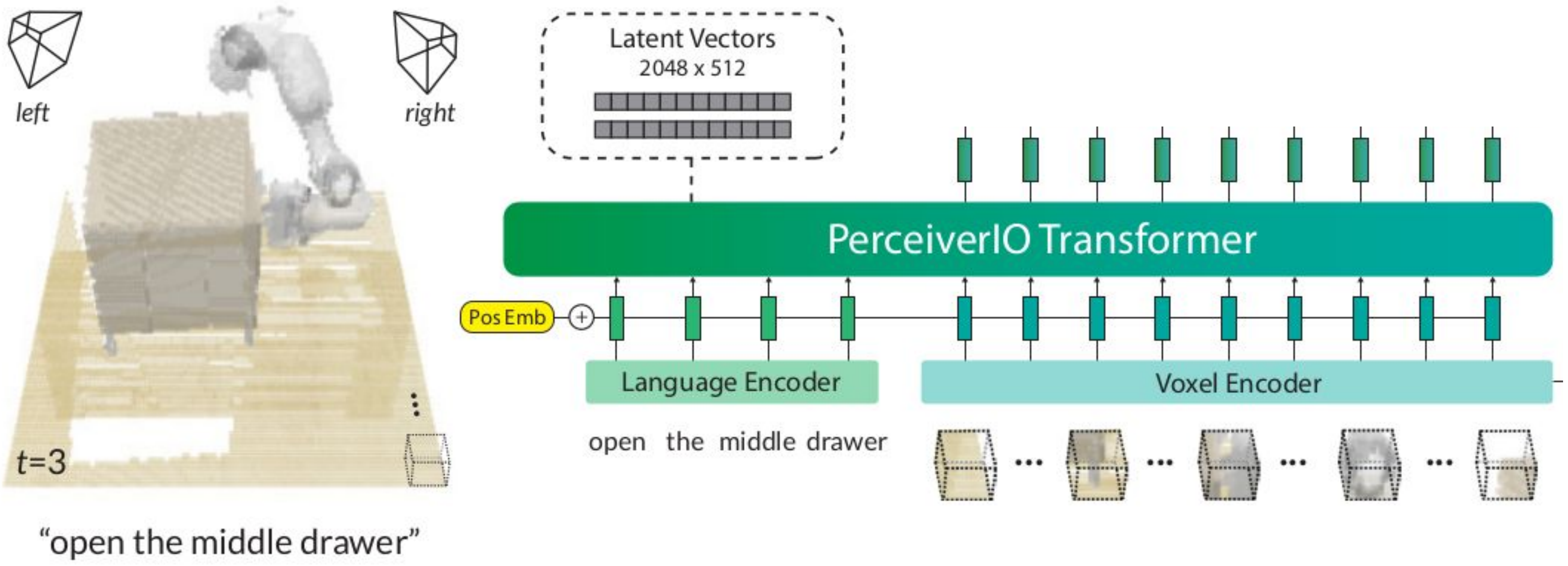
Transformer-based model

↓

Converts natural language text into a high-dimensional vector representation

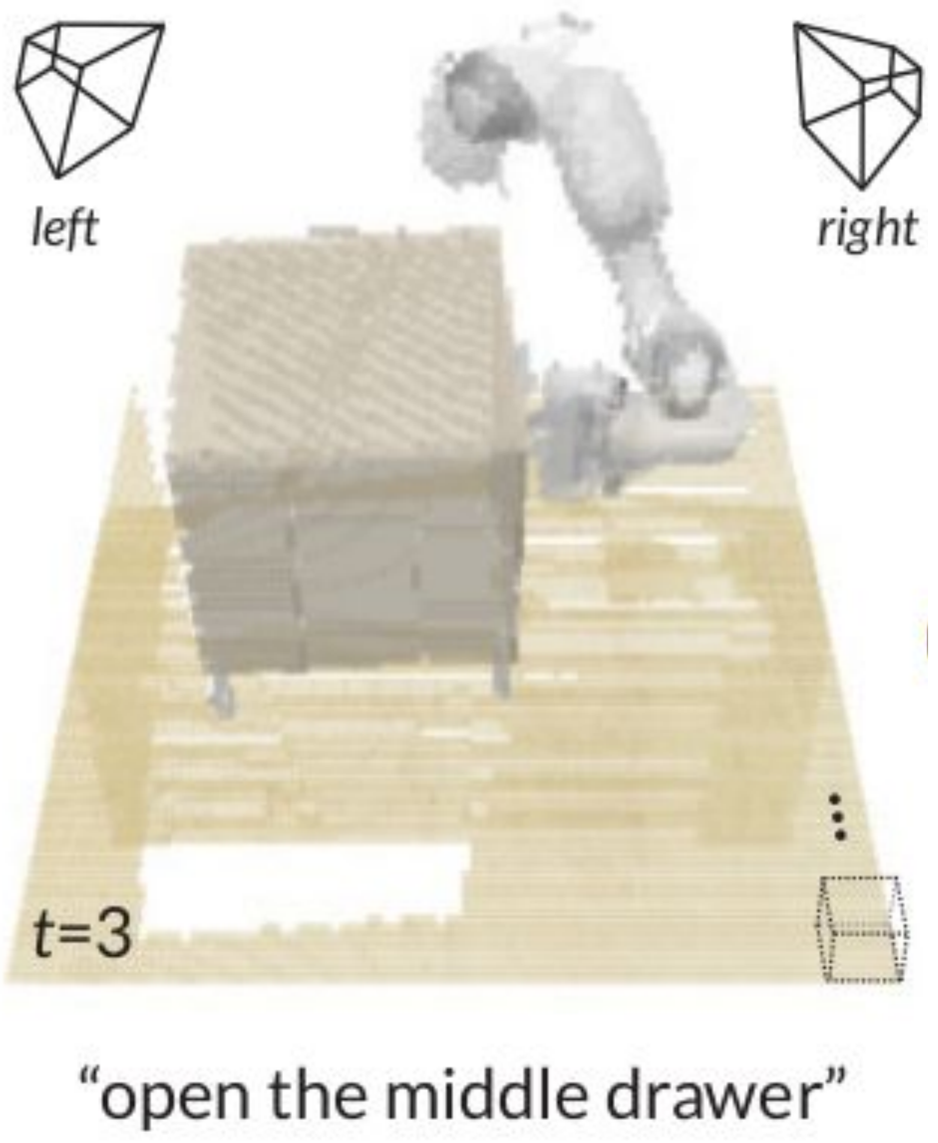
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Perceiver-Actor



Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Why PerceiverIO Transformer?

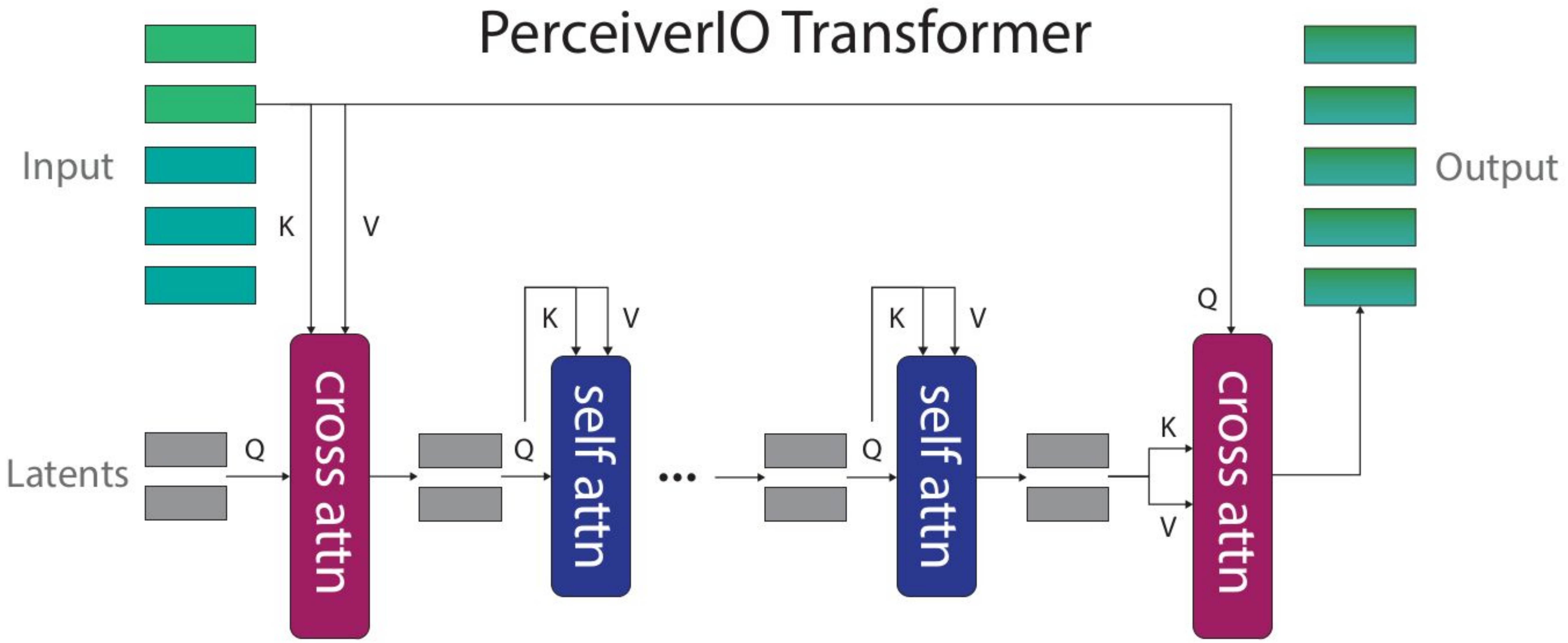


5x5x5 patches with 100x100x100 grid = 8000 patches

Hard to fit on the memory of a commodity GPU

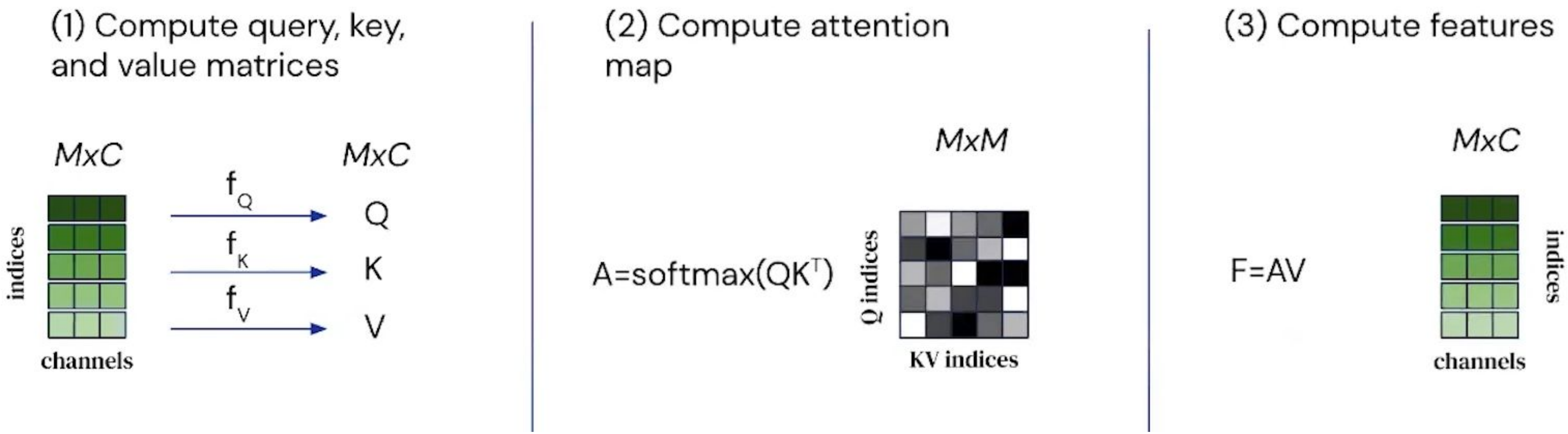
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

PerceiverIO Transformer Architecture



Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

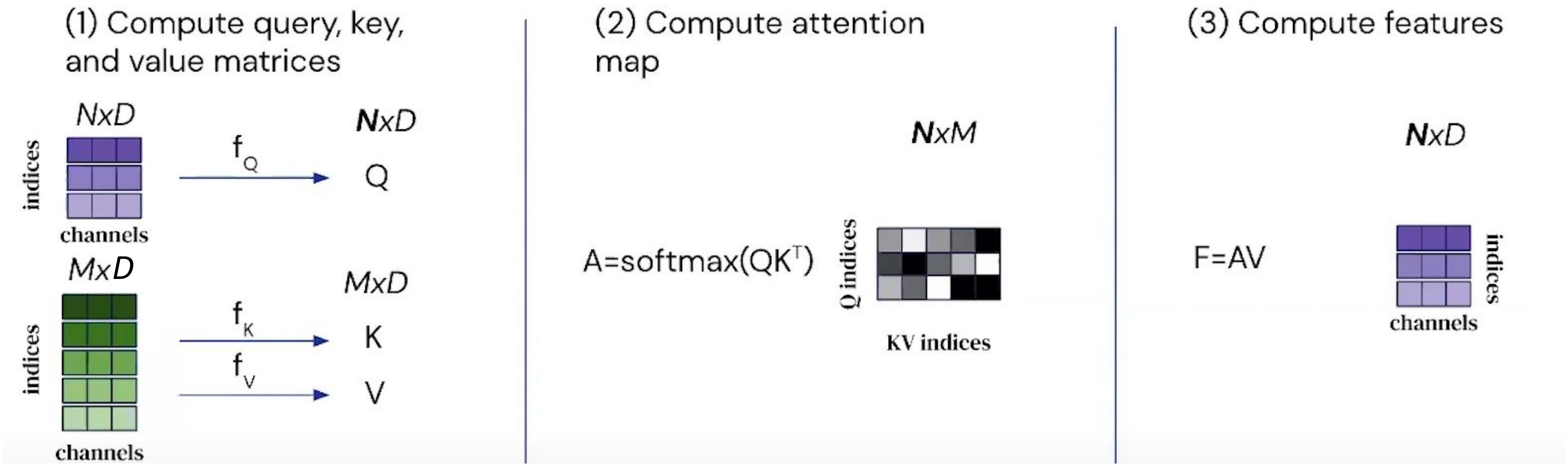
Self-attention in transformer



For standard 224x224 images the M value is 50,176

Vaswani, A. (2017). Attention is all you need. Advances in Neural Information Processing Systems.

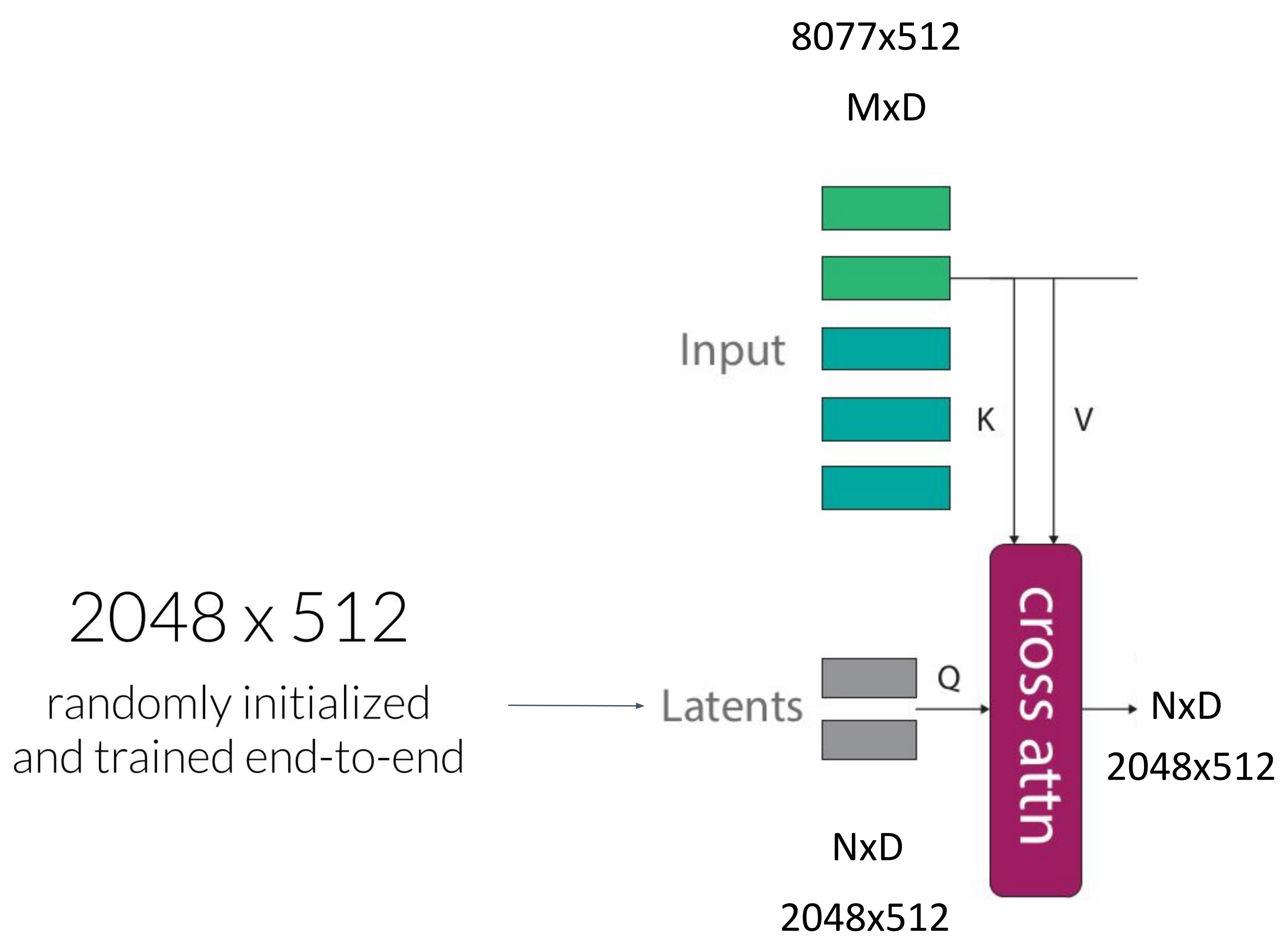
Cross-attention in PerciverIO transformer



For standard 224x224 images the M value is 50,176 and N=512 for ImageNet

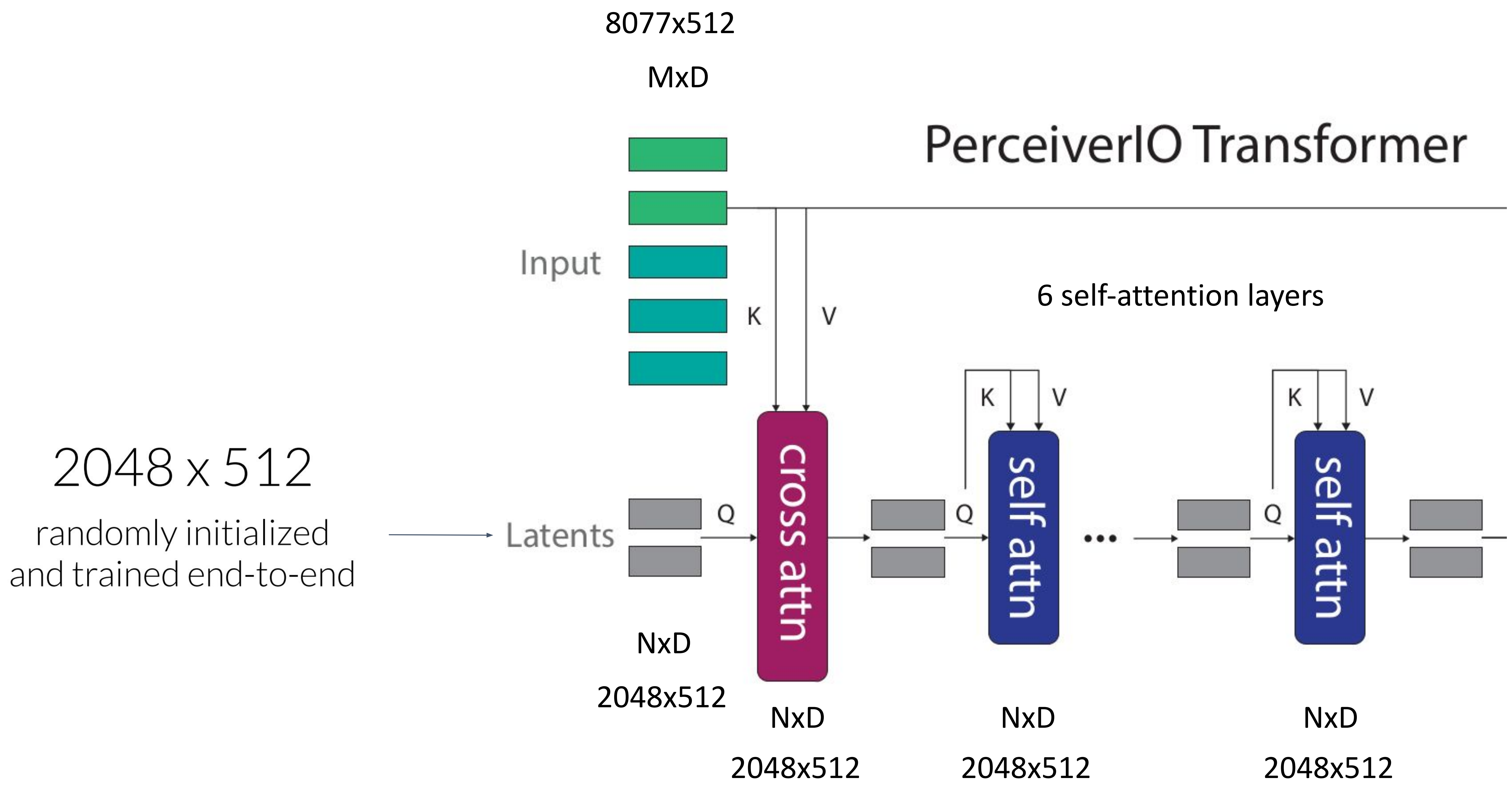
Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., & Carreira, J. (2021, July). Perceiver: General perception with iterative attention. In International conference on machine learning (pp. 4651-4664). PMLR.

PerceiverIO Transformer



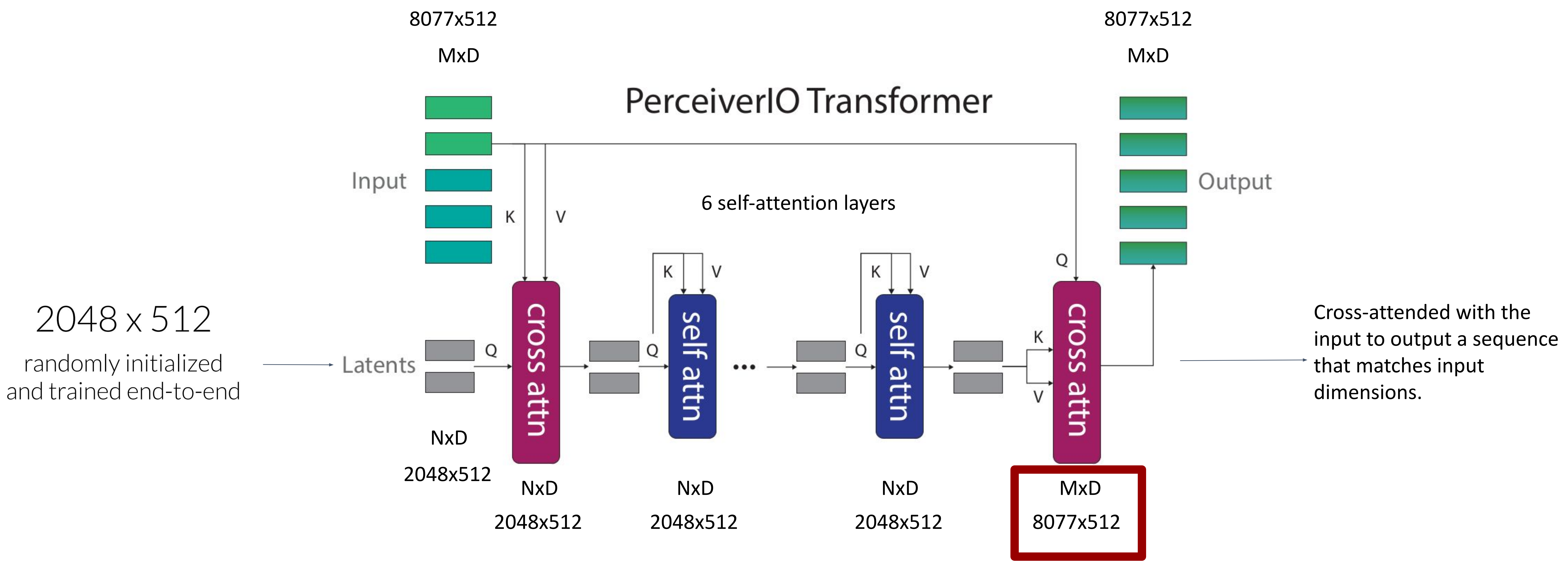
Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

PerceiverIO Transformer



Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

PerceiverIO Transformer



Q = MxD
K = NxD
V = NxD

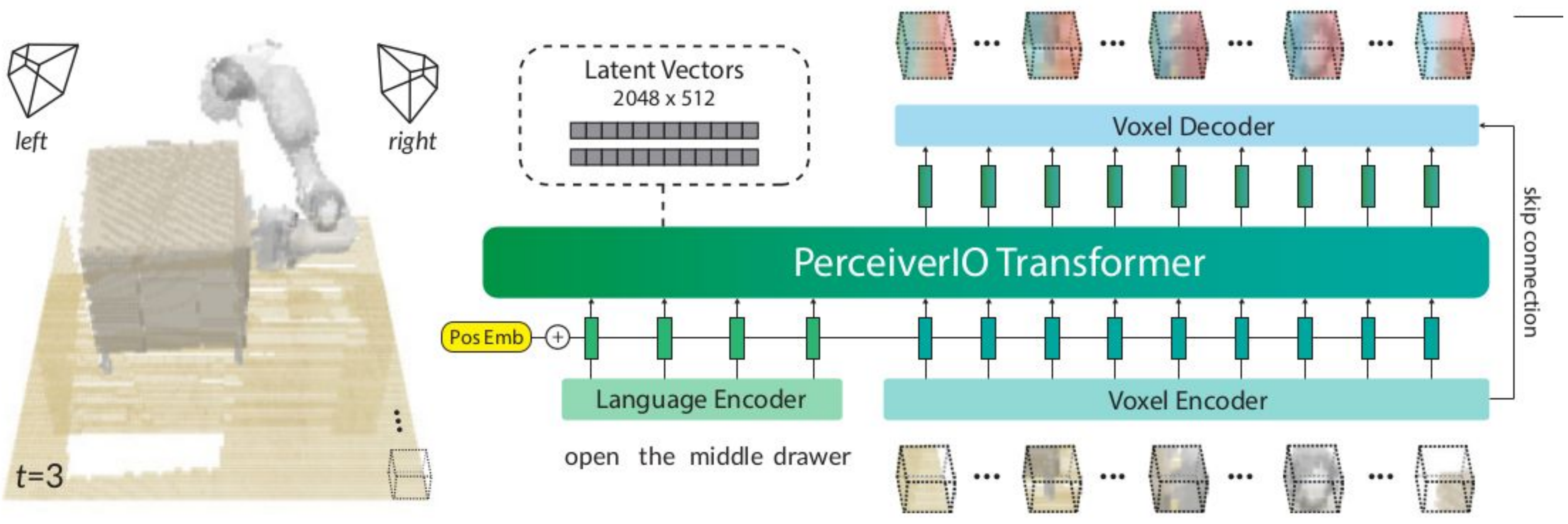
A = softmax(QK)
A = MxN

F = AxV
F = MxD

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Perceiver-Actor

100 x 100 x 100 x 64

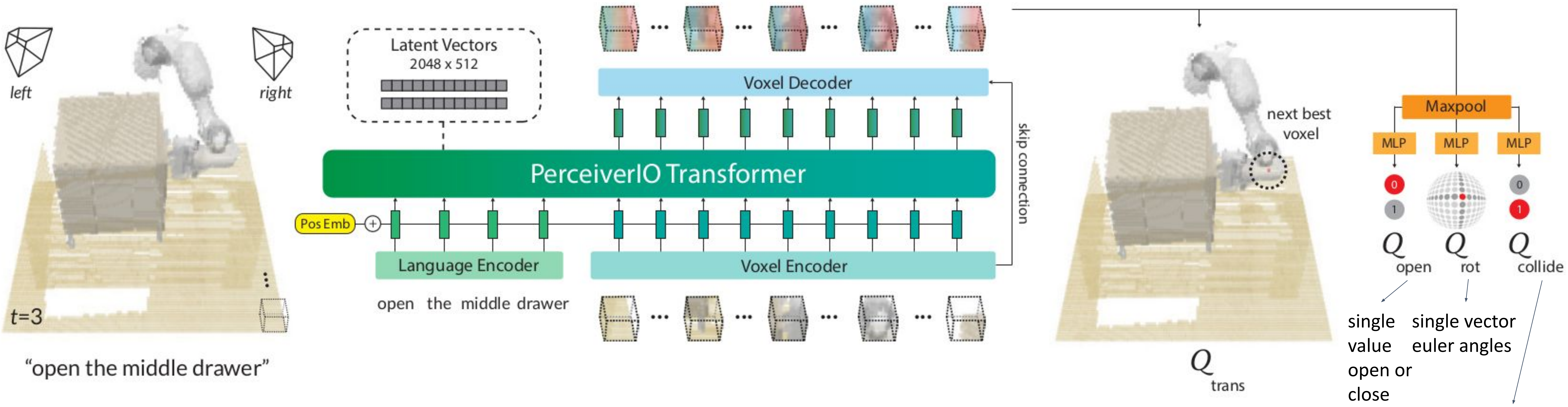


"open the middle drawer"

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Perceiver-Actor

100 x 100 x 100 x 64



x,y,z

collide variable

100 x 100 x 100 x 64 — 100 x 100 x 100 x 1

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Training details

Trained through supervised learning with discrete-time input-action tuples from a dataset of demonstrations

Tuples are composed of voxel observations, language goals, and keyframe actions {(v1 , l1 , k1), (v2 , l2 , k2), . . .}

Randomly sample a tuple and supervise agent to predict keyframe action k given the observation and goal (v, l)

Trained with a batch-size of 16 on 8 NVIDIA V100 GPUs for 16 days (600K iterations).

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Loss Function

Cross-entropy loss

$$\mathcal{L}_{\text{total}} = -\mathbb{E}_{Y_{\text{trans}}}[\log \mathcal{V}_{\text{trans}}] - \mathbb{E}_{Y_{\text{rot}}}[\log \mathcal{V}_{\text{rot}}] - \mathbb{E}_{Y_{\text{open}}}[\log \mathcal{V}_{\text{open}}] - \mathbb{E}_{Y_{\text{collide}}}[\log \mathcal{V}_{\text{collide}}],$$

$$\begin{aligned}
\mathcal{V}_{\text{trans}} &= \text{softmax}(\mathcal{Q}_{\text{trans}}((x, y, z)|\mathbf{v}, \mathbf{l})) \\
\mathcal{V}_{\text{rot}} &= \text{softmax}(\mathcal{Q}_{\text{rot}}((\psi, \theta, \phi)|\mathbf{v}, \mathbf{l})) \\
\mathcal{V}_{\text{open}} &= \text{softmax}(\mathcal{Q}_{\text{open}}(\omega|\mathbf{v}, \mathbf{l})) \\
\mathcal{V}_{\text{collide}} &= \text{softmax}(\mathcal{Q}_{\text{collide}}(\kappa|\mathbf{v}, \mathbf{l}))
\end{aligned}$$

voxel observation and language goal (v, l)

Ground Truth

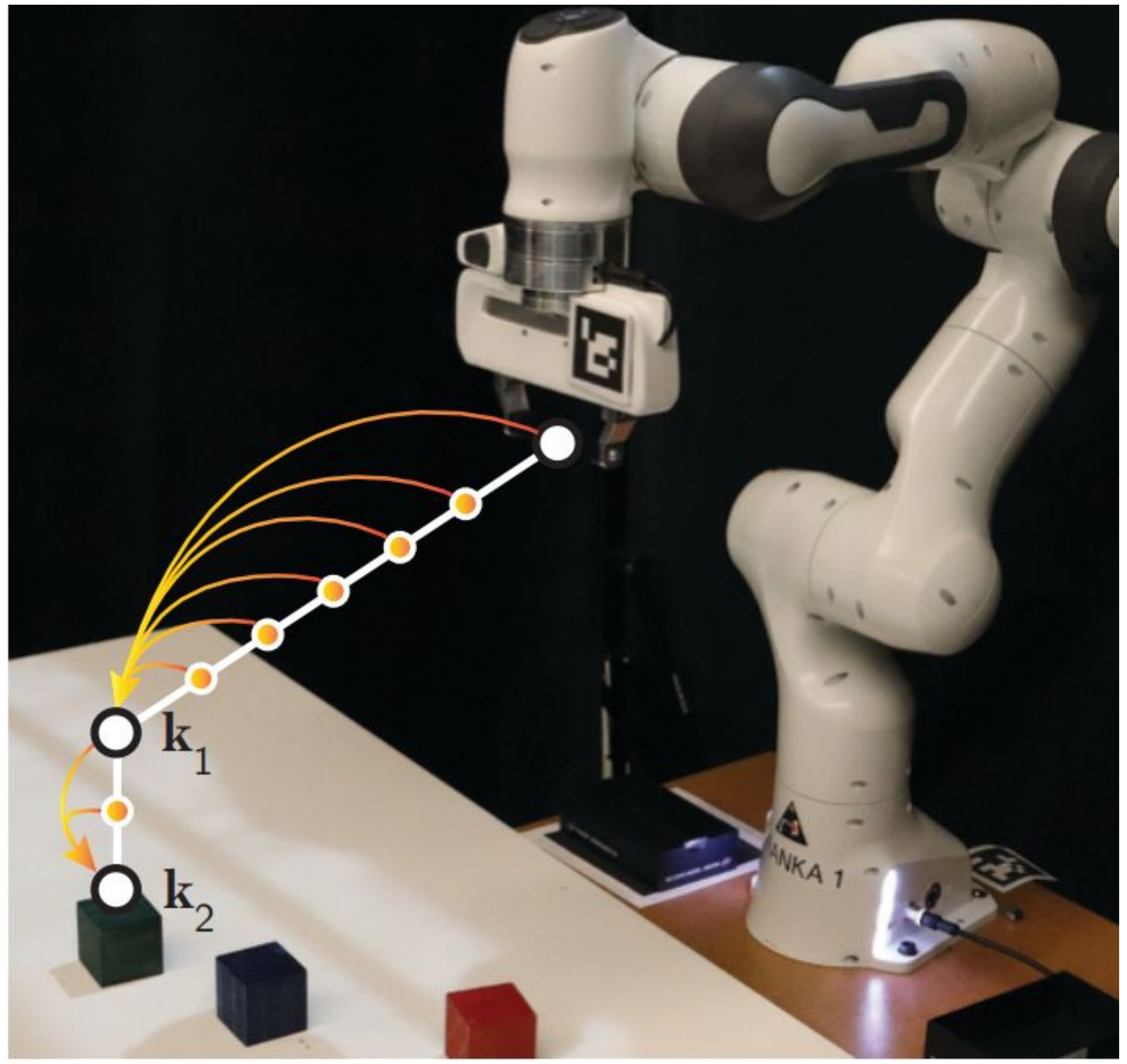
$$Y_{\text{trans}} : \mathbb{R}^{H \times W \times D},$$
$$Y_{\text{rot}} : \mathbb{R}^{(360/R) \times 3}$$
$$Y_{\text{open}} : \mathbb{R}^2, Y_{\text{collide}}$$

The loss function penalizes the model when it assigns a low probability to the correct action

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Dataset setup

Heuristic for Keyframe Extraction:
(1) Joint velocities are near zero
(2) Gripper open state has not changed



Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Task details

Task	Variation Type	# of Variations	Avg. Keyframes	Language Template
open drawer	placement	3	3.0	"open the __ drawer"
slide block	color	4	4.7	"slide the block to __ target"
sweep to dustpan	size	2	4.6	"sweep dirt to the __ dustpan"
meat off grill	category	2	5.0	"take the __ off the grill"
turn tap	placement	2	2.0	"turn __ tap"
put in drawer	placement	3	12.0	"put the item in the __ drawer"
close jar	color	20	6.0	"close the __ jar"
drag stick	color	20	6.0	"use the stick to drag the cube onto the __ target"
stack blocks	color, count	60	14.6	"stack __ __ blocks"
screw bulb	color	20	7.0	"screw in the __ light bulb"
put in safe	placement	3	5.0	"put the money away in the safe on the __ shelf"
place wine	placement	3	5.0	"stack the wine bottle to the __ of the rack"
put in cupboard	category	9	5.0	"put the __ in the cupboard"
sort shape	shape	5	5.0	"put the __ in the shape sorter"
push buttons	color	50	3.8	"push the __ button, [then the __ button]"
insert peg	color	20	5.0	"put the ring on the __ spoke"
stack cups	color	20	10.0	"stack the other cups on top of the __ cup"
place cups	count	3	11.5	"place __ cups on the cup holder"

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Multi-Task Test Results

Method
Image-BC (CNN)
Image-BC (ViT)
C2FARM-BC
PERACT (w/o Lang)
PERACT

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Multi-Task Test Results

	open drawer		slide block		sweep to dustpan		meat off grill		turn tap		put in drawer		close jar		drag stick		stack blocks	
Method	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100
Image-BC (CNN)																		
Image-BC (ViT)																		
C2FARM-BC																		
PerAct (w/o Lang)																		
PerAct																		

	screw bulb		put in safe		place wine		put in cupboard		sort shape		push buttons		insert peg		stack cups		place cups	
	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100
Image-BC (CNN)																		
Image-BC (ViT)																		
C2FARM-BC																		
PerAct (w/o Lang)																		
PerAct																		

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Multi-Task Test Results

Method	open drawer		slide block		sweep to dustpan		meat off grill		turn tap		put in drawer		close jar		drag stick		stack blocks	
	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100
Image-BC (CNN)	4	4	4	0	0	0	0	0	20	8	0	8	0	0	0	0	0	0
Image-BC (ViT)	16	0	8	0	8	0	0	0	24	16	0	0	0	0	0	0	0	0
C2FARM-BC	28	20	12	16	4	0	40	20	60	68	12	4	28	24	**72**	24	4	0
PerAct (w/o Lang)	20	28	8	12	20	16	40	48	36	60	16	16	16	12	48	60	0	0
PerAct	**68**	**80**	**32**	**72**	**72**	**56**	**68**	**84**	**72**	**80**	**16**	**68**	**32**	**60**	36	**68**	**12**	**36**

Method	screw bulb		put in safe		place wine		put in cupboard		sort shape		push buttons		insert peg		stack cups		place cups	
	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100	10	100
Image-BC (CNN)	0	0	0	4	0	0	0	0	0	0	4	0	0	0	0	0	0	0
Image-BC (ViT)	0	0	0	0	4	0	4	0	0	0	16	0	0	0	0	0	0	0
C2FARM-BC	12	8	0	12	**36**	8	**4**	0	8	8	**88**	**72**	0	**4**	0	0	0	0
PerAct (w/o Lang)	0	**24**	8	20	8	**20**	0	0	0	0	60	68	4	0	0	0	0	0
PerAct	**28**	**24**	**16**	**44**	20	12	0	**16**	**16**	**20**	56	48	**4**	0	0	0	0	0

Shridhar, M., Manuelli, L., & Fox, D. (2022). *Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation.* In CoRL.

Limitations

Hard to extend to dynamic and dexterous manipulation
Struggles with unseen objects
Does not predict task-completion
Struggles with complex spatial relationships
Computationally expensive since it relies on voxels
Scope of language (especially verbs) is mostly limited to the training distribution

Can it be achieved without voxels?

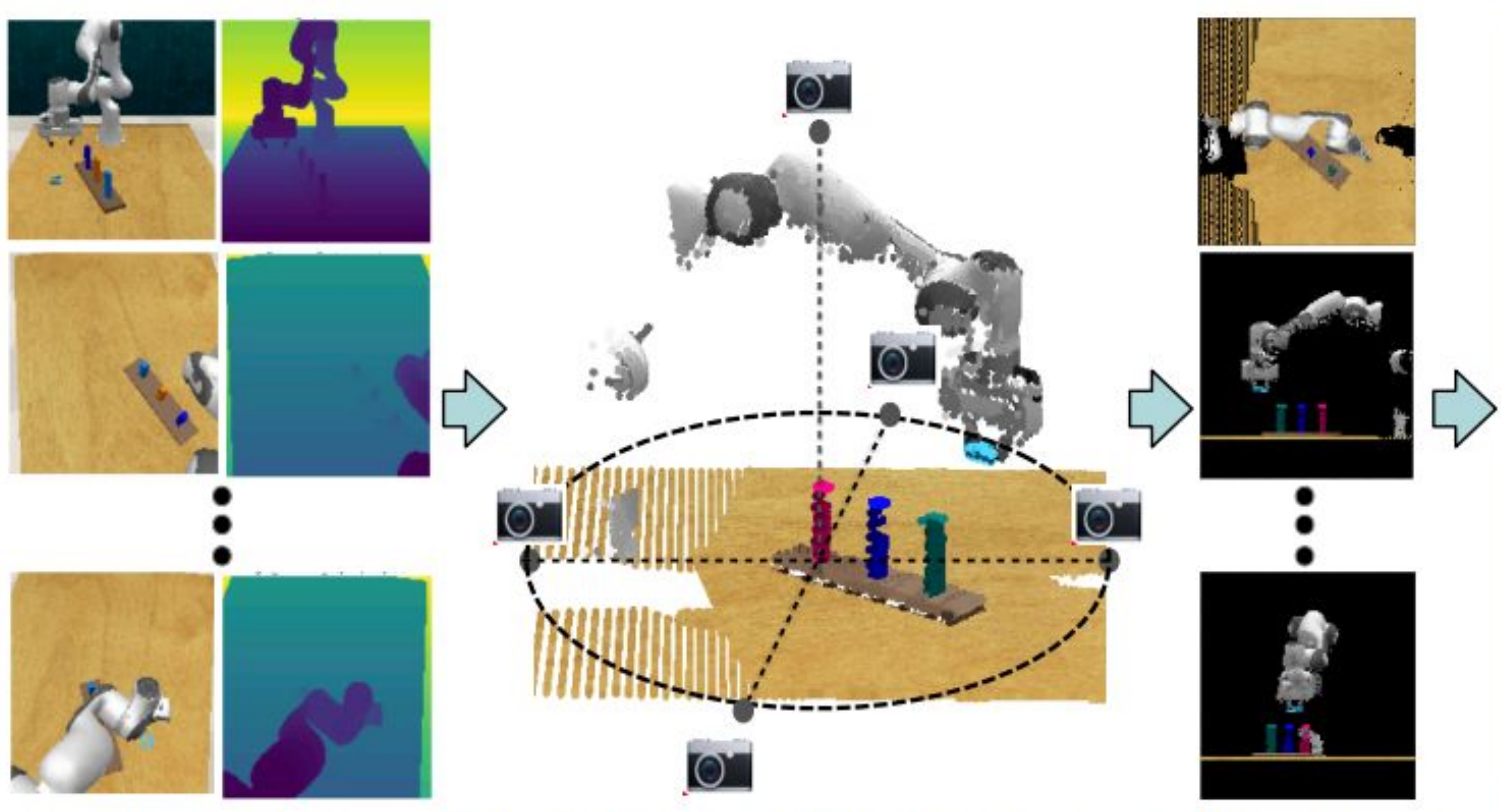
RVT: Robotic View Transformer for 3D Object Manipulation

Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, Dieter Fox

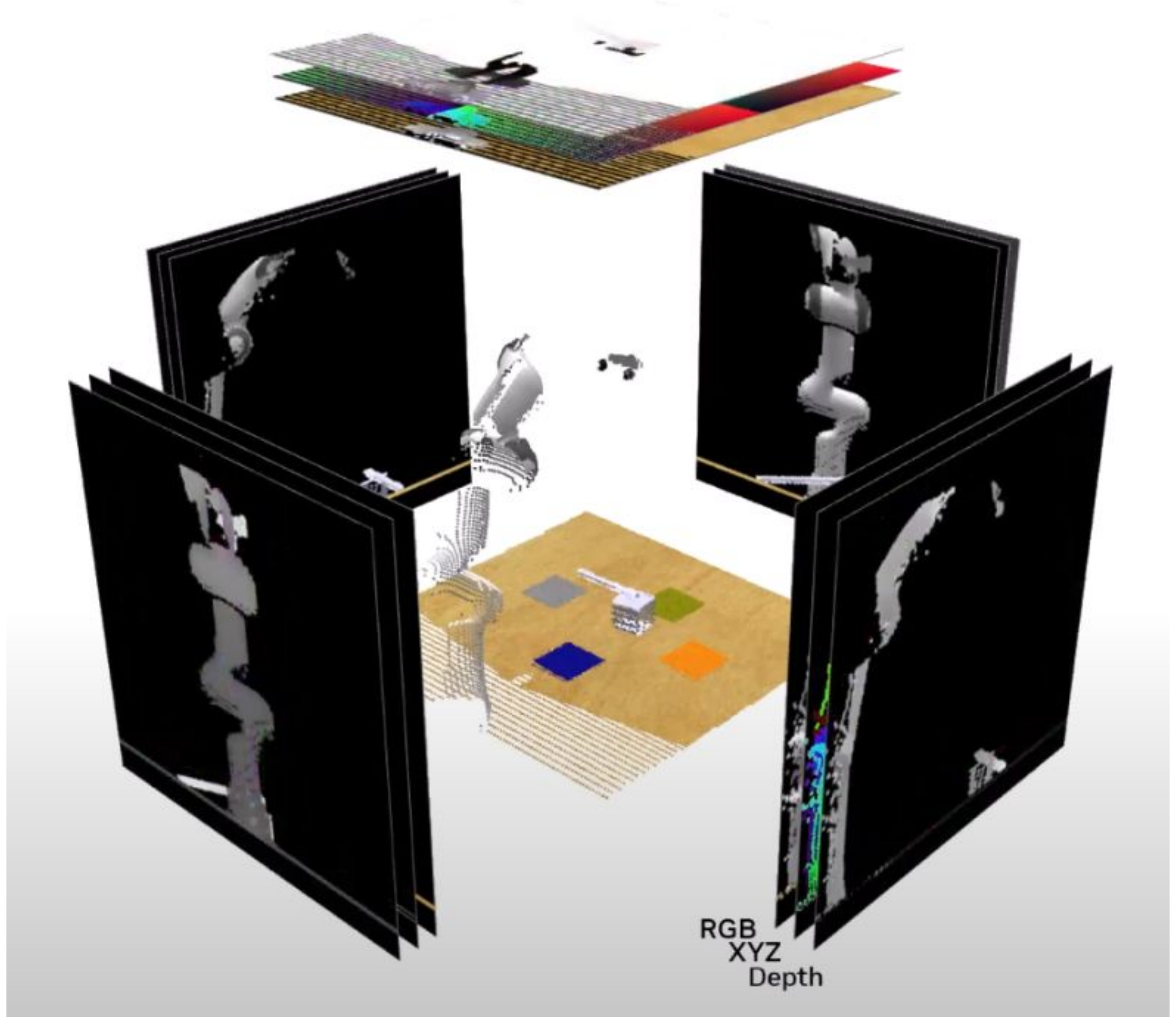
NVIDIA

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

RVT

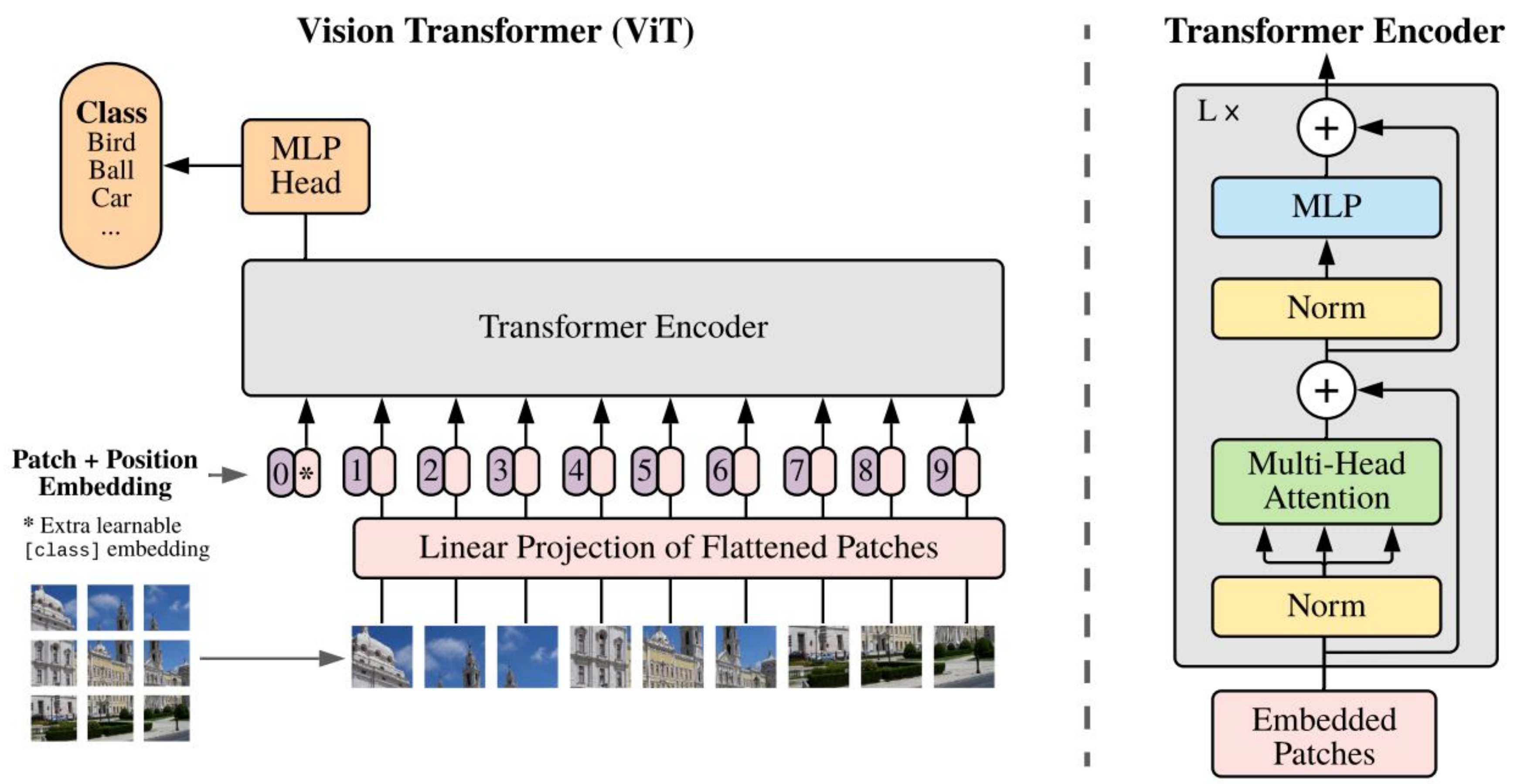


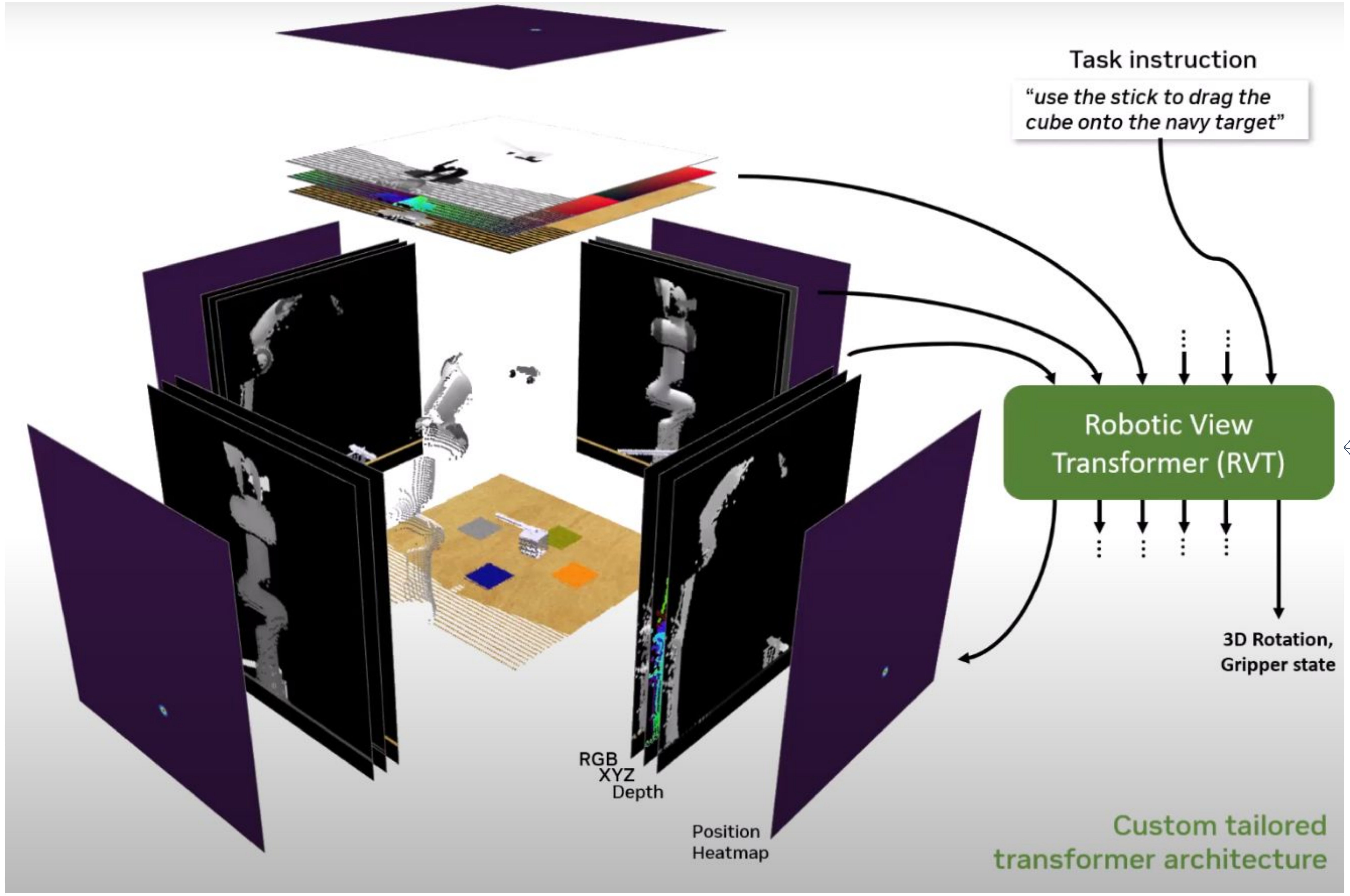
a. Sensor Data b. Point Cloud and Virtual Cameras c. Virtual Images



Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

RVT





Intra-Image Attention(self attention)
first 4 layers

Cross-Image Attention(cross attention)
Next 4 layers

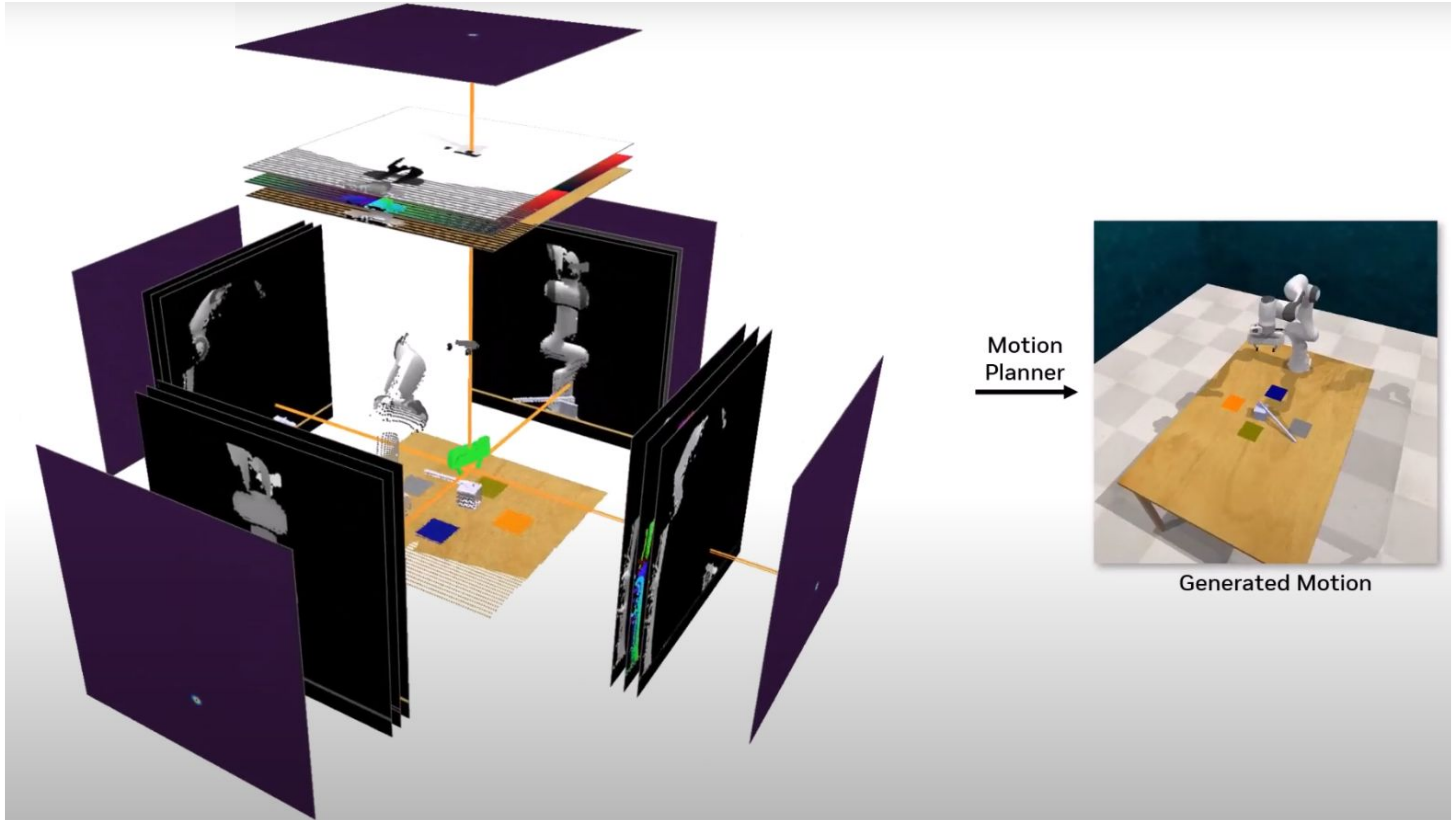
Total 8 layers

Heat maps from all locations used to project and find x,y,z

img_res 220x220

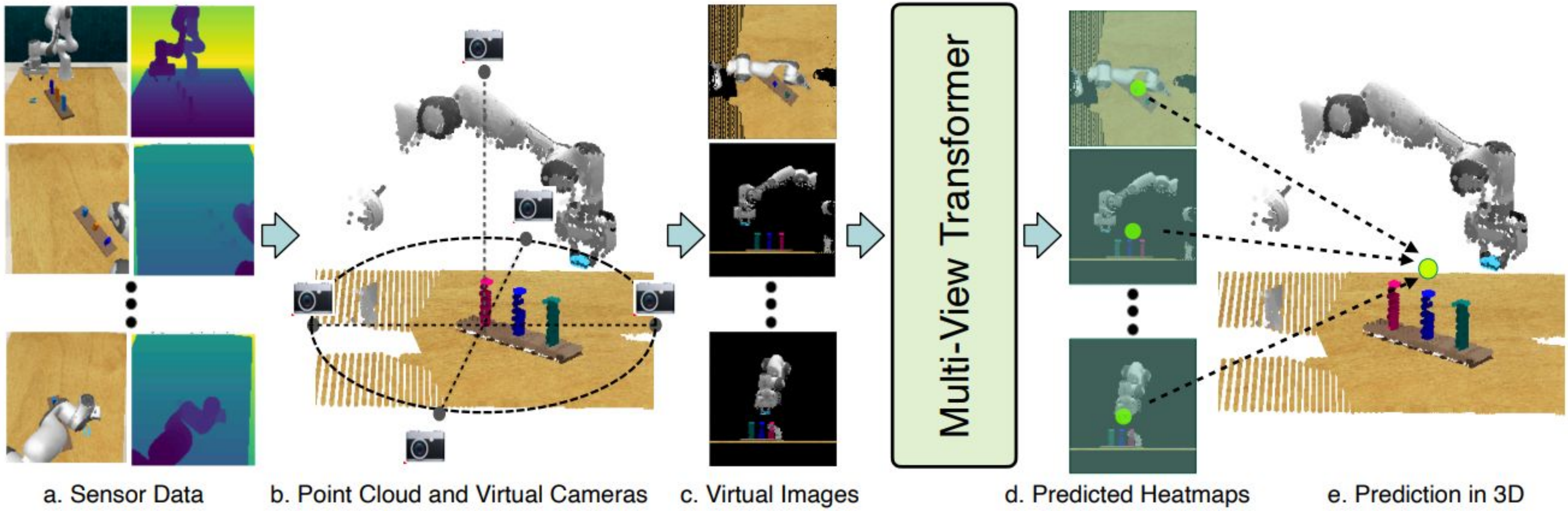
Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

RVT



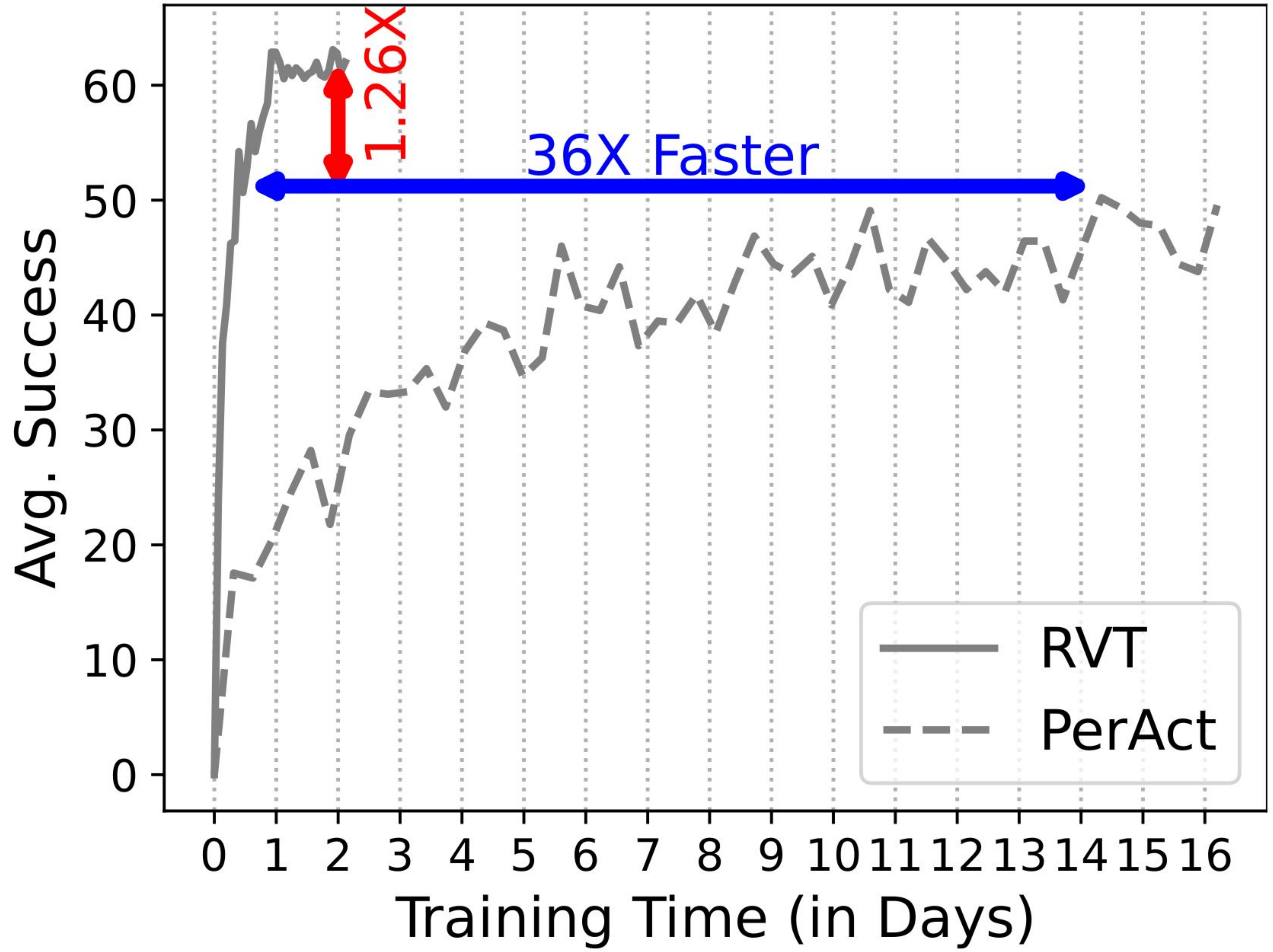
Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

RVT



Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

Comparison with PerAct



(NVIDIA Tesla V100) and number of GPUs (8)

26% higher success rate

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

Comparison with other models

NVIDIA RTX 3090

Models	Avg. Success ↑	Avg. Rank ↓	Train time (in days) ↓	Inf. Speed (in fps) ↑	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Open Drawer	Place Cups	Place Wine
Image-BC (CNN) [2, 6]	1.3	3.7	-	-	0	0	0	0	4	0	0
Image-BC (ViT) [2, 6]	1.3	3.8	-	-	0	0	0	0	0	0	0
C2F-ARM-BC [5, 6]	20.1	3.1	-	-	24	24	4	20	20	0	8
PerAct [6]	49.4	1.9	16.0	4.9	**55.2** ± 4.7	89.6 ± 4.1	5.6 ± 4.1	70.4 ± 2.0	**88.0** ± 5.7	2.4 ± 3.2	44.8 ± 7.8
RVT (ours)	**62.9**	**1.1**	**1.0**	**11.6**	52.0 ± 2.5	**99.2** ± 1.6	**11.2** ± 3.0	**88.0** ± 2.5	71.2 ± 6.9	**4.0** ± 2.5	**91.0** ± 5.2

Models	Push Buttons	Put in Cupboard	Put in Drawer	Put in Safe	Screw Bulb	Slide Block	Sort Shape	Stack Blocks	Stack Cups	Sweep to Dustpan	Turn Tap
Image-BC (CNN) [2, 6]	0	0	8	4	0	0	0	0	0	0	8
Image-BC (ViT) [2, 6]	0	0	0	0	0	0	0	0	0	0	16
C2F-ARM-BC [5, 6]	72	0	4	12	8	16	8	0	0	0	68
PerAct [6]	92.8 ± 3.0	28.0 ± 4.4	51.2 ± 4.7	84.0 ± 3.6	17.6 ± 2.0	74.0 ± 13.0	16.8 ± 4.7	26.4 ± 3.2	2.4 ± 2.0	52.0 ± 0.0	88.0 ± 4.4
RVT (ours)	**100.0** ± 0.0	**49.6** ± 3.2	**88.0** ± 5.7	**91.2** ± 3.0	**48.0** ± 5.7	**81.6** ± 5.4	**36.0** ± 2.5	**28.8** ± 3.9	**26.4** ± 8.2	**72.0** ± 0.0	**93.6** ± 4.1

Faster inference speed compared to PerAct

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

Comparison with other models

Models	Avg. Success ↑	Avg. Rank ↓	Train time (in days) ↓	Inf. Speed (in fps) ↑	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Open Drawer	Place Cups	Place Wine
Image-BC (CNN) [2, 6]	1.3	3.7	-	-	0	0	0	0	4	0	0
Image-BC (ViT) [2, 6]	1.3	3.8	-	-	0	0	0	0	0	0	0
C2F-ARM-BC [5, 6]	20.1	3.1	-	-	24	24	4	20	20	0	8
PerAct [6]	49.4	1.9	16.0	4.9	**55.2** ± 4.7	89.6 ± 4.1	5.6 ± 4.1	70.4 ± 2.0	**88.0** ± 5.7	2.4 ± 3.2	44.8 ± 7.8
RVT (ours)	**62.9**	**1.1**	**1.0**	**11.6**	52.0 ± 2.5	**99.2** ± 1.6	**11.2** ± 3.0	**88.0** ± 2.5	71.2 ± 6.9	**4.0** ± 2.5	**91.0** ± 5.2

Models	Push Buttons	Put in Cupboard	Put in Drawer	Put in Safe	Screw Bulb	Slide Block	Sort Shape	Stack Blocks	Stack Cups	Sweep to Dustpan	Turn Tap
Image-BC (CNN) [2, 6]	0	0	8	4	0	0	0	0	0	0	8
Image-BC (ViT) [2, 6]	0	0	0	0	0	0	0	0	0	0	16
C2F-ARM-BC [5, 6]	72	0	4	12	8	16	8	0	0	0	68
PerAct [6]	92.8 ± 3.0	28.0 ± 4.4	51.2 ± 4.7	84.0 ± 3.6	17.6 ± 2.0	74.0 ± 13.0	16.8 ± 4.7	26.4 ± 3.2	2.4 ± 2.0	52.0 ± 0.0	88.0 ± 4.4
RVT (ours)	**100.0** ± 0.0	**49.6** ± 3.2	**88.0** ± 5.7	**91.2** ± 3.0	**48.0** ± 5.7	**81.6** ± 5.4	**36.0** ± 2.5	**28.8** ± 3.9	**26.4** ± 8.2	**72.0** ± 0.0	**93.6** ± 4.1

Showcases Ability to handle complex spatial relationships better than PerAct

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

Comparison with other models

Models	Avg. Success ↑	Avg. Rank ↓	Train time (in days) ↓	Inf. Speed (in fps) ↑	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Open Drawer	Place Cups	Place Wine
Image-BC (CNN) [2, 6]	1.3	3.7	-	-	0	0	0	0	4	0	0
Image-BC (ViT) [2, 6]	1.3	3.8	-	-	0	0	0	0	0	0	0
C2F-ARM-BC [5, 6]	20.1	3.1	-	-	24	24	4	20	20	0	8
PerAct [6]	49.4	1.9	16.0	4.9	**55.2** ± 4.7	89.6 ± 4.1	5.6 ± 4.1	70.4 ± 2.0	**88.0** ± 5.7	2.4 ± 3.2	44.8 ± 7.8
RVT (ours)	**62.9**	**1.1**	**1.0**	**11.6**	52.0 ± 2.5	**99.2** ± 1.6	**11.2** ± 3.0	**88.0** ± 2.5	71.2 ± 6.9	**4.0** ± 2.5	**91.0** ± 5.2

Models	Push Buttons	Put in Cupboard	Put in Drawer	Put in Safe	Screw Bulb	Slide Block	Sort Shape	Stack Blocks	Stack Cups	Sweep to Dustpan	Turn Tap
Image-BC (CNN) [2, 6]	0	0	8	4	0	0	0	0	0	0	8
Image-BC (ViT) [2, 6]	0	0	0	0	0	0	0	0	0	0	16
C2F-ARM-BC [5, 6]	72	0	4	12	8	16	8	0	0	0	68
PerAct [6]	92.8 ± 3.0	28.0 ± 4.4	51.2 ± 4.7	84.0 ± 3.6	17.6 ± 2.0	74.0 ± 13.0	16.8 ± 4.7	26.4 ± 3.2	2.4 ± 2.0	52.0 ± 0.0	88.0 ± 4.4
RVT (ours)	**100.0** ± 0.0	**49.6** ± 3.2	**88.0** ± 5.7	**91.2** ± 3.0	**48.0** ± 5.7	**81.6** ± 5.4	**36.0** ± 2.5	**28.8** ± 3.9	**26.4** ± 8.2	**72.0** ± 0.0	**93.6** ± 4.1

Showcases Ability to handle complex spatial relationships better than PerAct

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

Comparison with other models

Models	Avg. Success ↑	Avg. Rank ↓	Train time (in days) ↓	Inf. Speed (in fps) ↑	Close Jar	Drag Stick	Insert Peg	Meat off Grill	Open Drawer	Place Cups	Place Wine
Image-BC (CNN) [2, 6]	1.3	3.7	-	-	0	0	0	0	4	0	0
Image-BC (ViT) [2, 6]	1.3	3.8	-	-	0	0	0	0	0	0	0
C2F-ARM-BC [5, 6]	20.1	3.1	-	-	24	24	4	20	20	0	8
PerAct [6]	49.4	1.9	16.0	4.9	**55.2** ± 4.7	89.6 ± 4.1	5.6 ± 4.1	70.4 ± 2.0	**88.0** ± 5.7	2.4 ± 3.2	44.8 ± 7.8
RVT (ours)	**62.9**	**1.1**	**1.0**	**11.6**	52.0 ± 2.5	**99.2** ± 1.6	**11.2** ± 3.0	**88.0** ± 2.5	71.2 ± 6.9	**4.0** ± 2.5	**91.0** ± 5.2

Models	Push Buttons	Put in Cupboard	Put in Drawer	Put in Safe	Screw Bulb	Slide Block	Sort Shape	Stack Blocks	Stack Cups	Sweep to Dustpan	Turn Tap
Image-BC (CNN) [2, 6]	0	0	8	4	0	0	0	0	0	0	8
Image-BC (ViT) [2, 6]	0	0	0	0	0	0	0	0	0	0	16
C2F-ARM-BC [5, 6]	72	0	4	12	8	16	8	0	0	0	68
PerAct [6]	92.8 ± 3.0	28.0 ± 4.4	51.2 ± 4.7	84.0 ± 3.6	17.6 ± 2.0	74.0 ± 13.0	16.8 ± 4.7	26.4 ± 3.2	2.4 ± 2.0	52.0 ± 0.0	88.0 ± 4.4
RVT (ours)	**100.0** ± 0.0	**49.6** ± 3.2	**88.0** ± 5.7	**91.2** ± 3.0	**48.0** ± 5.7	**81.6** ± 5.4	**36.0** ± 2.5	**28.8** ± 3.9	**26.4** ± 8.2	**72.0** ± 0.0	**93.6** ± 4.1

Showcases Ability to handle complex spatial relationships better than PerAct

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y. W., & Fox, D. (2023, December). Rvt: Robotic view transformer for 3d object manipulation. In Conference on Robot Learning (pp. 694-710). PMLR.

Something similar to this?

Act3D: 3D Feature Field Transformers for Multi-Task Robotic Manipulation

Ellis, M. D., Sukal, T., DeMott, T., & Dewald, J. P. (2007, June). ACT 3D exercise targets gravity-induced discoordination and improves reaching work area in individuals with stroke. In 2007 IEEE 10th International Conference on Rehabilitation Robotics (pp. 890-895). IEEE.

THANK YOU

Team task - Data viz - Due Today

As a first step to narrowing down your final project, I want you to start researching the data **X** that will be used.

Please upload a ***video*** showing all the data streams in your project that will be used to train the deep-learning models **y=f(X)**.

- If you use an existing dataset for your project, I expect your video to contain samples of these sensor observations and the correct labels.
- If you are using a simulator, I expect you to collect the data from the simulator and then show the data streams that will be used for training your model.
- The same goes for real-world experiments as well.

Next Class:
Final Project Check-ins

- A google slide deck with all the data viz videos uploaded, will be created by Prof. Desingh
- Each group will introduce their project.
- Play their video and discuss their progress for ~10 mins and answer questions.
- Full attendance is expected!

P4 - Due Nov 13th

- Instructions available on the webpage
  - Here:
    https://rpm-lab.github.io/CSCI5980-F24-DeepRob/projects/project4/
  - Uses PROPS Pose Estimation Dataset
- Implement PoseCNN
- Autograder is available.
- Due Wednesday, November 13th, 11:59 PM CT

DeepRob

[Student] Lecture 2
by Nikil Krishnakumar, Nanditha Naik
Pointnet and 3D Networks for Manipulation
University of Minnesota