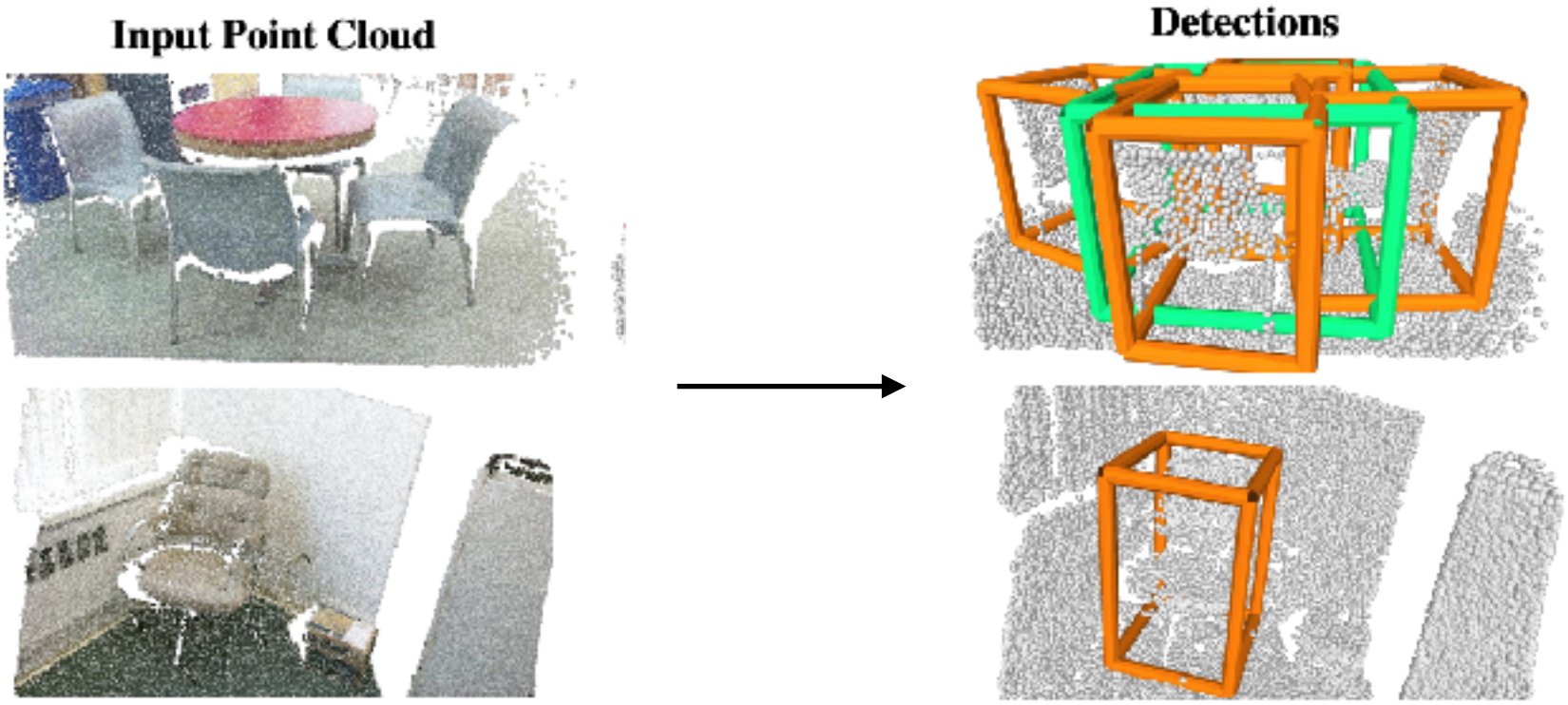
An End-to-End Transformer Model for 3D Object Detection

ICCV 2021

Ishan Misra, Rohit Girdhar, Armand Joulin

Problem Overview

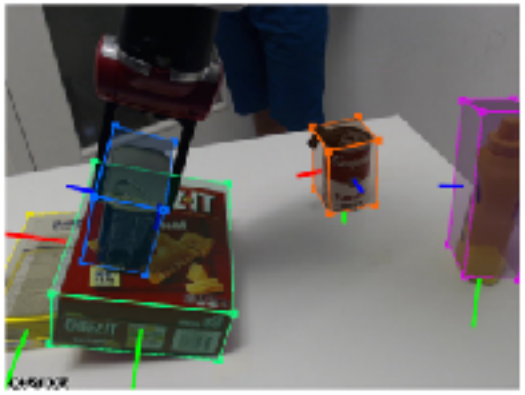
The goal is to predict 3D bounding boxes from raw point cloud (a set of points).



Motivation

3D Perception is useful for many modern applications.

Autonomous Driving



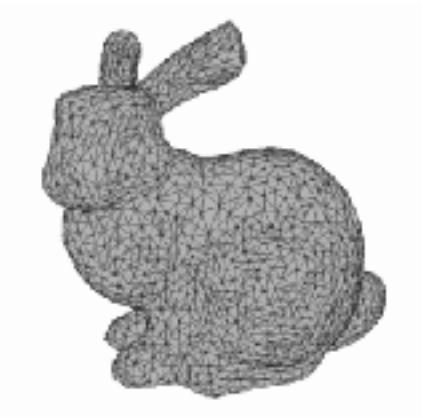
Robot Perception

Augmented & Virtual Reality

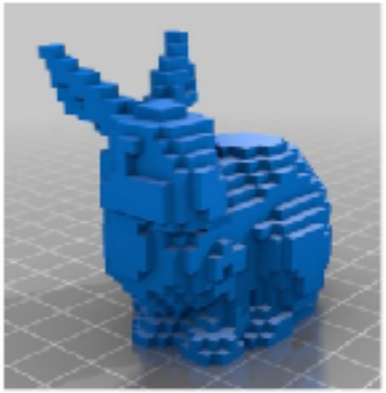
Urban Analytics

3D Input Representations

Due to its irregular format, point clouds are often converted to 3D voxel grids, meshes or collections of images.



Mesh

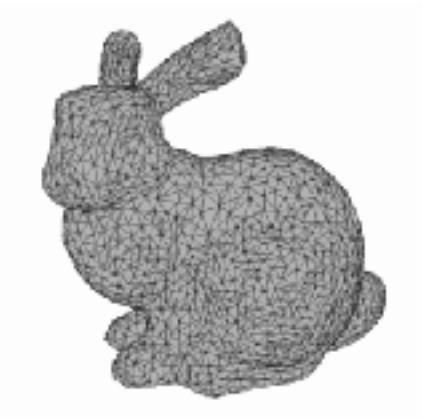


Volumetric
(e.g., Voxels)

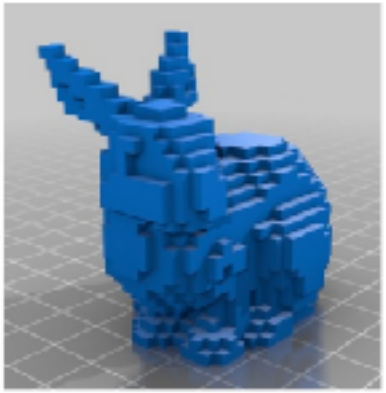
Projected RGBD

3D Input Representations

Due to its irregular format, point clouds are often converted to 3D voxel grids, meshes or collections of images.



Mesh



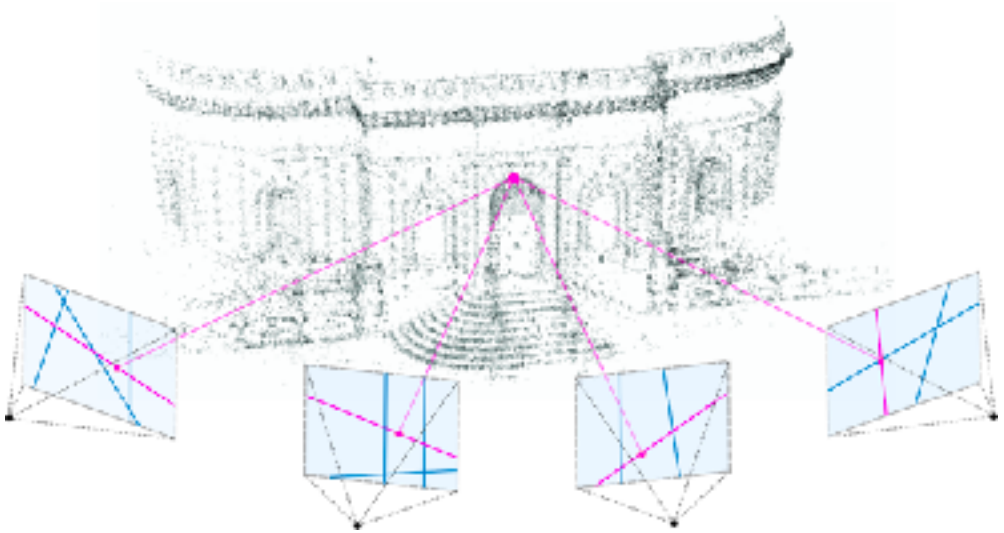
Volumetric
(e.g., Voxels)

Projected RGBD

This typically leads to quantization and other approximation-related issues

Why Point Clouds?

Many traditional geometry-based (i.e., unsupervised) 3D reconstruction methods produce 3D point clouds as part of their outputs.



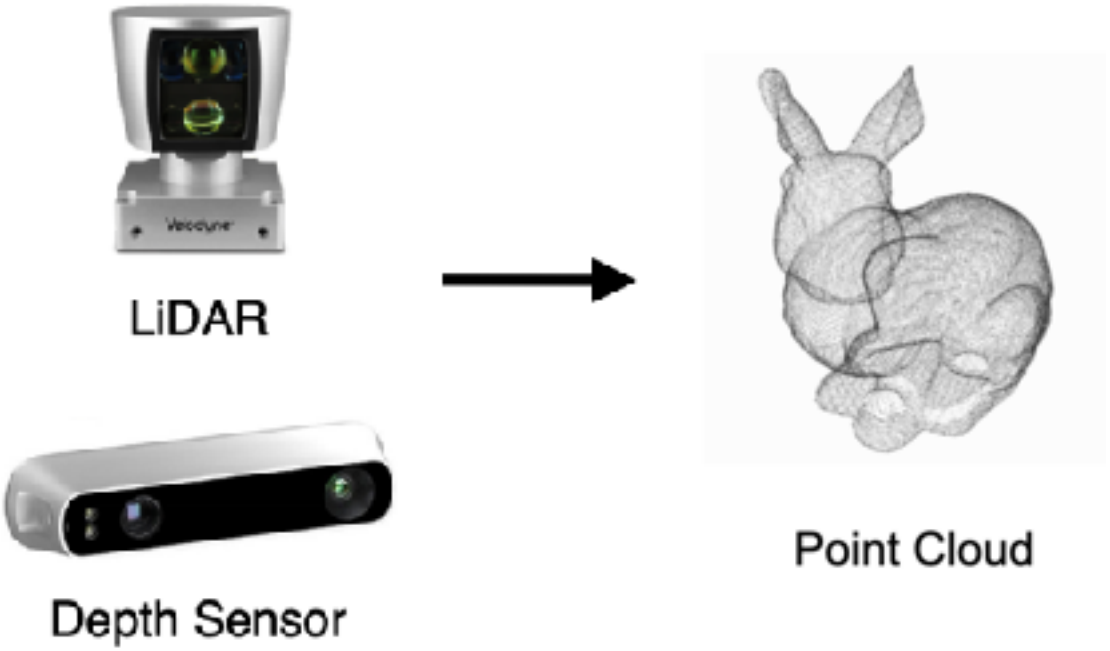
3D Reconstruction using
Structure from Motion (SfM)

Point Cloud Reconstruction
of the Colosseum

No discretization errors!

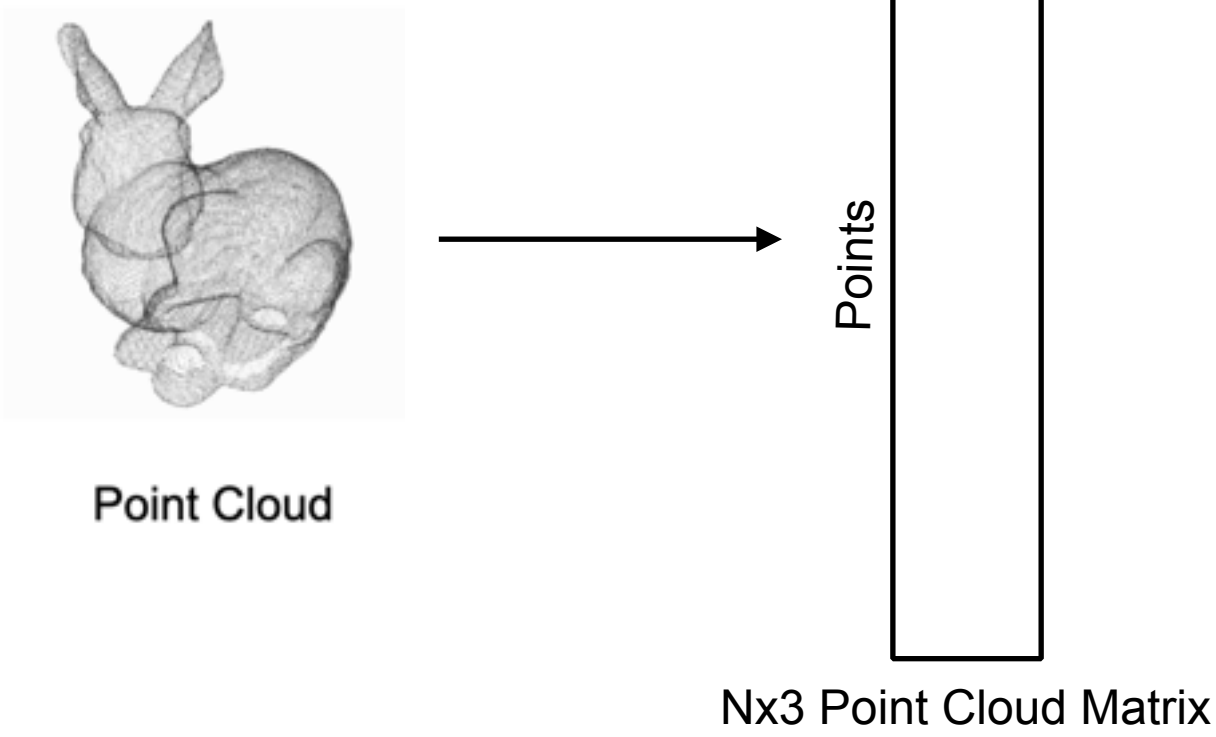
Why Point Clouds?

Furthermore, many traditionally available sensors produce point clouds as its outputs.



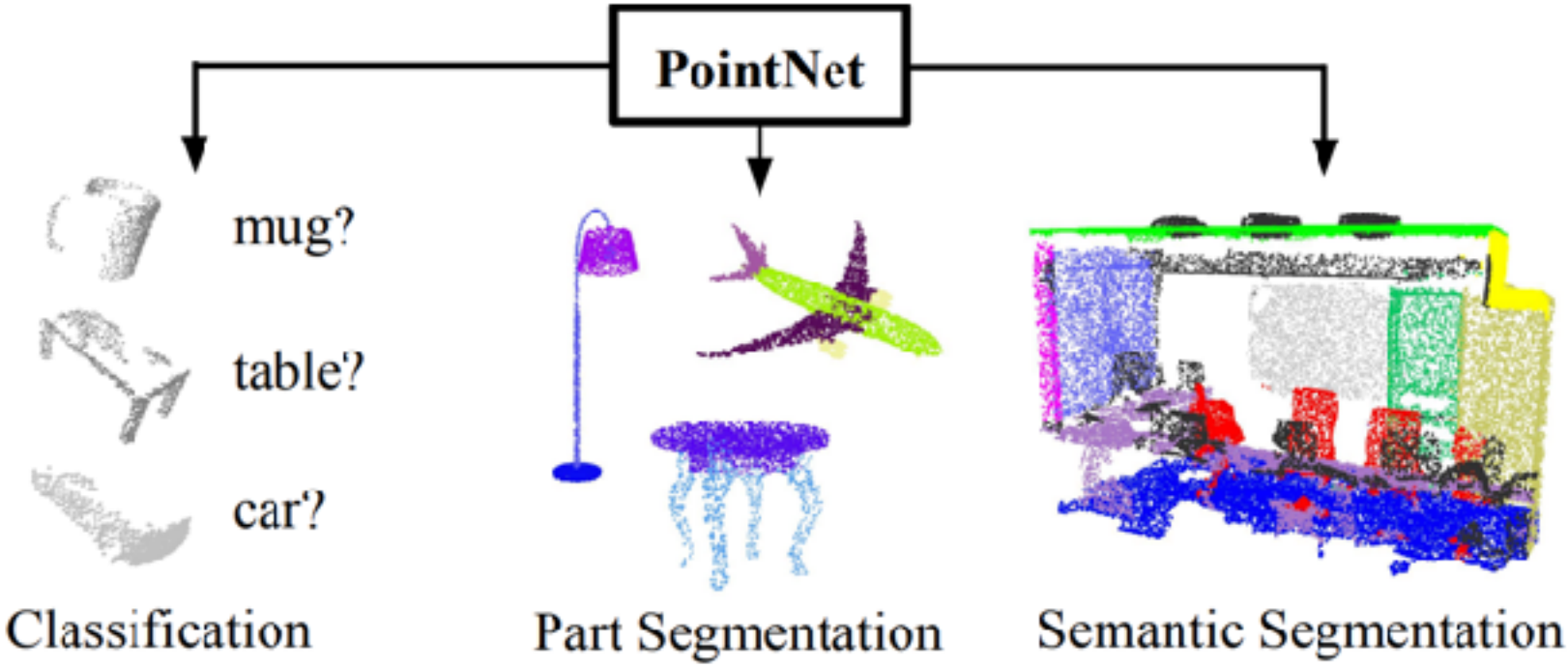
Point Cloud Representation

Point clouds can be represented as a Nx3 matrix where N is the number of points.



Prior Work on Point Clouds

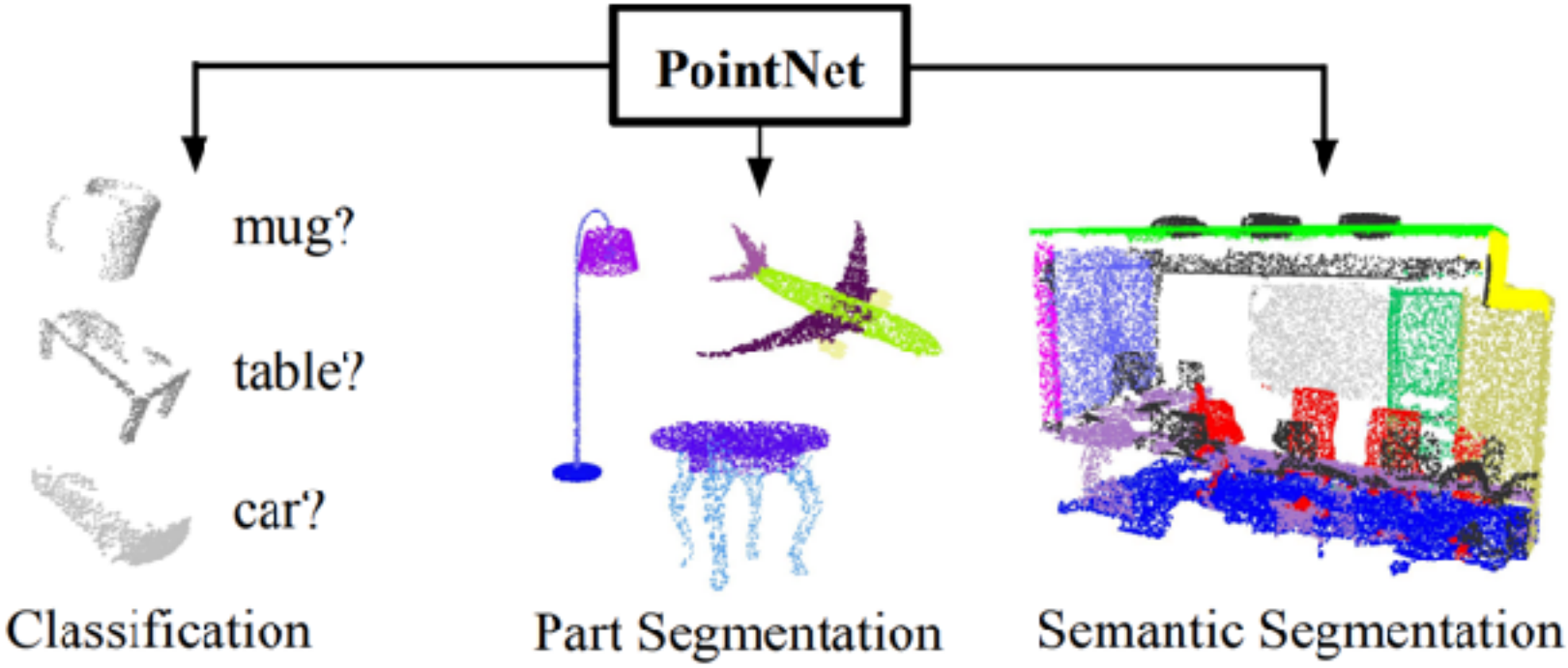
A deep point-based network applied to 3D classification and segmentation tasks.



Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” CVPR 2017

Prior Work on Point Clouds

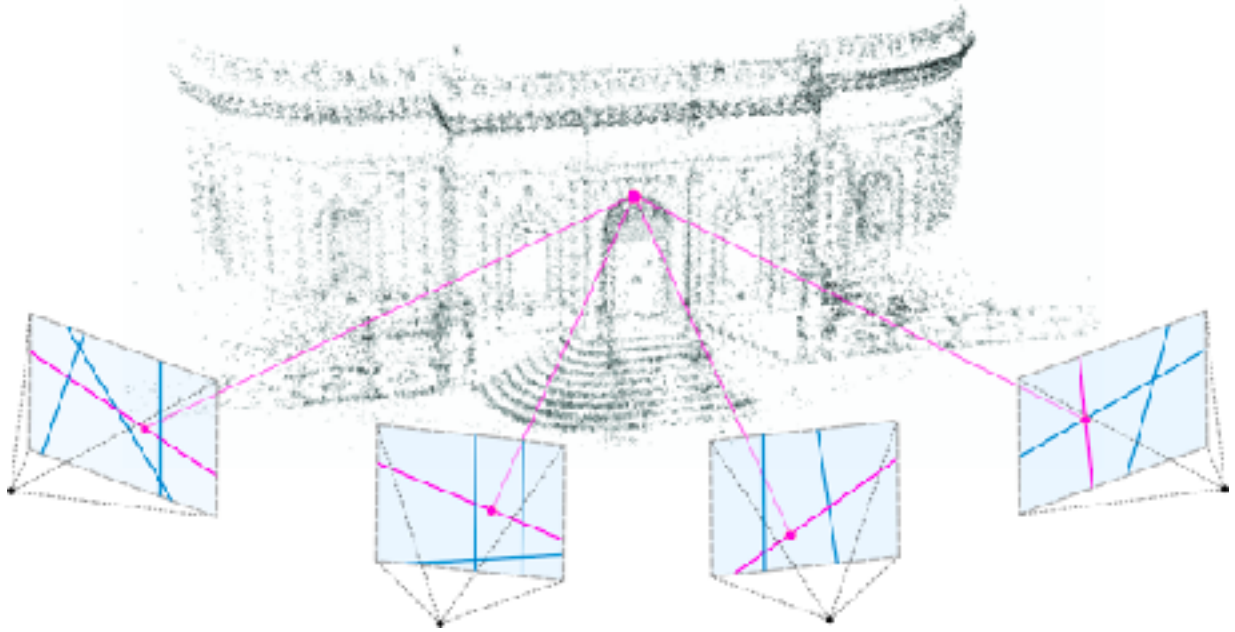
A deep point-based network applied to 3D classification and segmentation tasks.



Can’t easily do 3D bounding box detection using this architecture.

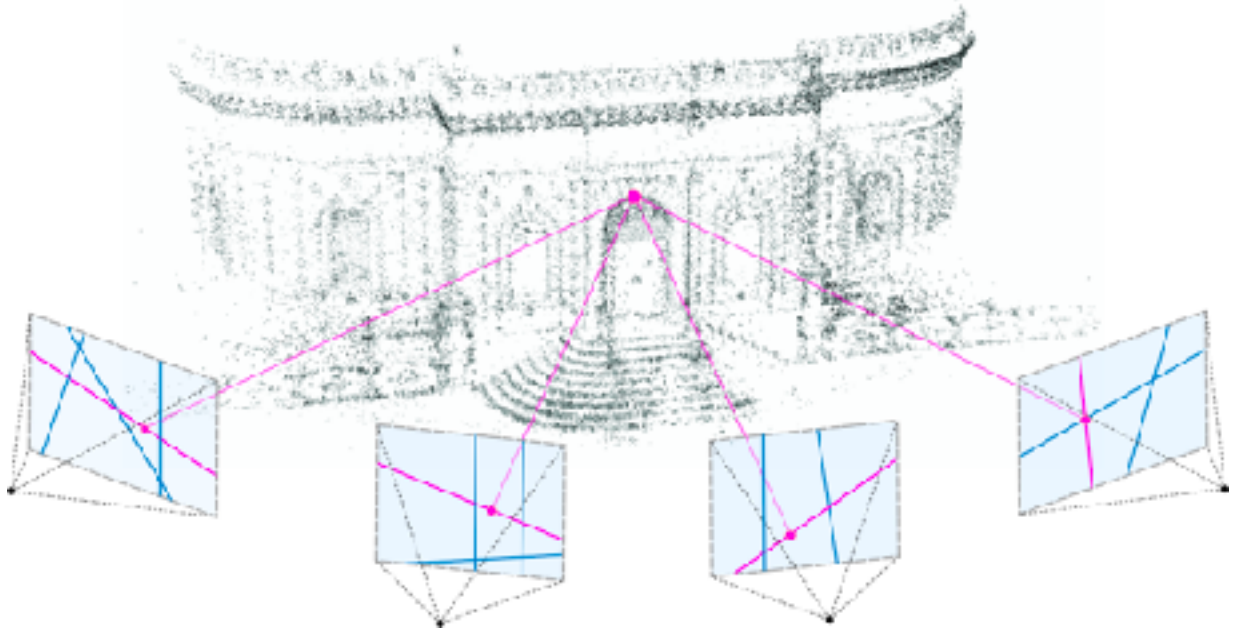
Challenges

- Model needs to be invariant to N! permutations.
- Need to output a set of detections from an unordered set of points.



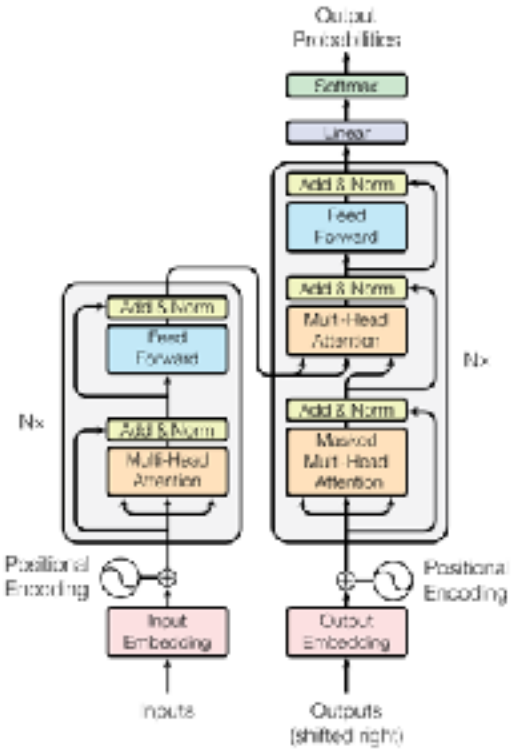
Challenges

- Model needs to be invariant to N! permutations.
- Need to output a set of detections from an unordered set of points.



Challenges

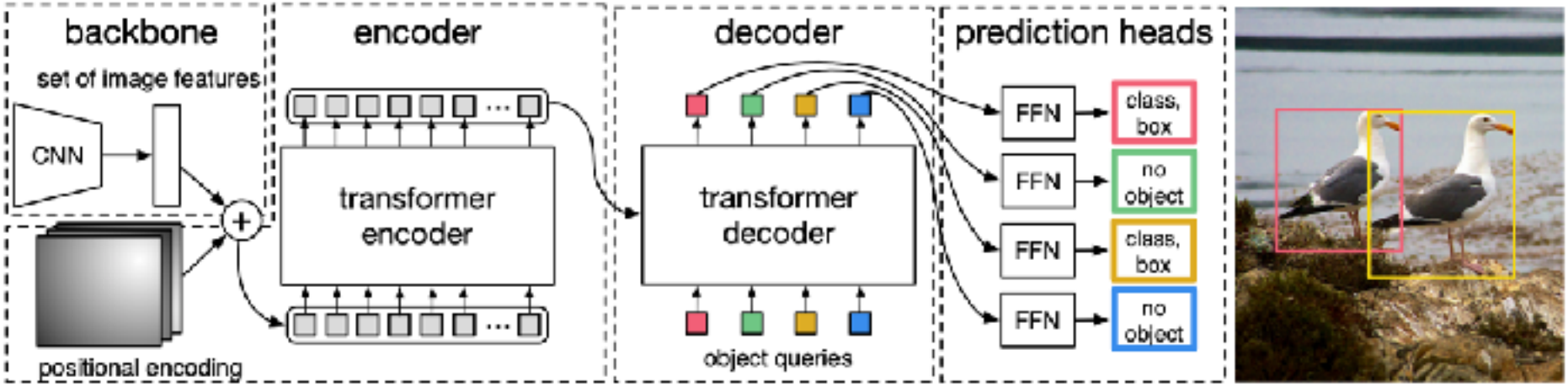
- Model needs to be invariant to N! permutations.
- Need to output a set of detections from an unordered set of points.



Vaswani et al., ”Attention is All You Need”, NIPS 2017

Challenges

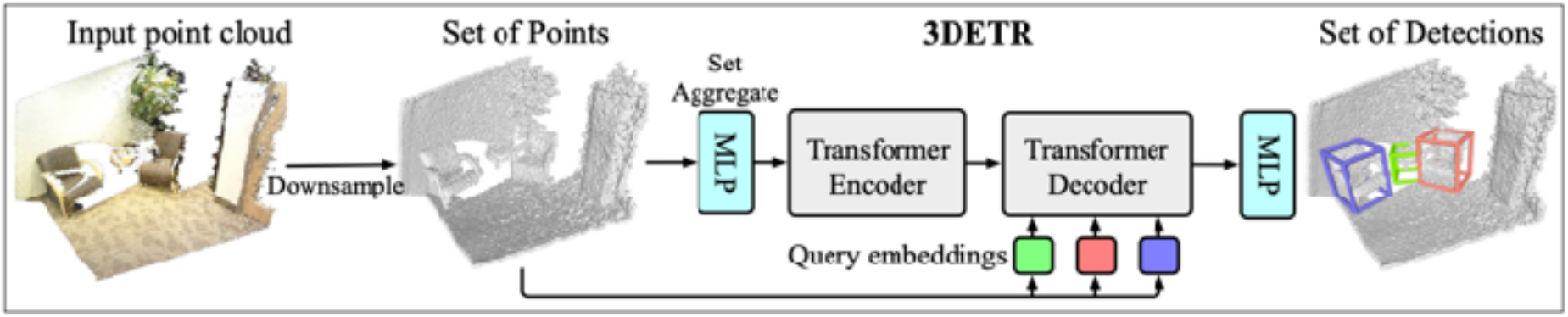
- Model needs to be invariant to N! permutations.
- Need to output a set of detections from an unordered set of points.



Carion et al., ”DE:TR: End-to-End Object Detection with Transformers,” ECCV 2020

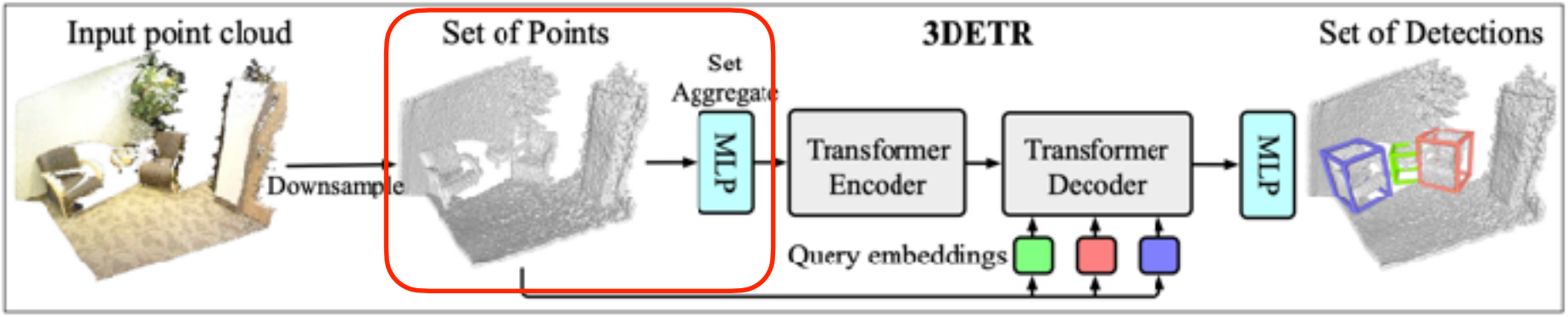
3DETR

- 3DETR is an end-to-end trainable Transformer that takes a set of 3D points (point cloud) as input and outputs a set of 3D bounding boxes.



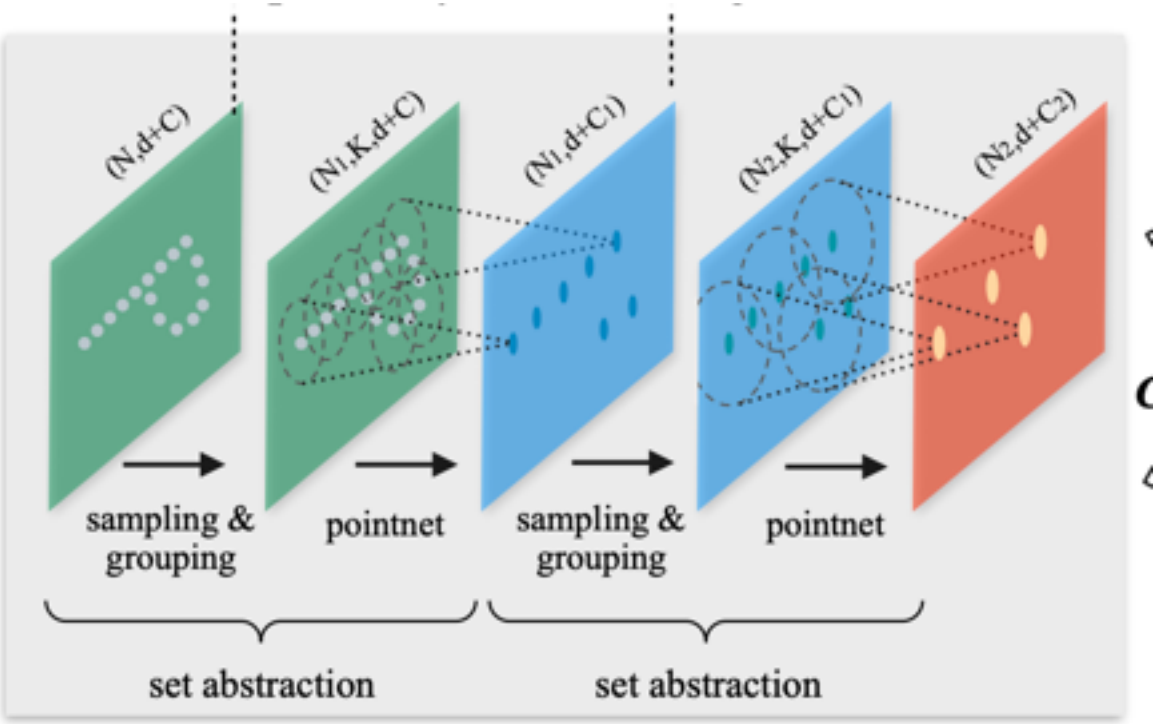
3DETR

- 3DETR is an end-to-end trainable Transformer that takes a set of 3D points (point cloud) as input and outputs a set of 3D bounding boxes.



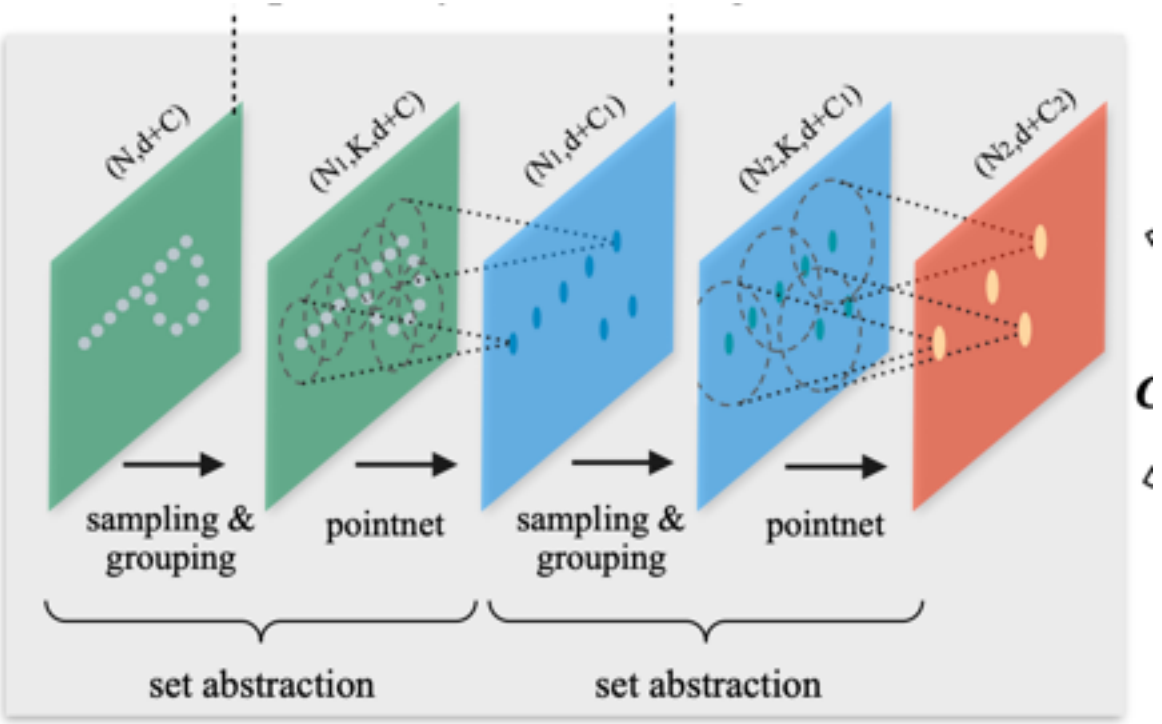
Point Set Aggregation

- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer



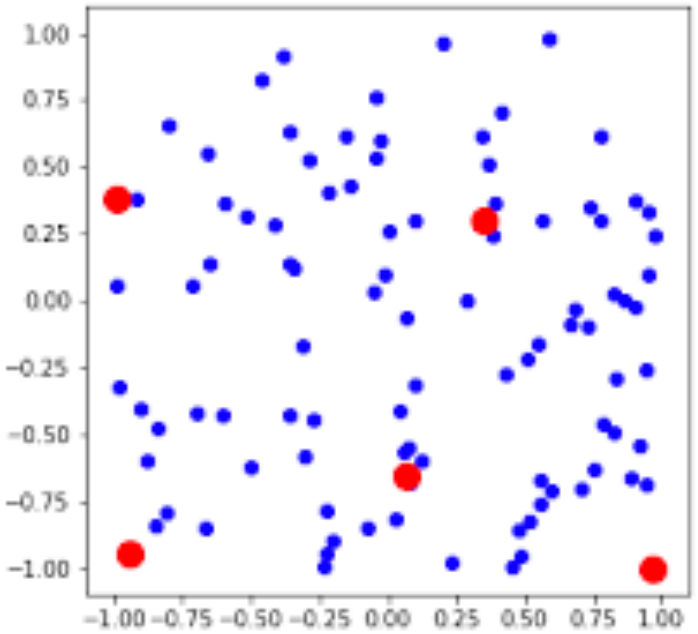
Point Set Aggregation

- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer



Point Set Aggregation

- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer



Point Set Aggregation

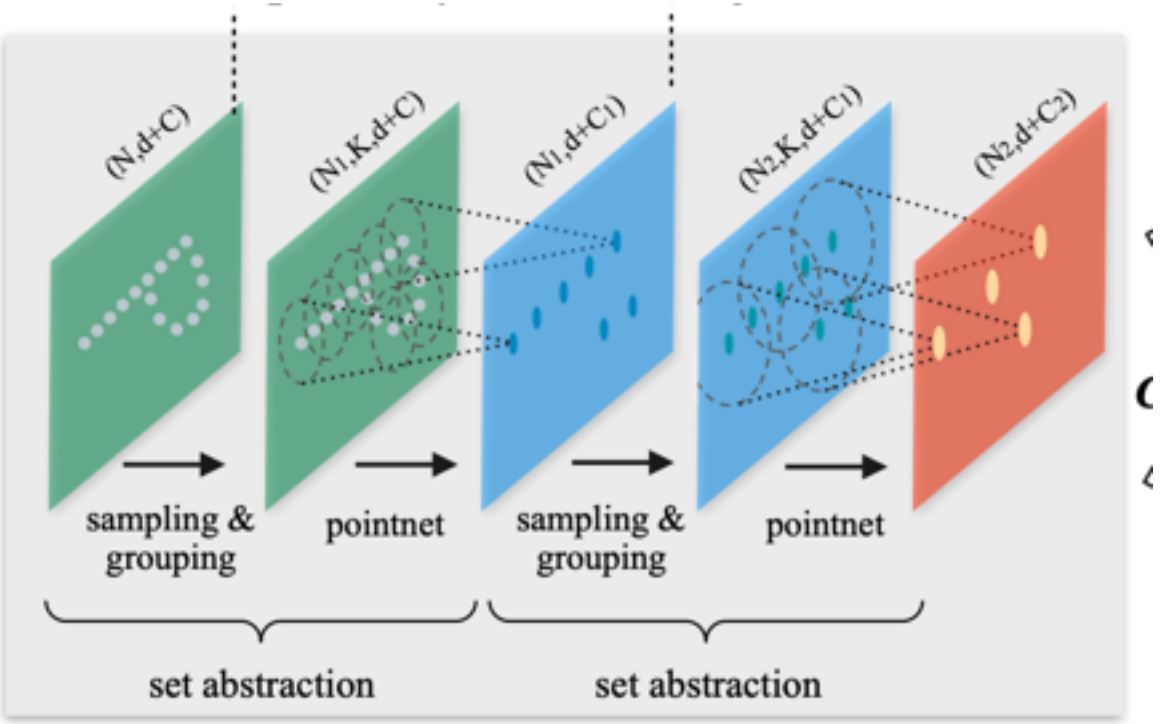
- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer

Farthest Point Sampling

Uniform Sampling

Point Set Aggregation

- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer

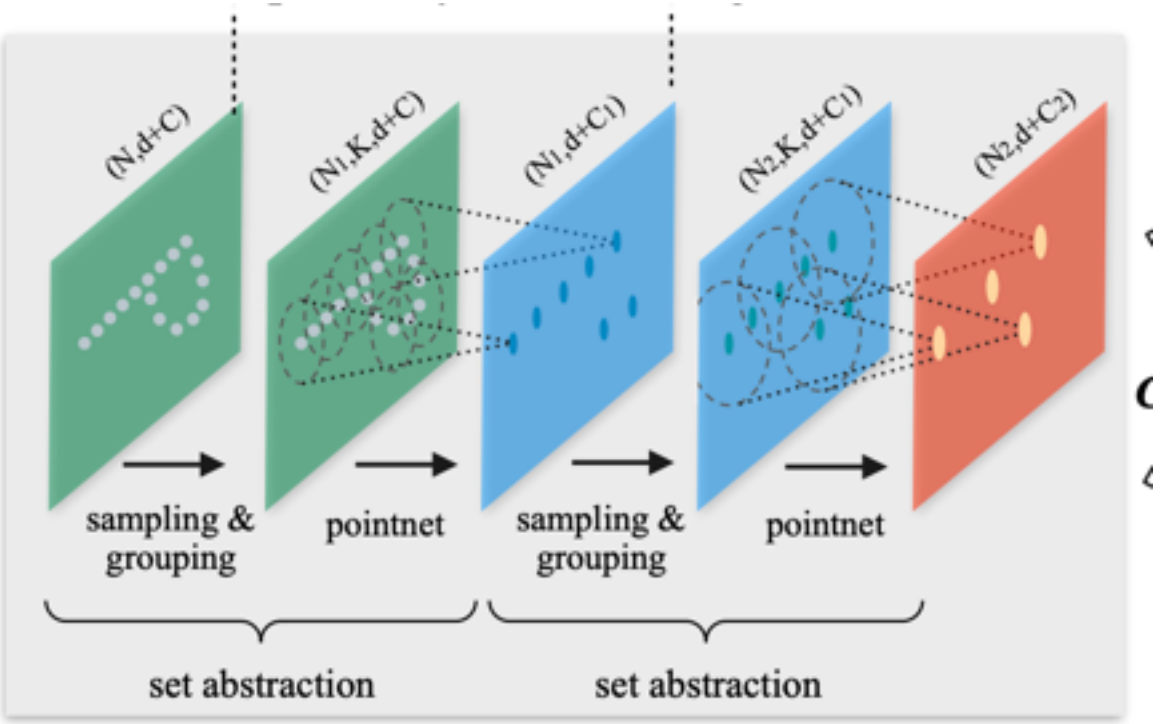


Point Set Aggregation

- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer

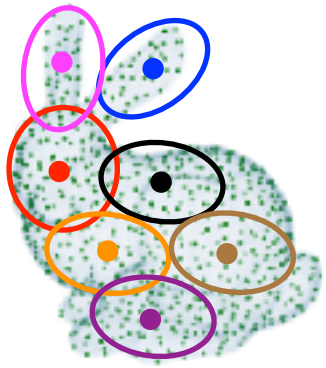
Point Set Aggregation

- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer



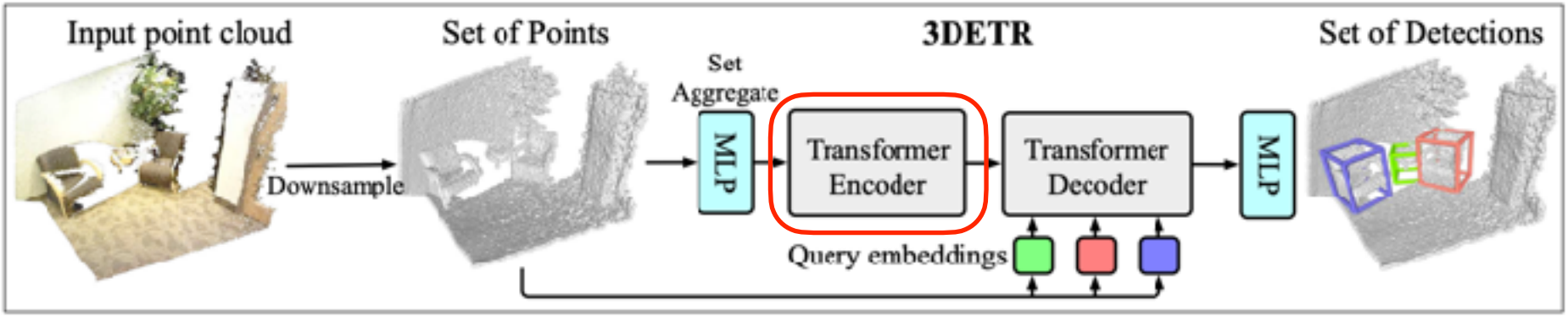
Point Set Aggregation

- The set aggregation scheme is made of three key layers: (1) Sampling layer, (2) Grouping layer and (3) PointNet layer



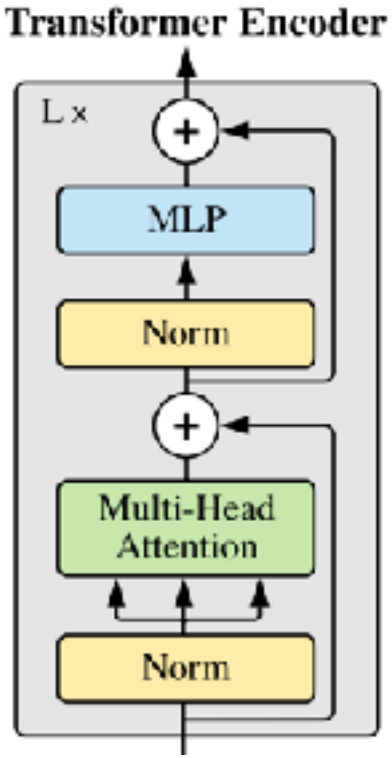
3DETR

- 3DETR is an end-to-end trainable Transformer that takes a set of 3D points (point cloud) as input and outputs a set of 3D bounding boxes.



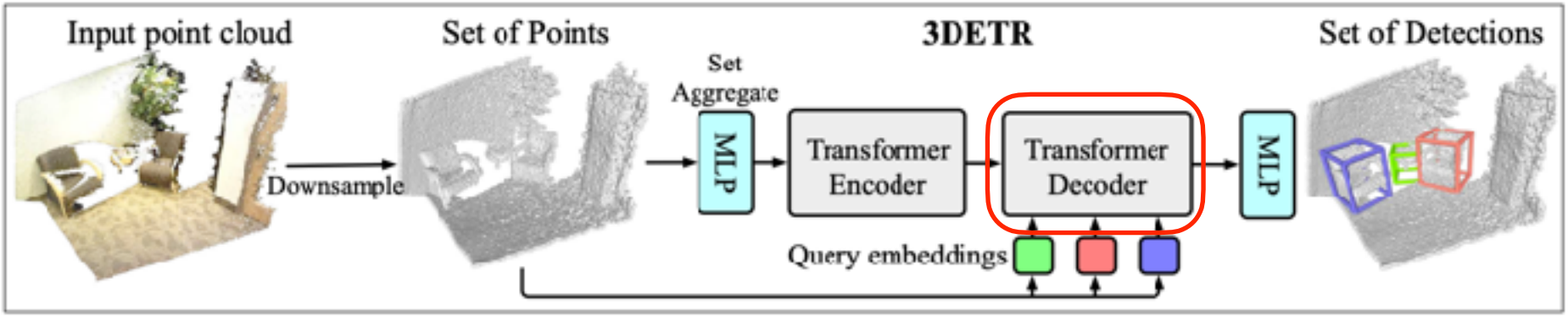
Transformer Encoder

- 3DETR uses a standard multi-head attention transformer encoder.



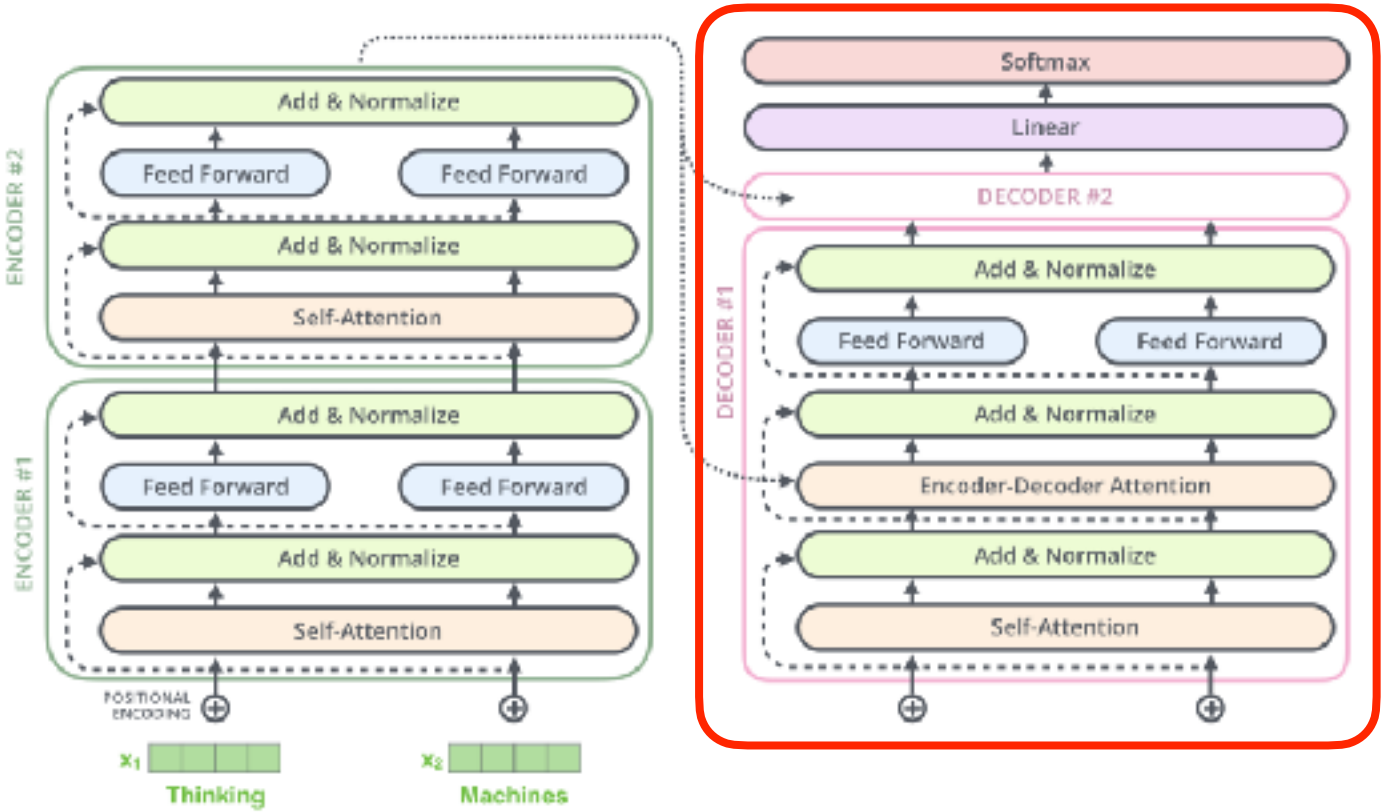
3DETR

- 3DETR is an end-to-end trainable Transformer that takes a set of 3D points (point cloud) as input and outputs a set of 3D bounding boxes.



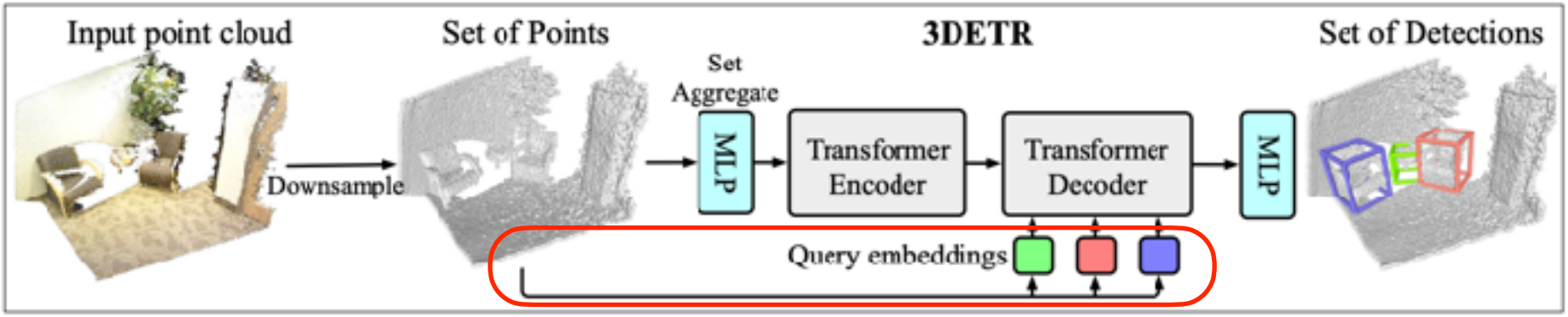
Transformer Decoder

- Compared to the encoder, the decoder additionally has an encoder-decoder attention layer.



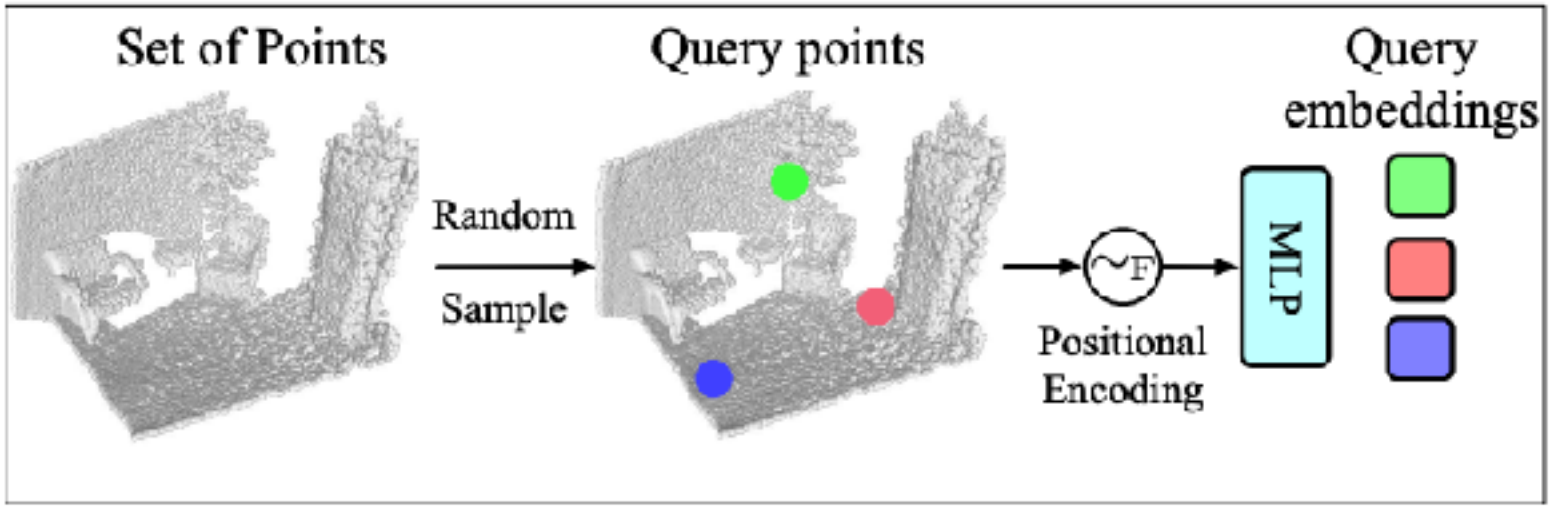
3DETR

- 3DETR is an end-to-end trainable Transformer that takes a set of 3D points (point cloud) as input and outputs a set of 3D bounding boxes.



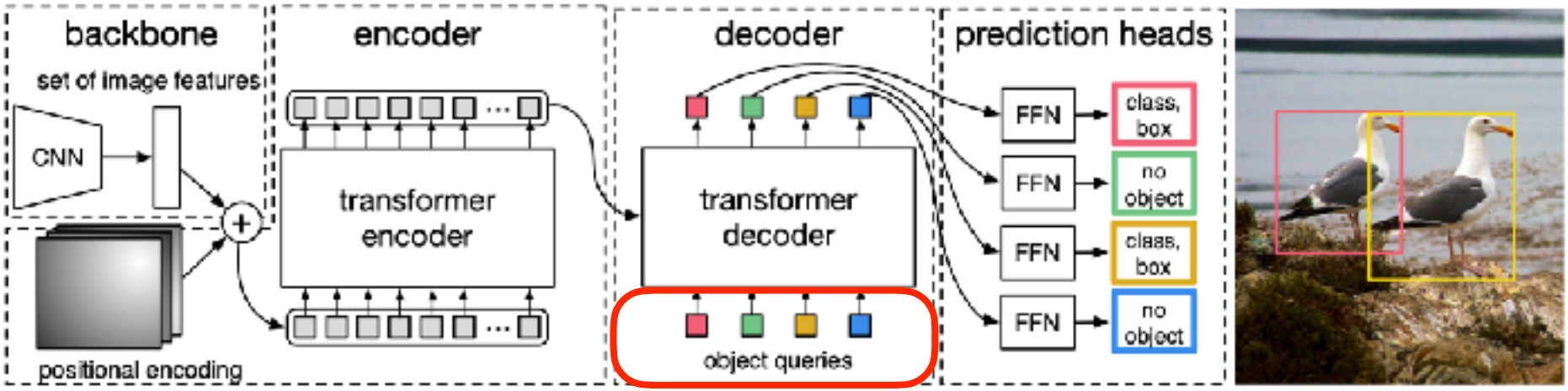
Non-Parametric Decoder Queries

- The ‘query’ points are randomly sampled, and embedded.
- The decoder then used them to produce bounding box predictions.



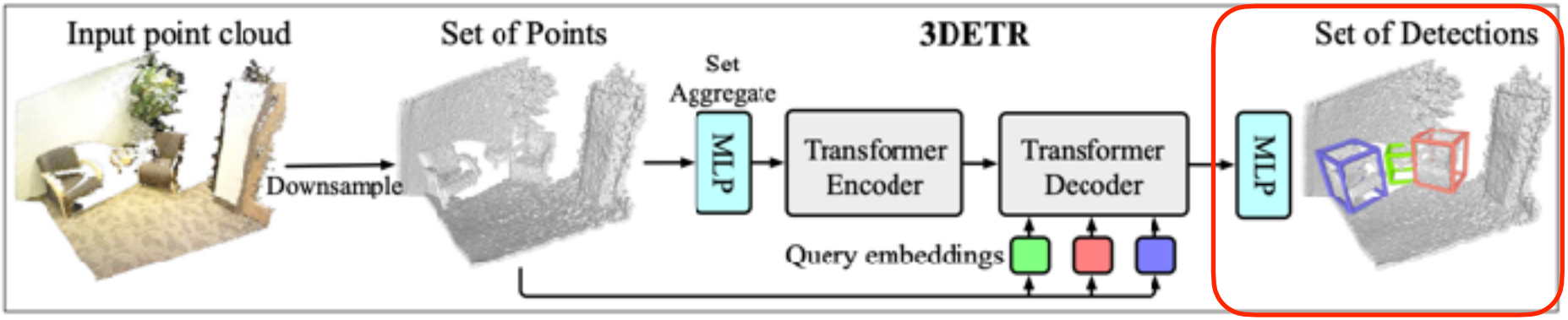
Parametric Decoder Queries

- DETR uses learnable (e.g., parametric) object queries as inputs to the decoder.



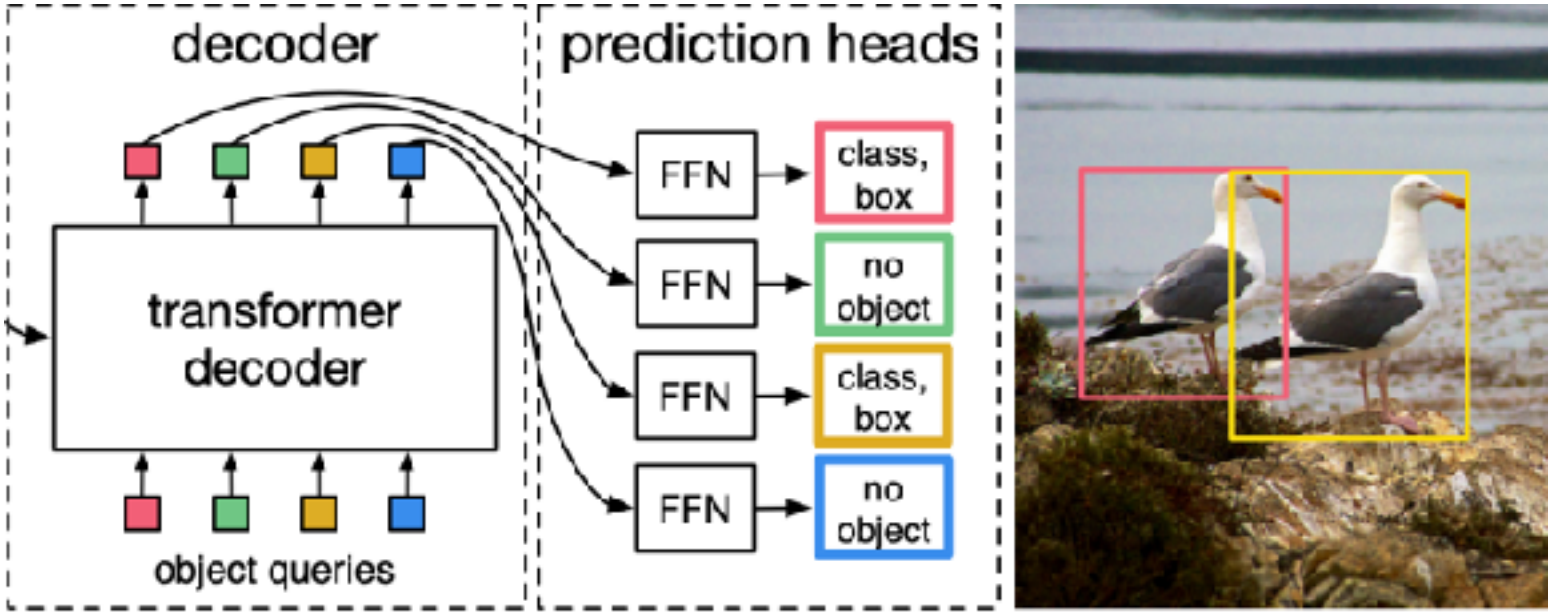
3DETR

- 3DETR is an end-to-end trainable Transformer that takes a set of 3D points (point cloud) as input and outputs a set of 3D bounding boxes.



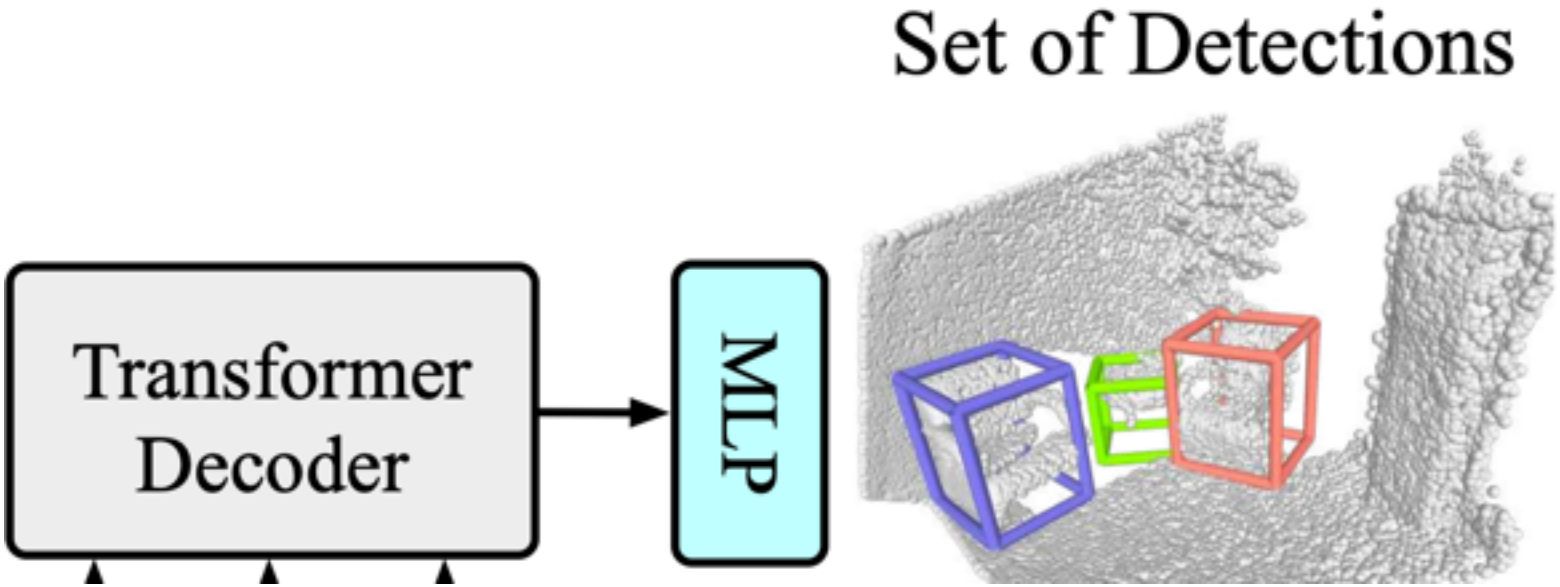
2D Detection

- Mapping of queries to box and class predictions using multilayer perceptrons (MLP).



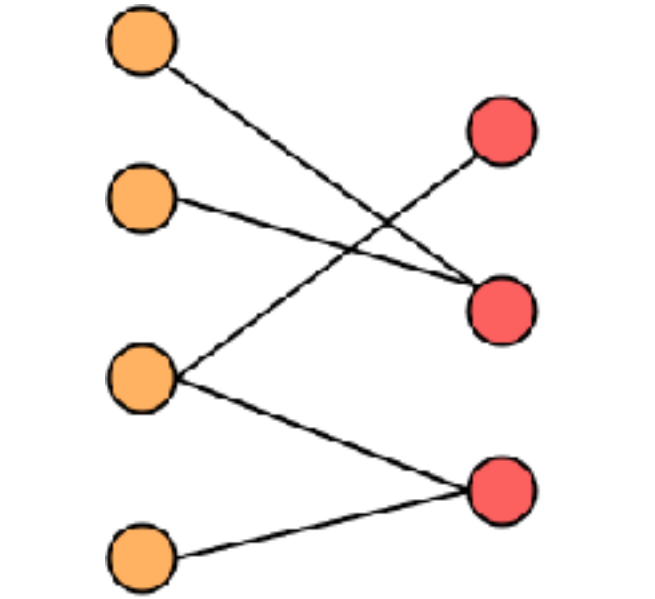
3D Detection

- Mapping of queries to box and class predictions using multilayer perceptrons (MLP).



Bipartite Matching

- The mapping from ground truth 3D objects to the predictions is determined using bipartite matching.
- The loss is then computed using best matches between the predictions and ground truth.



Loss Function

- The final 3D detection loss is a weighted combination of the three following terms:

$$\mathcal{L}_{\text{3DETR}} = \lambda_c \|\hat{\mathbf{c}} - \mathbf{c}\|_1 + \lambda_d \|\hat{\mathbf{d}} - \mathbf{d}\|_1 - \lambda_s \mathbf{s}_c^\intercal \log \hat{\mathbf{s}}_c$$

Loss Function

- The final 3D detection loss is a weighted combination of the three following terms:

L1 regression losses for the center coordinates and box dimensions

$$\mathcal{L}_{\text{3DETR}} = \lambda_c \|\hat{\mathbf{c}} - \mathbf{c}\|_1 + \lambda_d \|\hat{\mathbf{d}} - \mathbf{d}\|_1 - \lambda_s \mathbf{s}_c^\intercal \log \hat{\mathbf{s}}_c$$

Loss Function

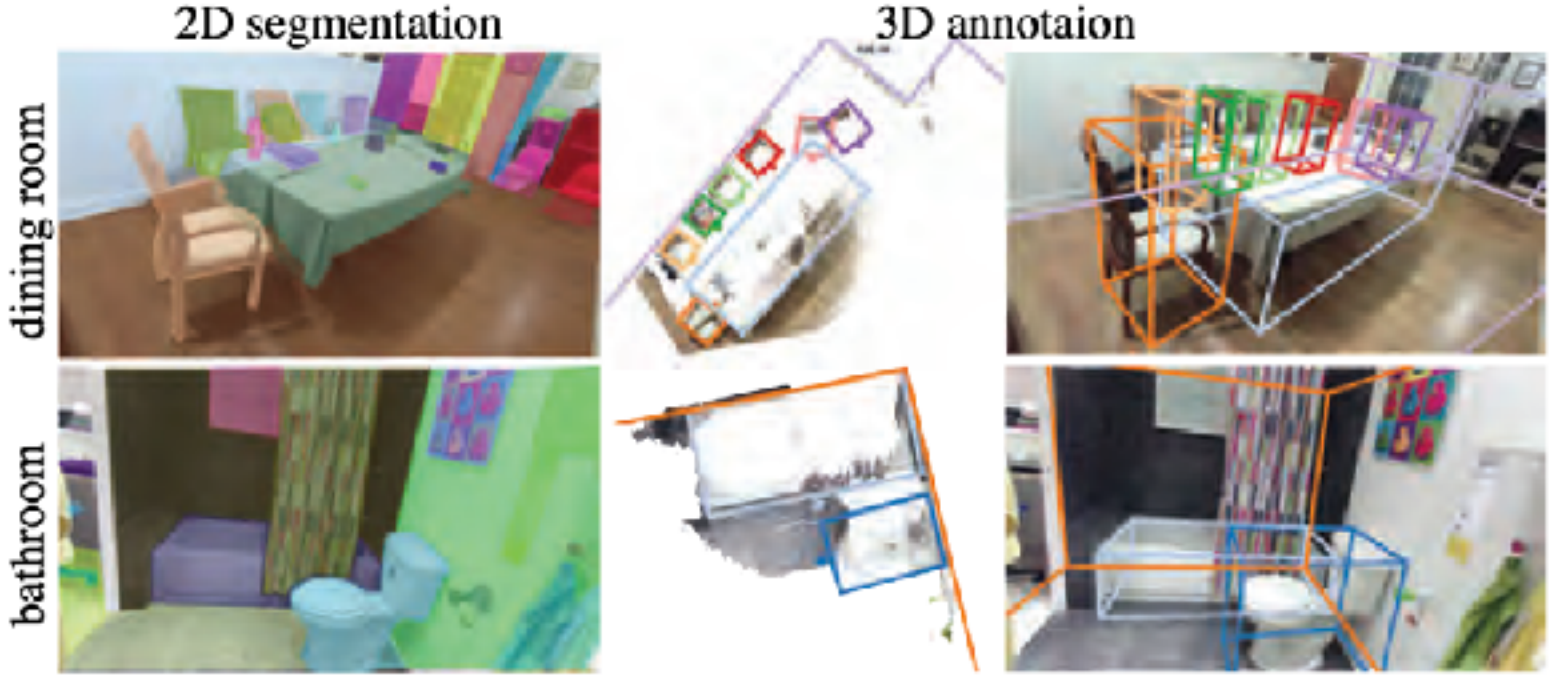
- The final 3D detection loss is a weighted combination of the three following terms:

A cross-entropy loss for semantic classification

$$\mathcal{L}_{\text{3DETR}} = \lambda_c \|\hat{\mathbf{c}} - \mathbf{c}\|_1 + \lambda_d \|\hat{\mathbf{d}} - \mathbf{d}\|_1 - \lambda_s \mathbf{s}_c^{\intercal} \log \hat{\mathbf{s}}_c$$

Experiments

- The evaluation is done on two 3D indoor detection benchmarks - ScanNetV2 and SUN RGB-D-v1.
- The detection performance is evaluated using mean Average Precision at two IoU thresholds of 0.25 and 0.5.



Experiments

- 3DETR achieves comparable results to BoxNet or VoteNet despite having fewer hand-coded 3D specific decisions.

Method	ScanNetV2		SUN RGB-D	
	AP_{25}	AP_{50}	AP_{25}	AP_{50}
BoxNet† [42]	49.0	21.1	52.4	25.1
3DETR	62.7	37.5	56.8	30.1
VoteNet† [42]	60.4	37.5	58.3	33.4
3DETR-m	65.0	47.0	59.0	32.7
H3DNet [89]	67.2	48.1	60.1	39.0

Experiments

- 3DETR achieves comparable results to BoxNet or VoteNet despite having fewer hand-coded 3D specific decisions.

Method	ScanNetV2		SUN RGB-D	
	AP_{25}	AP_{50}	AP_{25}	AP_{50}
BoxNet† [42]	49.0	21.1	52.4	25.1
3DETR	62.7	37.5	56.8	30.1
VoteNet† [42]	60.4	37.5	58.3	33.4
3DETR-m	65.0	47.0	59.0	32.7
H3DNet [89]	67.2	48.1	60.1	39.0

Experiments

- 3DETR achieves comparable results to BoxNet or VoteNet despite having fewer hand-coded 3D specific decisions.

Method	ScanNetV2		SUN RGB-D	
	AP_{25}	AP_{50}	AP_{25}	AP_{50}
BoxNet† [42]	49.0	21.1	52.4	25.1
3DETR	62.7	37.5	56.8	30.1
VoteNet† [42]	60.4	37.5	58.3	33.4
3DETR-m	65.0	47.0	59.0	32.7
H3DNet [89]	67.2	48.1	60.1	39.0

Experiments

- 3DETR achieves comparable results to BoxNet or VoteNet despite having fewer hand-coded 3D specific decisions.

Method	ScanNetV2		SUN RGB-D	
	AP_{25}	AP_{50}	AP_{25}	AP_{50}
BoxNet† [42]	49.0	21.1	52.4	25.1
3DETR	62.7	37.5	56.8	30.1
VoteNet† [42]	60.4	37.5	58.3	33.4
3DETR-m	65.0	47.0	59.0	32.7
H3DNet [89]	67.2	48.1	60.1	39.0

Ablations

Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
				AP_{25}	AP_{50}	AP_{25}	AP_{50}
3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
	PN++	Tx.	Set	61.4	34.7	56.8	26.9

#	Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
					AP_{25}	AP_{50}	AP_{25}	AP_{50}
Comparing different decoders								
1	3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
2		Tx.	Box	Box	31.0	10.2	36.4	14.4
3		Tx.	Vote	Vote	46.1	23.4	47.5	24.9
Comparing different losses								
4		Tx.	Tx.	Box	49.6	20.5	49.5	21.1
5		Tx.	Tx.	Vote	54.0	31.9	53.4	28.3

Ablations

Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
				AP_{25}	AP_{50}	AP_{25}	AP_{50}
3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
	PN++	Tx.	Set	61.4	34.7	56.8	26.9

#	Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
					AP_{25}	AP_{50}	AP_{25}	AP_{50}
Comparing different decoders								
1	3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
2		Tx.	Box	Box	31.0	10.2	36.4	14.4
3		Tx.	Vote	Vote	46.1	23.4	47.5	24.9
Comparing different losses								
4		Tx.	Tx.	Box	49.6	20.5	49.5	21.1
5		Tx.	Tx.	Vote	54.0	31.9	53.4	28.3

Ablations

Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
				AP_{25}	AP_{50}	AP_{25}	AP_{50}
3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
	PN++	Tx.	Set	61.4	34.7	56.8	26.9

#	Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
					AP_{25}	AP_{50}	AP_{25}	AP_{50}
Comparing different decoders								
1	3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
2		Tx.	Box	Box	31.0	10.2	36.4	14.4
3		Tx.	Vote	Vote	46.1	23.4	47.5	24.9
Comparing different losses								
4		Tx.	Tx.	Box	49.6	20.5	49.5	21.1
5		Tx.	Tx.	Vote	54.0	31.9	53.4	28.3

Ablations

Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
				AP_{25}	AP_{50}	AP_{25}	AP_{50}
3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
	PN++	Tx.	Set	61.4	34.7	56.8	26.9

#	Method	Encoder	Decoder	Loss	ScanNetV2		SUN RGB-D	
					AP_{25}	AP_{50}	AP_{25}	AP_{50}
Comparing different decoders								
1	3DETR	Tx.	Tx.	Set	62.7	37.5	56.8	30.1
2		Tx.	Box	Box	31.0	10.2	36.4	14.4
3		Tx.	Vote	Vote	46.1	23.4	47.5	24.9
Comparing different losses								
4		Tx.	Tx.	Box	49.6	20.5	49.5	21.1
5		Tx.	Tx.	Vote	54.0	31.9	53.4	28.3

Ablations

- Using non-parametric queries significantly affect detection performance.

#	Method	Positional Embedding		Query Type	ScanNetV2	
		Encoder	Decoder		AP_{25}	AP_{50}
1	3DETR					
2						
3		Sine	Sine	np + Sine	55.8	30.9
4		-				
5	DETR [4]†	Sine	Sine	parametric [4]	15.4	5.3

Ablations

- Using non-parametric queries significantly affect detection performance.

#	Method	Positional Embedding		Query Type	ScanNetV2	
		Encoder	Decoder		AP_{25}	AP_{50}
1	3DETR					
2						
3		Sine	Sine	np + Sine	55.8	30.9
4		-				
5	DETR [4]†	Sine	Sine	parametric [4]	15.4	5.3

Discussion Questions

- What are the main differences with DETR? Is the technical contribution significant enough?

Discussion Questions

- What are the main differences with DETR? Is the technical contribution significant enough?

- Why such a big difference between non-parametric and parametric decoder queries?

Summary

- A simple end-to-end Transformer model for 3D detection on point clouds.
- 3DETR requires few 3D specific design decisions or hyper-parameters.
- Not a lot of technical novelty but timely and relevant topic with solid results.