Advanced Deep Learning for 3D Spatial Data

- Deep Learning in 3D for Robotics (a.k.a. too much for 4 hours) -

Prof Matteo Matteucci (matteo.matteucci@polimi.it)

Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

«Me, Myself, and I»

Matteo Matteucci, PhD

Full Professor

Dept. of Electronics, Information & Bioengineering
Politecnico di Milano

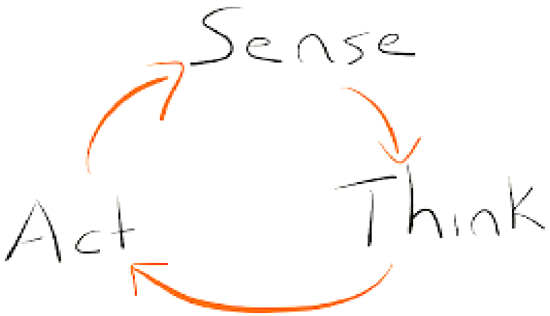
matteo.matteucci@polimi.it

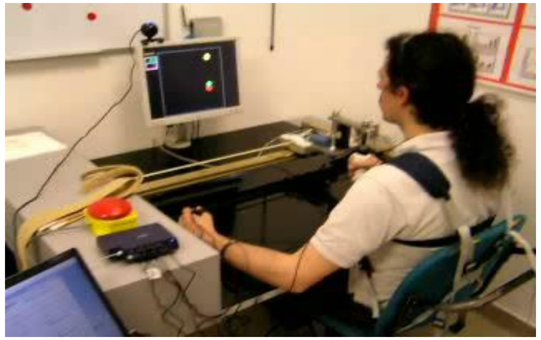
My research interests

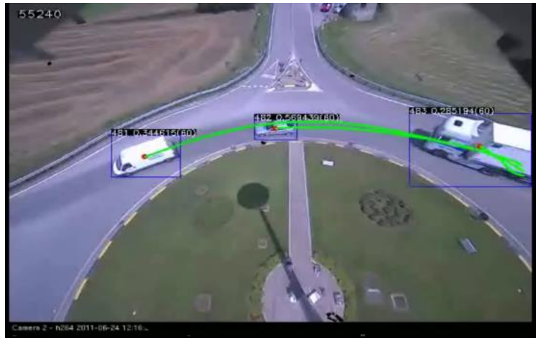
- Robotics & Autonomous Systems
- Machine Learning
- Pattern Recognition
- Computer Vision & Perception

Courses I teach

- Robotics (BS + MS)
- Cognitive Robotics (MS)
- Machine Learning (MS)
- Deep Learning (PhD)







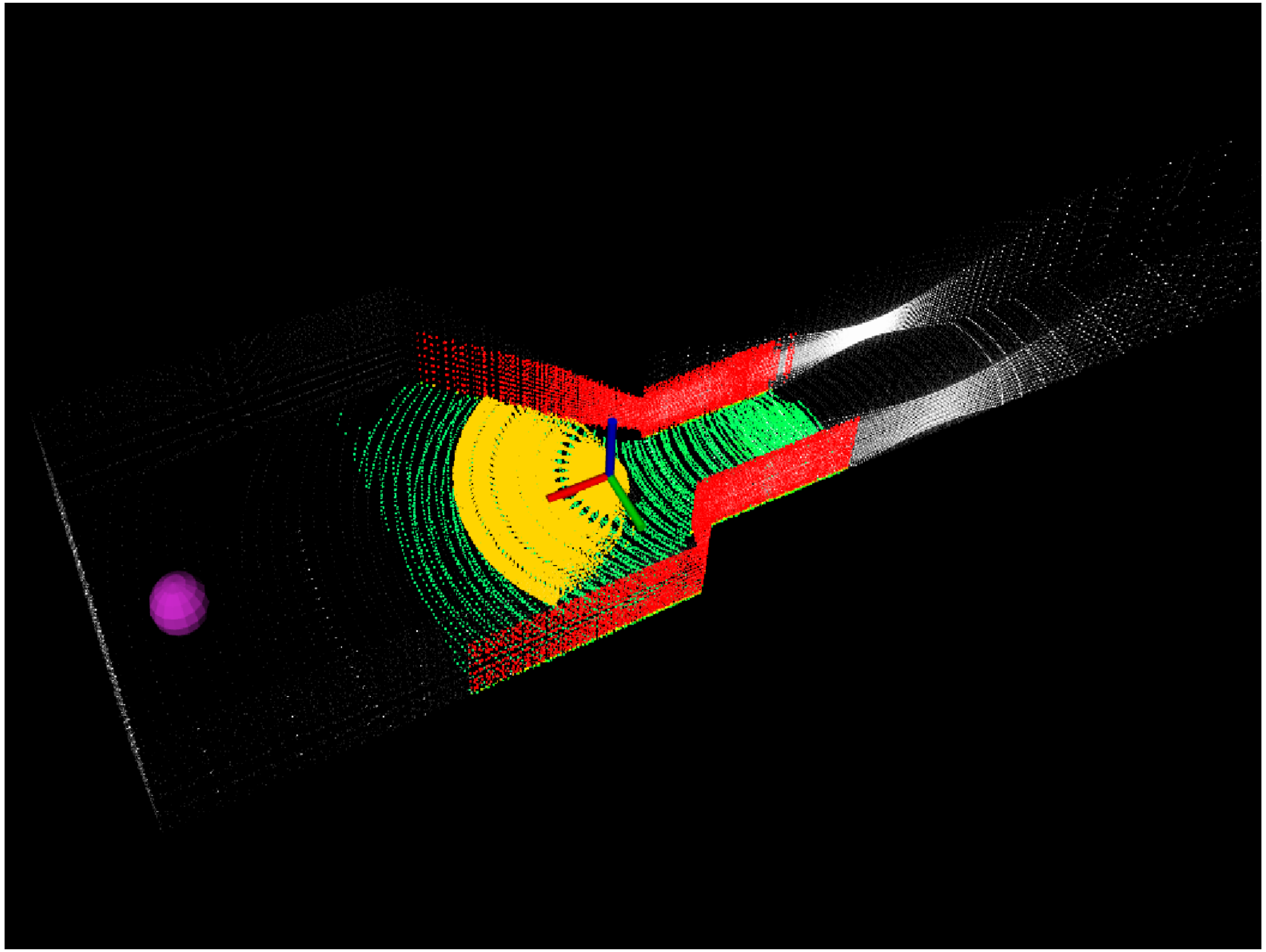
Enable physical and software autonomous systems to perceive, plan, and act without human intervention in the real world

«Me, Myself, and I»

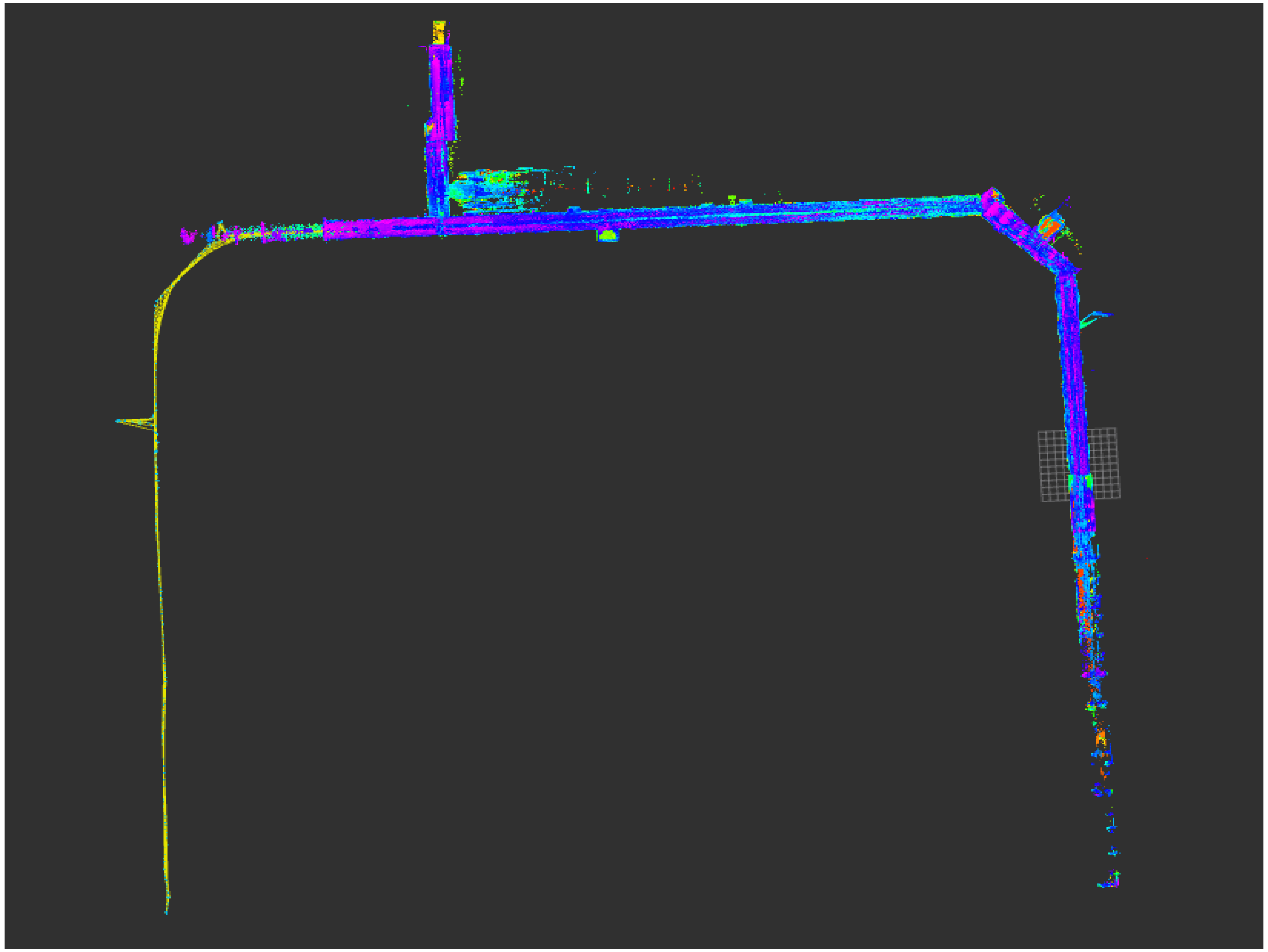
A Recent Example




A Recent Example



A Recent Example

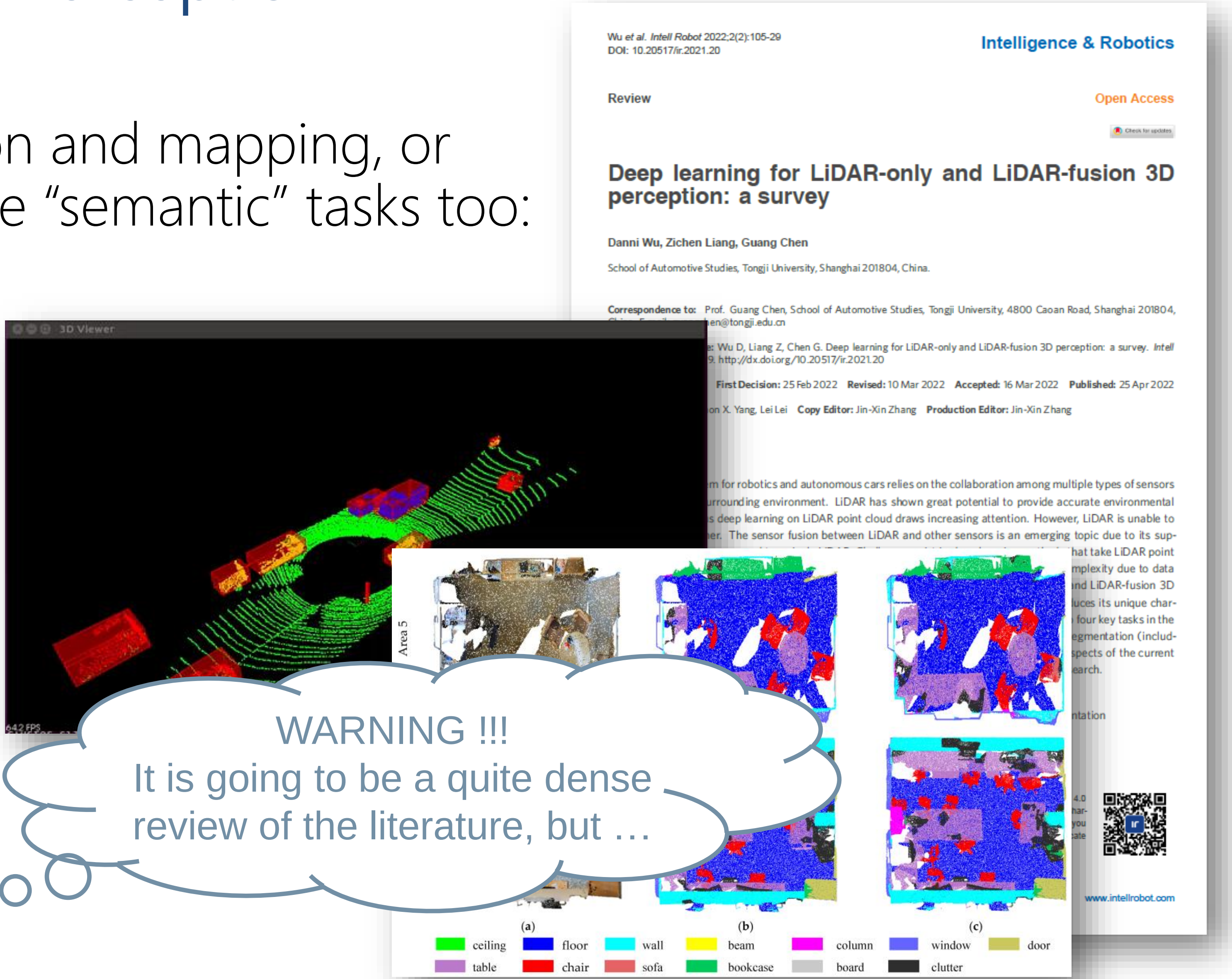


The DARPA Subterranean Challenge

Tasks for 3D Data in Robot Perception

Beside (simultaneous) localization and mapping, or autonomous navigation, we have "semantic" tasks too:

- 3D Shape Classification
- <u>3D Object Detection</u>
- 3D Object Tracking
- <u>3D Segmentation</u>
- 3D Instance Segmentation
- <u>3D Cooperative Perception</u>
- <u>3D Place Recognition</u>
- …



WARNING !!!
It is going to be a quite dense review of the literature, but …

We will look at some of these …

I'm not alone!

This lecture has been prepared with the contribution of (in order of appearance)

Simone Mentasti
simone.mentasti@polimi.it

Matteo Frosi
matteo.frosi@polimi.it

Lorenzo Cazzella
lorenzo.cazzella@polimi.it

Daniele Cattaneo
daniele.cattaneo@disco.unimib.it

POLITECNICO
MILANO 1863

Deep Learning in 3D for Robotics

- Object Detection in 3D Point Clouds -

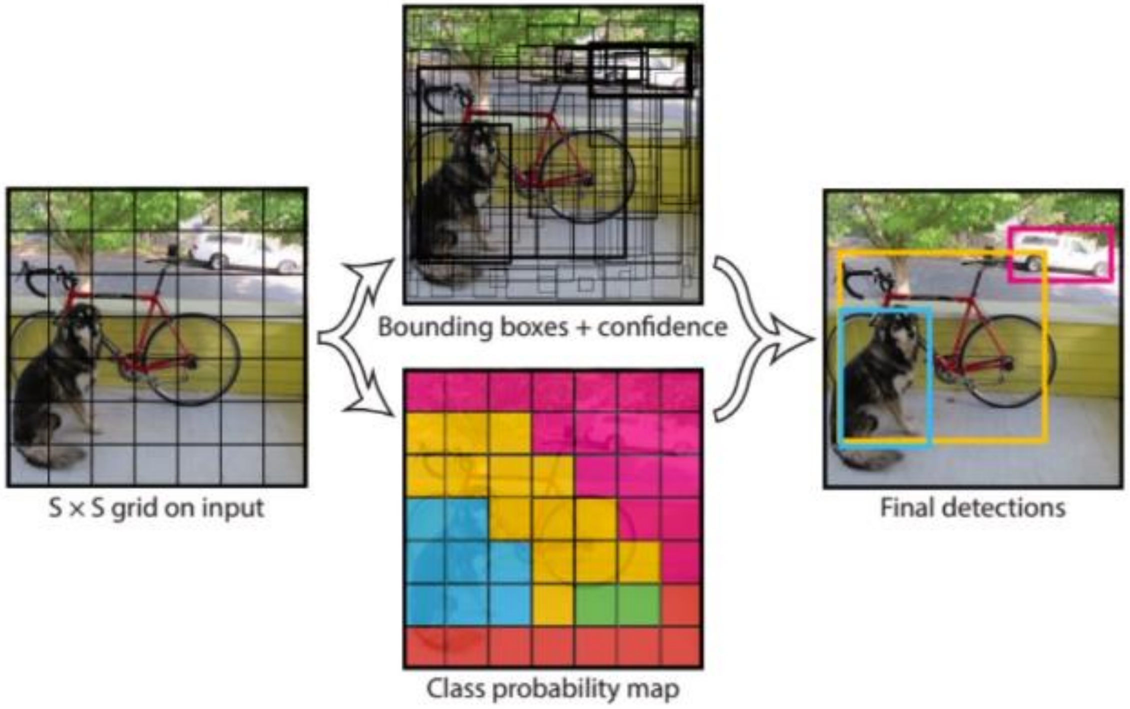
Matteo Matteucci (matteo.matteucci@polimi.it) and Simone Mentasti (simone.mentasti@polimi.it)

Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

AIRLAB
ARTIFICIAL INTELLIGENCE AND ROBOTICS LAB

What is object detection?

The 2D scenario we all know....



**You Only Look Once:
Unified, Real-Time Object Detection**

Joseph Redmon*, Santosh Divvala*†, Ross Girshick¶, Ali Farhadi*†
University of Washington*, Allen Institute for AI†, Facebook AI Research¶
http://pjreddie.com/yolo/

Abstract

We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.

1. Introduction

Humans glance at an image and instantly know what objects are in the image, where they are, and how they interact. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast, accurate algorithms for object detection would allow computers to drive cars without specialized sensors, enable assistive devices to convey real-time scene information to human users, and unlock the potential for general purpose, responsive robotic systems.

Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image [10].

More recent approaches like R-CNN use region proposal

1. Resize image.
2. Run convolutional network.
3. Non-max suppression.

Figure 1: The YOLO Detection System. Processing images with YOLO is simple and straightforward. Our system (1) resizes the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene [13]. These complex pipelines are slow and hard to optimize because each individual component must be trained separately.

We reframe object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Using our system, you only look once (YOLO) at an image to predict what objects are present and where they are.

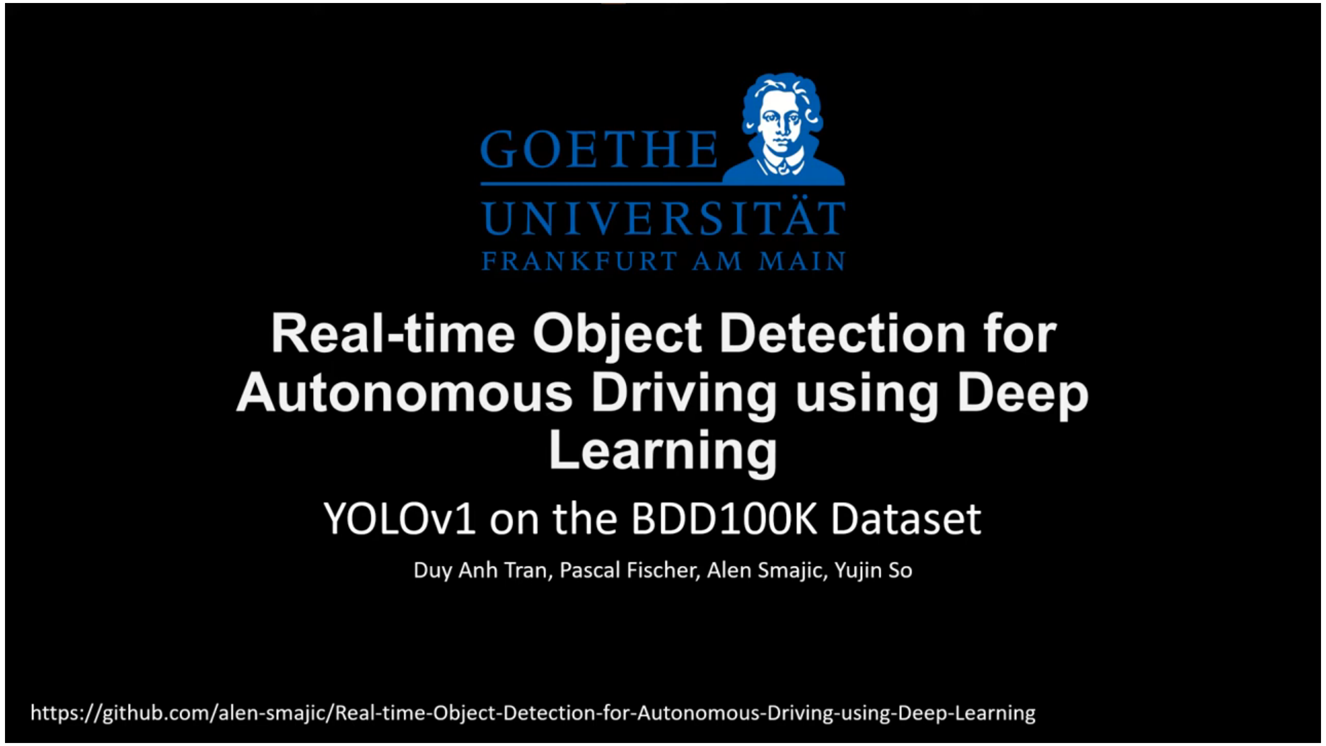
YOLO is refreshingly simple: see Figure 1. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection.

First, YOLO is extremely fast. Since we frame detection as a regression problem we don't need a complex pipeline. We simply run our neural network on a new image at test time to predict detections. Our base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. This means we can process streaming video in real-time with less than 25 milliseconds of latency. Furthermore, YOLO achieves more than twice the mean average precision of other real-time systems. For a demo of our system running in real-time on a webcam please see our project webpage: http://pjreddie.com/yolo/.

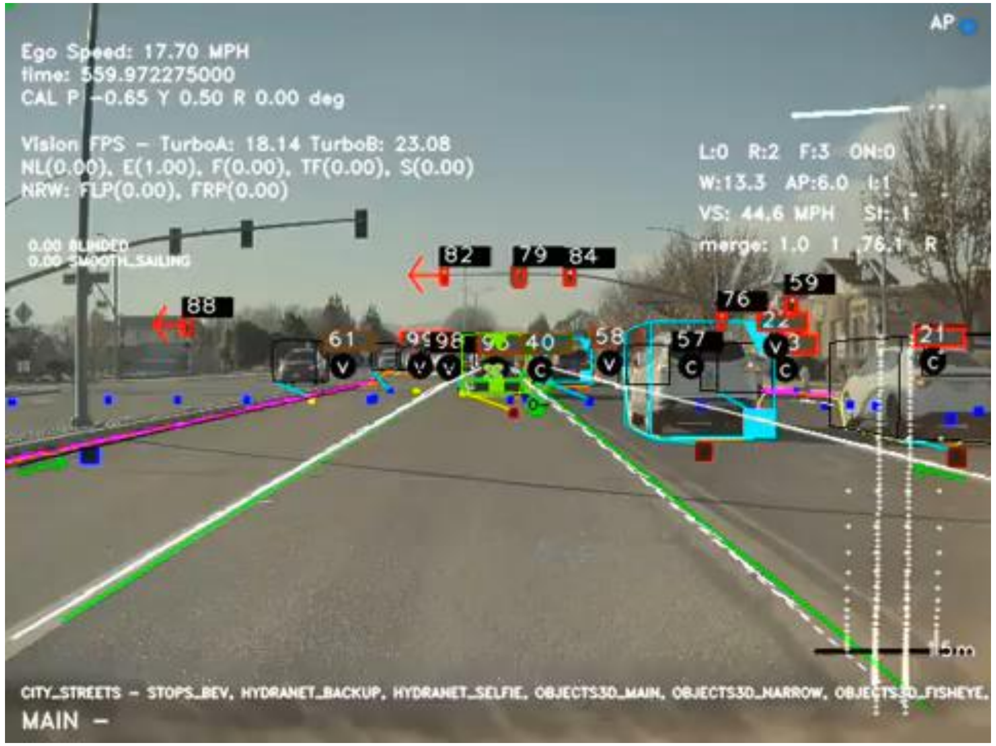
Second, YOLO reasons globally about the image when

What is object detection?

Yolo on BDD100k



What TESLA was seeing (2022)

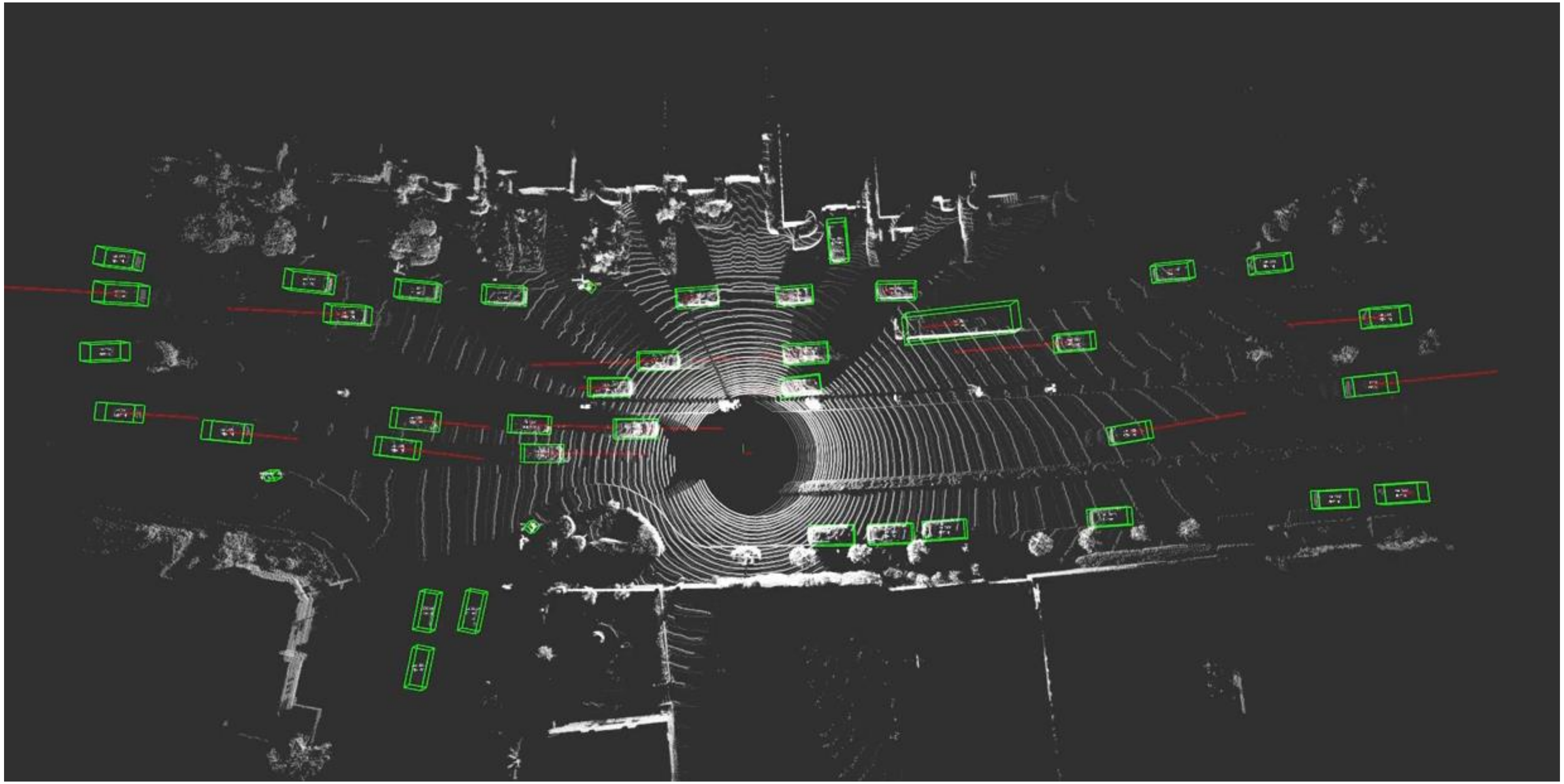


What changes with PointCloud?

3D data are a bit different....

What changes with PointCloud?

… but we expect at least 3D bounding boxes

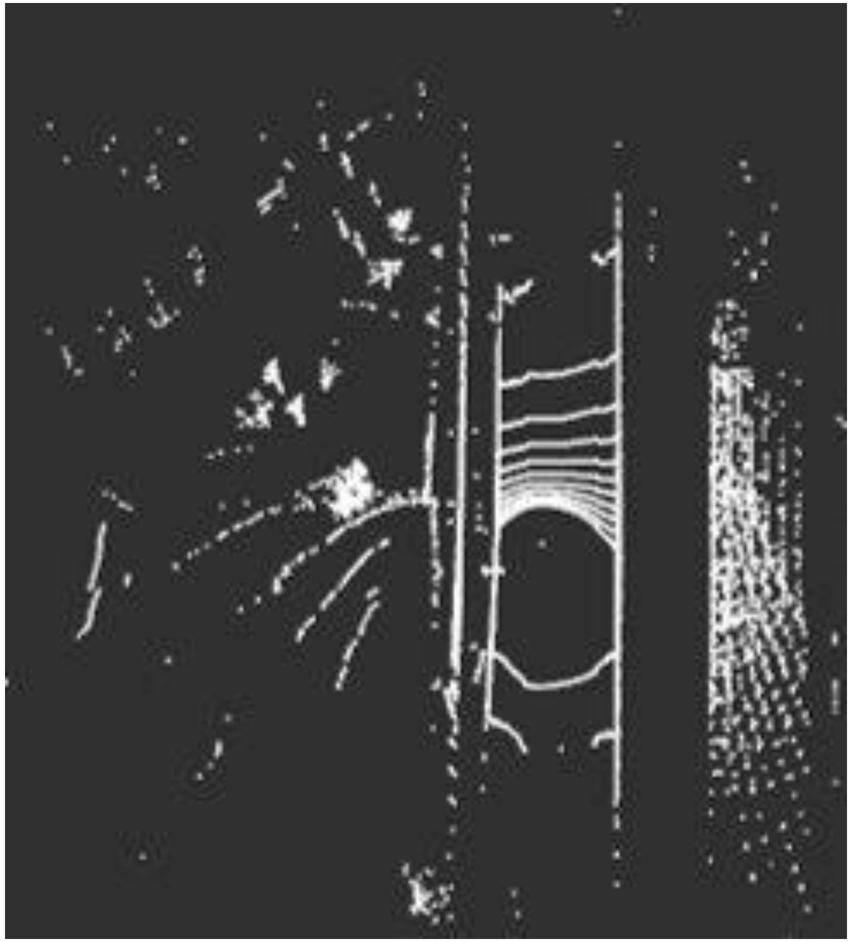


How was it done before deep learning?

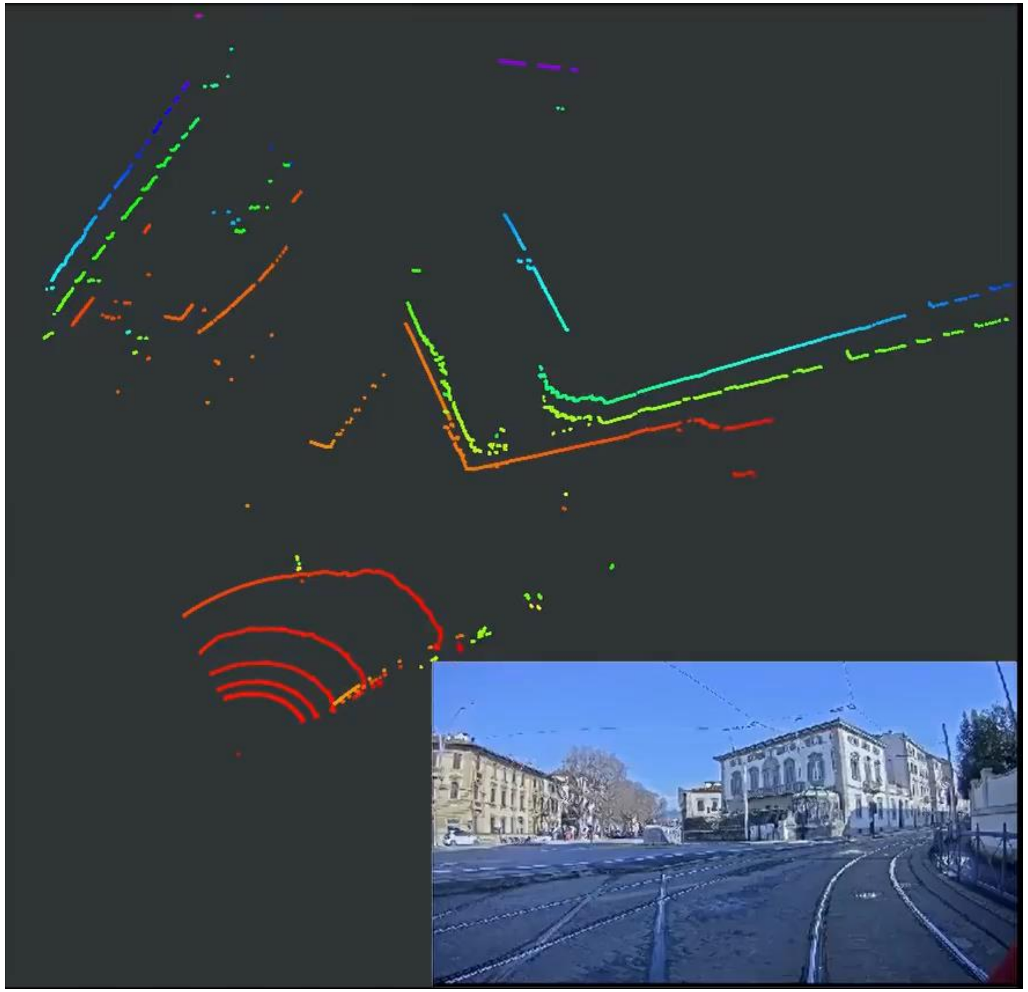
Sometimes data are not so informative

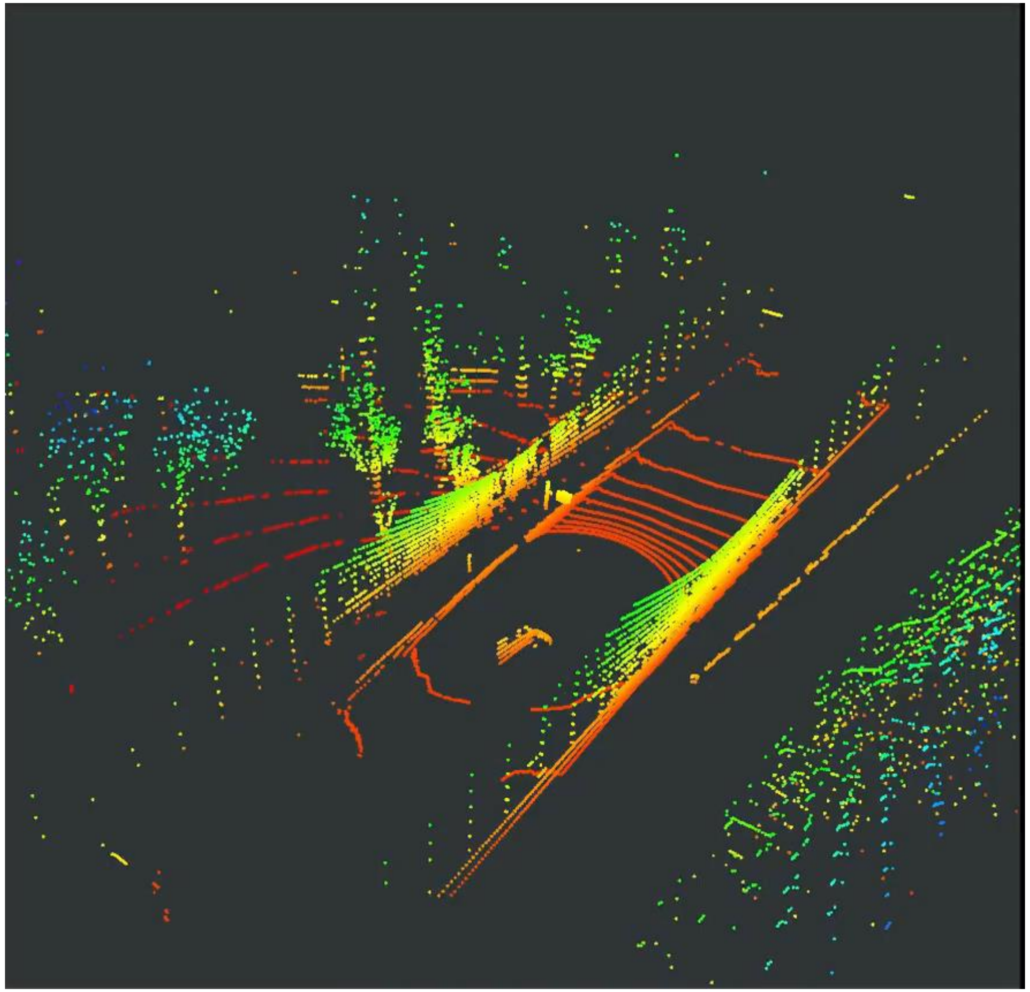
8 plane lidar 90° fov

16 plane lidar 360° fov

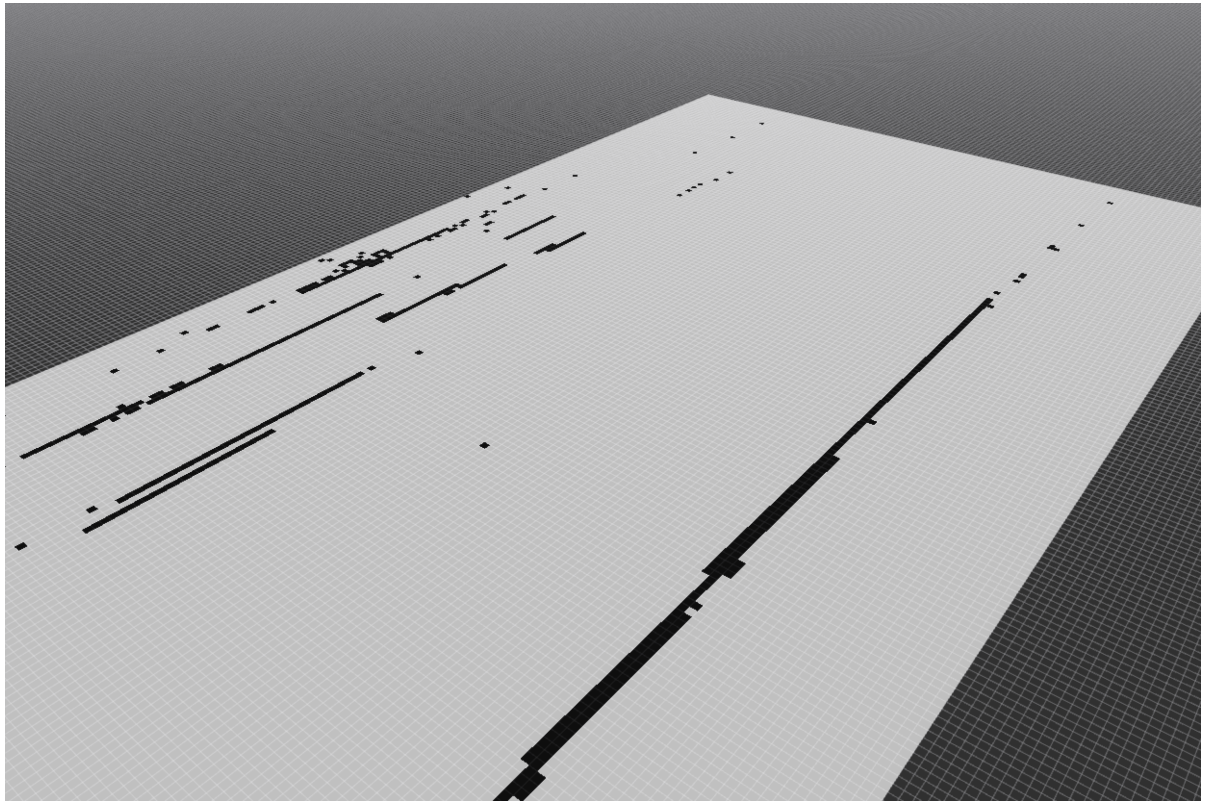


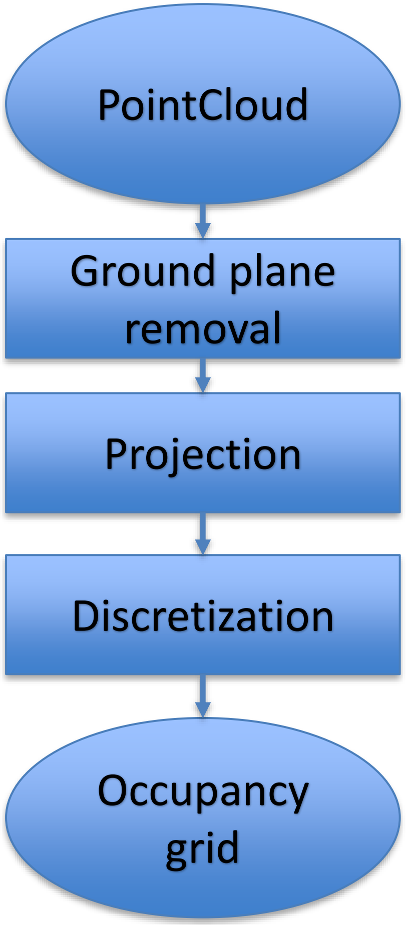
Sometimes data are not so informative



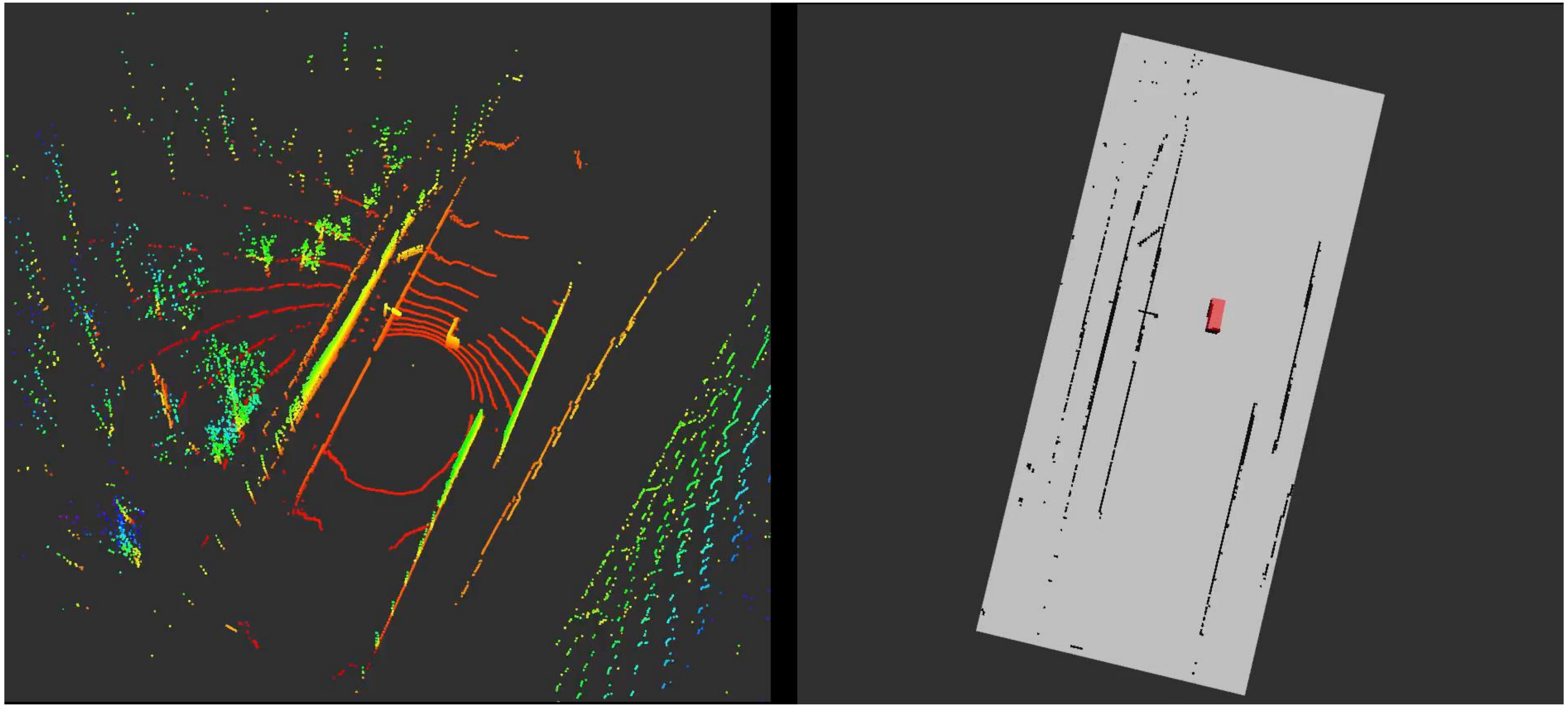


Geometric-based solution still works





Geometric-based solution still works



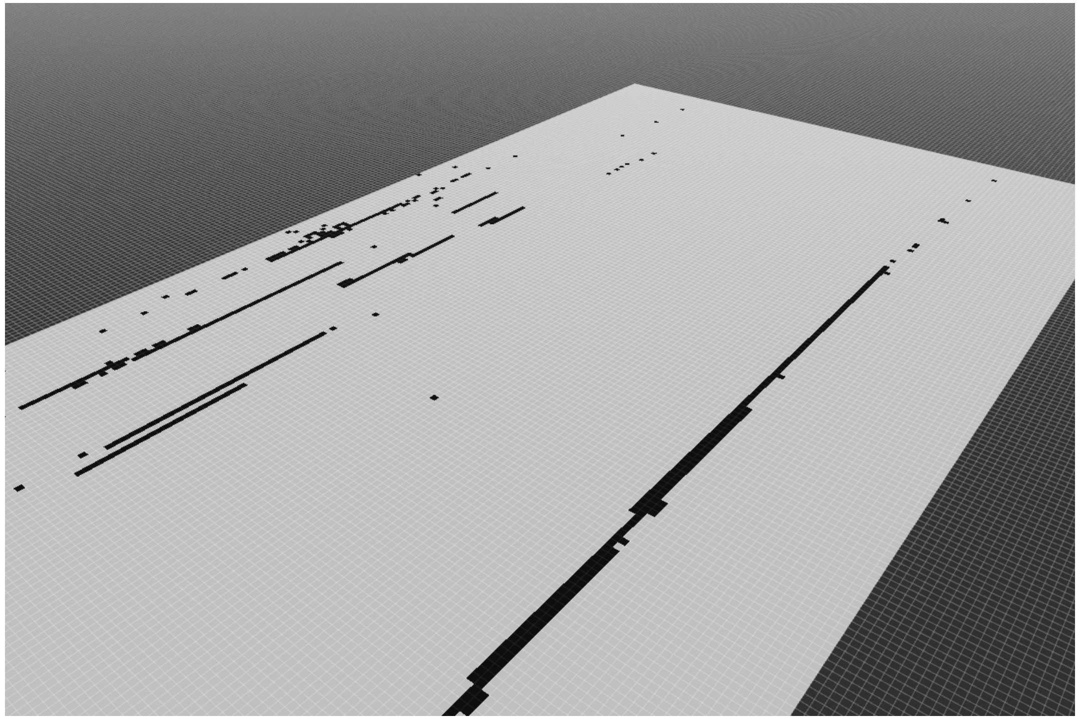
Find clusters

Filtering based on:
- *position*
- *Size*

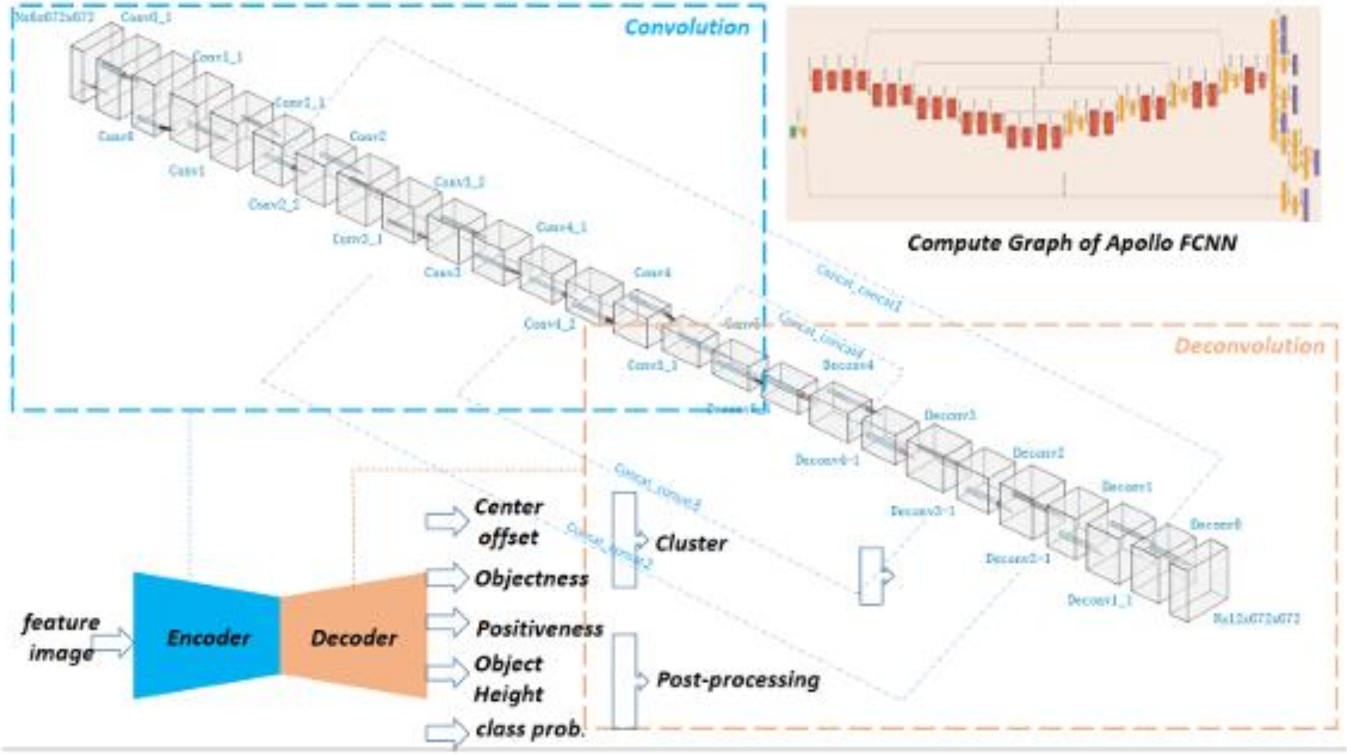
Retrieve a list of obstacle:
- *(x,y) position*
- *(l,w,h) size*
- *Difficult to provide class*

Occupancy grid are 2D images

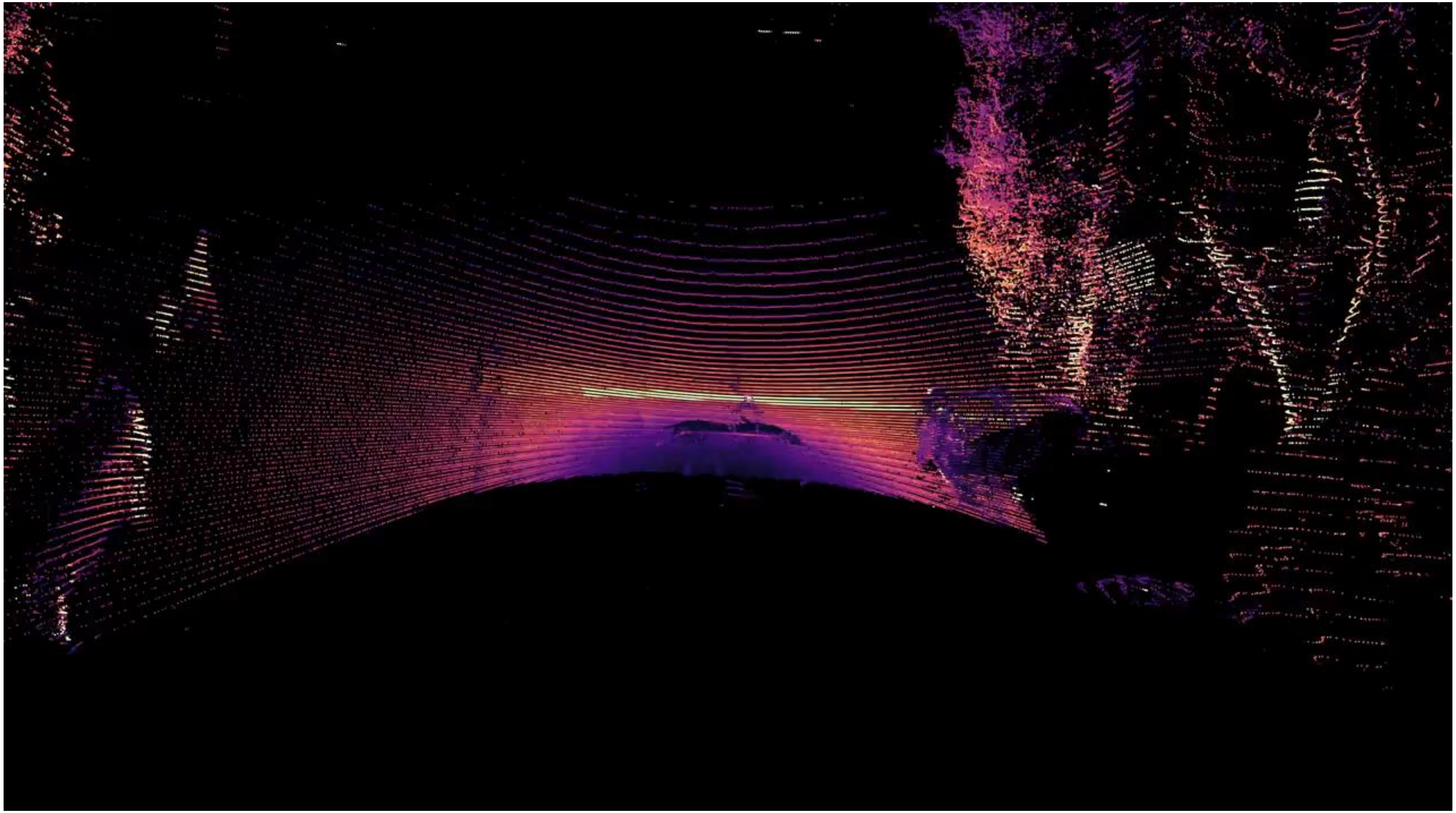
you can use deep learning…



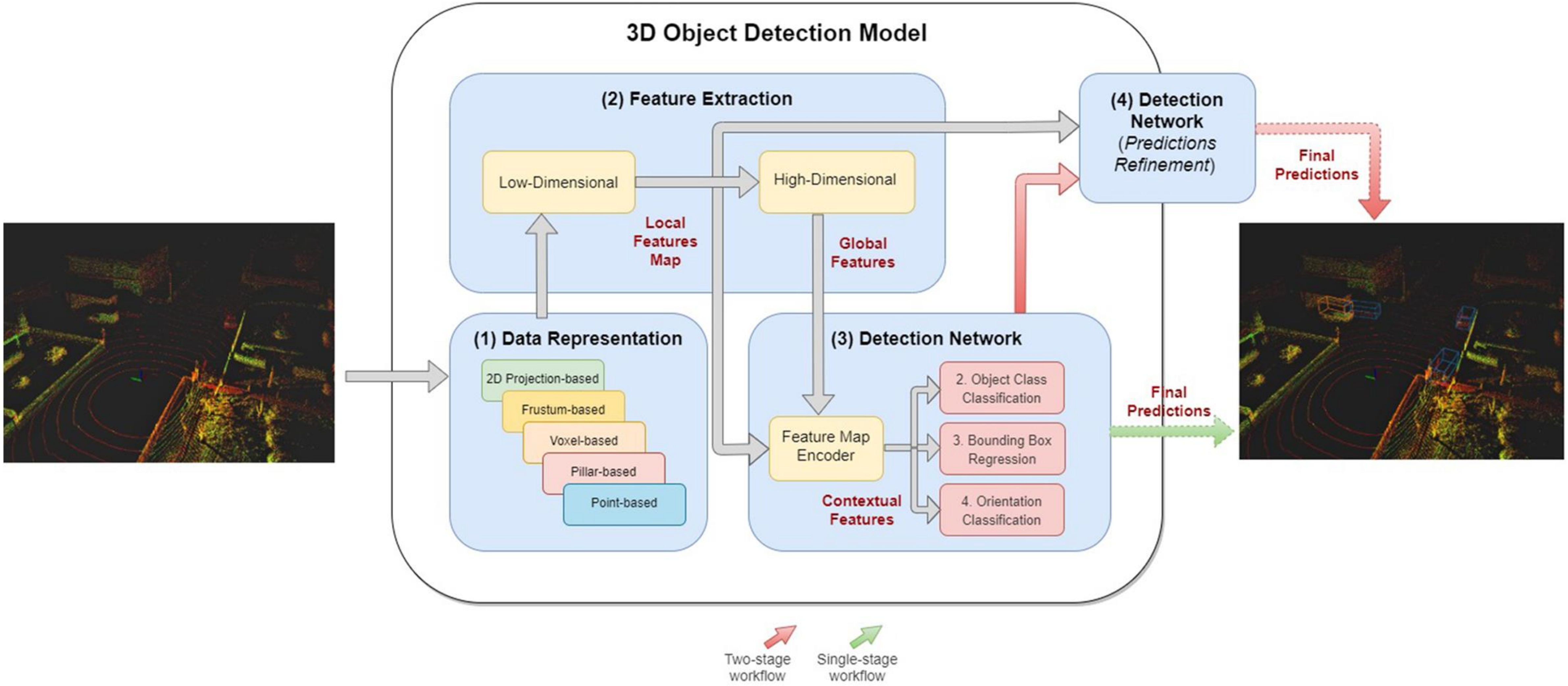
Apollo FCNN-based Model (2D grid-based detector)



What about high resolution lidars and deep learning?



3D data processing pipeline

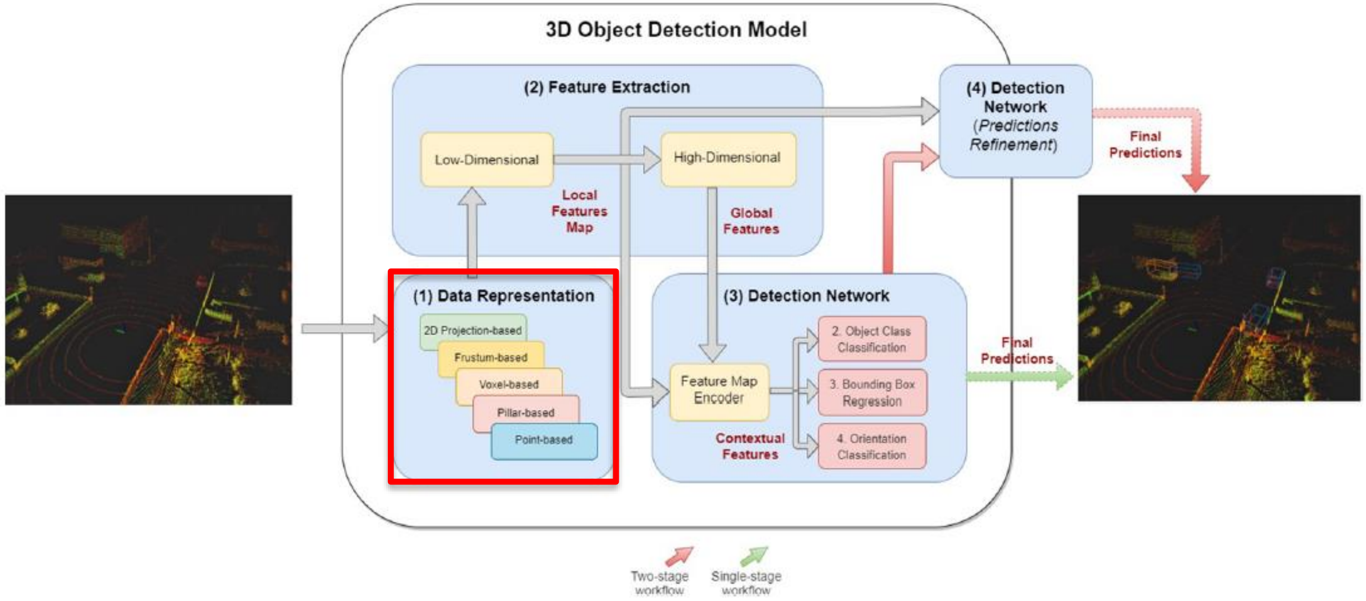


Fernandes, D., Silva, A., Névoa, R., Simões, C., Gonzalez, D., Guevara, M., ... & Melo-Pinto, P. (2021). Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, *68*, 161-191.

3D data processing pipeline

1) Data Representation:

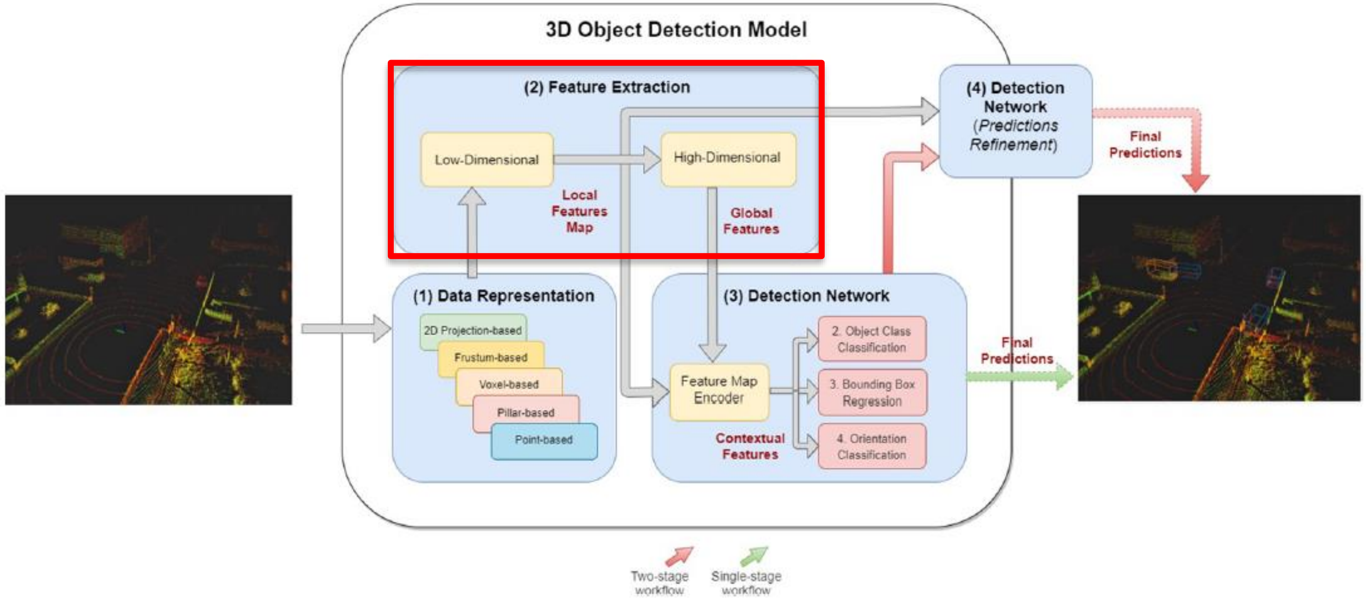
- Voxels
- Frustums
- Pillars
- 2D projection
- Raw 3D points



3D data processing pipeline

2) Feature extraction:

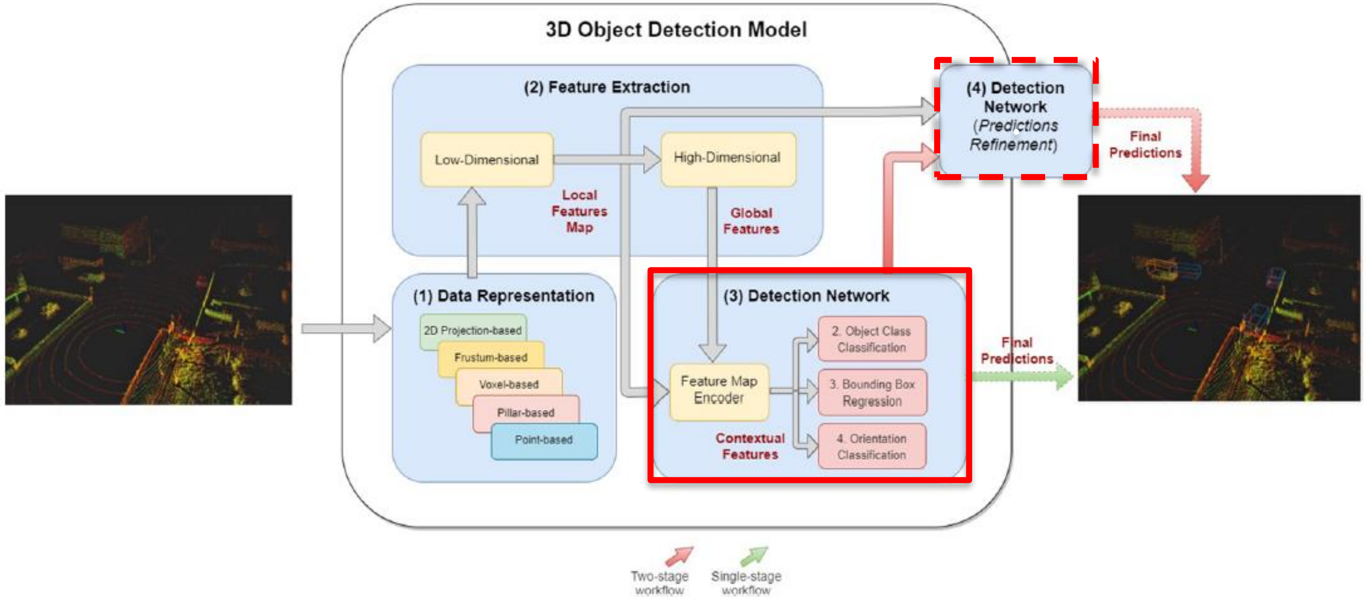
- Low-dimensional features
- High dimensional features



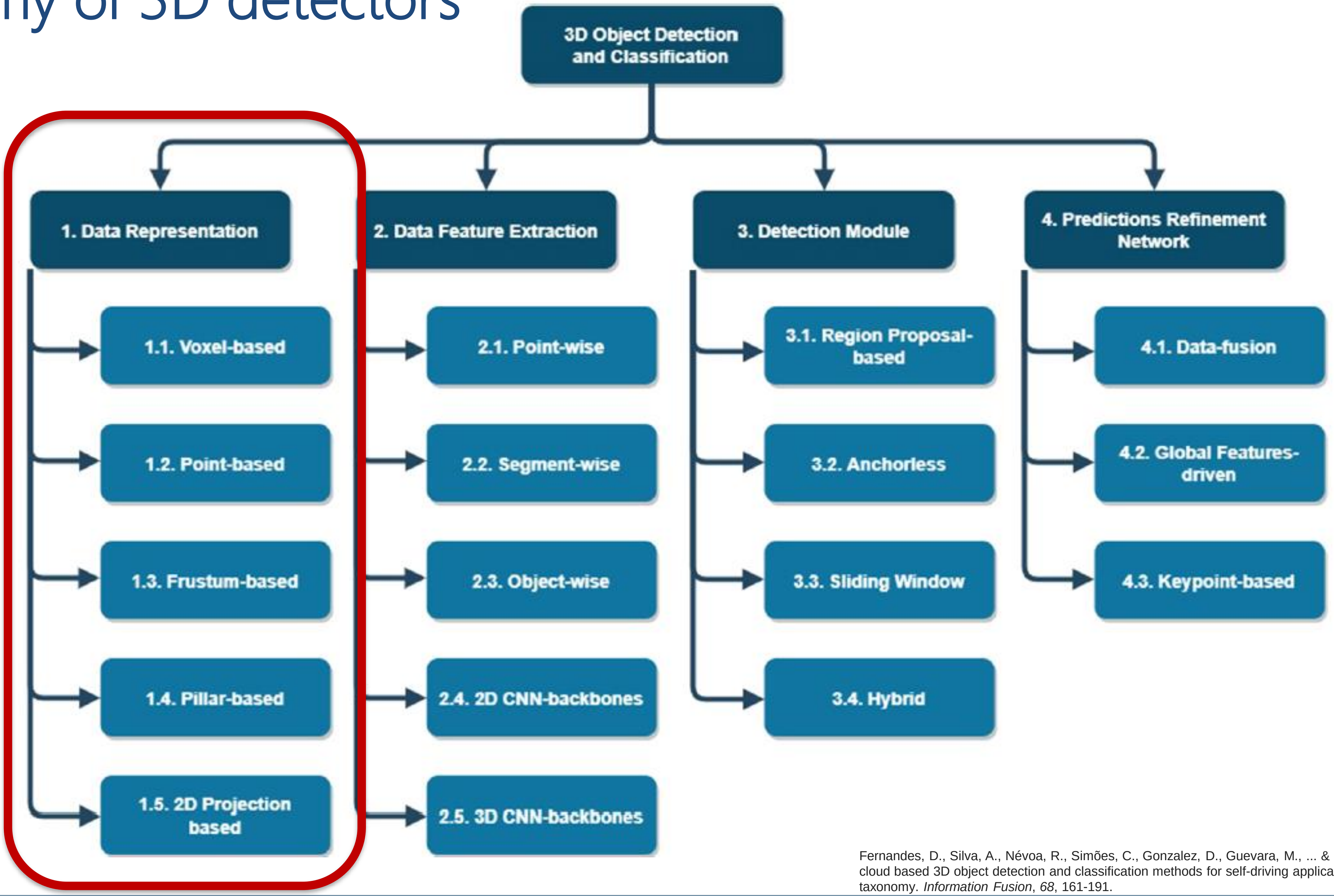
3D data processing pipeline

3) Detection Network:

- Heterogeneous architecture
- Second level feature extractor
- Two stage architectures:
  - Object proposal
  - Prediction refinement
- Produce:
  - Class
  - 3D Bounding box
  - Orientation
  - Speed



Taxonomy of 3D detectors

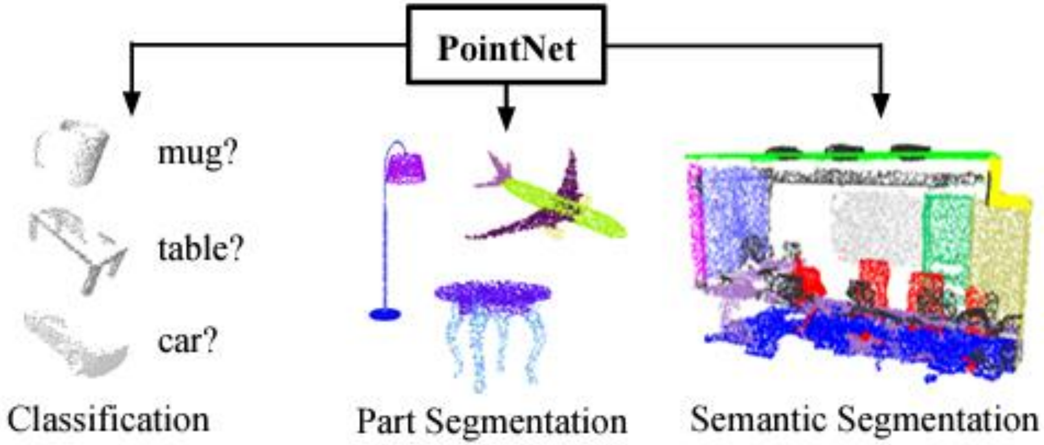


Fernandes, D., Silva, A., Névoa, R., Simões, C., Gonzalez, D., Guevara, M., ... & Melo-Pinto, P. (2021). Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, *68*, 161-191.

Data representation

Point-based:

- Works directly on the PointCloud
- Sparse representation
- Extract a feature vector for each point
- First extract low-dimensional features from each point independently
- Then aggregate these to form high-dimensional features
- Mostly based on PointNet backbone



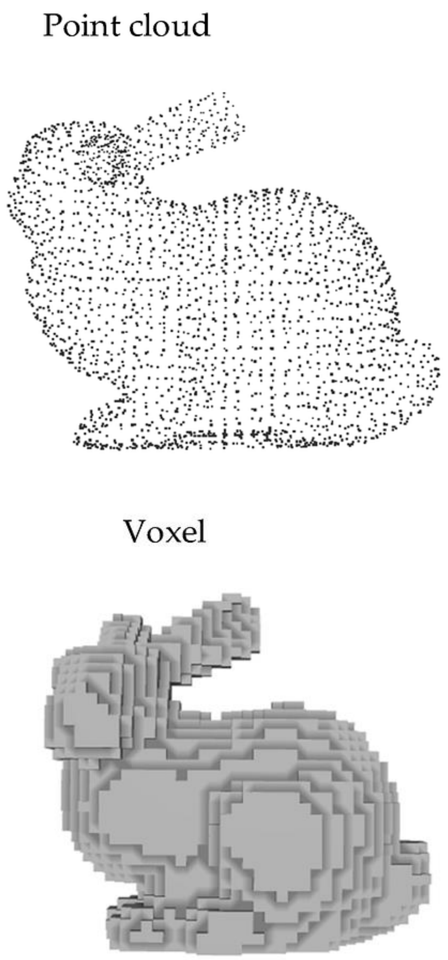
PointNet is just the starting point

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). **Pointnet**: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652-660).
Meyer, G. P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., & Wellington, C. K. (2019). **Lasernet**: An efficient probabilistic 3d object detector for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12677-12686).
Xu, D., Anguelov, D., & Jain, A. (2018). **Pointfusion**: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 244-253).

Data representation

Voxel-based

- Volumetric picture element
- PointCloud divided into equally spaced 3D voxels
- Feature extraction is applied to groups of points inside each voxel
- Reduce PointCloud dimension
- More efficient
- Less memory required



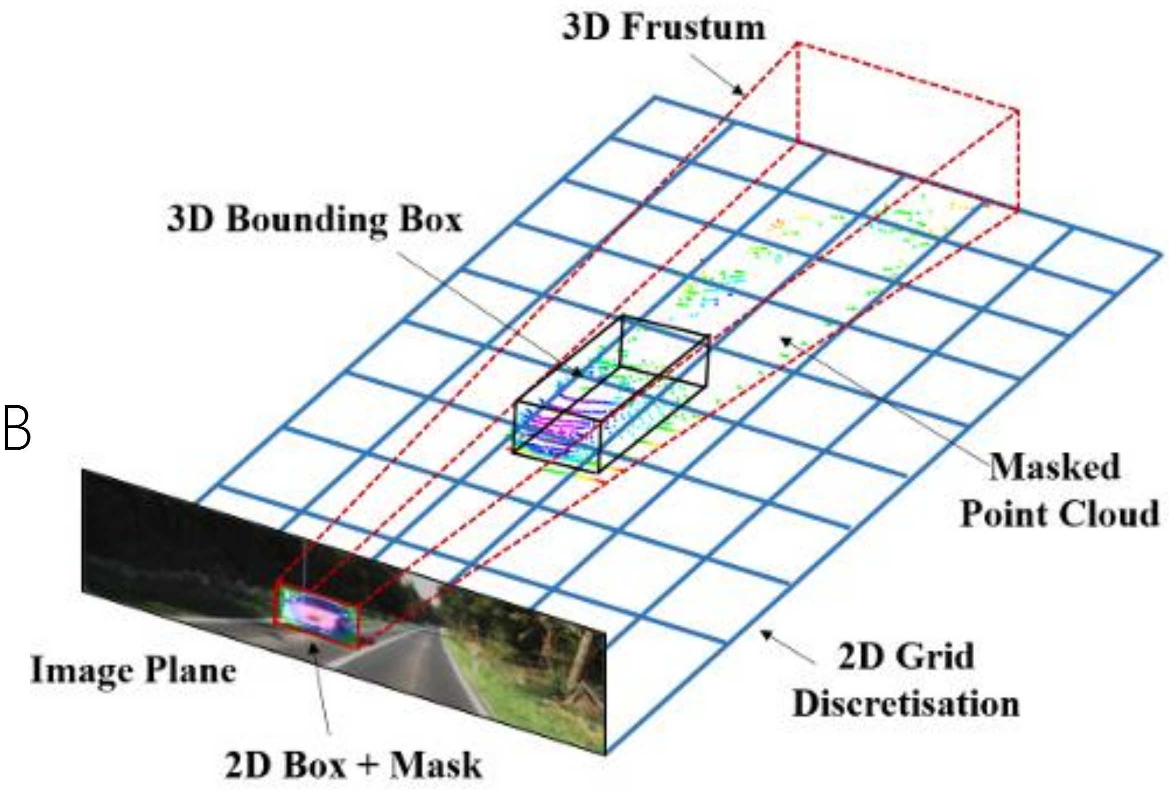
Zhou, Y., & Tuzel, O. (2018). **Voxelnet**: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4490-4499).

Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., & Li, H. (2021, May). **Voxel r-cnn**: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 2, pp. 1201-1209).

Data representation

Frustum-based

- Portion of a solid (usually a cone or pyramid) that lies between one or two parallel planes cutting it
- Crop PointCloud regions based on RGB detector
- Cropped areas are frustums

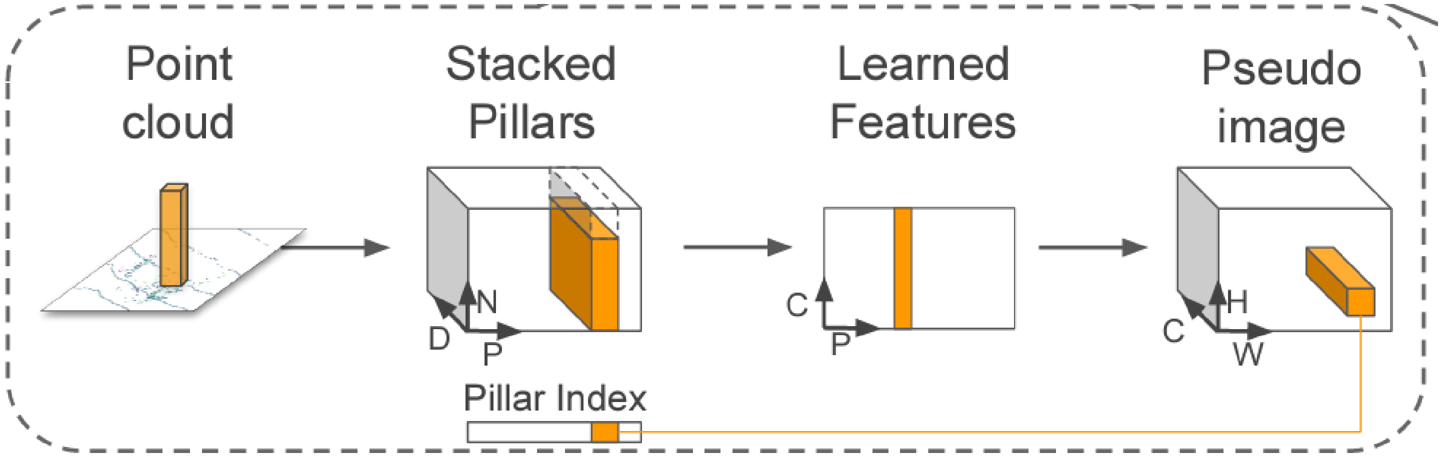


Paigwar, A., Sierra-Gonzalez, D., Erkent, Ö., & Laugier, C. (2021). **Frustum-pointpillars**: A multi-stage approach for 3d object detection using rgb camera and lidar. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2926-2933).
Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2018). **Frustum pointnets** for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 918-927).

Data representation

Pillar-based

- Data organized in vertical columns
- Leverage mounting position of LiDARS (horizontal)
- 2D discretization on the plane
- Condense Z information
- Compact representation

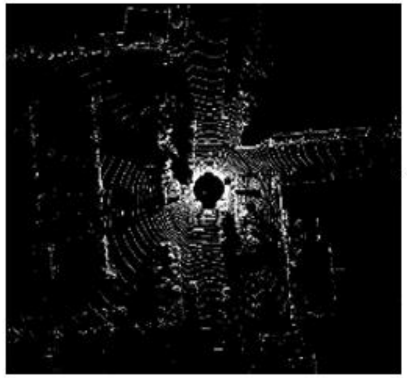


Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12697-12705).

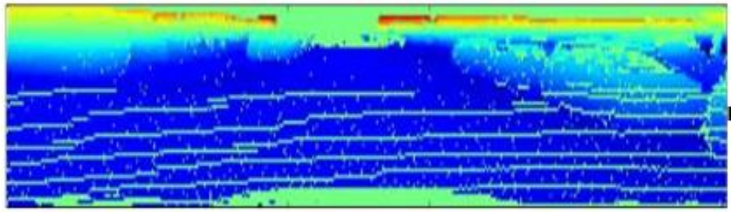
Data representation

Projection-based (previously seen)

- Different projections:
  - Bird's eye view
  - Front view
  - Range view
- Possible combination of different projections
- Compact and efficient representation
- Real-time and low power scenario
- Loss of information



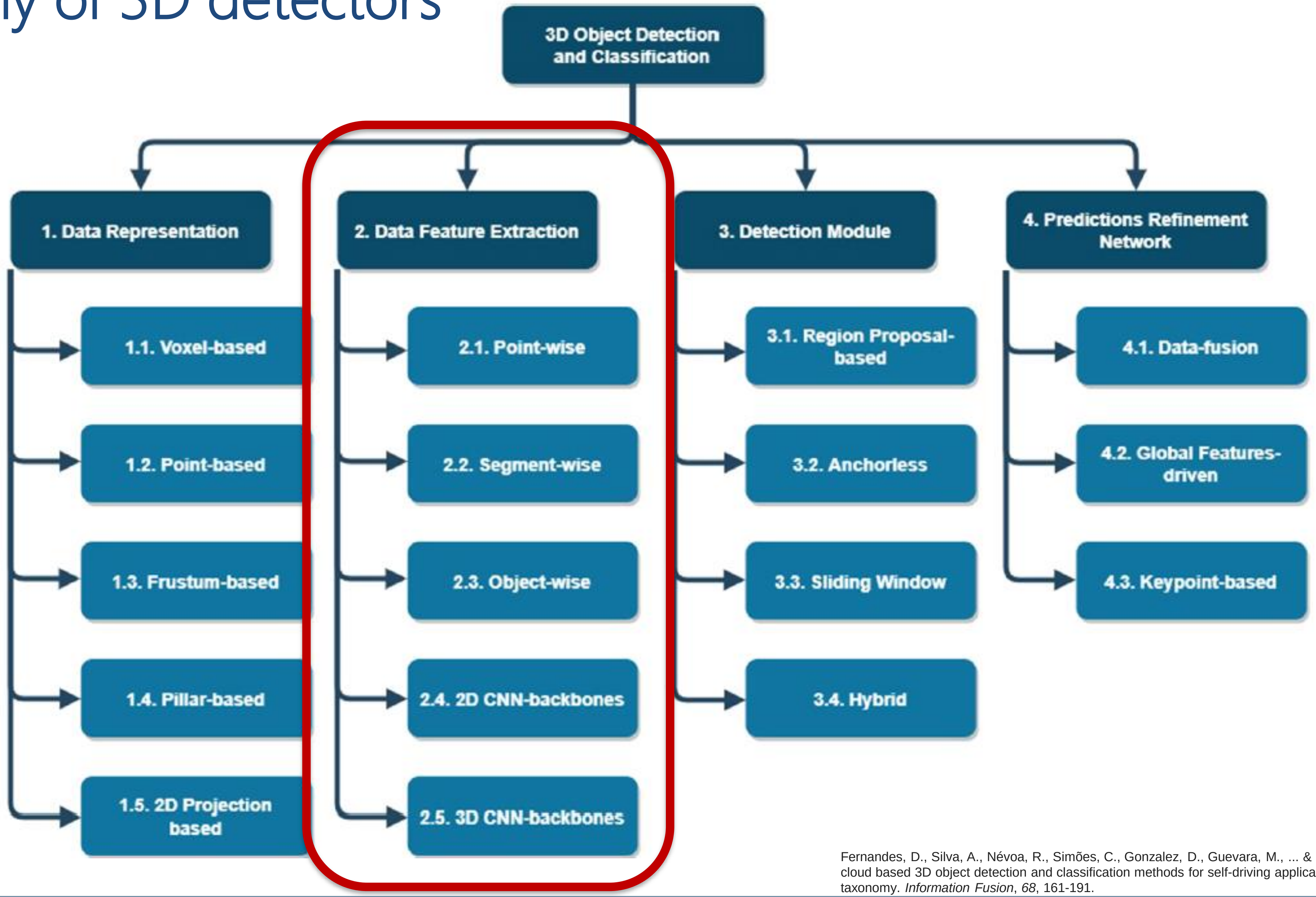
LIDAR Bird view (BV)



LIDAR Front view (FV)

Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). **Multi-view 3d object detection** network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1907-1915).
Yang, B., Luo, W., & Urtasun, R. (2018). **Pixor**: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*

Taxonomy of 3D detectors



Fernandes, D., Silva, A., Névoa, R., Simões, C., Gonzalez, D., Guevara, M., ... & Melo-Pinto, P. (2021). Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, *68*, 161-191.

Feature extraction

Local (low level) features:

- First extracted in the pipeline
- Position of points

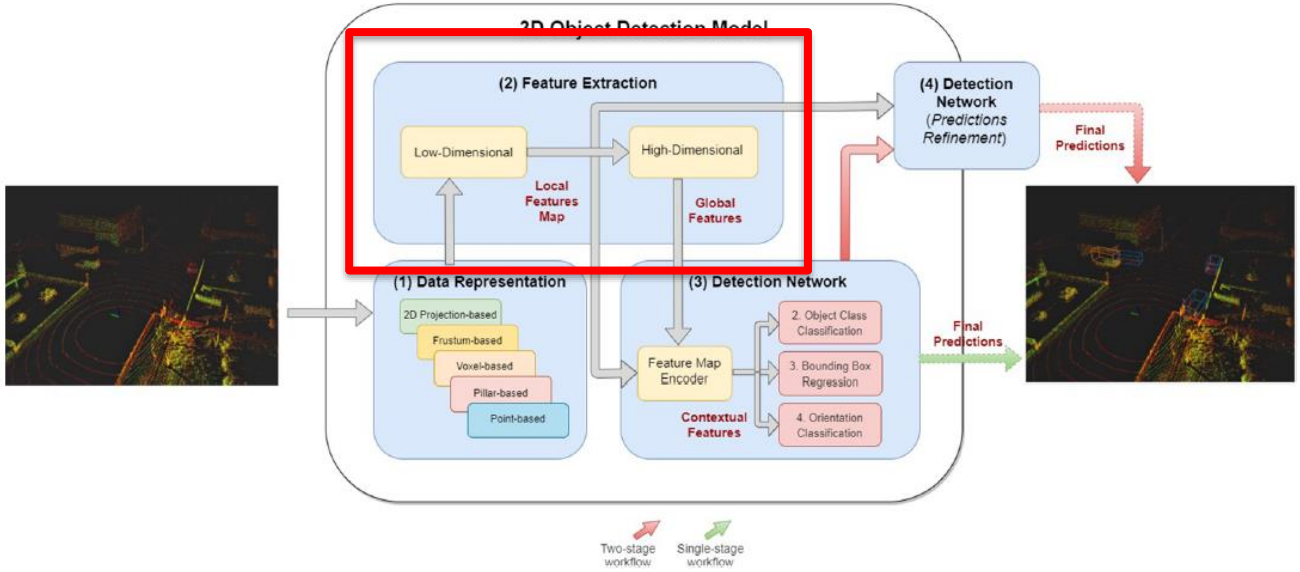
Global (high level) features:

- Geometric structure
- Relative position of points

Different feature extractor:

- Point-wise, segment-wise, object-wise, CNN-based

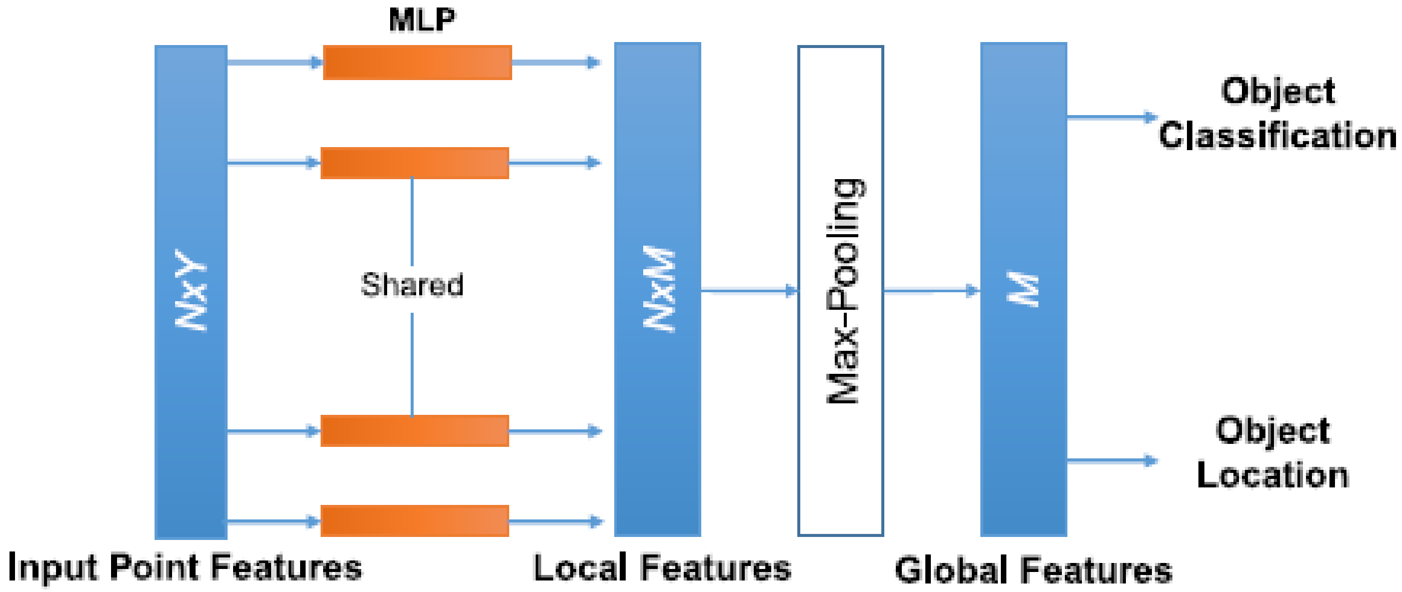
Multiple extractor can be combined in the same model (compound)



Feature extraction

Point-wise:

- Take as input the whole PointCloud
- Analyze and label each point
- PointNet, PointNet++
- N points times Y features
- Computational heavy



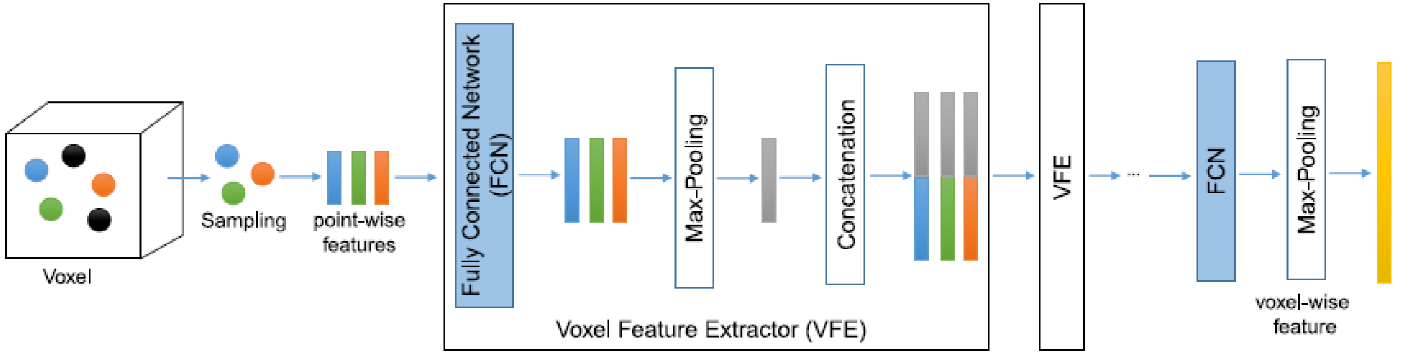
Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). **Pointnet**: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652-660).

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). **Pointnet++**: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, *30*.

Feature extraction

Segment-wise:

- Exploits voxel, pillars, frustum
- Segment the PointCloud into volumetric scale scenes
- Pointwise classification model applied to each segment
- Can work with multiple layers to improve resolution



Zhou, Y., & Tuzel, O. (2018). **Voxelnet**: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4490-4499).

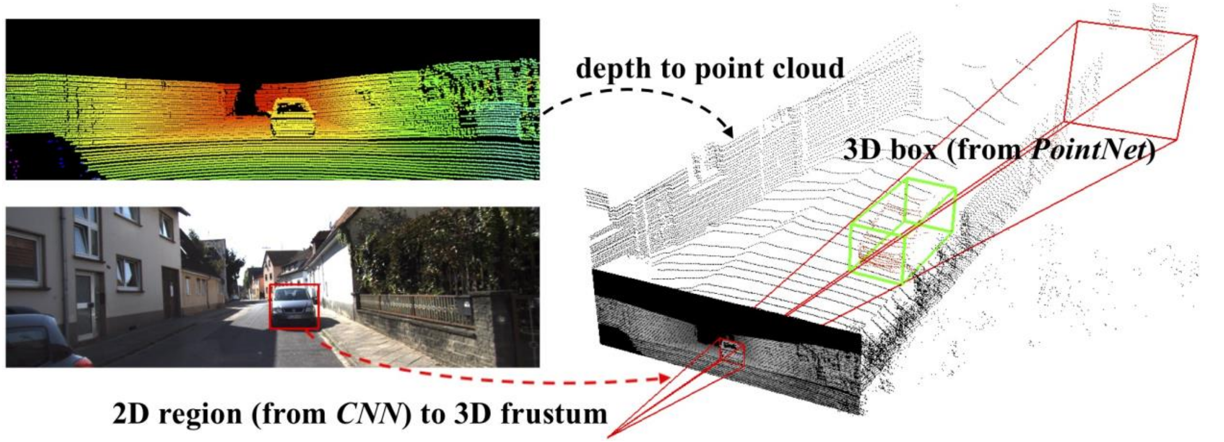
Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). **Pointpillars**: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12697-12705).

Yan, Y., Mao, Y., & Li, B. (2018). **Second**: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337.

Feature extraction

Object-wise

- Leverage a-priori information of the scene
- Combine 2D detector with 3D data
- Process only areas of the PointCloud where object are detected by other sensors
- Drastically reduce computational requirements
- Dependent on the accuracy of the input detector
- Frustum-based detector generally belong to this class

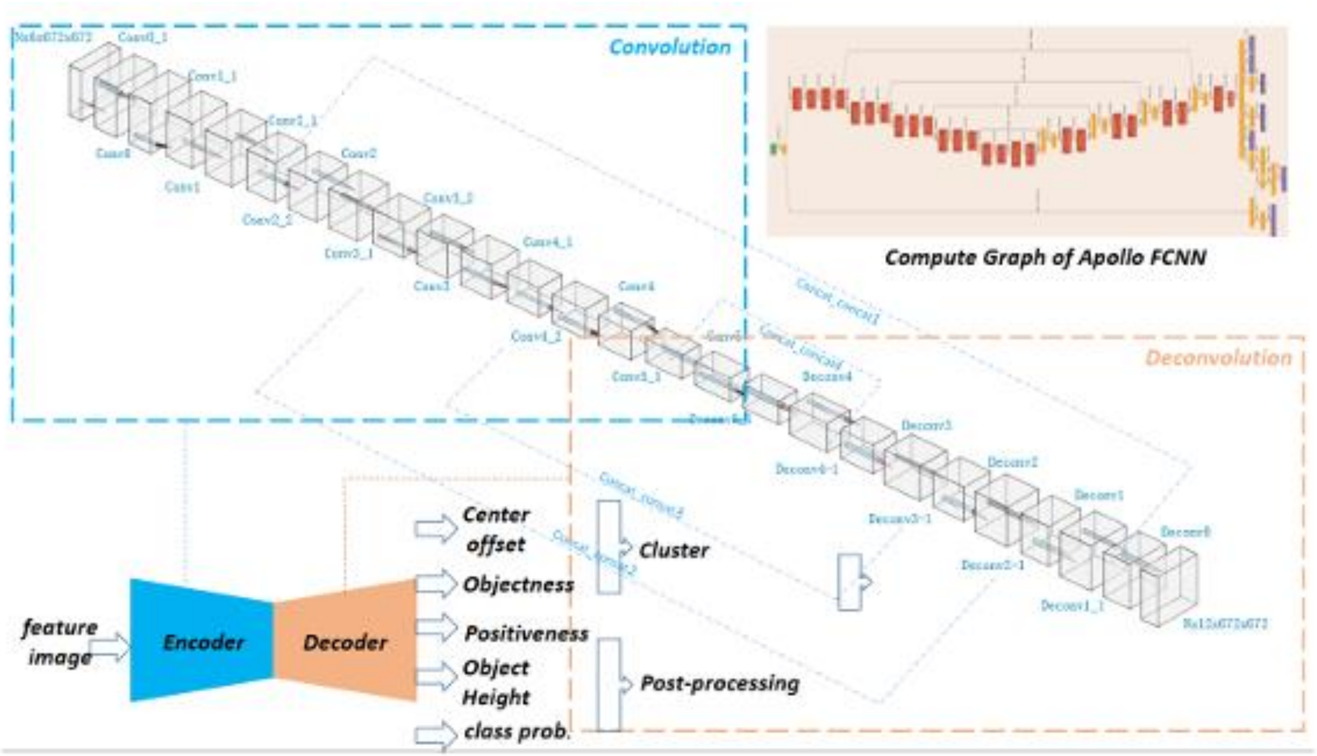


Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2018). **Frustum pointnets** for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 918-927).
Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017). **Multi-view 3d object detection** network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 1907-1915).

Feature extraction

CNN-based (2D)

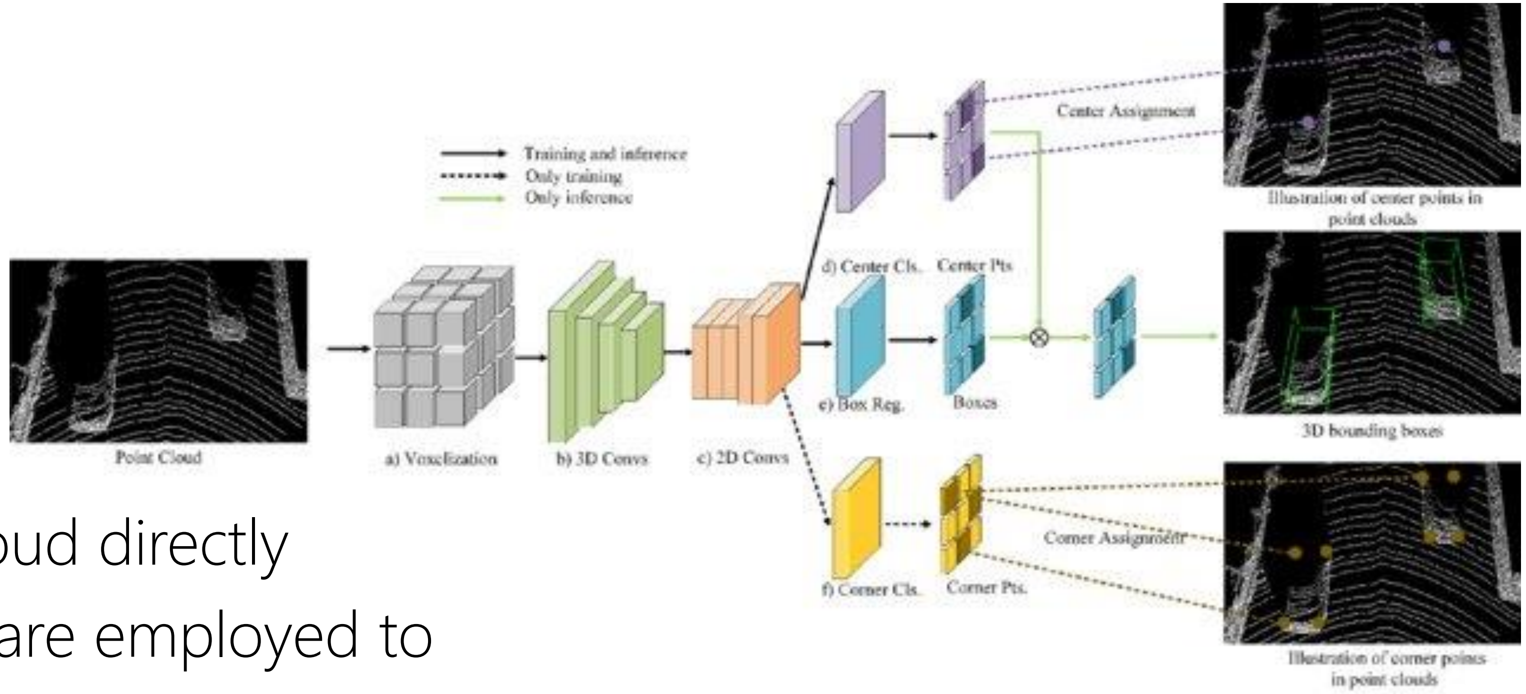
- 2D backbone from image processing
- Exploit projection-based data representation
- Treat the PointCloud as image
- Efficient and lightweight
- Loss of information
- RCNN/Yolo-based approaches



Feature extraction

CNN-based (3D)

- 3D backbone
- Sparse data
- Can’t use 3D convolution on PointCloud directly
- Sparse representations are employed to maintain efficiency
- Sparse Convolution, Submanifold Sparse Convolution

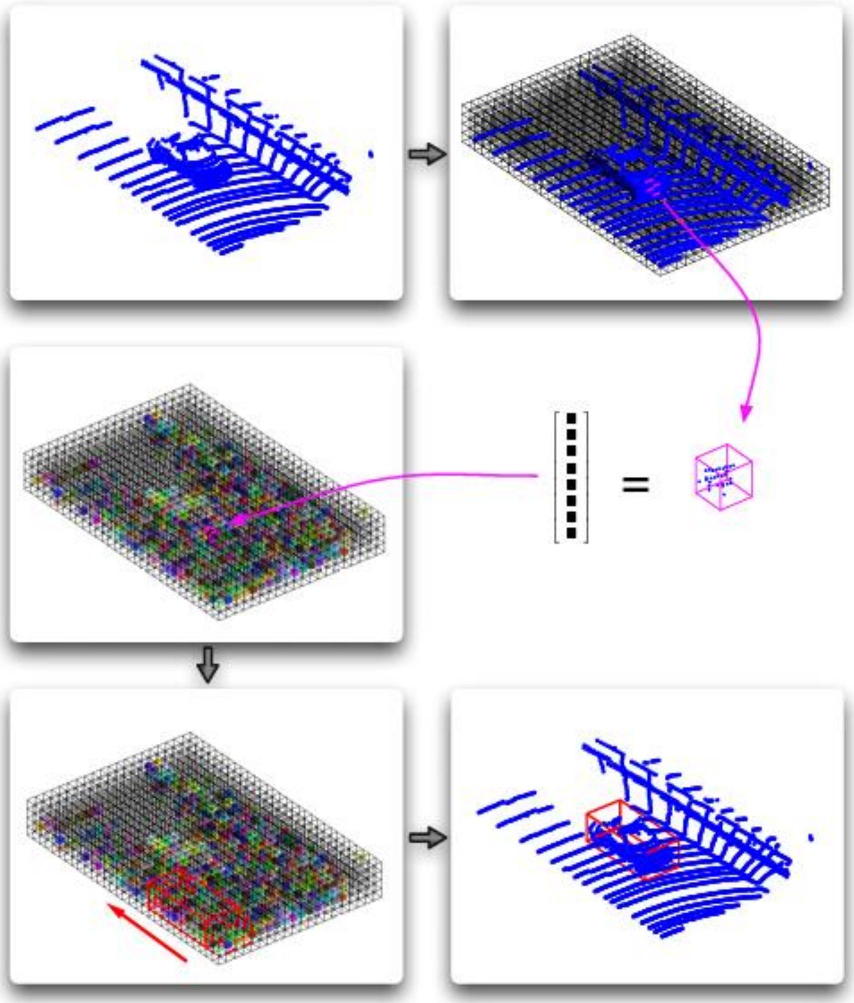


Wang, G., Tian, B., Ai, Y., Xu, T., Chen, L., & Cao, D. (2020). **Centernet3d**: An anchor free object detector for autonomous driving. *arXiv preprint arXiv:2007.07214*.
Yan, Y., Mao, Y., & Li, B. (2018). **Second**: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337.

Feature extraction

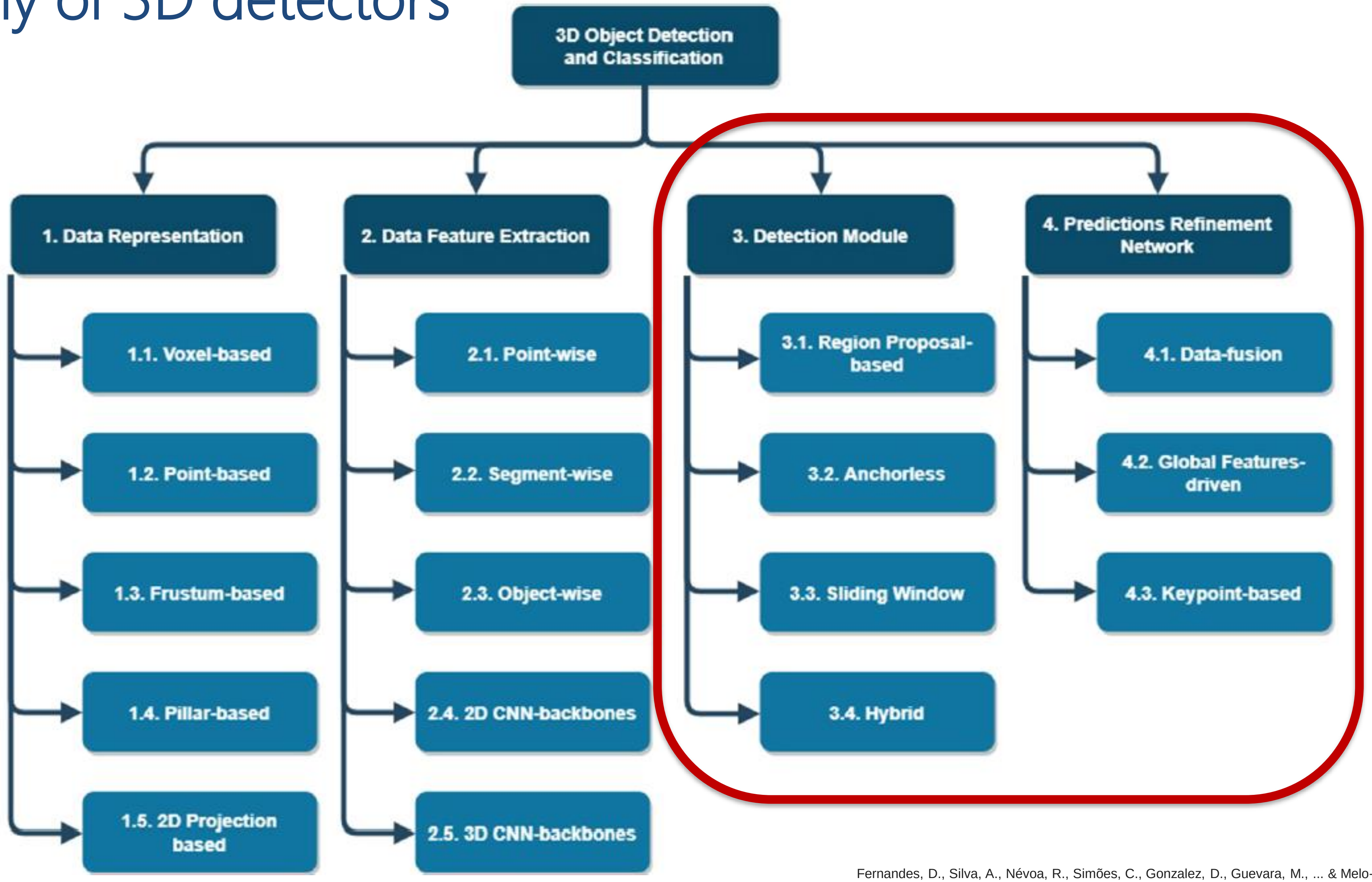
CNN-based (Voting scheme)

- Solve the problem of 3D convolution
- 3D grid discretization
- Feature vector built from 3D grid
- Cells in empty space are not stored
- Only non-zero vectors cast a vote
- Sparse voting is mathematically equivalent to a convolution on a sparse grid



Wang, D. Z., & Posner, I. (2015, July). Voting for voting in online point cloud object detection. In *Robotics: science and systems* (Vol. 1, No. 3, pp. 10-15).
Che, E., Jung, J., & Olsen, M. J. (2019). Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors*, *19*(4), 810.

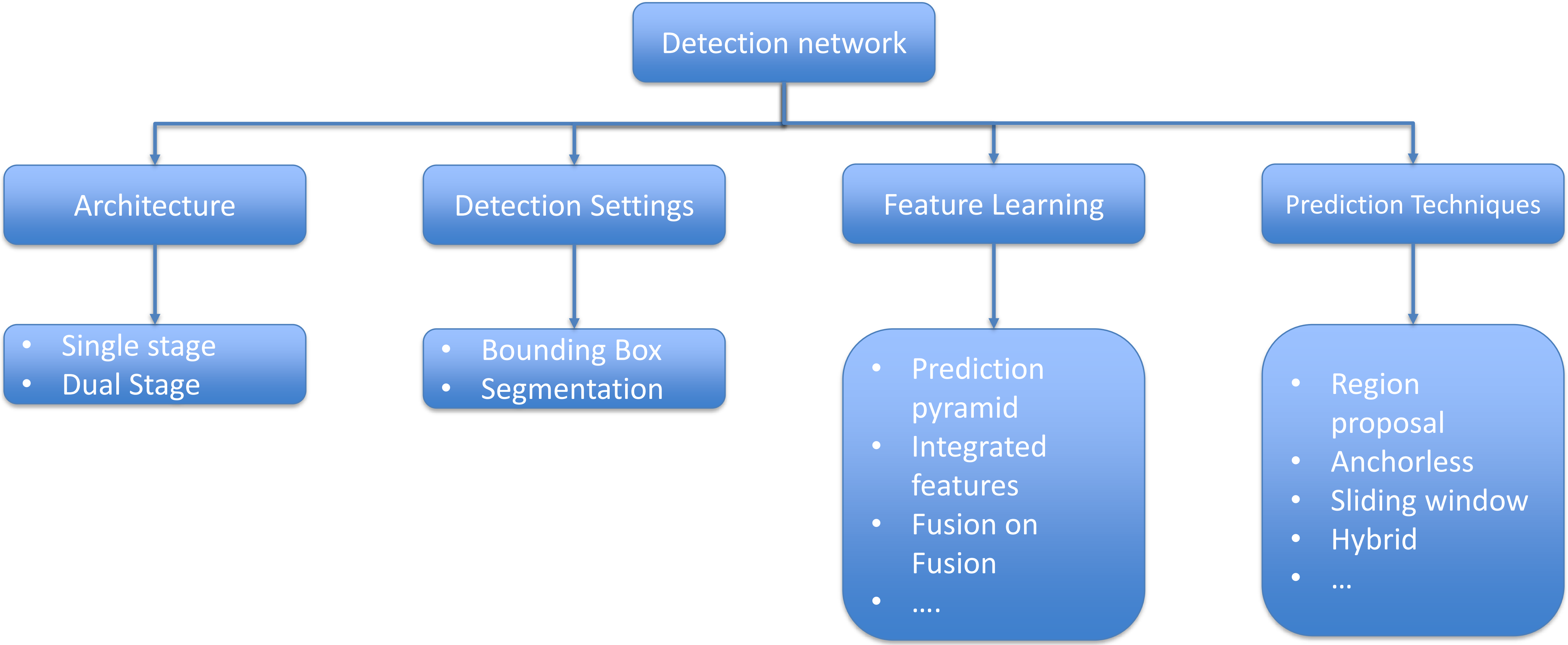
Taxonomy of 3D detectors



Fernandes, D., Silva, A., Névoa, R., Simões, C., Gonzalez, D., Guevara, M., ... & Melo-Pinto, P. (2021). Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, *68*, 161-191.

Detection and prediction refinement network

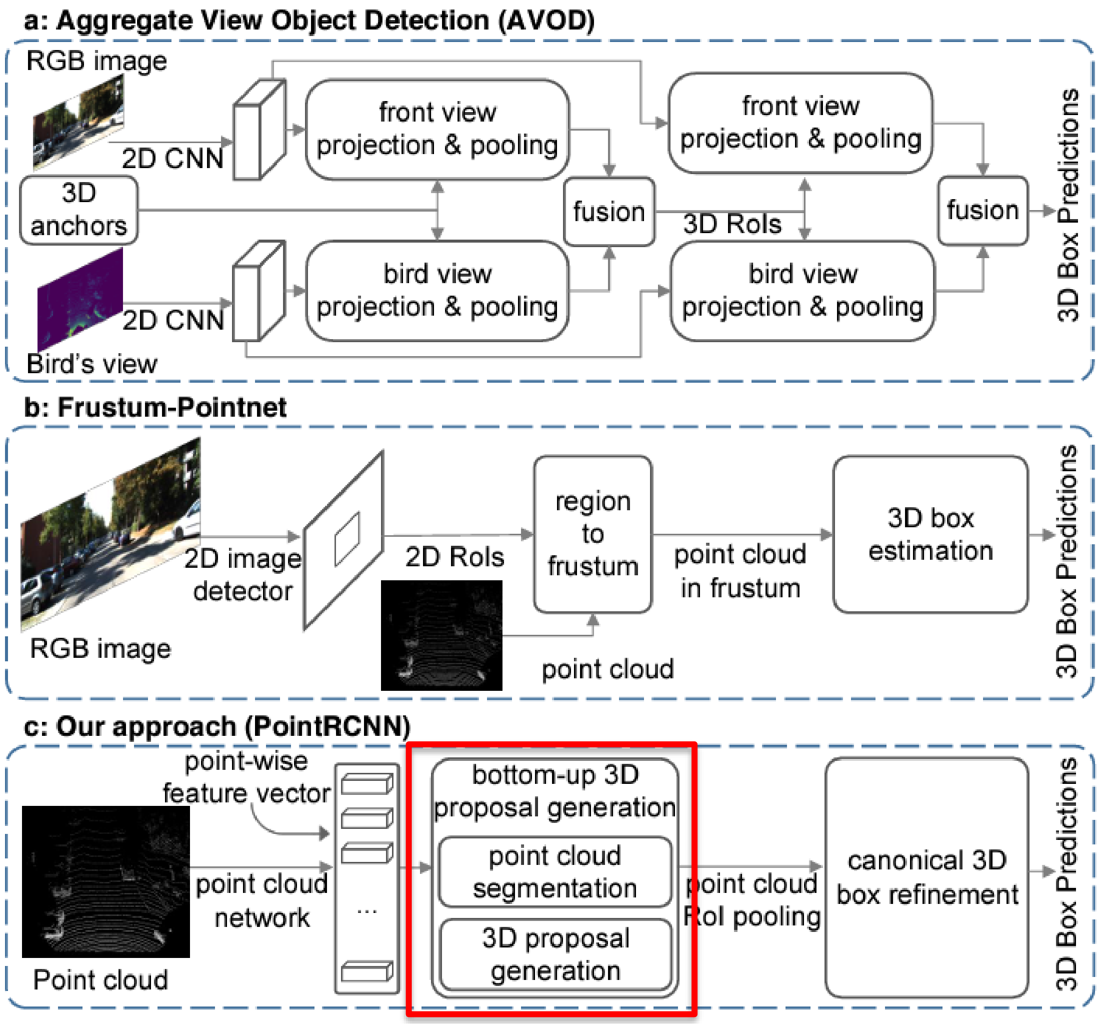
more taxonomy



Detection and prediction refinement network

Architecture:

- Similar to image:
  - Dual stage (R-CNN)
  - Single stage (SDD)
- Heads are still required to refine the region proposal output
- Single stage used in real time applications thanks to efficiency



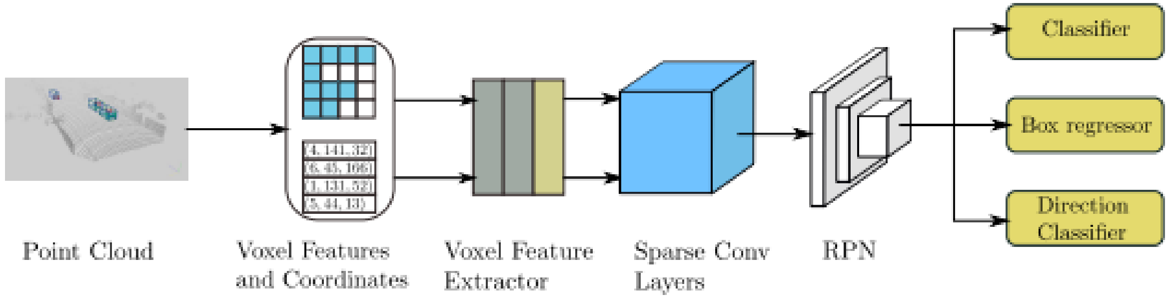
Shi, S., Wang, X., & Li, H. (2019). Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 770-779).

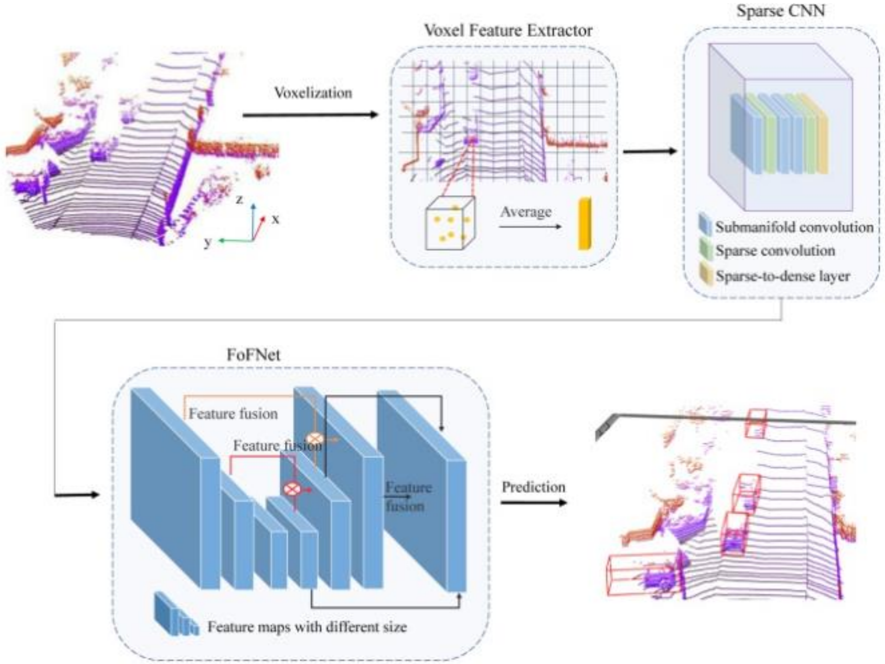
Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, *396*, 39-64.

Detection and prediction refinement network

Detector settings:

- Like for images:
  - <u>Cuboid</u>
  - Segmentation mask
- Cuboid based retrieve 3D bounding boxes
- Are the most common approach
- Most dataset provide ground truth as bounding boxes



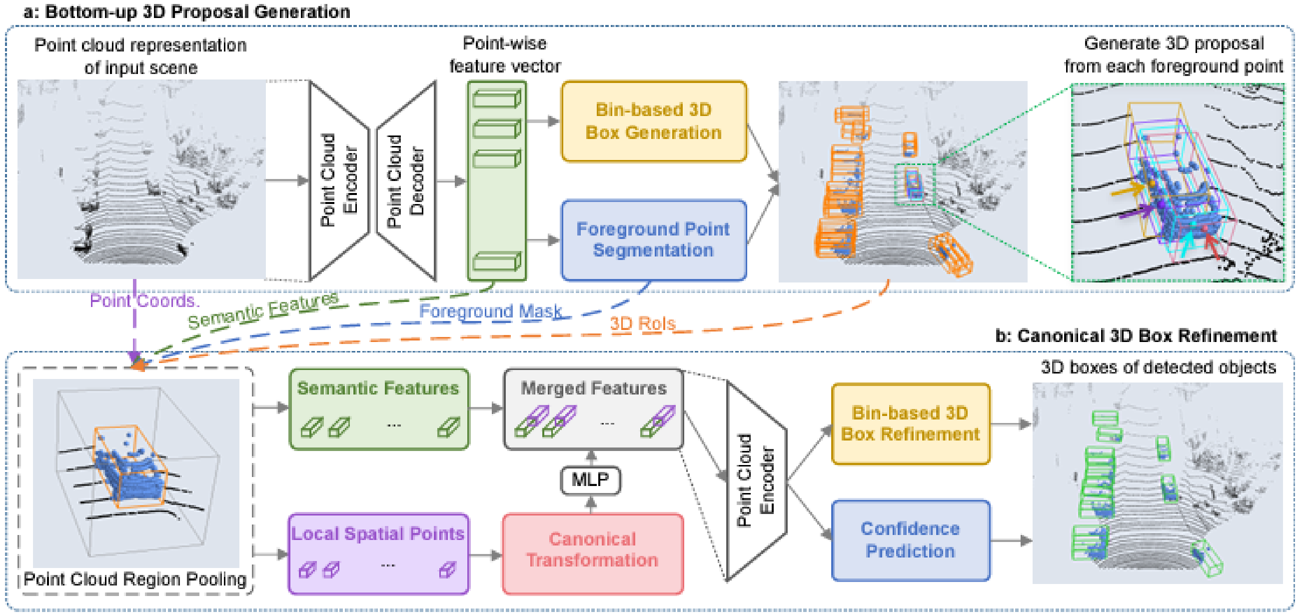


Yan, Y., Mao, Y., & Li, B. (2018). **Second**: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337.
Zhou, Y., & Tuzel, O. (2018). **Voxelnet**: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4490-4499).
Wang, L., Fan, X., Chen, J., Cheng, J., Tan, J., & Ma, X. (2020). 3D object detection based on sparse convolution neural network and feature fusion for autonomous driving in smart cities. *Sustainable Cities and Society*, *54*, 102002.

Detection and prediction refinement network

Detector settings:

- Like for images:
  - Cuboid
  - <u>Segmentation mask</u>
- Pixel-wise mask
- Foreground/background points
- Employ point-based feature extractors (e.g., PointNet++)
- Specific tasks, e.g.,road segmentation

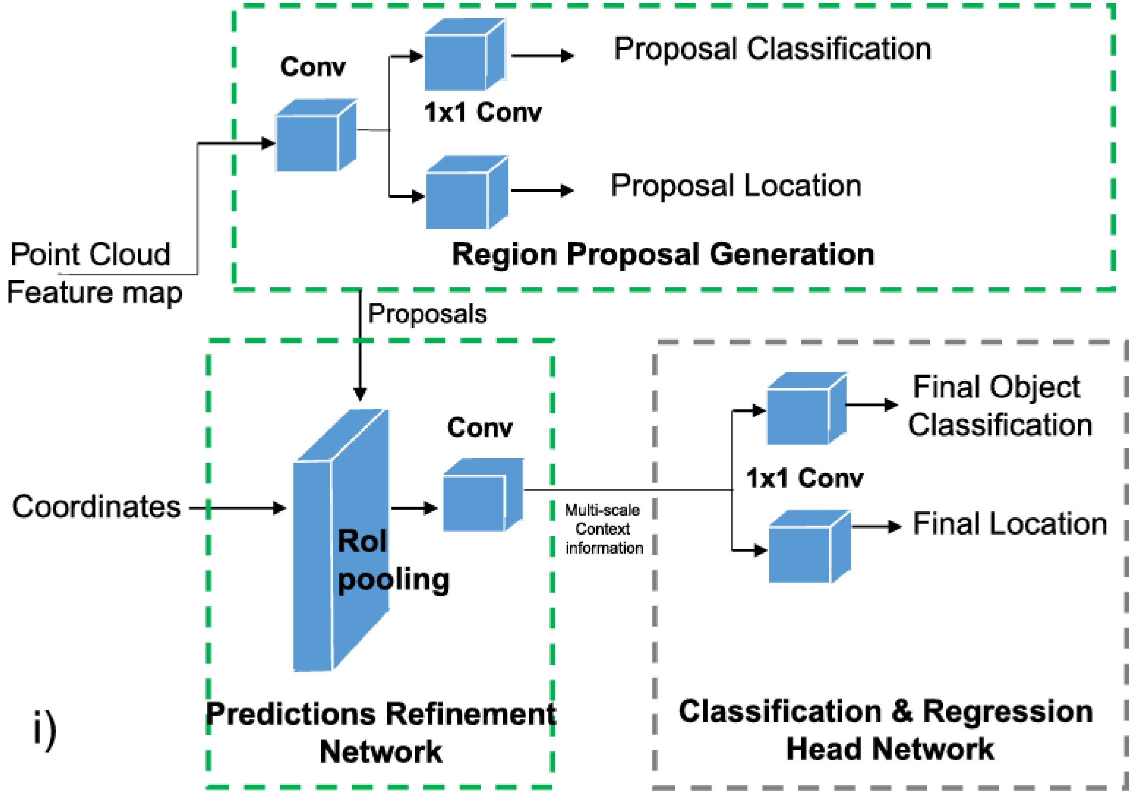


Shi, S., Wang, X., & Li, H. (2019). **Pointrcnn**: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 770-779).
Zarzar, J., Giancola, S., & Ghanem, B. (2019). **PointRGCN**: Graph convolution networks for 3D vehicles detection refinement. *arXiv preprint arXiv:1911.12236*.

Detection and prediction refinement network

Prediction techniques:

- Region proposal-based:
  - Handle multiple scales
  - Same size filters
  - Translation invariant
  - Low number of anchors
  - Efficient
  - Requires as input sparse 4D tensor



Li, B. (2017, September). 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1513-1518). IEEE.

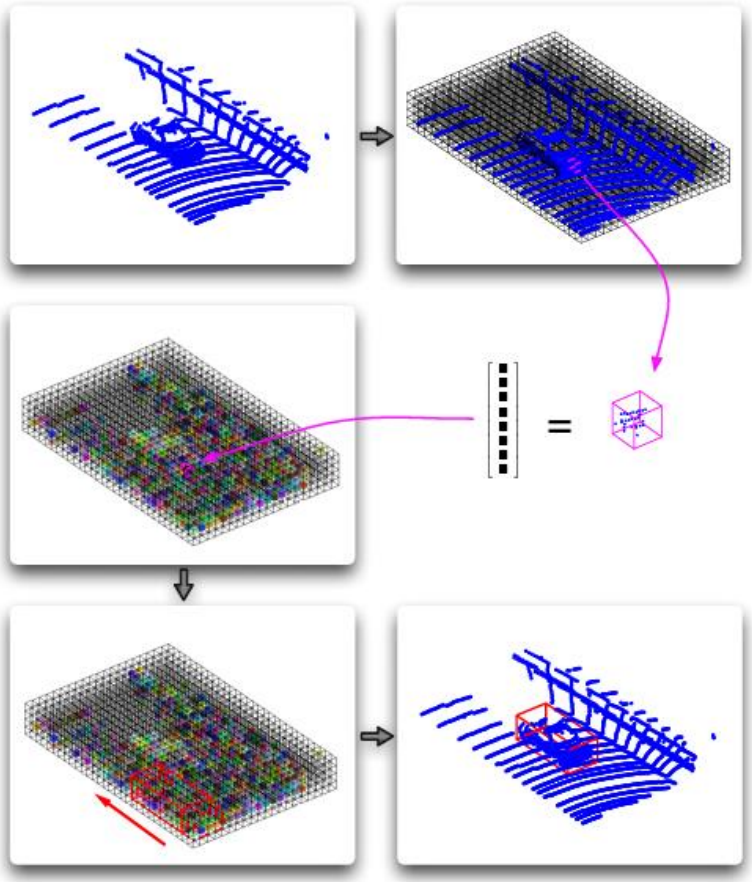
Li, B. (2017, September). 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1513-1518). IEEE.

Yan, Y., Mao, Y., & Li, B. (2018). Second: Sparsely embedded convolutional detection. *Sensors*, *18*(10), 3337.

Detection and prediction refinement network

Prediction techniques:

- Sliding window-based:
  - Widely used in computer vision
  - Rarely used for PointClouds
  - Window search in 3D is very exhaustive
  - Heavy computation
  - Combined with voting techniques to reduce computation time

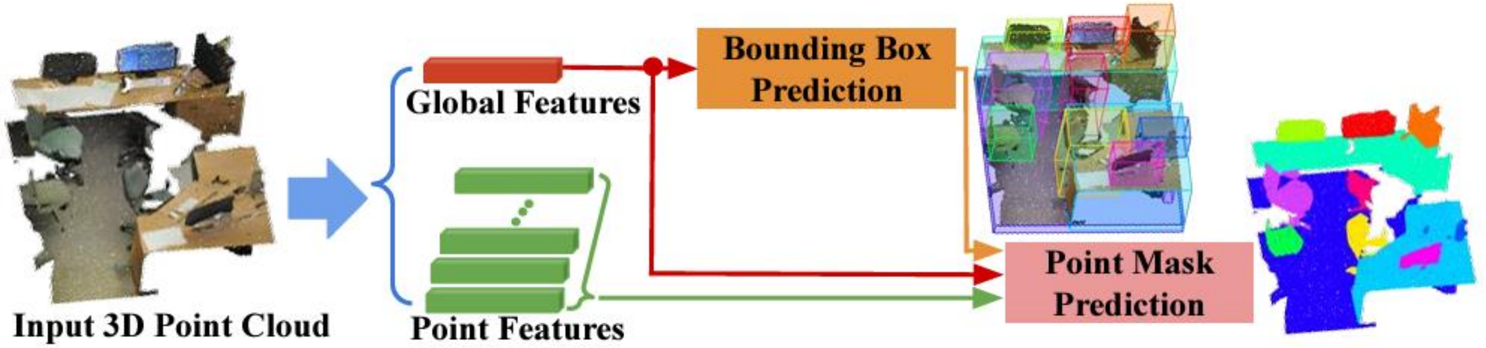


Wang, D. Z., & Posner, I. (2015, July). Voting for voting in online point cloud object detection. In *Robotics: science and systems* (Vol. 1, No. 3, pp. 10-15).
Engelcke, M., Rao, D., Wang, D. Z., Tong, C. H., & Posner, I. (2017, May). **Vote3deep**: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1355-1361). IEEE.

Detection and prediction refinement network

Prediction techniques:

- Anchorless detectors:
  - Each point contribute to the 3D reconstruction
  - Initially designed for static/indoor scenes
  - As for images can struggle with occlusions
  - Solve the issue of the large number of anchors generate by anchor-based models (100k anchors)



Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., & Trigoni, N. (2019). Learning object bounding boxes for 3D instance segmentation on point clouds. *Advances in neural information processing systems*, *32*.

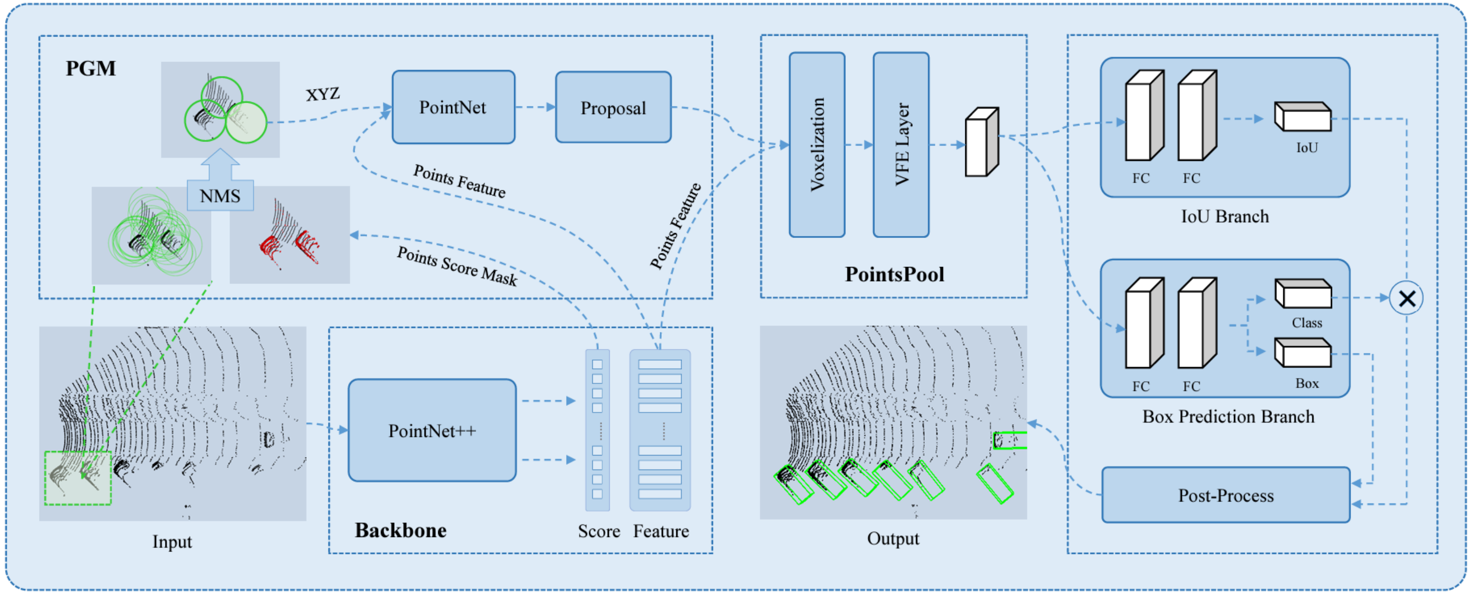
Wang, W., Yu, R., Huang, Q., & Neumann, U. (2018). **Sgpn**: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2569-2578).

Yang, B., Luo, W., & Urtasun, R. (2018). **Pixor**: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 7652-7660).

Detection and prediction refinement network

Prediction techniques:

- Hybrid detectors:
  - Rely on anchors and point masks
  - Dual stage architectures
  - First: anchor generation and filtering
  - Second: PointNet architecture for offset, orientation, score

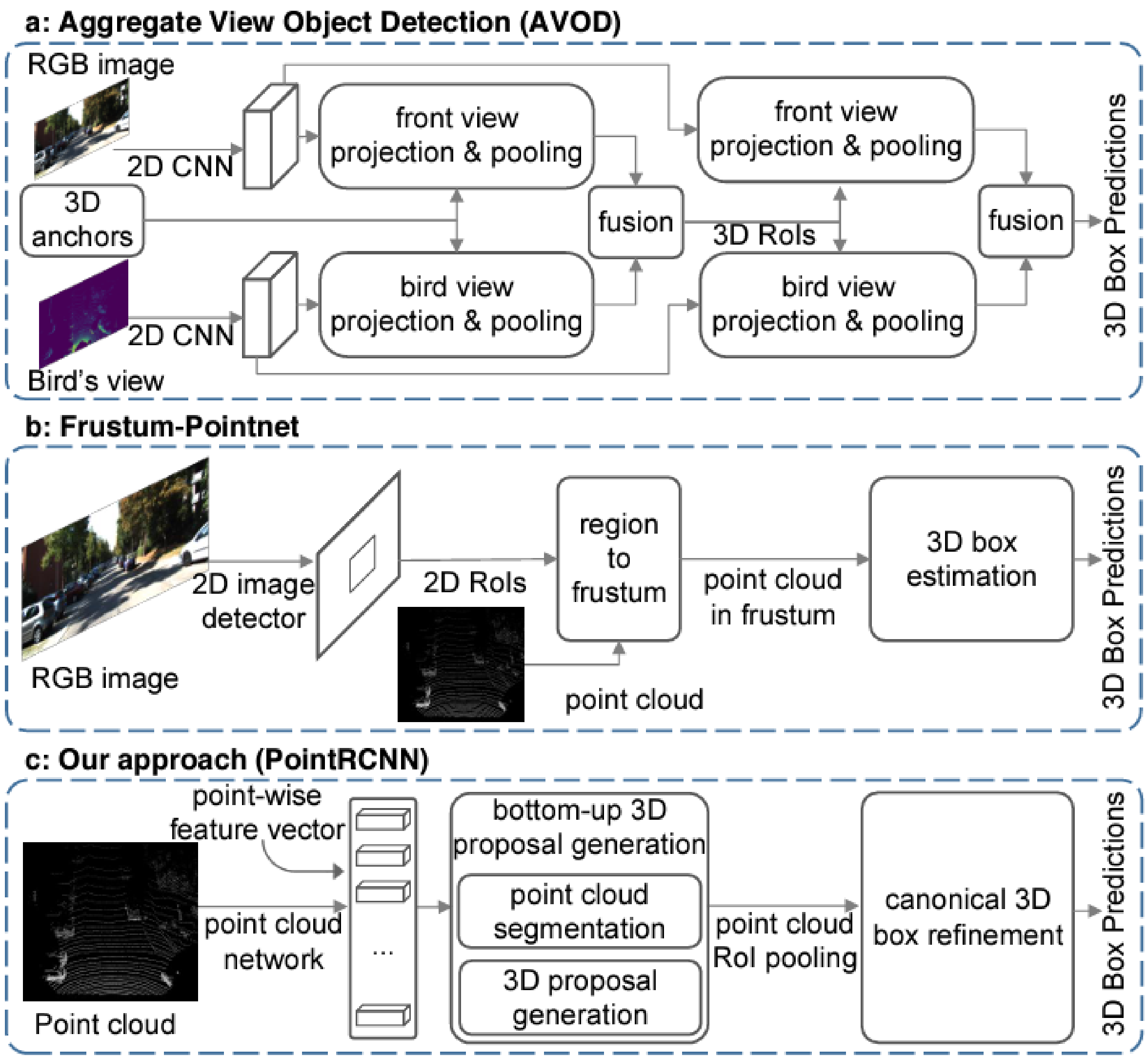


Yang, Z., Sun, Y., Liu, S., Shen, X., & Jia, J. (2019). **Std**: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1951-1960).

Detection and prediction refinement network

Refinement networks:

- RoI are noisy and not accurate
- Refinement network refine the imperfect bounding box proposals
- Combine global and local features
- Common in many multi-stage models



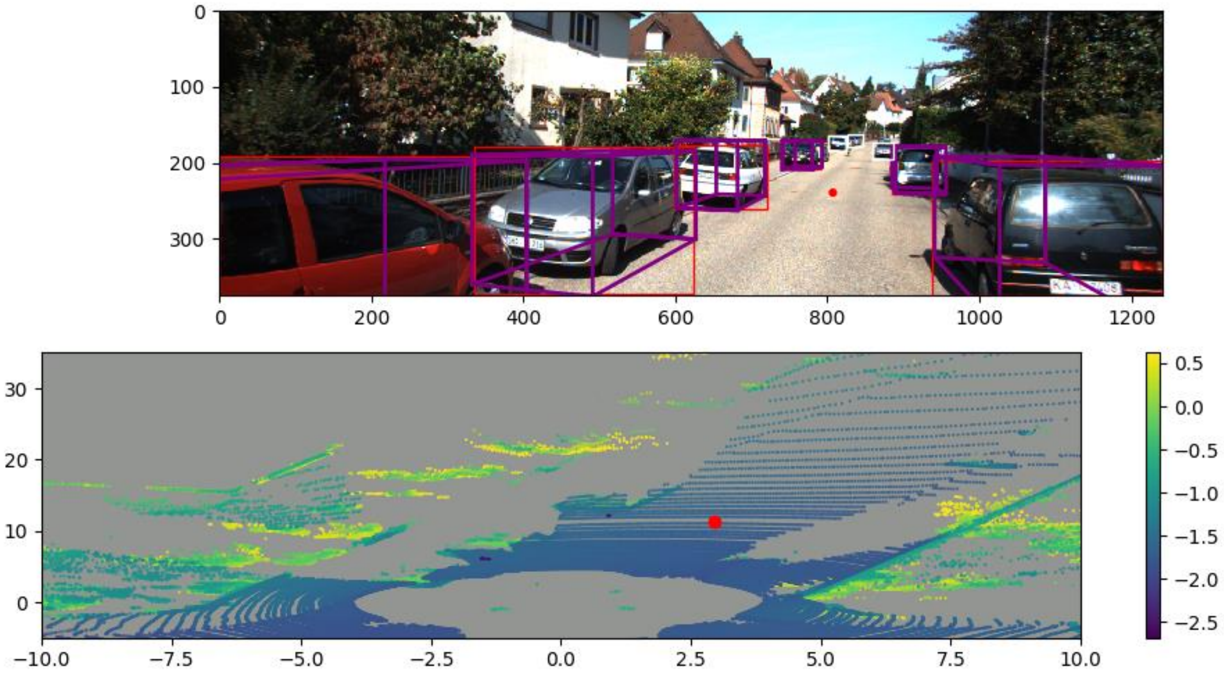
Shi, S., Wang, X., & Li, H. (2019). **Pointrcnn**: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 770-779).
Chen, Y., Liu, S., Shen, X., & Jia, J. (2019). Fast **point r-cnn**. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9775-9784).
Zarzar, J., Giancola, S., & Ghanem, B. (2019). **PointRGCN**: Graph convolution networks for 3D vehicles detection refinement. *arXiv preprint arXiv:1911.12236*.

Some famous models

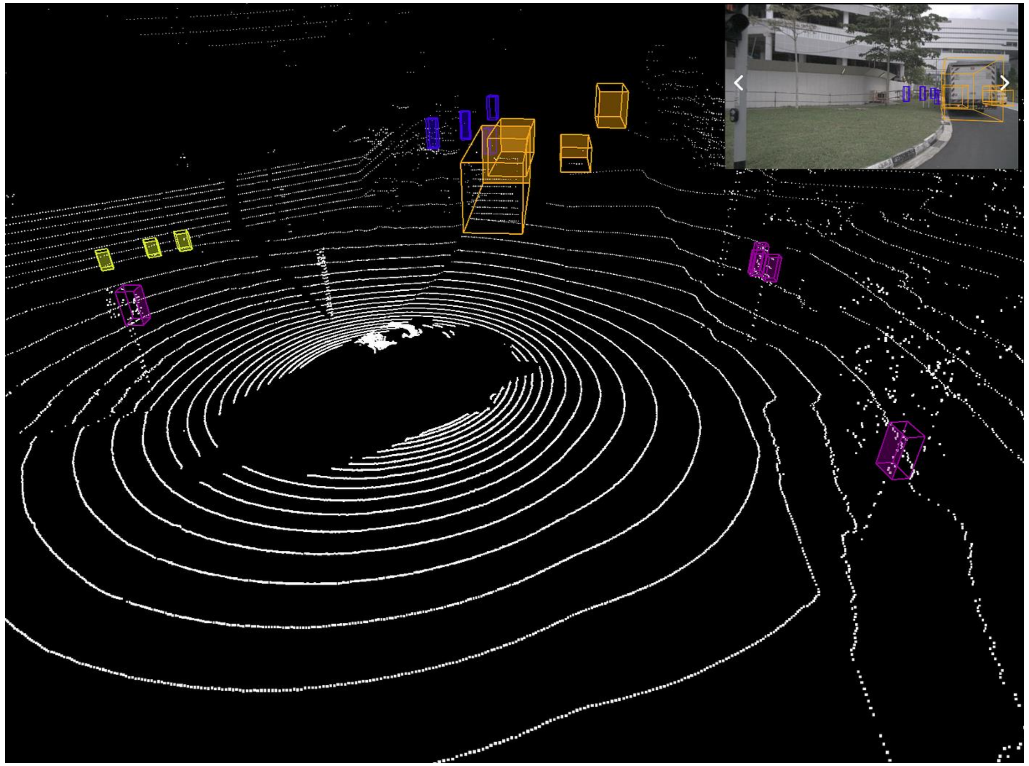
Data representation	Model	Architecture	Data feature extraction	Detection Encoder	Multi-scale feature learning	Detection settings	Prediction refinement net.
Volumetric	3D FCN	Single-stage	3D CNN	Anchorless	-	Masks	-
	VoxelNet	Single-stage	Compound	Region proposal	Integrated features	Bounding Box	-
	SECOND	Single-stage	Compound	Region proposal	Integrated features	Bounding Box	-
	PointPillars	Single-stage	Compound	Region proposal	Multiple prediction pyramid	Bounding Box	-
	Voxel-fpn	Single-stage	Segment	Region proposal	Multiple prediction pyramid	Bounding Box	Global features
Points	PointRCNN	Dual-stage	Segment	Anchorless	Prediction pyramid	Mask	Per-region data fusion
	STD	Dual-stage	Segment	Anchorless	-	Masks	Per-region data fusion
	LaserNet	Single-stage	3D CNN	Anchorless	-	Bounding Box	-
Projection	HDNet	Single-stage	2D CNN	Anchorless	Prediction pyramid	Bounding Box	-
	RT3D	Dual-stage	2D CNN	Region proposal	Prediction pyramid	Bounding Box	-
	Pixor	Single-stage	2D CNN	Anchorless	Prediction pyramid	Bounding Box	-

How to train (Public Datasets)

KITTI



NuScenes



Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The **kitti** dataset. *The International Journal of Robotics Research*, *32*(11), 1231-1237.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., ... & Beijbom, O. (2020). **nuscenes**: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11621-11631).

How to train (Public Datasets)

Waymo

A2D2

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., ... & Anguelov, D. (2020). Scalability in perception for autonomous driving: **Waymo** open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2446-2454).Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A. S., ... & Schuberth, P. (2020). **A2d2**: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*.

How to train (Public Datasets)

	Kitti [119,120]	NuScenes[121]	Waymo [122]	A2D2 [123]
Lidar Sensor	1 (64 channels)	1 (32 channels)	1+4 aux. (64 channel)	5 (16 channel)
Horizontal FoV (degrees)	360°	360°	360°	360°
Cameras	4 (0.7 MP)	6 (1.4 MP)	3(2.5 MP)+ 2 (1.7 MP)	6(2.3 MP)
Vehicle Bus Data	GPS+IMU	-	Velocity and angular velocity	GPS, IMU, steering angle, brake, throttle, odometry, velocity, pitch roll
Location	urban, one city (Karlsruhe)	urban, two cities (Boston and Sinagpore)	3 urban regions (USA)	urban, highways, country, roads, three cities in Germany
Hours	day	day, night	day, night	day
Weather	sunny, cloudy	various weather	various weather	various weather
Objects	3D	3D	3D, 2D	3D, pixel
Last Update	2015	2019	2019	2020
N° classes	3 (car, pedestrian and cyclist)	Up to 23 ("animal", "human.pedestrian.adult", "vehicle.bicycle" or "vehicle.emergency.police", "vehicle.moving", "pedestrian. standing" or "pedestrian.moving", etc.)	4 (vehicle, pedestrian, cyclist and sign)	14 (car, truck, pedestrian, cyclist, Van, Bus, Trailer, motorcycle, Emergency vehicles, animals among others)
Annotated Frames	20k	40k	230k	12k
3D Boxes	200k	1.4M	12M	N.S
Size (Hours)	1.5	5.5	6.4	N.S
Frames per second	10	20	2	10
Average points per frame	120k	34k	177k	N.S

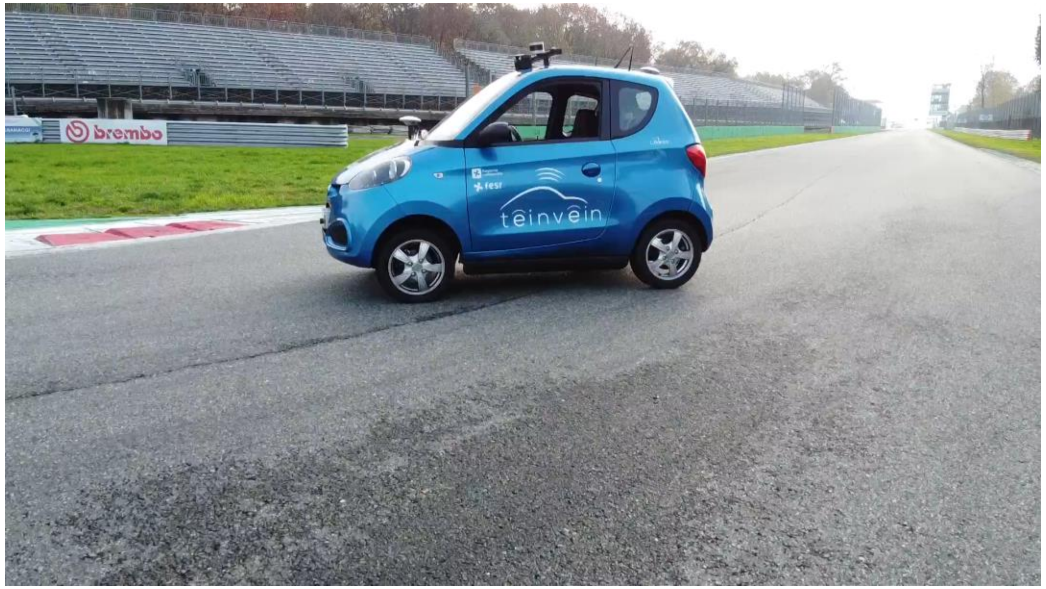
How do they perform?

Data Representation	Model (Year)	Inference Time (ms)	Cars E	M	H	Pedestrians E	M	H	Cyclist E	M	H
Volumetric	3D FCN (2016) [28]	1000	-	-	-	-	-	-	-	-	-
	VoxelNet (2017) [25]	225	77.47[*1]	65.11[*1]	57.73[*1]	39.48[*1]	33.69[*1]	31.51[*1]	61.22[*1]	48.36[*1]	44.37[*1]
	Vote3Deep (2017)[24]	1100	-	-	-	-	-	-	-	-	-
	SECOND-V1.5 (2018) [29]	20	84.65	75.96	68.71	-	-	-	-	-	-
	HR-SECOND (2018) [29]	110	84.78	75.32	68.70	45.31	35.52	33.14	75.83	60.82	53.67
	Patch Refinement – Patches - EMP (2018) [30]	50	89.84	78.41	73.15	-	-	-	-	-	-
	Patch Refinement – Patches (2018) [30]	150	88.67	77.20	71.82	-	-	-	-	-	-
	PointPillars (2018)[49]	16	82.58	74.31	68.99	51.45	41.92	38.89	77.10	58.65	51.92
	Fast Point R-CNN (2019) [31]	60	85.29	77.40	70.24	-	-	-	-	-	-
	VOXEL-FPN (2019) [32]	50	85.64	76.70	69.44	-	-	-	-	-	-
	PV-RCNN (2019) [33]	80	90.25	81.43	76.82	52.17	43.29	40.29	78.60	63.71	57.65
	MEGVII (2019)[27]	-	-	-	-	-	-	-	-	-	-
	HotSpotNet-Dense (2019)[95]	-	88.12[*1]	78.34[*1]	73.49[*1]	47.14[*1]	39.72[*1]	37.25[*1]	79.09[*1]	62.72[*1]	56.76[*1]
	HotSpotNet-Direct (2019) [95]	-	86.49[*1]	77.74[*1]	72.97[*1]	51.29[*1]	44.81[*1]	41.13[*1]	77.70[*1]	63.16[*1]	57.16[*1]
	3DBN (2019) [34]	130	83.77	73.53	66.23	-	-	-	-	-	-
	Fusion of Fusion Net (2020) [35]	50	84.15	74.45	66.97	49.44	41.21	36.42	75.36	59.65	53.03
	Point A^2-anchor (2020)[36]	80	87.81	78.49	73.51	53.10	43.35	40.06	79.17	63.52	56.93
	Point A^2-free (2020)[36]	80	88.48[*3]	78.96[*3]	78.36[*3]	70.73[*3]	64.13[*3]	57.45[*3]	88.18[*3]	73.35[*3]	70.75[*3]
	HVNet (2020) [62]	31	87.21[*3]	77.58[*3]	71.79[*3]	69.13[*3]	64.81[*3]	59.42[*3]	87.21[*3]	73.75[*3]	68.98[*3]
Points	IPOD (2018) [39]	20	79.75[*2]	72.57[*2]	66.33[*2]	56.92[*2]	44.68[*2]	42.39[*2]	71.40[*2]	53.46[*2]	48.34[*2]
	STD (2019) [40]	80	87.95	79.71	74.16	53.29	42.47	38.35	78.69	61.59	55.30
	PointRCNN(2019)[10]	100	86.96	75.64	70.70	47.98	39.37	36.01	74.96	58.82	52.53
	R-GCN only (2019) [41]	160	83.42	75.26	68.73	-	-	-	-	-	-
	PointRGCN (2019) [41]	260	85.97	75.73	70.60	-	-	-	-	-	-
	R-GCN (2019) [41]	160	83.42	75.26	68.73	-	-	-	-	-	-
	C-GCN (2019) [41]	147	83.49	73.62	67.01	-	-	-	-	-	-
	LaserNet (2019) [42]	30	-	-	-	-	-	-	-	-	-
Projection	Vehicle FCN detection (2016) [50]	-	-	-	-	-	-	-	-	-	-
	HDNet (2018) [53]	50	-	-	-	-	-	-	-	-	-
	BirdNet (2018) [52]	99	40.99	27.26	25.32	22.04	17.08	15.82	43.98	30.25	27.21
	RT3D (2018) [54]	90	23.74	19.14	18.86	-	-	-	-	-	-
	Pixor (2019) [55]	35	-	-	-	-	-	-	-	-	-

Future directions

Still an open problem, multiple improvements:

- Leverage data sparsity:
  - Improved kernel and convolution techniques
- Data representation:
  - Compressed representation without loss of information
- Multimodal perception:
  - Combine multiple sensors data
- Motion information integration
- Employ more recent architectures, e.g., transformers
- Optimization for real time requirements
- Deploy in real scenarios and drive



POLITECNICO
MILANO 1863

Deep Learning in 3D for Robotics

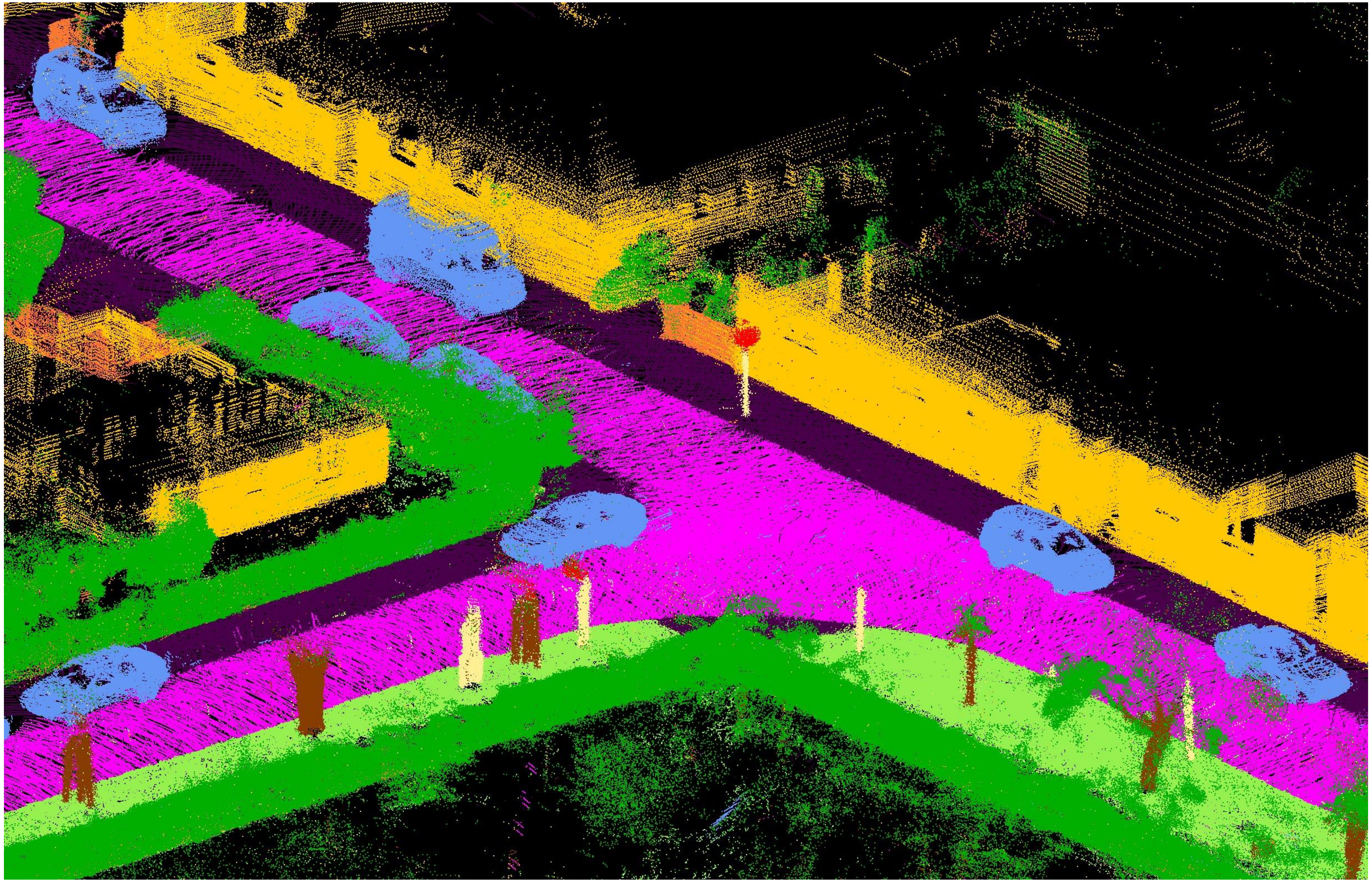
- 3D Point Clouds Segmentation -

Matteo Matteucci (matteo.matteucci@polimi.it), Matteo Frosi (matteo.frosi@polimi.it)

Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

AIRLAB
ARTIFICIAL INTELLIGENCE AND ROBOTICS LAB

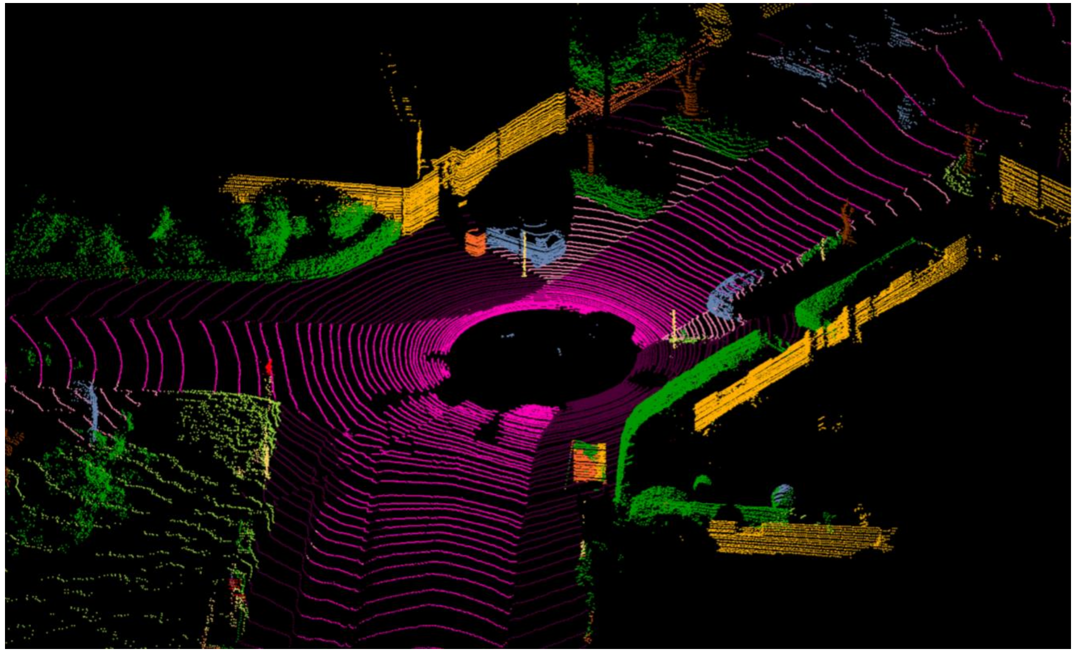
What is point cloud segmentation?



Segmentation requires the understanding of both the global geometric structure and the fine-grained details of each point.

According to the granularity, 3D point cloud segmentation methods can be classified into three categories: *semantic segmentation* (scene level), *instance segmentation* (object level) and *part segmentation* (part level).

From images to point clouds



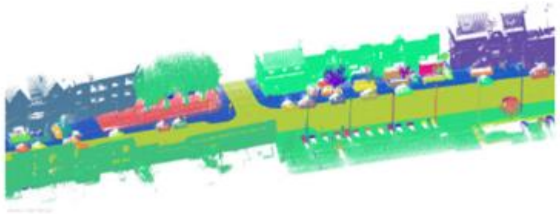
Cloud segmentation is challenging!

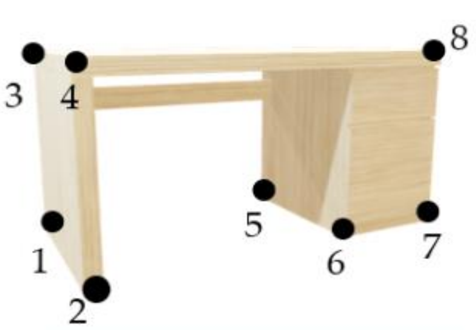
1. Unlike pixels, points are **unstructured**, making it difficult to apply well known architectures
2. Clouds have **translational variance**, i.e., the same object can appear different if located at different positions
3. **Sparsity** and **disorder**
4. **Computational inefficiency**, due to the high amount of data in a single point cloud

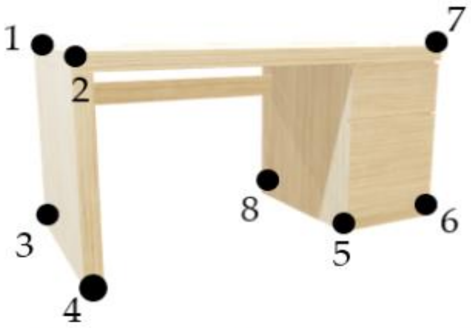
Why are point clouds so hard to use?











	x	y	z
1	x_1	y_1	z_1
2	x_2	y_2	z_2
3	x_3	y_3	z_3
4	x_4	y_4	z_4
5	x_5	y_5	z_5
6	x_6	y_6	z_6
7	x_7	y_7	z_7
8	x_8	y_8	z_8

	x	y	z
1	x_1	y_1	z_1
2	x_2	y_2	z_2
3	x_3	y_3	z_3
4	x_4	y_4	z_4
5	x_5	y_5	z_5
6	x_6	y_6	z_6
7	x_7	y_7	z_7
8	x_8	y_8	z_8

Segmentation evolution timeline

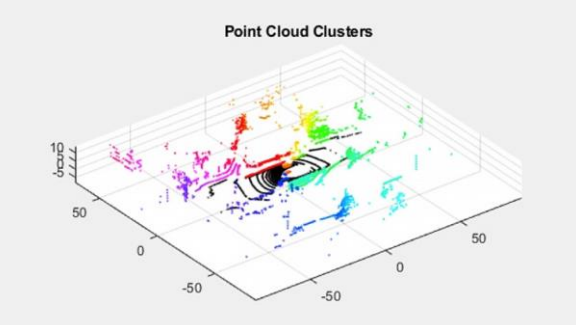
< 1980

around 2013

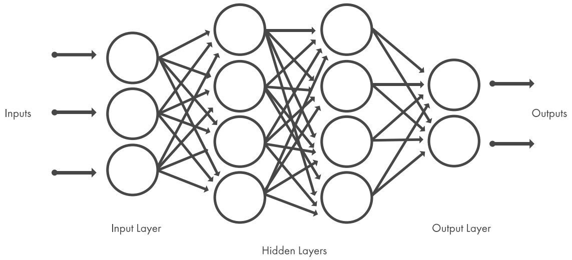
around 2020

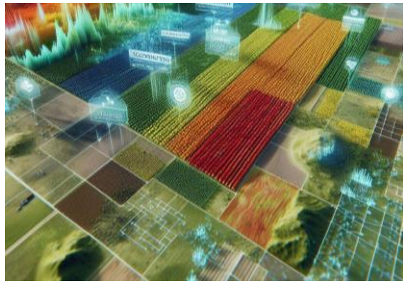
TODAY

CLASSIC ERA



DEEP ERA





GENERATIVE ERA

Some references

3D Point Cloud Segmentation: A survey

Anh Nguyen[1] Bac Le[2]

Abstract—**3D point cloud segmentation is the process of classifying point clouds into multiple homogeneous regions, the points in the same region will have the same properties. The segmentation is challenging because of high redundancy, uneven sampling density, and lack explicit structure of point cloud data. This problem has many applications in robotics such as intelligent vehicles, autonomous mapping and navigation. Many authors have introduced different approaches and algorithms. In this survey, we examine methods that have been proposed to segment 3D point clouds. The advantages, disadvantages, and design mechanisms of these methods are analyzed and discussed. Finally, we outline the promising future research directions.**

I. Introduction

Fully three dimensional scanners are now widely available. In particular, with scanners such as Light Detection and Ranging (LIDAR) and Microsoft Kinect, 3D point clouds can be easily acquired for different purposes. The explosion of point cloud data need a library to process them. Point Cloud Library (PCL) [11] was introduced in 2011. This library contains state of the art algorithms for 3D perception. With the development of hardware and PCL, processing point clouds gains more and more attraction in robotics, as well as other fields.

The segmentation of point clouds into foreground and background is a fundamental step in processing 3D point clouds. Given the set of point clouds, the objective of the segmentation process is to cluster points with similar characteristics into homogeneous regions. These isolated regions should be meaningful. The segmentation process could be helpful for analyzing the scene in various aspects such as locating and recognizing objects, classification, and feature extraction.

In computer graphics, intensive researches have been done to decompose 3D model into functionally meaningful regions. The general way is build a graph from the input mesh, and cluster the graph to produce a segmentation by using information such as normal direction, smoothness, or concavity along boundaries. Shamir [7] survey variety of methods have been proposed for this problem: convex decomposition, watershed analysis, hierarchical clustering, region growing, and spectral clustering. Many of these approaches have been used widely to segment point cloud data, especially in region based methods [26] [32] [21] [19] [43].

In computer vision, segmenting 2D images is a classic problem and has been studied for several decades. It attracts a significant amount of work [10] [40] [27]. One of the most popular approach is graph clustering (e.g. Graph Cuts [4] including Normalised Cuts [36] and Min Cuts [14]). The idea of these methods have been used widely to segmenting 3D point cloud data [9] [12] [44]. However, Anand [2] showed that when a 2D image is formed from the corresponding 3D world, we will lost a lot of valuable information about the 3D shape and geometric layout of objects.

The work of Anguelov [9] suggested a 3D point cloud segmentation algorithm should have three important properties. First, the algorithm should be able to take advantage of several qualitatively different kinds of features, such as trees will have distinguished features from cars. When the number of features grows, segmentation algorithm should be able to learn how to trade them off automatically. Second, segmentation algorithm should be able to infer the label of points which lie in sparsely sampled regions based on the information of their neighbors. Third, the segmentation algorithm should adapt to the particular 3D scanner used, because different laser scanners produce qualitatively different point cloud data, and they may have different properties even with the same scene.

In the next section, we outline the main challenges of the field as these motivate the various approaches. Then, we briefly describe the common available 3D point cloud datasets. We classify and discuss in detail segmentation methods in section 3. While many works have been proposed, we do not intend to give complete coverage of all works in the area. In section 4, we discuss limitations of the state of the art and outline future directions.

II. Challenges and Datasets

A. Challenges

We can precisely determine the shape, size and other properties of the objects in 3D data. However, segmenting objects in 3D point clouds is not a trivial task. The point cloud data are usually noisy, sparse, and unorganized. The sampling density of points is also typically uneven due to varying linear and angular rates of the scanner. In addition, the surface shape can be arbitrary with sharp features and there is no statistical distribution pattern in the data [31]. Moreover, due to the limitations of the 3D sensors, the foreground is often highly entangled with the background. These problems present a difficult challenge when designing a segmentation algorithm.

[1]Anh Nguyen is Graduate Research Fellow at Computer Science Department, Information Technology Faculty, University of Science, Vietnam nganh@fit.hcmus.edu.vn
[2]Dr. Bac Le is Associate Professor and dean of Computer Science Department, Information Technology Faculty, University of Science, Vietnam lhbac@fit.hcmus.edu.vn

978-1-4799-1201-8/13/$31.00 ©2013 IEEE 225

Deep Learning for 3D Point Clouds: A Survey

Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun

Abstract—Point cloud learning has lately attracted increasing attention due to its wide applications in many areas, such as computer vision, autonomous driving, and robotics. As a dominating technique in AI, deep learning has been successfully used to solve various 2D vision problems. However, deep learning on point clouds is still in its infancy due to the unique challenges faced by the processing of point clouds with deep neural networks. Recently, deep learning on point clouds has become even thriving, with numerous methods being proposed to address different problems in this area. To stimulate future research, this paper presents a comprehensive review of recent progress in deep learning methods for point clouds. It covers three major tasks, including 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation. It also presents comparative results on several publicly available datasets, together with insightful observations and inspiring future research directions.

Index Terms—Deep learning, point clouds, 3D data, shape classification, shape retrieval, object detection, object tracking, scene flow, instance segmentation, semantic segmentation, part segmentation

1 Introduction

With the rapid development of 3D acquisition technologies, 3D sensors are becoming increasingly available and affordable, including various types of 3D scanners, LiDARs, and RGB-D cameras (such as Kinect, RealSense and Apple depth cameras) [1]. 3D data acquired by these sensors can provide rich geometric, shape and scale information [2], [3]. Complemented with 2D images, 3D data provides an opportunity for a better understanding of the surrounding environment for machines. 3D data has numerous applications in different areas, including autonomous driving, robotics, remote sensing, and medical treatment [4].

3D data can usually be represented with different formats, including depth images, point clouds, meshes, and volumetric grids. As a commonly used format, point cloud representation preserves the original geometric information in 3D space without any discretization. Therefore, it is the preferred representation for many scene understanding related applications such as autonomous driving and robotics. Recently, deep learning techniques have dominated many research areas, such as computer vision, speech recognition, and natural language processing. However, deep learning on 3D point clouds still face several significant challenges [5], such as the small scale of datasets, the high dimensionality and the unstructured nature of 3D point clouds. On this basis, this paper focuses on the analysis of deep learning methods which have been used to process 3D point clouds.

Deep learning on point clouds has been attracting more and more attention, especially in the last five years. Several publicly available datasets are also released, such as ModelNet [6], ScanObjectNN [7], ShapeNet [8], PartNet [9], S3DIS [10], ScanNet [11], Semantic3D [12], ApolloCar3D [13], and the KITTI Vision Benchmark Suite [14], [15]. These datasets have further boosted the research of deep learning on 3D point clouds, with an increasingly number of methods being proposed to address various problems related to point cloud processing, including 3D shape classification, 3D object detection and tracking, 3D point cloud segmentation, 3D point cloud registration, 6-DOF pose estimation, and 3D reconstruction [16], [17], [18]. Few surveys of deep learning on 3D data are also available, such as [19], [20], [21], [22]. However, our paper is the first to specifically focus on deep learning methods for point cloud understanding. A taxonomy of existing deep learning methods for 3D point clouds is shown in Fig. 1.

Compared with the existing literatures, the major contributions of this work can be summarized as follows:

1) To the best of our knowledge, this is the *first* survey paper to comprehensively cover deep learning methods for several important point cloud understanding tasks, including 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation.
2) As opposed to existing reviews [19], [20], we specifically focus on deep learning methods for *3D point clouds* rather than all types of 3D data.

- *Yulan Guo is with the School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou 510275, China, and also with the College of Electronic Science and Technology, National University of Defense Technology, Changsha, China. E-mail: yulan.guo@nudt.edu.cn.*
- *Hanyun Wang is with the School of Surveying and Mapping, Information Engineering University, Zhengzhou 450001, China. E-mail: why.scholar@gmail.com.*
- *Qingyong Hu is with Department of Computer Science, University of Oxford, OX1 3PP Oxford, U.K. E-mail: qingyong.hu@cs.ox.ac.uk.*
- *Hao Liu is with the School of Electronics and Communication Engineering, Sun Yat-sen University, Guangzhou 510275, China. E-mail: liuh327@mail2.sysu.edu.cn.*
- *Li Liu is with the College of System Engineering, National University of Defense Technology, Changsha, China, and also with the Center for Machine Vision and Signal Analysis, University of Oulu, 90570 Oulu, Finland. E-mail: lilyliu_nudt@163.com.*
- *Mohammed Bennamoun is with the Department of Computer Science and Software Engineering, the University of Western Australia, Crawley, WA 6009, Australia. E-mail: mohammed.bennamoun@uwa.edu.au.*

Manuscript received 12 Jan. 2020; revised 19 June 2020; accepted 22 June 2020. Date of publication 29 June 2020; date of current version 3 Nov. 2021. (Corresponding author: Yulan Guo.) Recommended for acceptance by E. B Sudderth. Digital Object Identifier no. 10.1109/TPAMI.2020.3005434

electronics MDPI

Review

Deep-Learning-Based Point Cloud Semantic Segmentation: A Survey

Rui Zhang †, Yichao Wu *,†, Wei Jin and Xiaoman Meng

School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450046, China; zhangrui@ncwu.edu.cn (R.Z.); Z20211090841@stu.ncwu.edu.cn (W.J.); X20201090790@stu.ncwu.edu.cn (X.M.)
* Correspondence: wu546300070@gmail.com
† These authors contributed equally to this work.

Abstract: With the rapid development of sensor technologies and the widespread use of laser scanning equipment, point clouds, as the main data form and an important information carrier for 3D scene analysis and understanding, play an essential role in the realization of national strategic needs, such as traffic scene perception, natural resource management, and forest biomass carbon stock estimation. As an important research direction in 3D computer vision, point cloud semantic segmentation has attracted more and more researchers' attention. In this paper, we systematically outline the main research problems and related research methods in point cloud semantic segmentation and summarize the mainstream public datasets and common performance evaluation metrics. Point cloud semantic segmentation methods are classified into rule-based methods and point-based methods according to the representation of the input data. On this basis, the core ideas of each type of segmentation method are introduced, the representative and innovative algorithms of each type of method are elaborated, and the experimental results on the datasets are compared and analyzed. Finally, some promising research directions and potential tendencies are proposed.

Keywords: deep learning; point cloud semantic segmentation; convolutional neural network; feature representation learning; computer vision

check for updates

Citation: Zhang, R.; Wu, Y.; Jin, W.; Meng, X. Deep-Learning-Based Point Cloud Semantic Segmentation: A Survey. *Electronics* **2023**, *12*, 3642. https://doi.org/10.3390/electronics12173642

Academic Editor: Luca Mesin

Received: 12 July 2023
Revised: 6 August 2023
Accepted: 14 August 2023
Published: 29 August 2023

1. Introduction

In recent years, with the booming development of a large group of emerging industries, such as smart cities, automotive navigation systems, augmented reality, and environmental assessment, a large amount of research related to 3D scene perception has been motivated. This research invariably requires the processing and analysis of huge amounts of 3D data. How to enhance the understanding of 3D scenes and extract effective high-level features has become an important scientific problem in 3D computer vision.

As a key form and essential information carrier of 3D data, a point cloud is a collection of points representing the information of objects in 3D scenes, which can be used as a digital representation of the real world. Point clouds usually contain coordinates, color, intensity values, and other attributes so that the original geometric structure of the object in 3D scenes can be retained to the maximum extent. As a key step in understanding 3D scenes, point cloud semantic segmentation is a technique that divides the original point cloud into several subsets with different semantic information and classifies each point into specific groups according to the degree of attribute similarity. At present, point cloud semantic segmentation has been widely applied to national strategic needs, such as autonomous driving [1], augmented reality [2], and transmission line inspection [3]. It has important research significance and broad development prospects.

In recent years, deep learning techniques have made breakthroughs in computer vision, and more and more computer vision tasks rely on convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs), and

Classic era: how was it done before deep learning?

Classic era: How was it done before deep learning?

Classic approaches

Edge-based methods

Region-based methods

Attribute-based methods

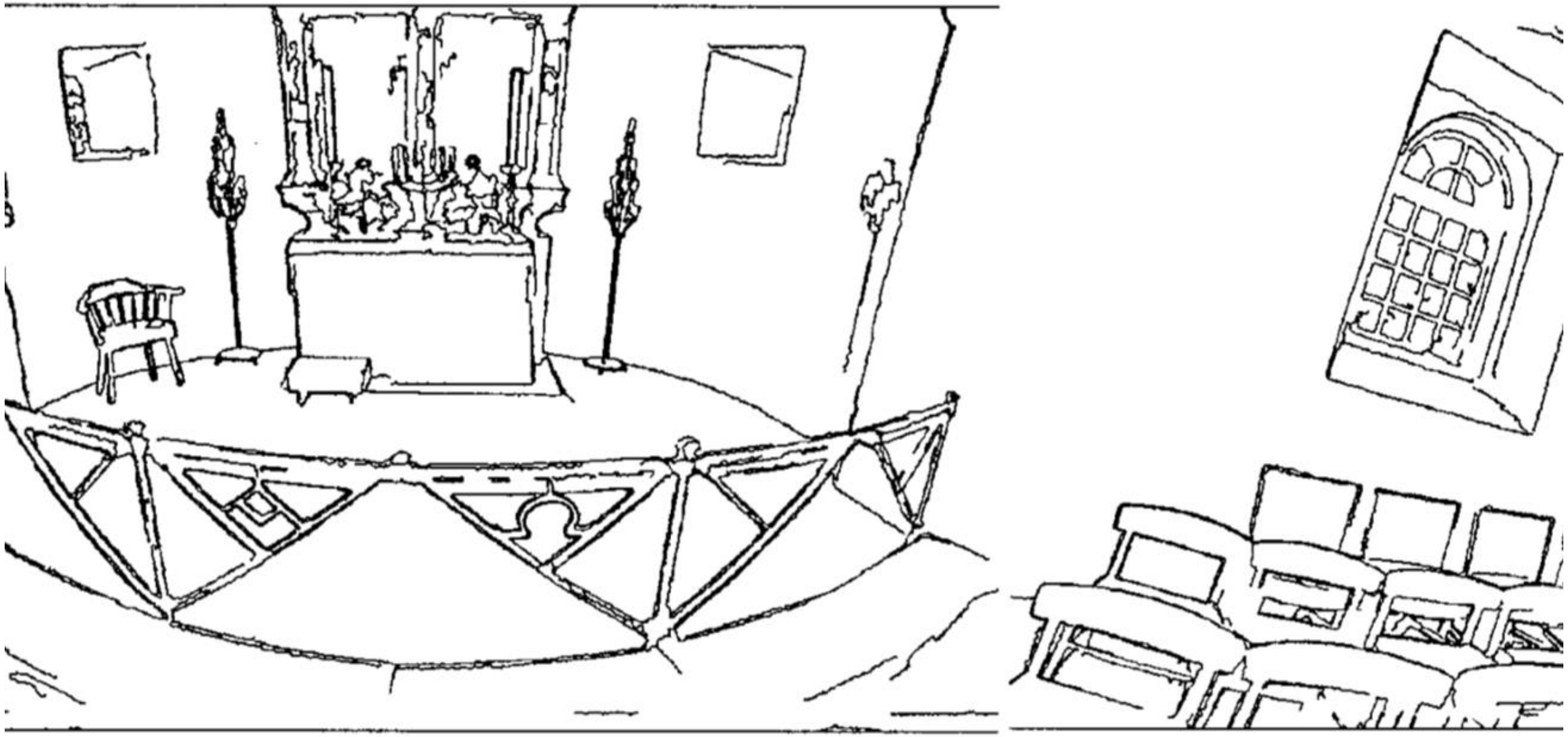
Model-based methods

Graph-based approaches

Nguyen, Anh, and Bac Le. "3D point cloud segmentation: A survey." *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*. IEEE, 2013.

Edge-based approaches

They detect the boundaries of several regions in the point clouds to obtain regions. The principle of the methods is to locate the points that have rapid change in intensity.



Advantages:

- Fast segmentation

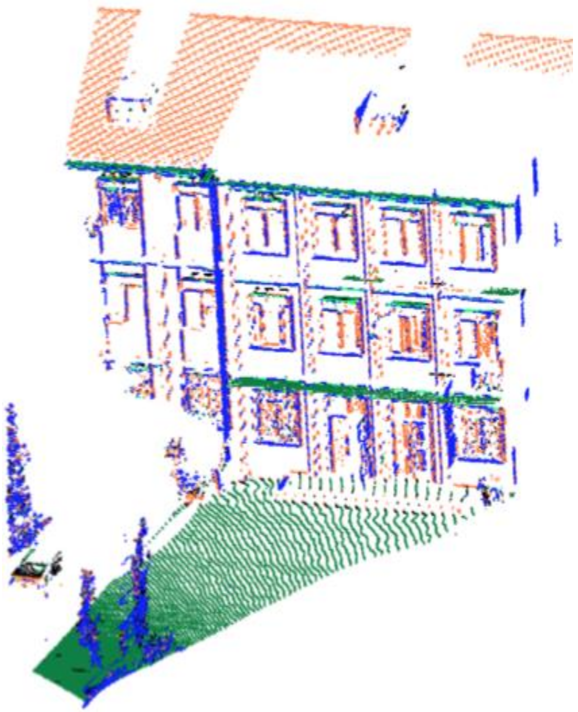
Disadvantages:

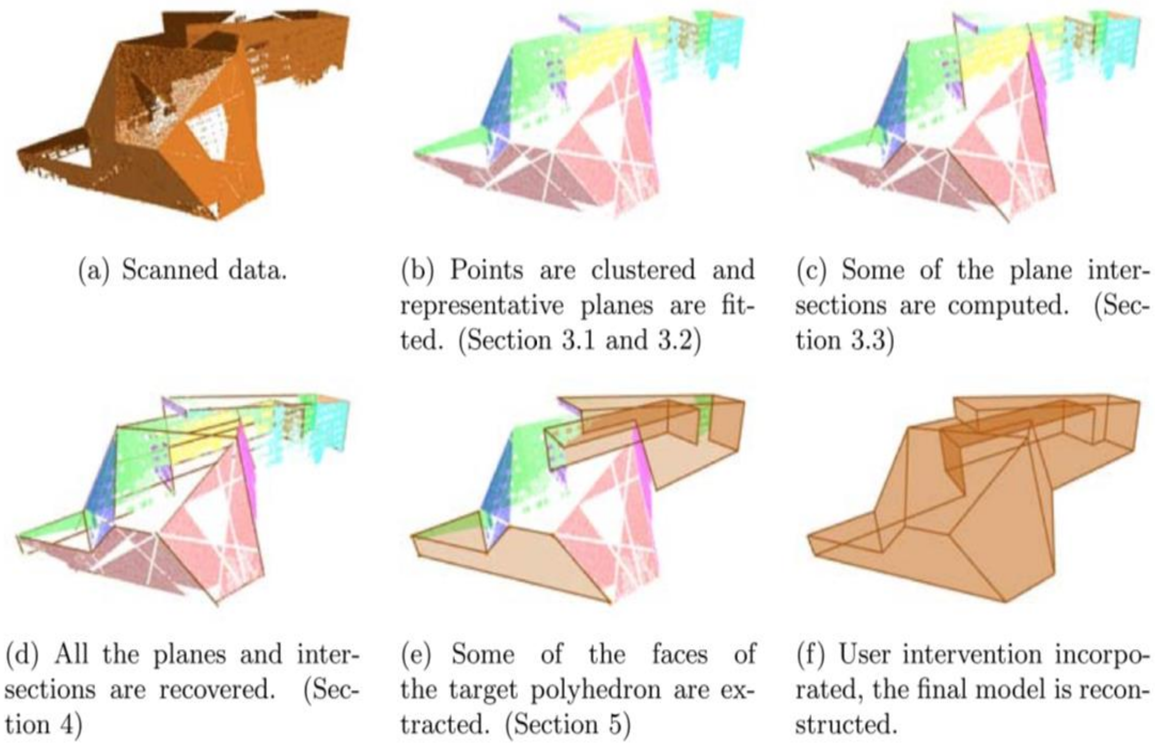
- Very low accuracy
- Sensitive to noise and density
- Require a middleman representation (e.g., range images)

Sappa, A., and M. Devy. "Fast range image segmentation byan edge detection strategy." *Proceedings of the3rd International Conference on 3D Digital Imagingand Modeling*. 2001.

Region-based approaches

They use neighborhood information to combine nearby points with similar properties, to obtain isolated regions, and to find dissimilarity between different regions. They are further classified in seeded (left) and unseeded (right) methods.





Advantages:
- Resistant to noise

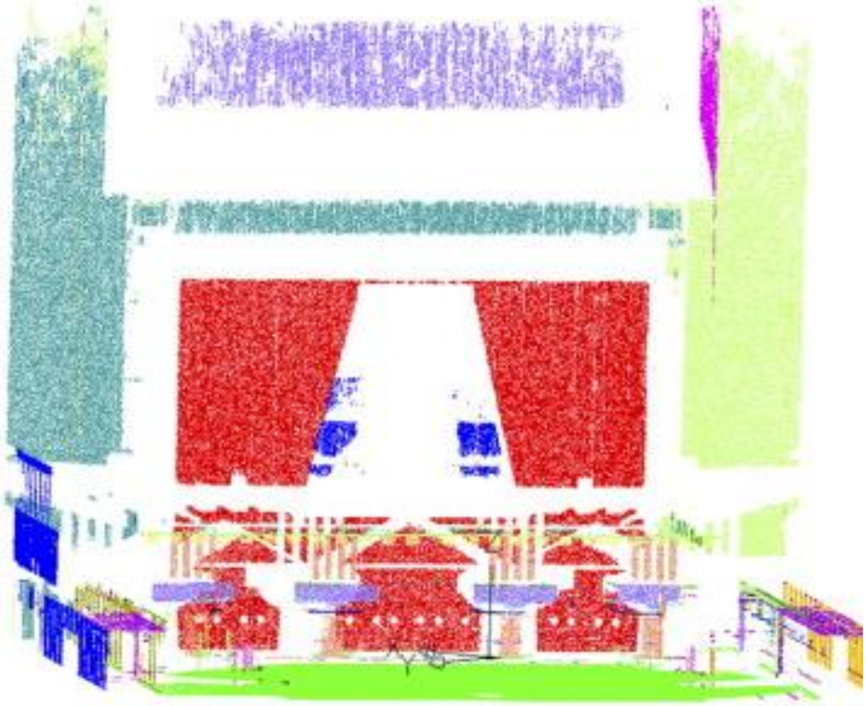
Disadvantages:
- Over and under segmentation issues
- Borders are fuzzy
- Slower than other methods

Ning, Xiaojuan, et al. "Segmentation of architecture shape information from 3D point cloud." *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*. 2009.

Chen, Jie, and Baoquan Chen. "Architectural modeling from sparsely scanned range data." *International Journal of Computer Vision* 78 (2008): 223-236.

Attribute-based approaches

These methods include two separate steps: attribute computation (e.g., Euclidean distance, density, normals) and attribute-based clustering.



Advantages:
- Spatial relations are considered
- Multi-cue clustering

Disadvantages:
- Accuracy heavily depends on attribute quality
- Precise computation can be slow

Biosca, Josep Miquel, and José Luis Lerma. "Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods." *ISPRS Journal of Photogrammetry and Remote Sensing* 63.1 (2008): 84-98.

Model-based approaches

They use geometric primitive shapes (e.g., sphere and plane) for grouping points. The points which have the same mathematical representation are grouped as one segment.

Advantages:

- Fast
- Robust to outliers

Disadvantages:

- Inaccurate when dealing with different point cloud sources

Schnabel, Ruwen, Roland Wahl, and Reinhard Klein. "Efficient RANSAC for point-cloud shape detection." *Computer graphics forum*. Vol. 26. No. 2. Oxford, UK: Blackwell Publishing Ltd, 2007.

Graph-based approaches

They consider the clouds in terms of a graph. In a simple model, each vertex corresponds to a point and the edges connect to certain pairs of neighboring points

(a) Colored lidar scan (b) True-color segmentation results

Advantages:
- Can segment complex scenes
- Can handle noise or uneven density

Disadvantages:
- Cannot run in real-time
- Computationally demanding

Strom, Johannes, Andrew Richardson, and Edwin Olson. "Graph-based segmentation for colored 3D laser point clouds." *2010 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2010.

Deep era: attention is all you ne… wait

Let the metrics resume

The **overall accuracy** (OA) is the ratio of the number of samples correctly predicted by the segmentation algorithms to the total number of samples.

$$OA = \frac{\sum_{i=0}^{N} M_{ii}}{\sum_{i=0}^{N} \sum_{j=0}^{N} M_{ij}}$$

The **mean class accuracy** (mAcc) is an improvement of OA, which calculates the precision for each category separately, and then averages the summed results according to the number of categories.

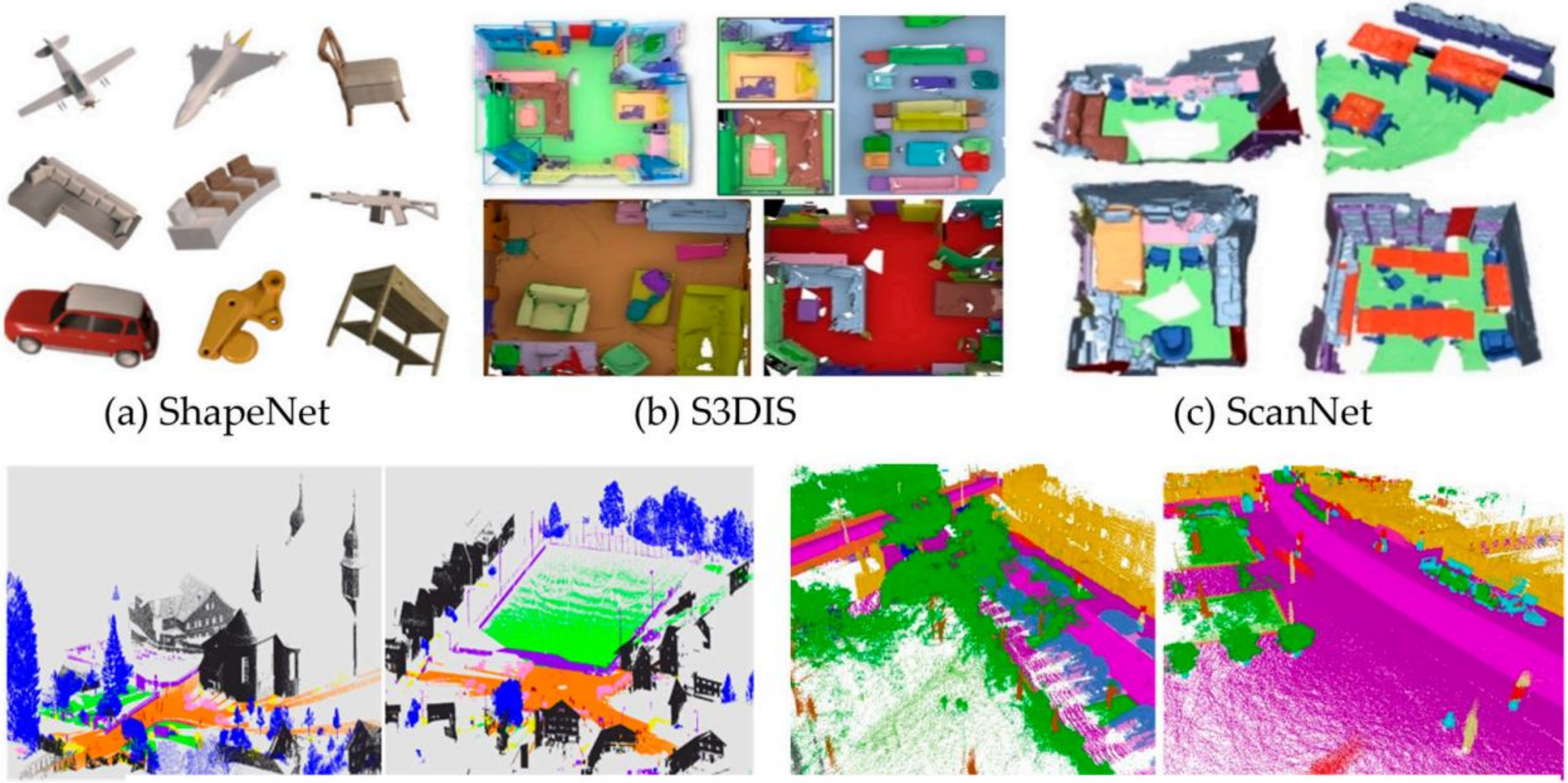
$$mAcc = \frac{1}{N+1} \sum_{i=0}^{N} \frac{M_{ii}}{\sum_{j=0}^{N} M_{ij}}$$

The mean **intersection over union** (mIoU) is the most important index to evaluate the performance of the segmentation methods, which first calculates the ratio between the intersection of the predicted and true regions of the models for each category, and then calculates the average value of the summed results according to the number of categories.

$$mIoU = \frac{1}{N+1} \sum_{i=0}^{N} \frac{M_{ii}}{\sum_{j=0}^{N} M_{ij} + \sum_{i=0}^{N} M_{ji} - M_{ii}}$$

Assuming that there are N + 1 semantic classes (including empty class), Mij denotes the number of units with actual semantic type i but predicted type j and vice versa for Mji. Mii denotes the number of units with actual semantic type i and predicted type i.

Well known datasets



(a) ShapeNet (b) S3DIS (c) ScanNet

(d) Semantic3D (e) SemanticKITTI

WELL KNOWN datasets (up to 2022)

Name	Year	Type	Application Scenario	Category	Size	Sensor
ModelNet10 [15]	2015	S	Oc	10	4.9 Tm	-
ModelNet40 [15]	2015	S	Oc	10	12.3 Tm	-
ScanObjectNN [23]	2019	R	Oc	15	15 To	-
ShapeNet [19]	2015	S	Ps	55	51.3 Tm	-
ShapeNet Part [24]	2016	S	Ps	16	16.9 Tm	-
SUN RGB-D [14]	2015	R	Is	47	103.5 Tf	Kinect
S3DIS [16]	2016	R	Is	13	273.0 Mp	Matterport
ScanNet [20]	2017	R	Is	22	242.0 Mp	RGB-D
MIMAP [25]	2020	R	Is	-	22.5 Mp	XBeibao
ArCH [26]	2020	R	Hs	10	102.74 Mp	TLS
KITTI [27]	2012	R	Os	3	179.0 Mp	MLS
Semantic3D [21]	2017	R	Os	8	4000.0 Mp	MLS
Paris-rue-Madame [28]	2018	R	Os	17	20.0 Mp	MLS
Paris-Lille-3D [18]	2018	R	Os	9	143.0 Mp	MLS
ApolloScape [29]	2018	R	Os	24	140.7 Tf	RGB-D
SemanticKITTI [22]	2019	R	Os	25	4549.0 Mp	MLS
Toronto-3D [30]	2020	R	Os	8	78.3 Mp	MLS
A2D2 [17]	2020	R	Os	38	41.3 Tf	TLS
SemanticPOSS [31]	2020	R	Os	14	216 Mp	MLS
WHU-TLS [32]	2020	R	Os	-	1740.0 Mp	TLS
nuScenes [33]	2020	R	Os	31	34.1 Tf	Velodyne HDL-32E
PandaSet [34]	2021	R	Os	37	16.0 Tf	MLS
Panoptic nuScenes [35]	2022	R	Os	32	1100.0 Mp	MLS
TJ4DRadSet [36]	2022	R	Os	8	7.75 Tf	4D Radar
DALES [37]	2020	R	Us	8	505.0 Mp	ALS
LASDU [38]	2020	R	Us	5	3.12 Mp	ALS
SensatUrban [39]	2022	R	Us	13	2847.0 Mp	UAV Photogrammetry

- S --> Synthetic Environment
- R --> Real Environment

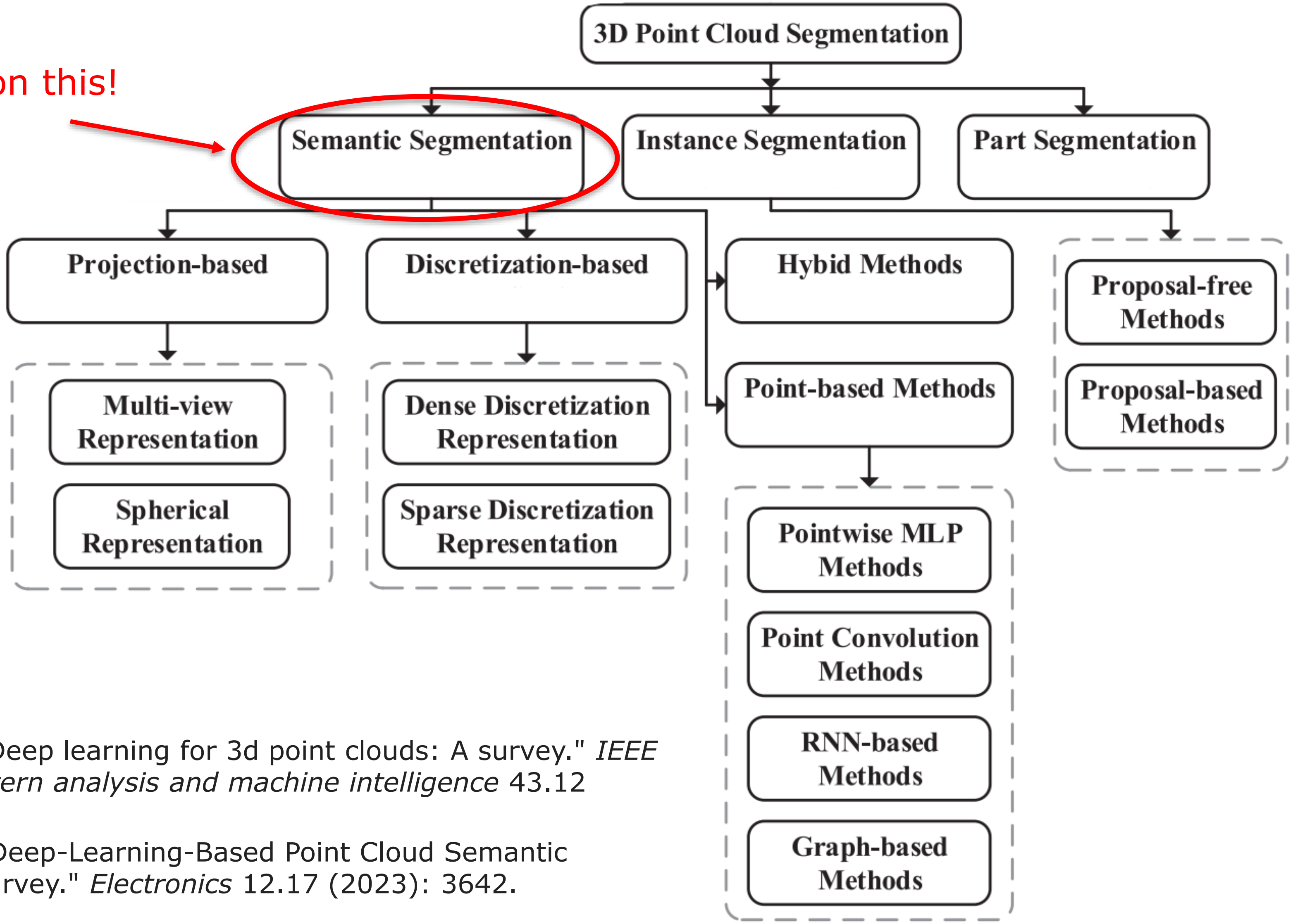
- Oc --> Object classification
- Ps --> Part segmentation
- Is --> Indoor segmentation
- Os --> Outdoor segmentation
- Hs --> Heritage segmentation
- Us --> Urban segmentation

- Tm --> Thousand models
- Tf --> Thousand frames
- To --> Thousand objects
- Mp --> Million points

- ALS --> Airborne Laser Scanning
- MLS --> Mobile Laser Scanning
- TLS --> Terrestrial Laser Scanning

Deep segmentation: a high-level classification

We will focus on this!



Guo, Yulan, et al. "Deep learning for 3d point clouds: A survey." *IEEE transactions on pattern analysis and machine intelligence* 43.12 (2020): 4338-4364.

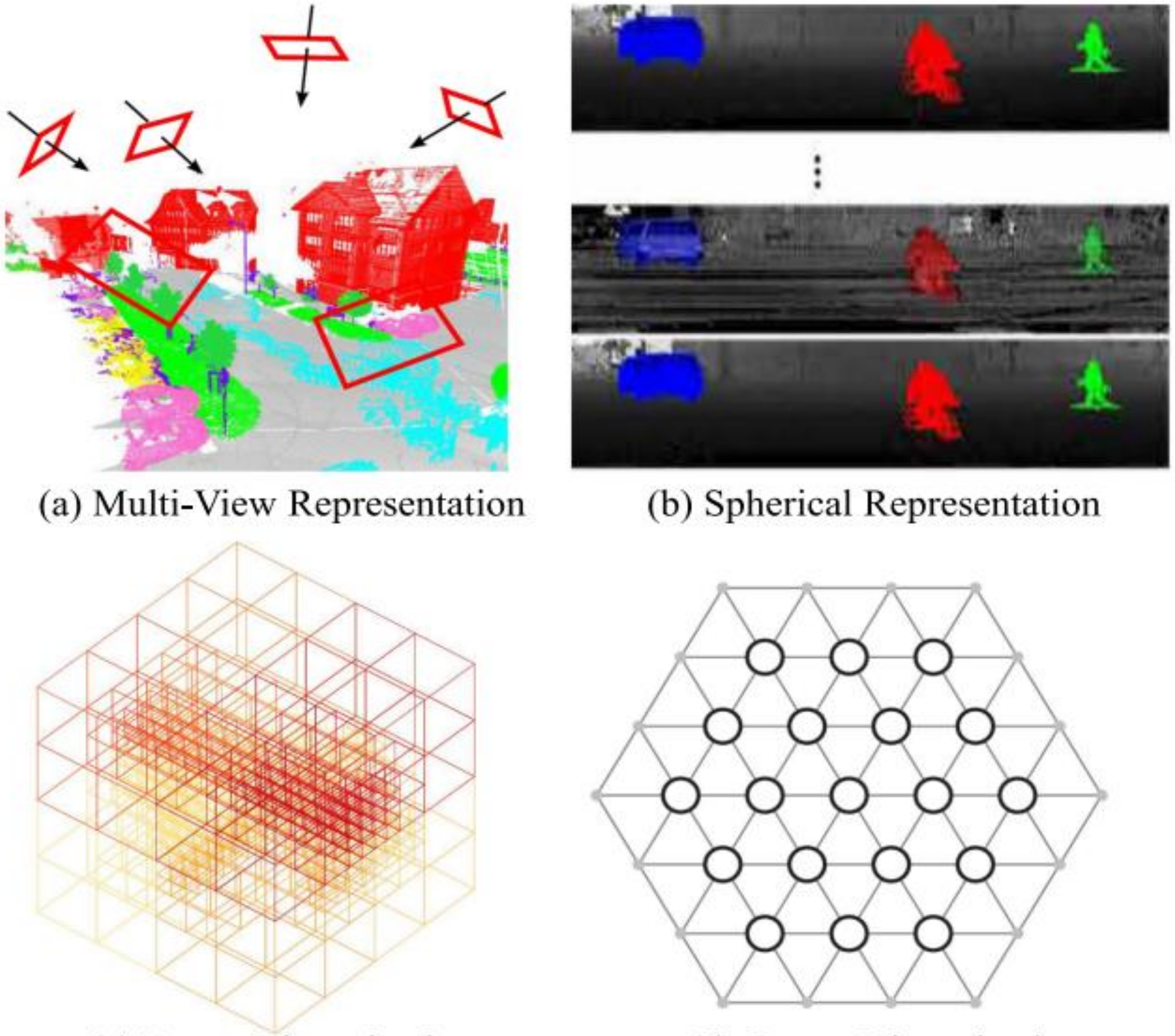
Zhang, Rui, et al. "Deep-Learning-Based Point Cloud Semantic Segmentation: A Survey." *Electronics* 12.17 (2023): 3642.

Semantic Segmentation

The goal of semantic segmentation is to separate a cloud into subsets according to the semantic meanings of points.

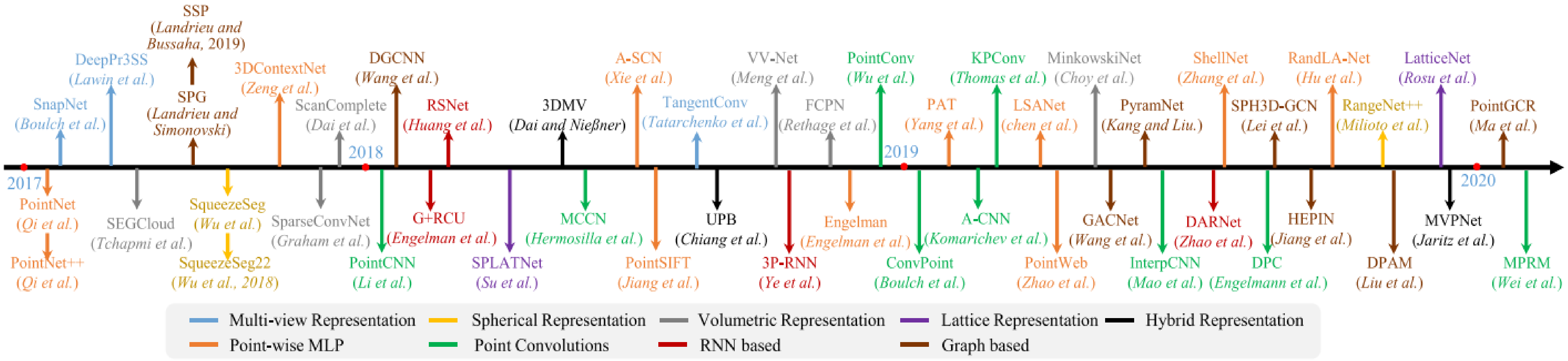
There are four paradigms for semantic segmentation: projection-based, discretization-based, point-based, and hybrid methods.

- Both the projection and discretization-based methods transform a point cloud to an intermediate representation, such as multi-view, spherical, volumetric, permutohedral lattice, and hybrid.
- Point-based methods directly work on irregular point clouds.



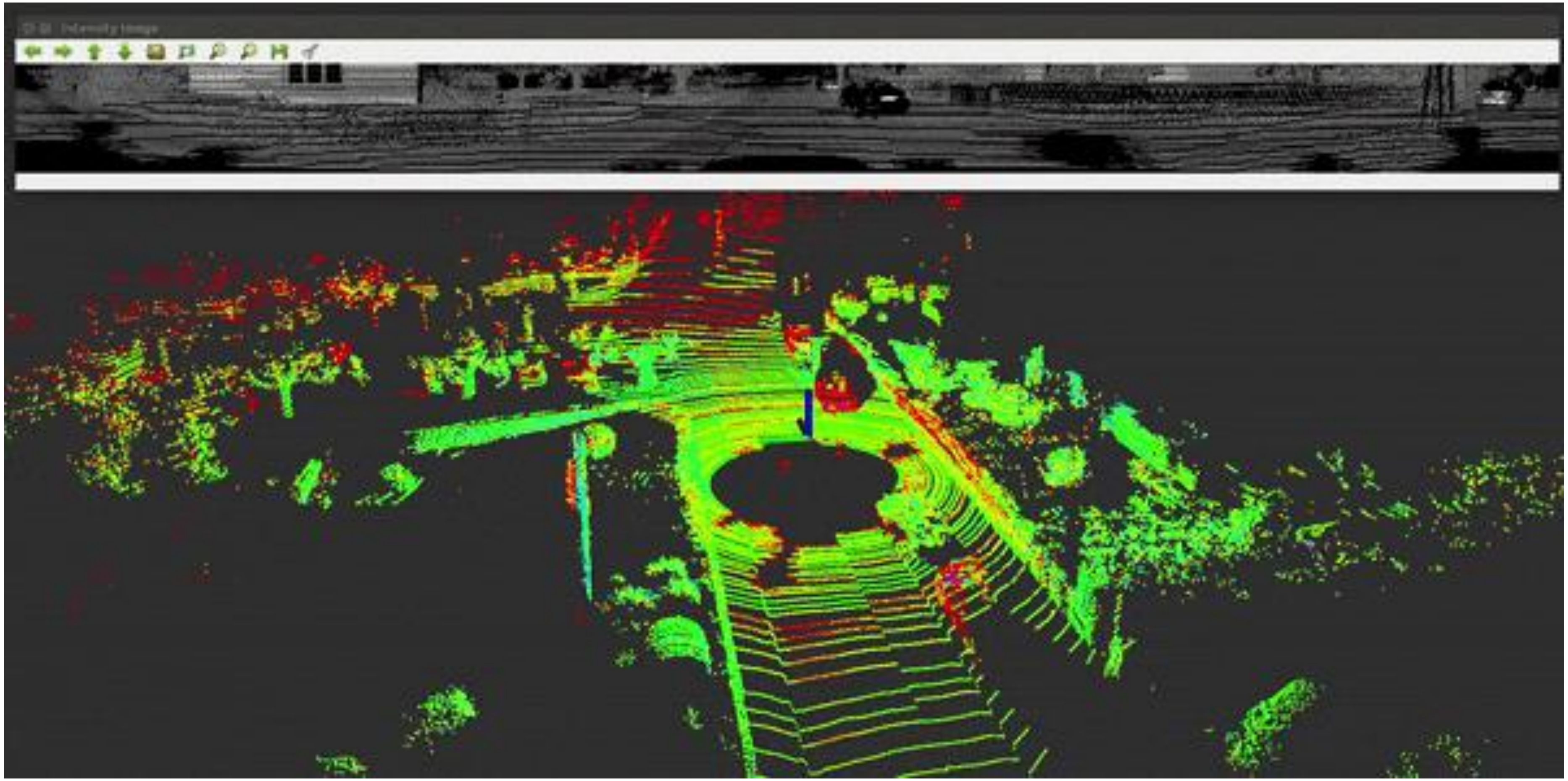
(a) Multi-View Representation (b) Spherical Representation (c) Dense Discretization Representation (d) Sparse Discretization Representation

Semantic Segmentation – From 2017 to 2021

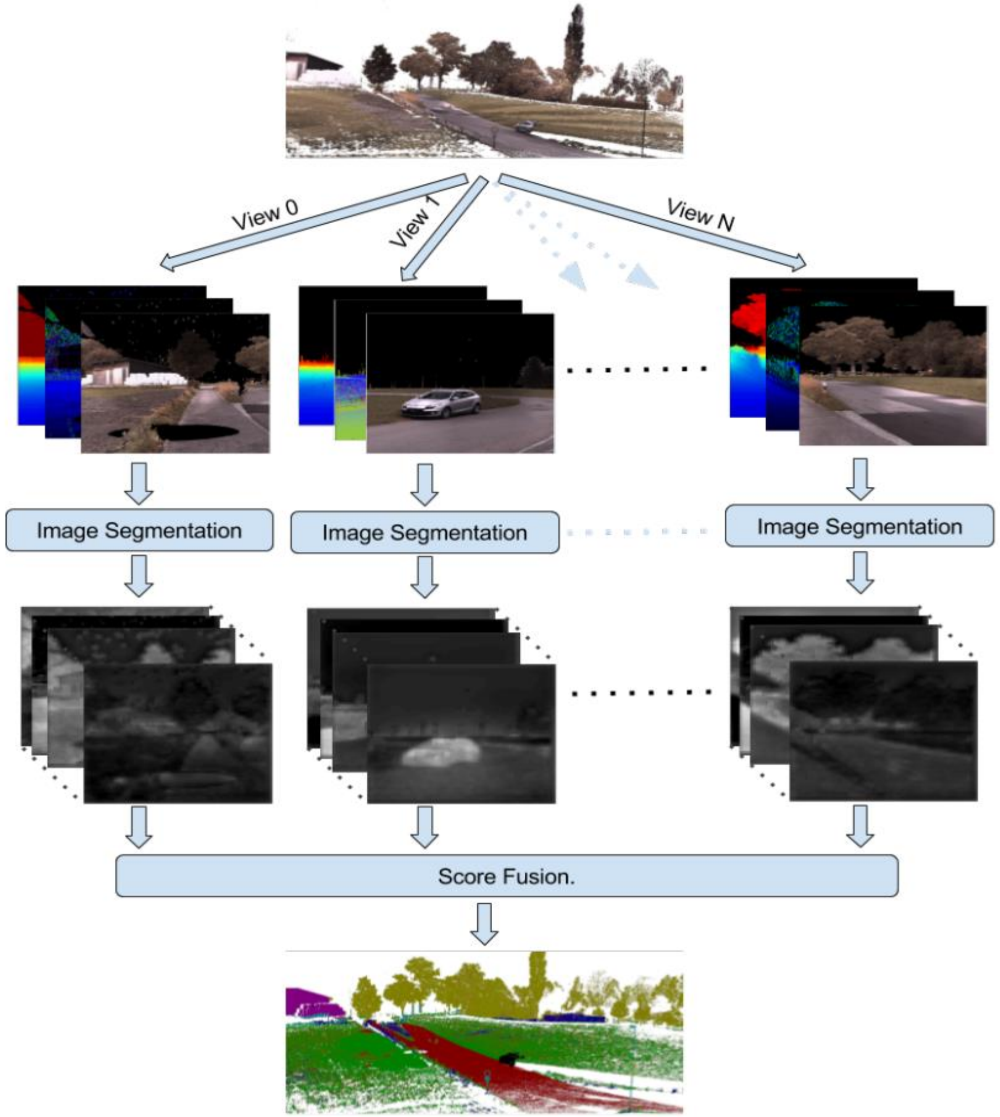


Projection-based methods

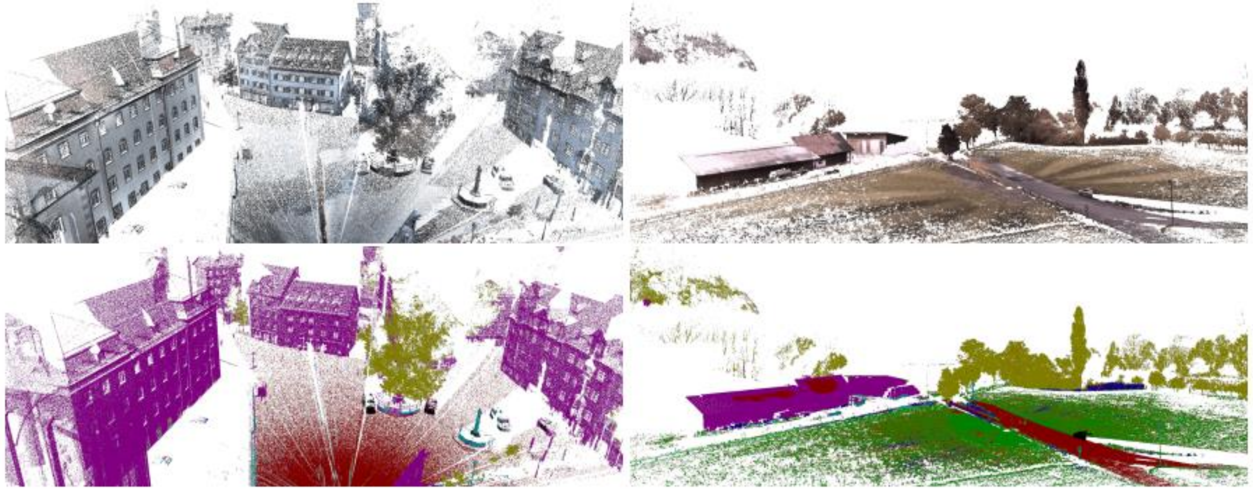
These methods usually project a 3D point cloud into 2D images, including multi-view and spherical images. Those images are then segmented using state-of-the-art methods for image segmentation, and the results are back-projected in 3D.



Projection-based methods: Multi-view representation (1)

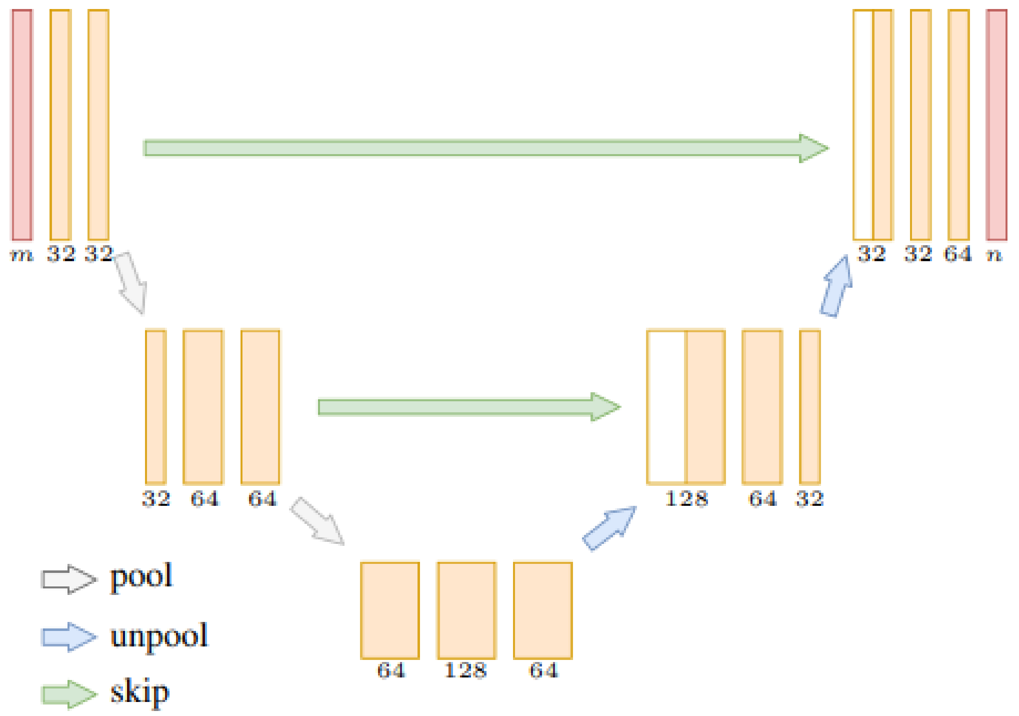


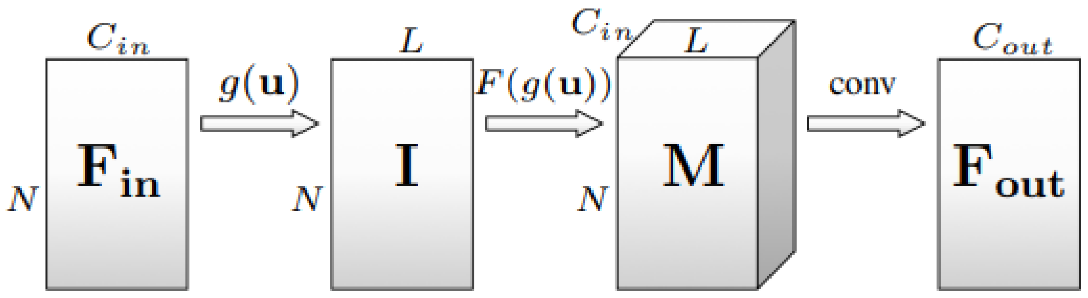
1. The input point cloud is projected into multiple virtual camera views, generating 2D color depth and surface normal images.
2. The images for each view are processed by a multi-stream CNN (VGG16) for segmentation.
3. The output predication scores from all views are fused into a single prediction for each point.



Lawin, Felix Järemo, et al. "Deep projective 3D semantic segmentation." *Computer Analysis of Images and Patterns: 17th International Conference, CAIP 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I 17*. Springer International Publishing, 2017.

Projection-based methods: Multi-view representation (2)

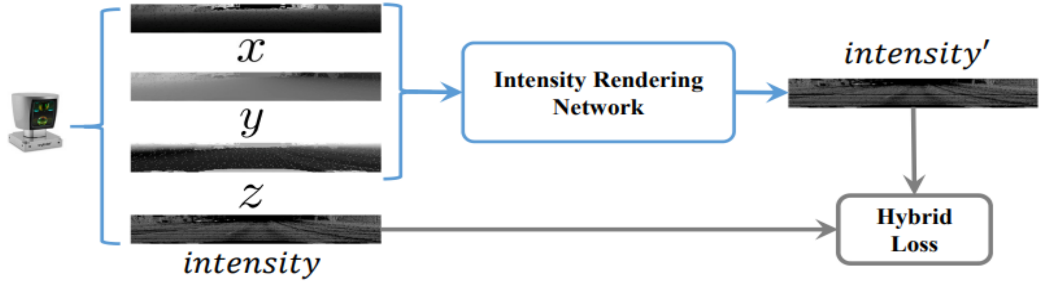




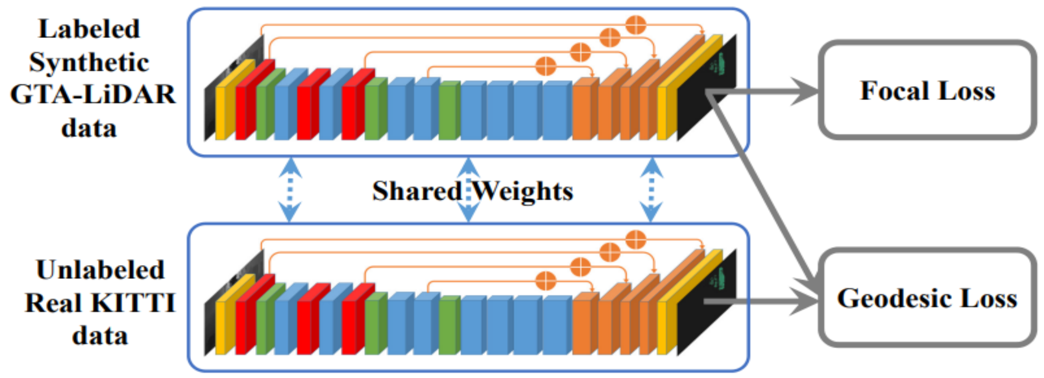
1. The local surface geometry around each point is projected to a virtual tangent plane, defining a set of tangent images.
2. Every tangent image is treated as a regular 2D grid that supports planar convolution.
3. Tangent convolutions are directly operated on the surface geometry.

Tatarchenko, Maxim, et al. "Tangent convolutions for dense prediction in 3d." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

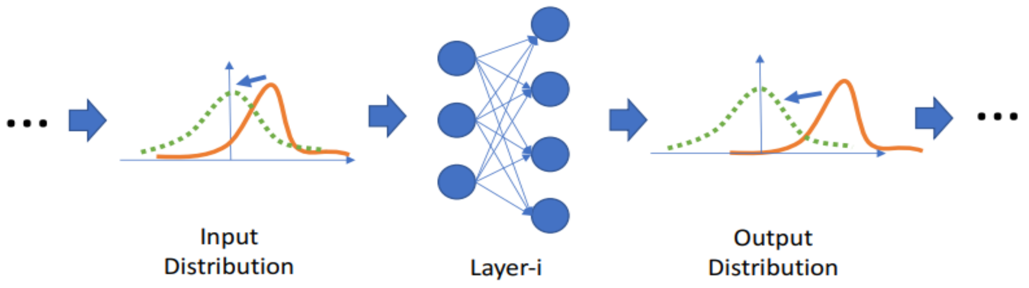
Projection-based methods: Spherical representation (1)



(a) Pre-training: Learned Intensity Rendering

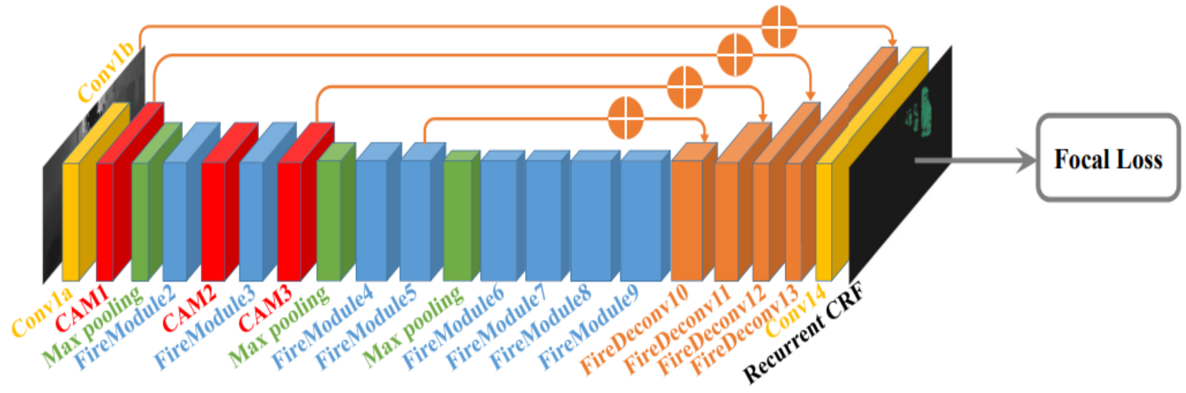


(b) Training: Geodesic Correlation Alignment



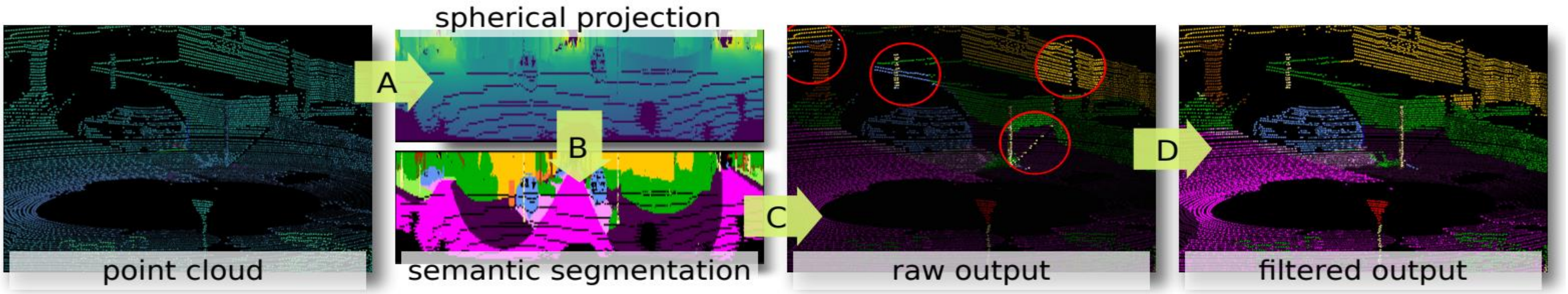
(c) Post-training: Progressive Domain Calibration

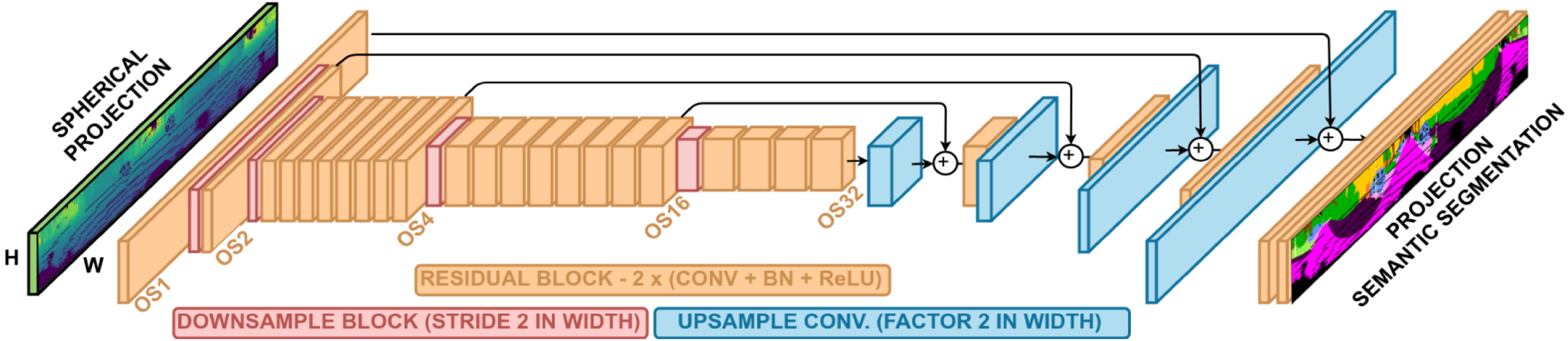
1. Improved architecture over SqueezeSeg over training loss, batch normalization, and extra input channel.
2. Domain adaptation training is exploited to allow generalization over synthetic data (GTA-V).
3. Pipeline comprises learned intensity rendering, geodesic correlation alignment and progressive domain calibration.



Wu, Bichen, et al. "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud." *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019.

Projection-based methods: Spherical representation (2a)





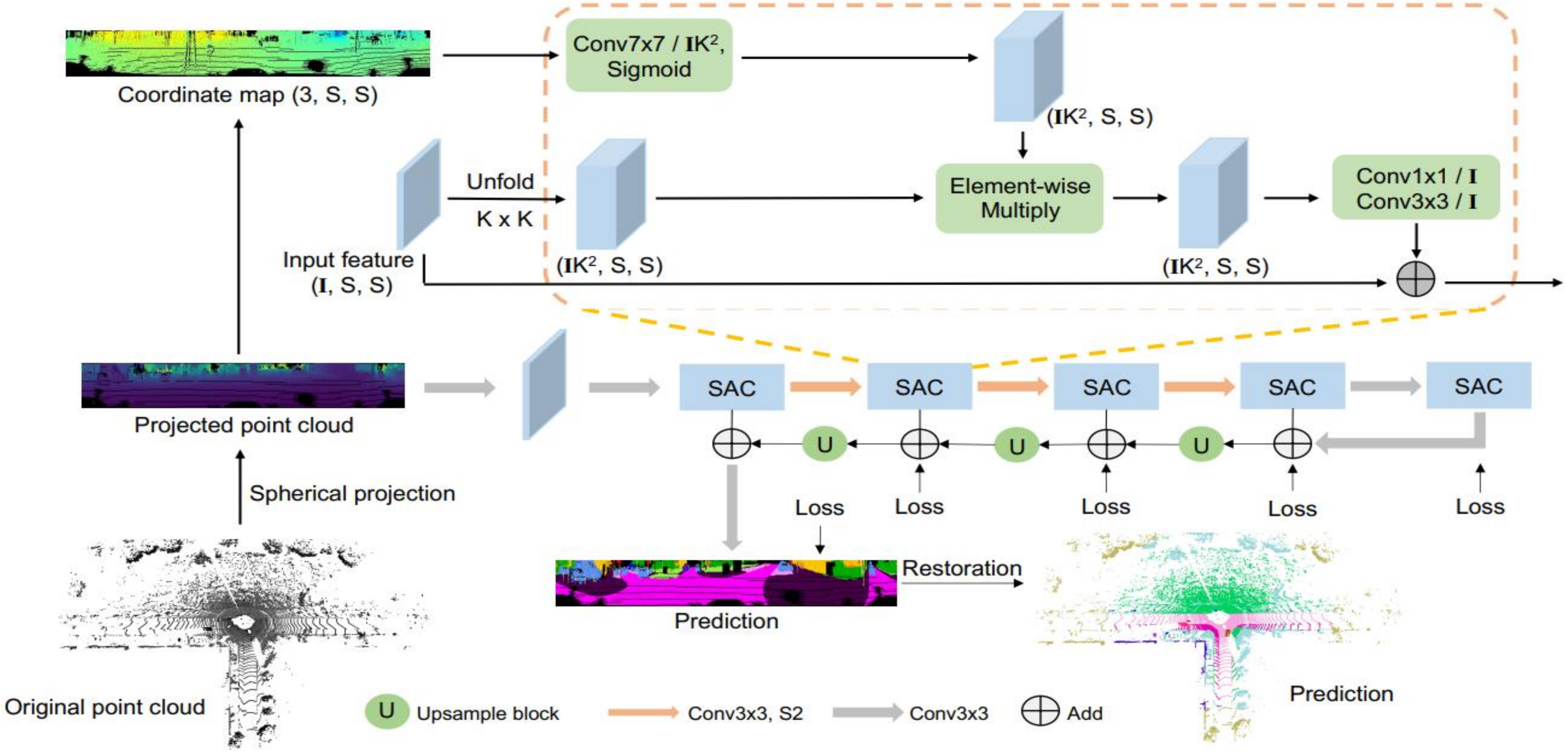
Milioto, Andres, et al. "Rangenet++: Fast and accurate lidar semantic segmentation." *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019.

Projection-based methods: Spherical representation (2b)

IoU *(SemanticKitti)*

Approach	Size	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean IoU	Scans/sec
Pointnet [14]		46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.6	2
Pointnet++ [15]		53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1	0.1
SPGraph [10]	50000pts	68.3	0.9	4.5	0.9	0.8	1.0	6.0	0.0	49.5	1.7	24.2	0.3	68.2	22.5	59.2	27.2	17.0	18.3	10.5	20.0	0.2
SPLATNet [19]		66.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.4	0.8	41.5	0.0	68.7	27.8	72.3	35.9	35.8	13.8	0.0	22.8	1
TangentConv [20]		86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2	35.9	0.3
SqueezeSeg [21]		68.8	16.0	4.1	3.3	3.6	12.9	13.1	0.9	85.4	26.9	54.3	4.5	57.4	29.0	60.0	24.3	53.7	17.5	24.5	29.5	**66**
SqueezeSeg-CRF [21]		68.3	18.1	5.1	4.1	4.8	16.5	17.3	1.2	84.9	28.4	54.7	04.6	61.5	29.2	59.6	25.5	54.7	11.2	36.3	30.8	55
SqueezeSegV2 [22]	64 × 2048 px	81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	39.7	50
SqueezeSegV2-CRF [22]		82.7	21.0	22.6	14.5	15.9	20.2	24.3	2.9	88.5	42.4	65.5	18.7	73.8	41.0	68.5	36.9	58.9	12.9	41.0	39.6	40
RangeNet21 [Ours]		85.4	26.2	26.5	18.6	15.6	31.8	33.6	4.0	91.4	57.0	74.0	26.4	81.9	52.3	77.6	48.4	63.6	36.0	50.0	47.4	20
RangeNet53 [Ours]	64 × 2048 px	86.4	24.5	32.7	25.5	22.6	36.2	33.6	4.7	**91.8**	64.8	74.6	**27.9**	84.1	55.0	78.3	50.1	64.0	38.9	52.2	49.9	13
	64 × 1024 px	84.6	20.0	25.3	24.8	17.3	27.5	27.7	7.1	90.4	51.8	72.1	22.8	80.4	50.0	75.1	46.0	62.7	33.4	43.4	45.4	25
	64 × 512 px	81.0	9.9	11.7	19.3	7.9	16.8	25.8	2.5	90.1	49.9	69.4	2.0	76.0	45.5	74.2	38.8	62.7	25.5	38.1	39.3	52
RangeNet53++ [Ours+kNN]	64 × 2048 px	**91.4**	**25.7**	**34.4**	**25.7**	**23.0**	**38.3**	**38.8**	**4.8**	**91.8**	**65.0**	**75.2**	27.8	**87.4**	**58.6**	**80.5**	**55.1**	**64.6**	**47.9**	**55.9**	**52.2**	12
	64 × 1024 px	90.3	20.6	27.1	25.2	17.6	29.6	34.2	7.1	90.4	52.3	72.7	22.8	83.9	53.3	77.7	52.5	63.7	43.8	47.2	48.0	21
	64 × 512 px	87.4	9.9	12.4	19.6	7.9	18.1	29.5	2.5	90.0	50.7	70.0	2.0	80.2	48.9	77.1	45.7	64.1	37.1	42.0	41.9	38

Projection-based methods: Spherical representation (3a)

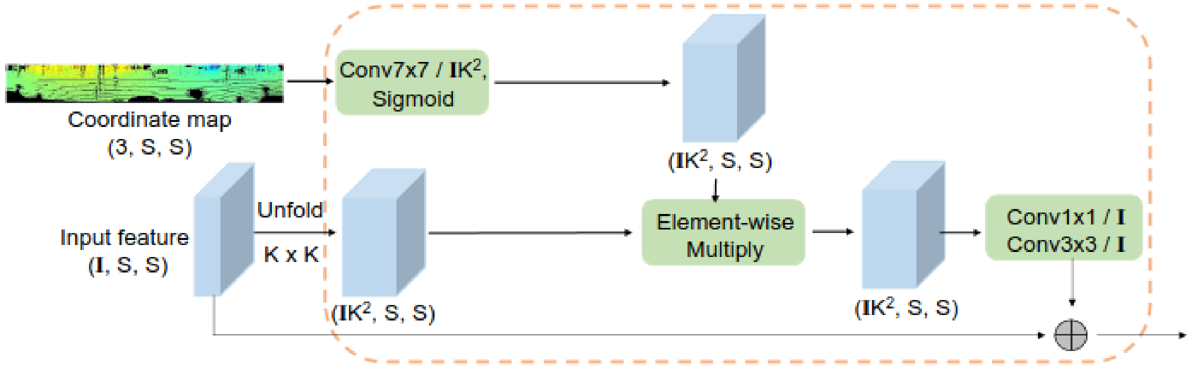


Xu, Chenfeng, et al. "Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*. Springer International Publishing, 2020.

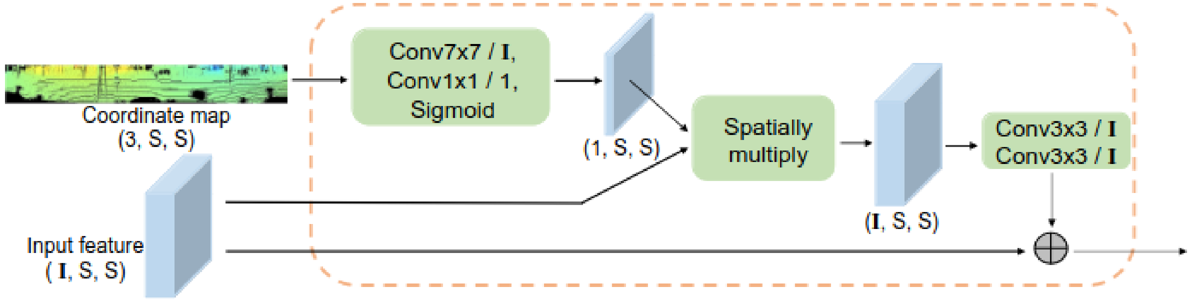
Projection-based methods: Spherical representation (3b)

Spatially Adaptive Convolution (SAC) is spatially-adaptive, since W depends on the location (p, q), and content-aware since W is a function of the raw input X0.

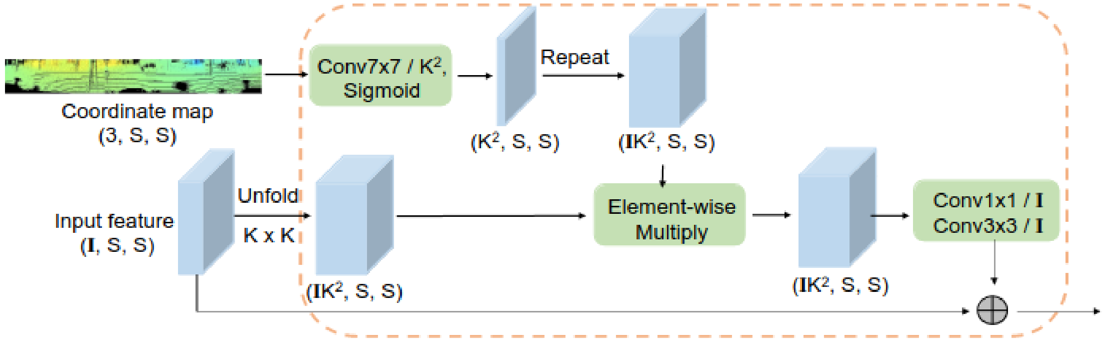
$$Y[m,p,q] = \sigma\Big(\sum_{i,j,n} W(X_0)[m,n,p,q,i,j] \times X[n,p+\hat{i},q+\hat{j}]\Big).$$



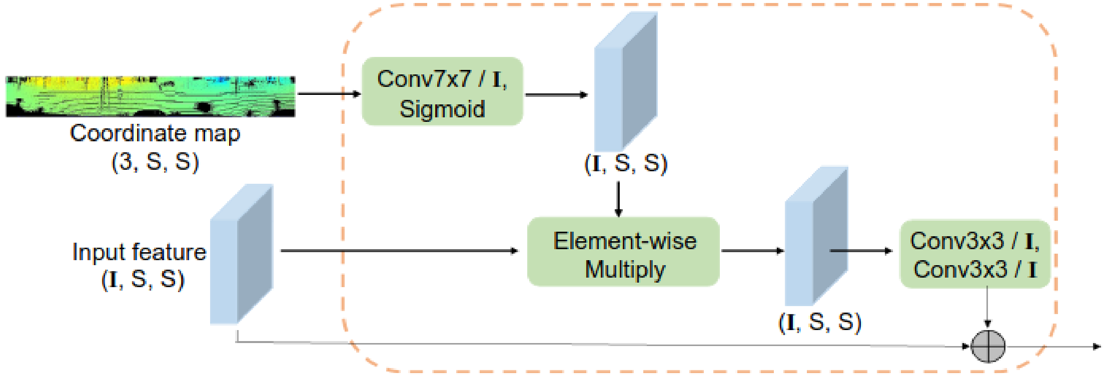
(a) SAC-ISK



(b) SAC-S



(c) SAC-SK



(d) SAC-IS

Projection-based methods: Spherical representation (3c)

$$L = \sum_{i=1}^{5} \frac{-\sum_{H_i,W_i} \sum_{c=1}^{C} w_c \cdot y_c \cdot log(\hat{y}_c)}{H_i \times W_i}.$$

Multi-layer Cross Entropy Loss

1. During training, from stage1 to stage5, a prediction layer at each stage's output is added
2. For each output, the ground truth label map is downsampled by 1x, 2x, 4x, 8x, and 10x, and the maps are used to train the output of stage1 to stage5, respectively
3. wc is a normalized factor, Hi and Wi are the height and width of the output in i-th stage, yc is the prediction for the c-th class in each pixel and ˆyc is the label
4. The intermediate supervisions guide the model to form features with more semantic meaning

Projection-based methods: Spherical representation (3d)

IoU
(SemanticKitti)

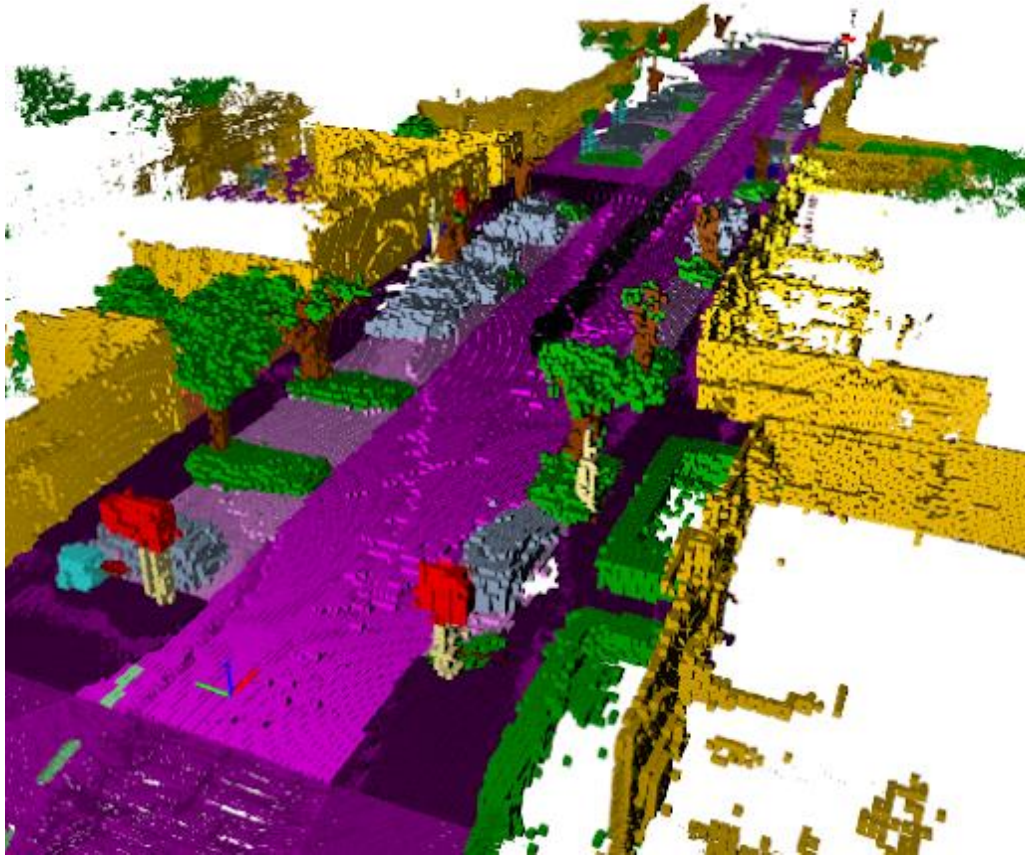
Method	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	Motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic-sign	mean IoU	Scans/sec
PNet [35]	46.3	1.3	0.3	0.1	0.8	0.2	0.2	0.0	61.6	15.8	35.7	1.4	41.4	12.9	31.0	4.6	17.6	2.4	3.7	14.6	2
PNet++ [36]	53.7	1.9	0.2	0.9	0.2	0.9	1.0	0.0	72.0	18.7	41.8	5.6	62.3	16.9	46.5	13.8	30.0	6.0	8.9	20.1	0.1
SPGraph [22]	68.3	0.9	4.5	0.9	0.8	1.0	6.0	0.0	49.5	1.7	24.2	0.3	68.2	22.5	59.2	27.2	17.0	18.3	10.5	20.0	0.2
SPLAT [43]	66.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	70.4	0.8	41.5	0.0	68.7	27.8	72.3	35.9	35.8	13.8	0.0	22.8	1
TgConv [46]	86.8	1.3	12.7	11.6	10.2	17.1	20.2	0.5	82.9	15.2	61.7	9.0	82.8	44.2	75.5	42.5	55.5	30.2	22.2	35.9	0.3
RLNet [15]	94.0	19.8	21.4	42.7	38.7	47.5	48.8	4.6	90.4	56.9	67.9	15.5	81.1	49.7	78.3	60.3	59.0	44.2	38.1	50.3	22
SSG [56]	68.8	16.0	4.1	3.3	3.6	12.9	13.1	0.9	85.4	26.9	54.3	4.5	57.4	29.0	60.0	24.3	53.7	17.5	24.5	29.5	**65**
SSG‡ [56]	68.3	18.1	5.1	4.1	4.8	16.5	17.3	1.2	84.9	28.4	54.7	4.6	61.5	29.2	59.6	25.5	54.7	11.2	36.3	30.8	53
SSGV2 [58]	81.8	18.5	17.9	13.4	14.0	20.1	25.1	3.9	88.6	45.8	67.6	17.7	73.7	41.1	71.8	35.8	60.2	20.2	36.3	39.7	50
SSGV2‡ [58]	82.7	21.0	22.6	14.5	15.9	20.2	24.3	2.9	88.5	42.4	65.5	18.7	73.8	41.0	68.5	36.9	58.9	12.9	41.0	39.6	39
RGN21 [30]	85.4	26.2	26.5	18.6	15.6	31.8	33.6	4.0	91.4	57.0	74.0	26.4	81.9	52.3	77.6	48.4	63.6	36.0	50.0	47.4	20
RGN53 [30]	86.4	24.5	32.7	25.5	22.6	36.2	33.6	4.7	**91.8**	64.8	74.6	**27.9**	84.1	55.0	78.3	50.1	64.0	38.9	52.2	49.9	12
RGN53* [30]	91.4	25.7	34.4	25.7	23.0	38.3	38.8	4.8	**91.8**	**65.0**	**75.2**	27.8	87.4	58.6	80.5	55.1	64.6	47.9	55.9	52.2	11
SSGV3-21	84.6	31.5	32.4	11.3	20.9	39.4	36.1	**21.3**	90.8	54.1	72.9	23.9	81.1	50.3	77.6	47.7	63.9	36.1	51.7	48.8	16
SSGV3-53	87.4	35.2	33.7	29.0	31.9	41.8	39.1	20.1	**91.8**	63.5	74.4	27.2	85.3	55.8	79.4	52.1	64.7	38.6	53.4	52.9	7
SSGV3-21*	89.4	33.7	34.9	11.3	21.5	42.6	44.9	21.2	90.8	54.1	73.3	23.2	84.8	53.6	80.2	53.3	64.5	46.4	57.6	51.6	15
SSGV3-53*	92.5	**38.7**	**36.5**	29.6	33.0	45.6	46.2	20.1	91.7	63.4	74.8	26.4	**89.0**	**59.4**	**82.0**	58.7	65.4	**49.6**	**58.9**	**55.9**	6

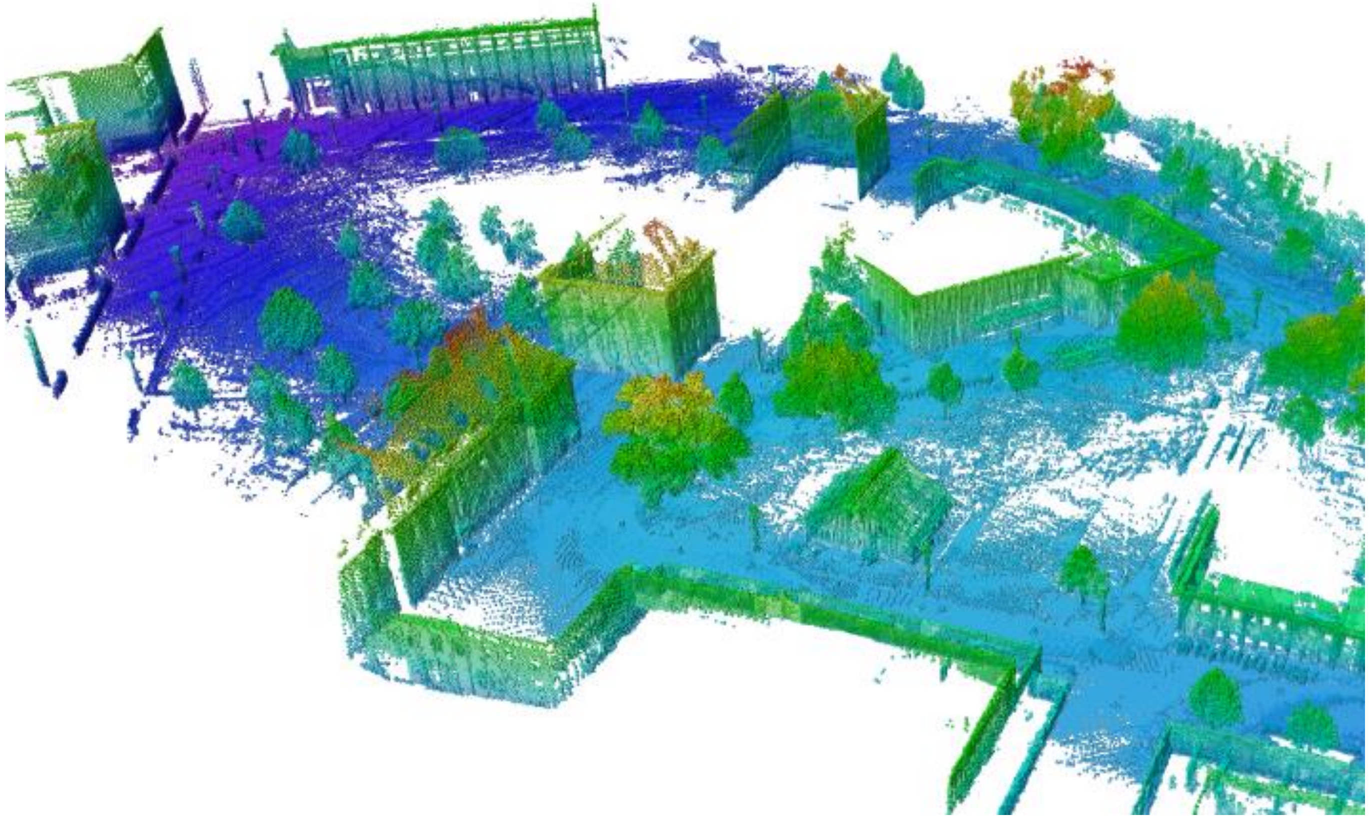
Projection-based methods: Comparison (early 2022)

Method	Year	Dataset	Performance			Contribution
			OA	mAcc	mIoU	
MVCNN [43]	2015	ModelNet40	90.1%	-	-	The first multiview CNN
SnapNet [48]	2017	Sun RGB-D	-	67.4%	-	Generate RGB and depth views by 2D image views
		Semantic3D	88.6%	70.8%	59.1%	
SnapNet-R [49]	2017	Sun RGB-D	78.1%	-	38.3%	Improvements to SnapNet
GVCNN [44]	2018	ModelNet40	93.1%	-	-	Grouping module to learn the connections and differences between views
SqueezeSeg [50]	2018	KITTI	-	-	29.5%	Data conversion from 3D to 2D using spherical projection
SqueezeSegV2 [52]	2018	KITTI	-	-	39.7%	Introducing a context aggregation module to SqueezeSeg
PVRNet [45]	2019	ModelNet40	93.6%	-	-	Consider relationships between points and views, and fuse features
RangeNet++ [46]	2019	KITTI	-	-	52.2%	GPU-accelerated postprocessing +RangNet++
SqueezeSegV3 [53]	2020	SemanticKITTI	-	-	55.9%	Proposing the spatially adaptive and context-aware convolution
Robert et al. [47]	2022	S3DIS	-	-	74.4%	Introducing an attention scheme for multiview image-based methods
		ScanNet	-	-	71.0%	

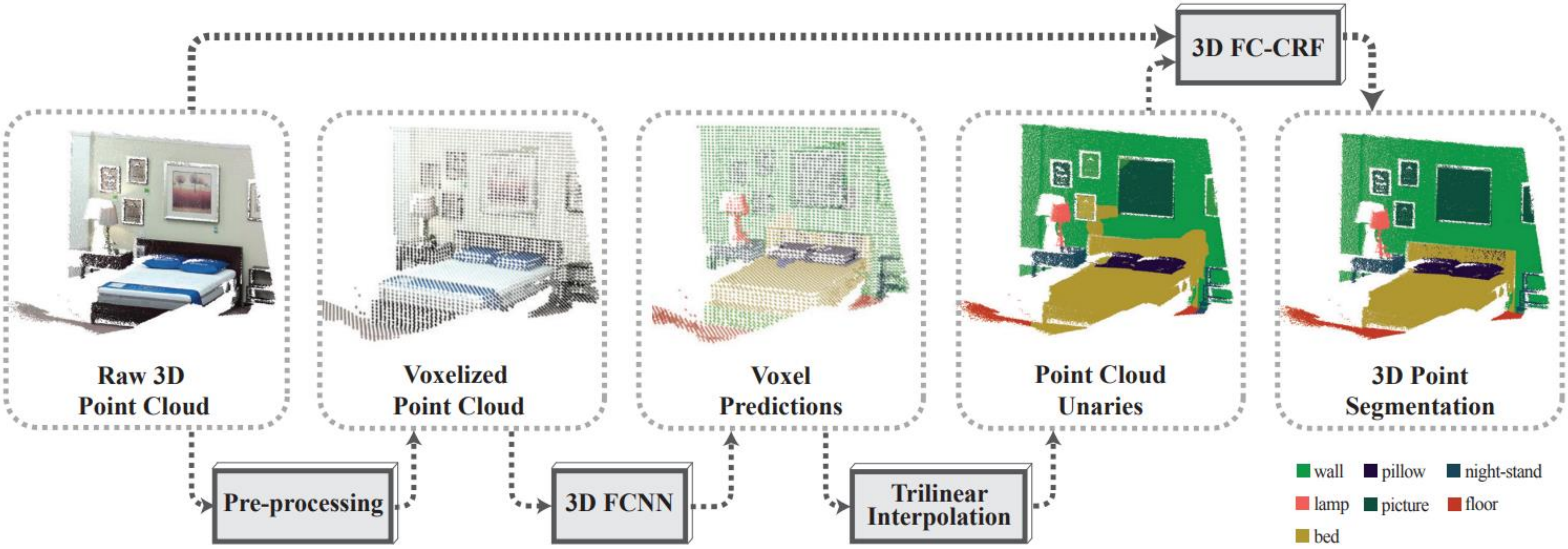
Discretization-based methods

These methods usually convert a point cloud into a dense/sparse discrete representation, such as volumetric and sparse permutohedral lattices.



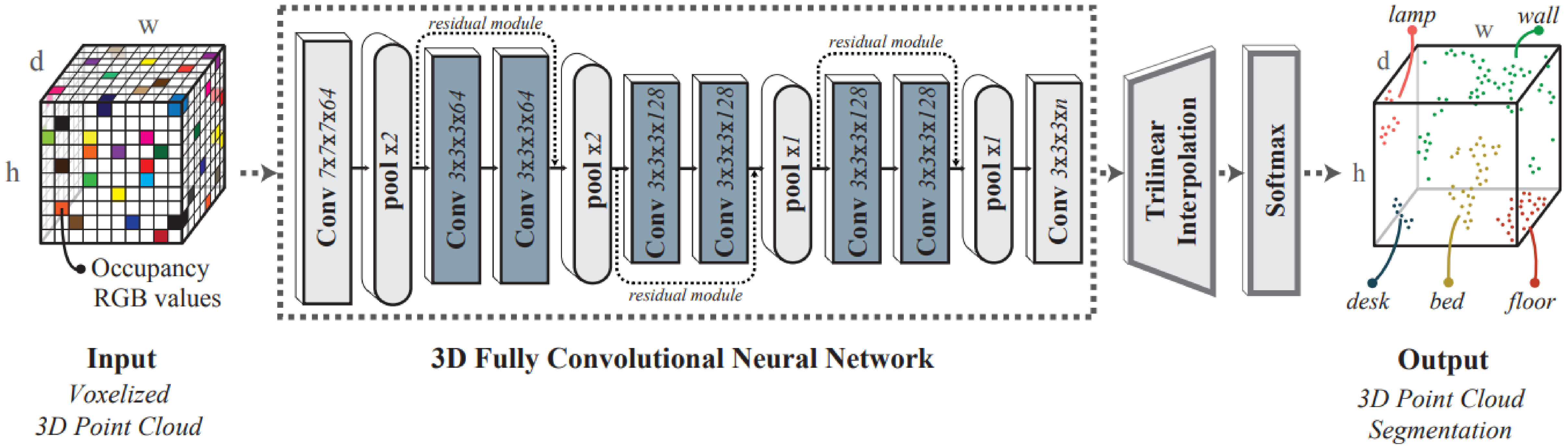


Discretization-based methods: Dense representation (1a)



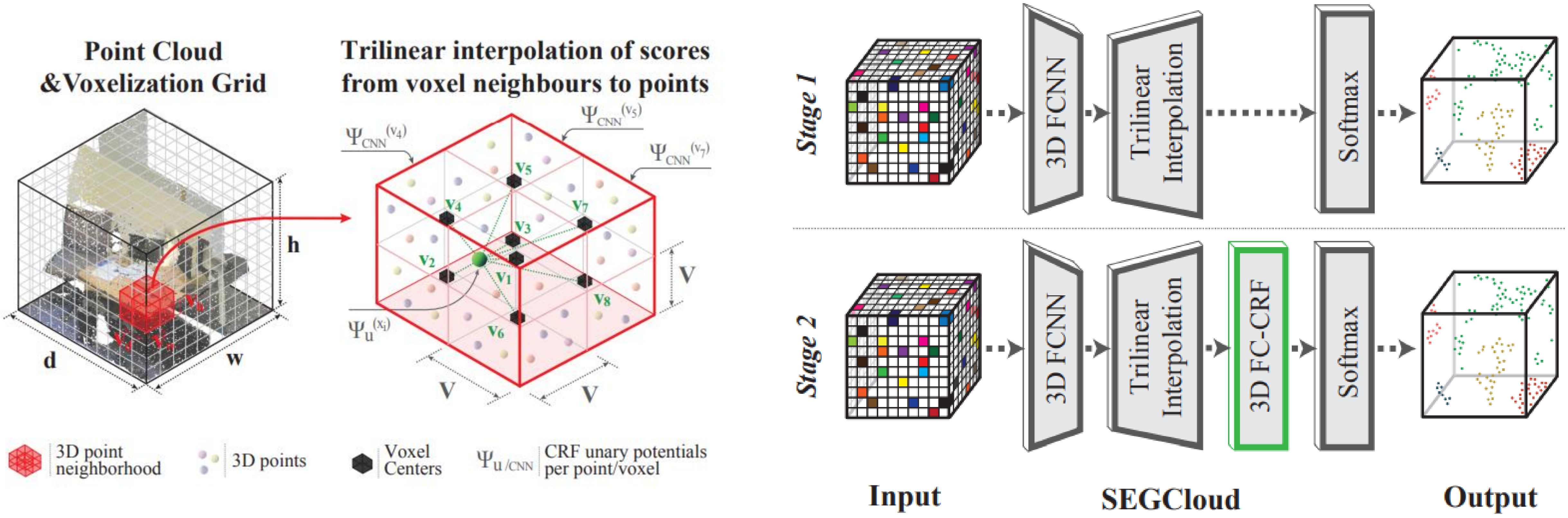
Tchapmi, Lyne, et al. "Segcloud: Semantic segmentation of 3d point clouds." *2017 international conference on 3D vision (3DV)*. IEEE, 2017.

Discretization-based methods: Dense representation (1b)



Tchapmi, Lyne, et al. "Segcloud: Semantic segmentation of 3d point clouds." *2017 international conference on 3D vision (3DV)*. IEEE, 2017.

Discretization-based methods: Dense representation (1c)



Tchapmi, Lyne, et al. "Segcloud: Semantic segmentation of 3d point clouds." *2017 international conference on 3D vision (3DV)*. IEEE, 2017.

Discretization-based methods: Dense representation (1d)

Table 1: **Results on the Semantic3D.net Benchmark (*reduced-8* challenge)**

Method	man-made terrain	natural terrain	high vegetation	low vegetation	buildings	hard scape	scanning artefacts	cars	mIOU	mAcc[3]
TMLC-MSR [27]	**89.80**	74.50	53.70	26.80	88.80	18.90	36.40	44.70	54.20	68.95
DeePr3SS [41]	85.60	**83.20**	74.20	32.40	89.70	18.50	25.10	59.20	58.50	88.90
SnapNet [6]	82.00	77.30	79.70	22.90	**91.10**	18.40	**37.30**	**64.40**	59.10	70.80
3D-FCNN-TI(Ours)	84.00	71.10	77.00	31.80	89.90	27.70	25.20	59.00	58.20	69.86
SEGCloud (Ours)	83.90	66.00	**86.00**	**40.50**	**91.10**	**30.90**	27.50	64.30	**61.30**	**73.08**

Table 2: **Results on the Large-Scale 3D Indoor Spaces Dataset (S3DIS)**

Method	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter	**mIOU**	**mAcc**
PointNet [53]	88.80	**97.33**	69.80	**0.05**	3.92	**46.26**	10.76	52.61	58.93	40.28	5.85	**26.38**	33.22	41.09	48.98
3D-FCNN-TI(Ours)	**90.17**	96.48	**70.16**	0.00	11.40	33.36	21.12	**76.12**	70.07	57.89	37.46	11.16	**41.61**	47.46	54.91
SEGCloud (Ours)	90.06	96.05	69.86	0.00	**18.37**	38.35	**23.12**	75.89	**70.40**	**58.42**	**40.88**	12.96	41.60	**48.92**	**57.35**

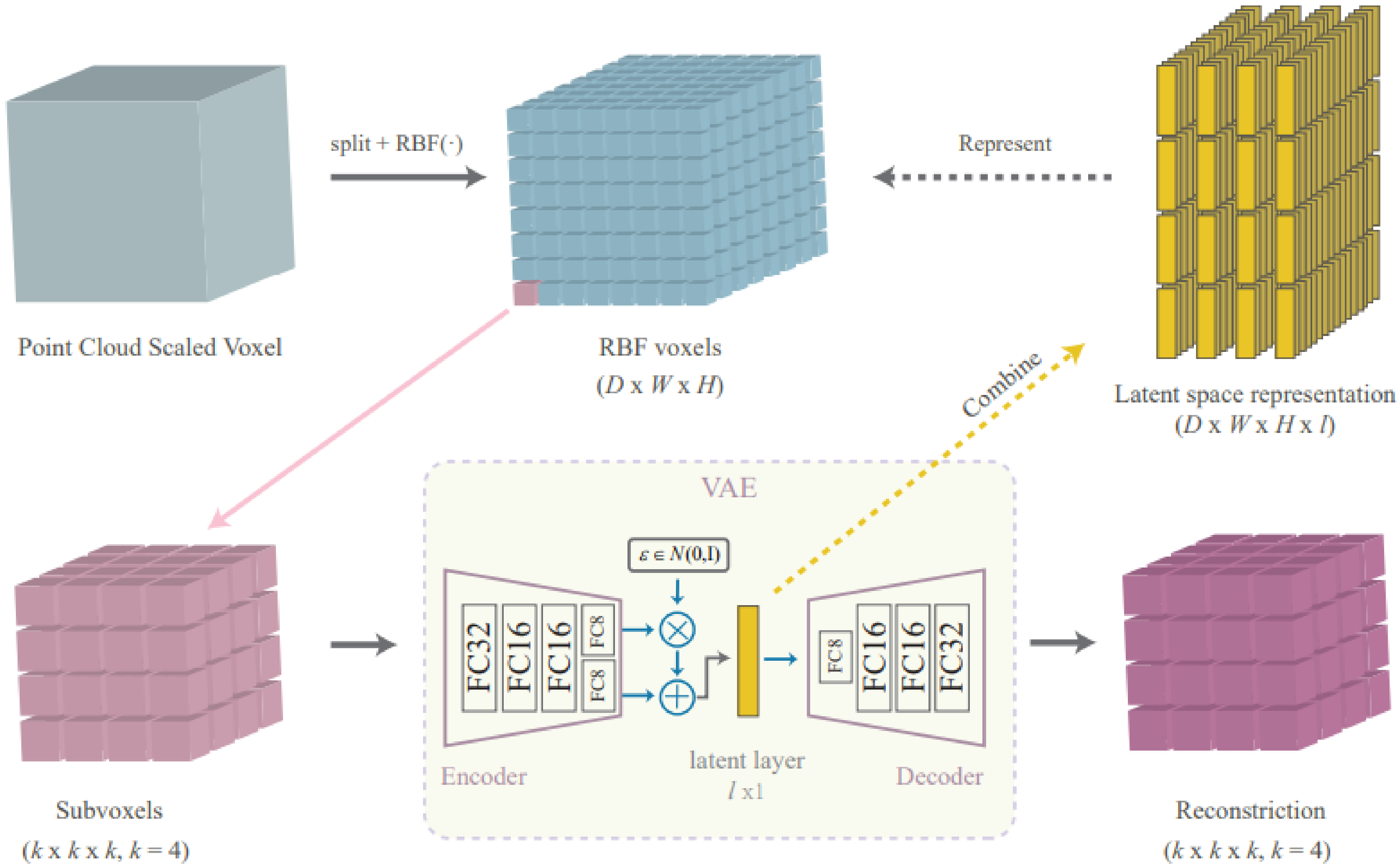
Table 3: **Results on the NYUV2 dataset**

Method	Bed	Objects	Chair	Furniture	Ceiling	Floor	Deco.	Sofa	Table	Wall	Window	Booksh.	TV	**mIOU**	**mAcc**	**glob Acc**
Couprie et al. [14]	38.1	8.7	34.1	42.4	62.6	87.3	40.4	24.6	10.2	86.1	15.9	13.7	6.0	-	36.2	52.4
Wang et al. [65]	47.6	12.4	23.5	16.7	68.1	84.1	26.4	39.1	35.4	65.9	52.2	45.0	32.4	-	42.2	-
Hermans et al. [29]	68.4	8.6	41.9	37.1	**83.4**	91.5	35.8	28.5	27.7	71.8	46.1	45.4	**38.4**	-	48.0	54.2
Wolf et al. [69]	74.56	17.62	62.16	47.85	82.42	**98.72**	26.36	**69.38**	**48.57**	83.65	25.56	**54.92**	31.05	39.51	55.6±0.2	64.9±0.3
3D-FCNN-TI(Ours)	69.3	**40.26**	**64.34**	**64.41**	73.05	95.55	21.15	55.51	45.09	**84.96**	20.76	42.24	23.95	42.13	53.9	**67.38**
SEGCloud (Ours)	**75.06**	39.28	62.92	61.8	69.16	95.21	**34.38**	62.78	45.78	78.89	**26.35**	53.46	28.5	**43.45**	**56.43**	66.82

Table 4: **Results on the KITTI dataset.**

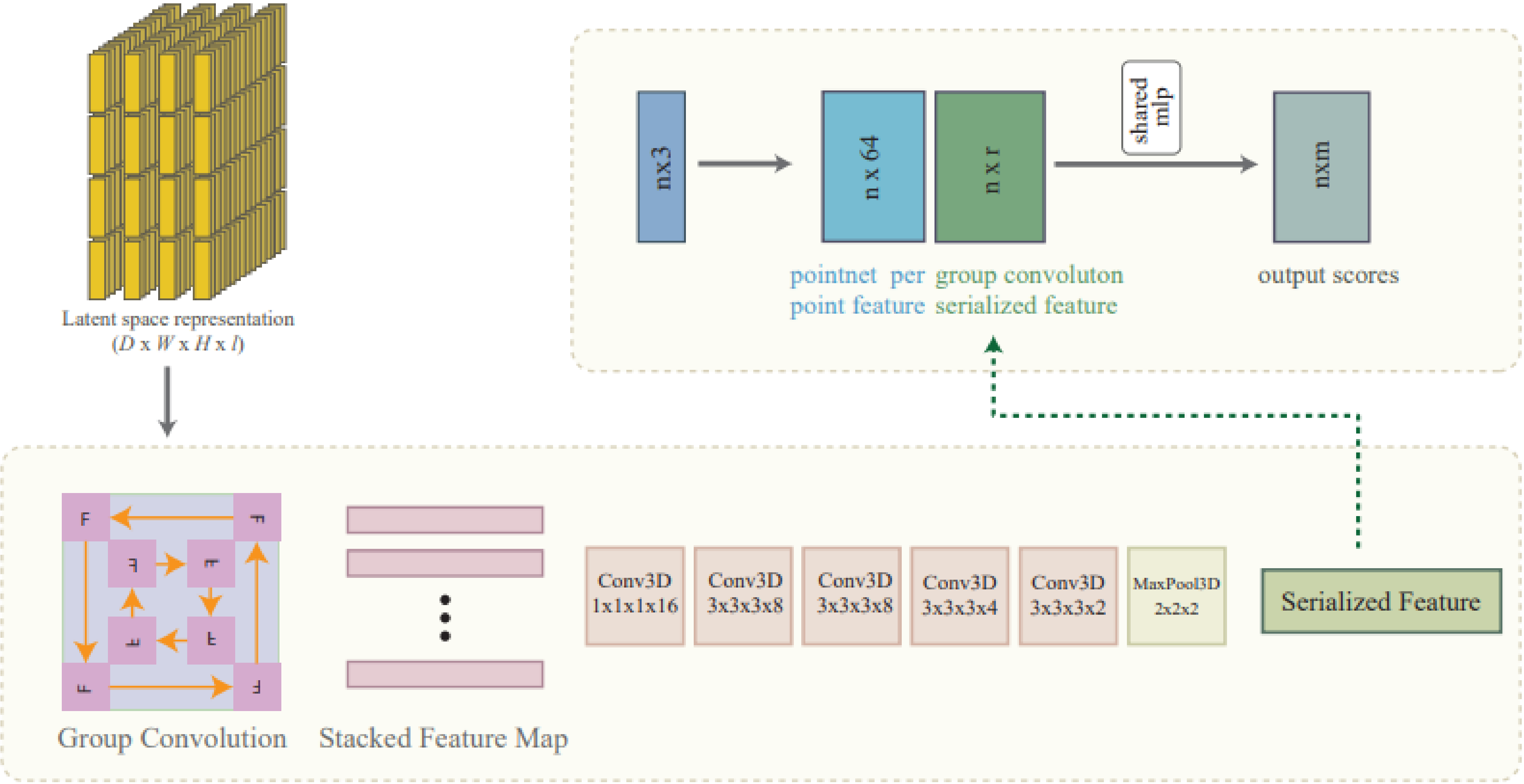
Method	building	sky	road	vegetation	sidewalk	car	pedestrian	cyclist	signage	fence	**mIOU**	**mAcc**
Zhang *et al.* [75]	**86.90**	-	89.20	55.00	26.20	50.0	49.00	**19.3**	**51.7**	**21.1**	-	**49.80**
3D-FCNN-TI(Ours)	85.83	-	**90.57**	**70.50**	25.56	65.68	46.35	7.78	28.40	4.51	35.65	47.24
SEGCloud (Ours)	85.86	-	88.84	68.73	**29.74**	**67.51**	**53.52**	7.27	39.62	4.05	**36.78**	49.46

Discretization-based methods: Dense representation (2a)



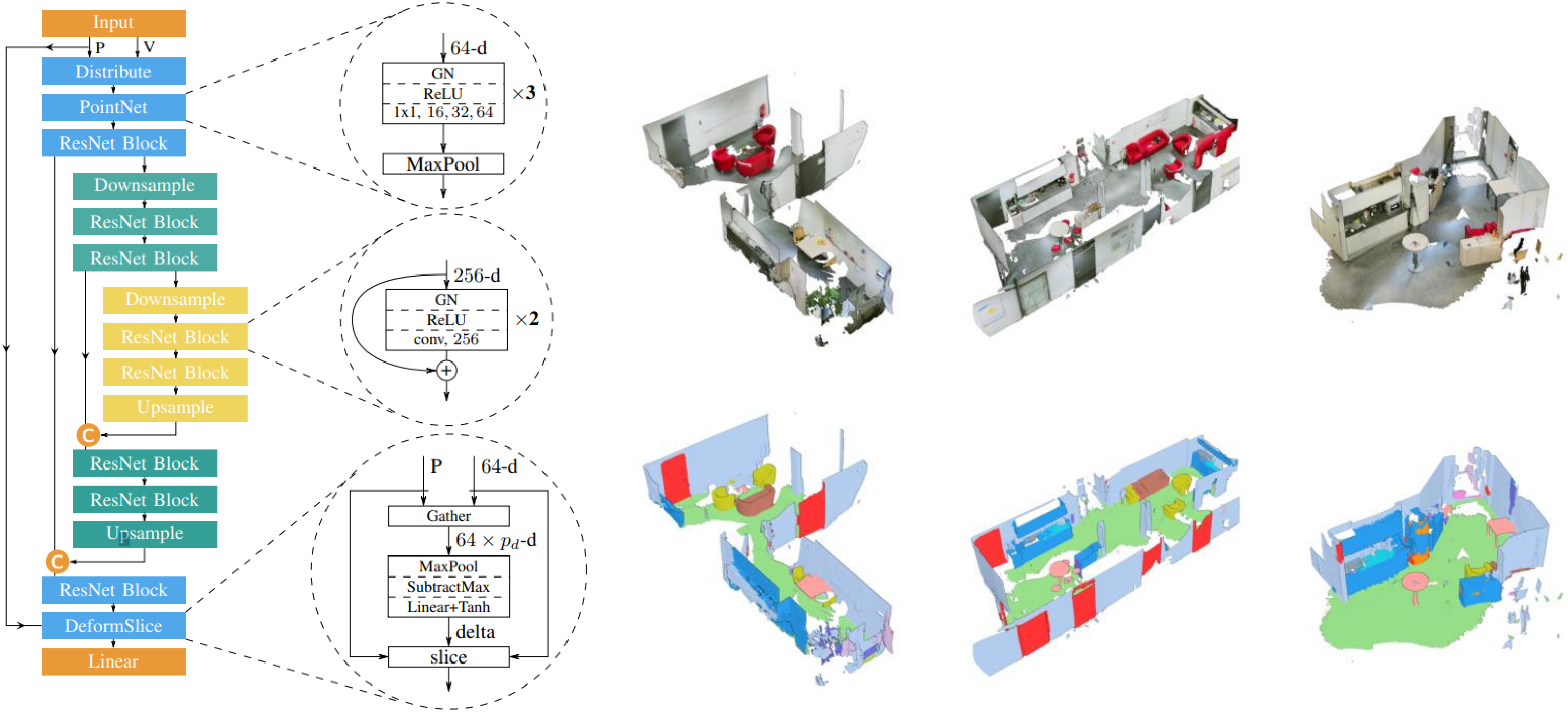
Meng, Hsien-Yu, et al. "Vv-net: Voxel vae net with group convolutions for point cloud segmentation." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

Discretization-based methods: Dense representation (2b)



Meng, Hsien-Yu, et al. "Vv-net: Voxel vae net with group convolutions for point cloud segmentation." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

Discretization-based methods: Sparse representation (1)

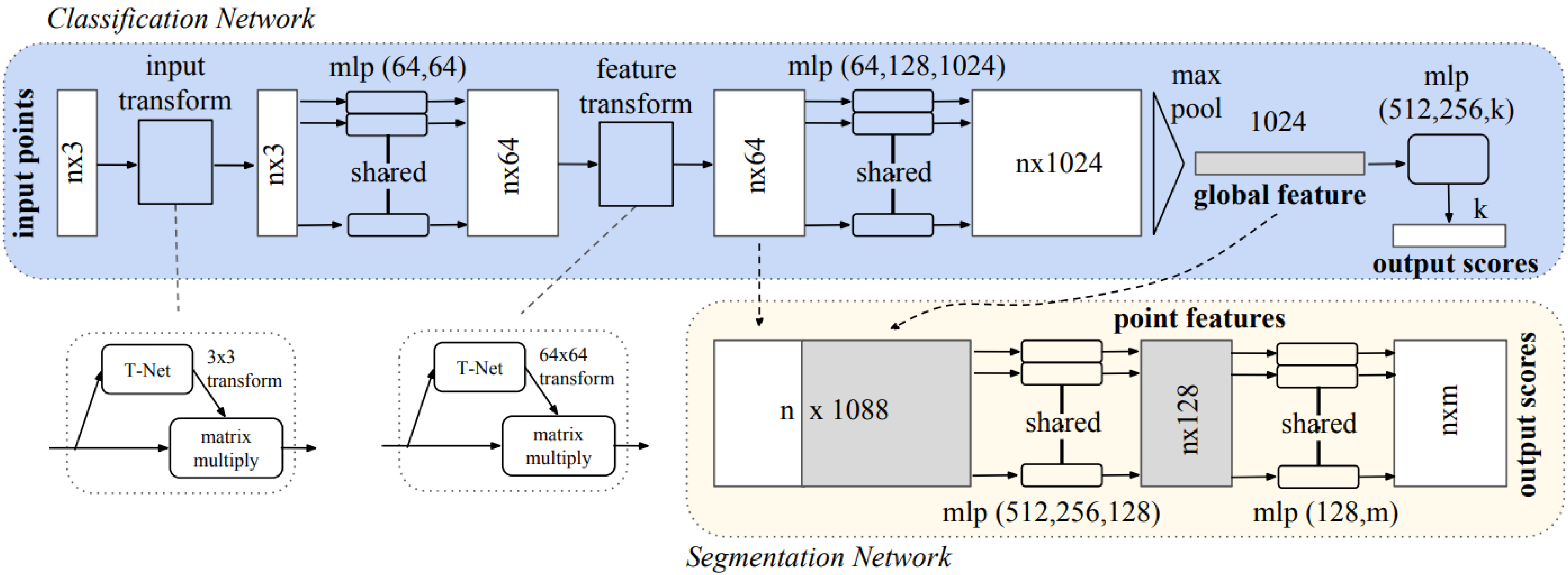


Rosu, Radu Alexandru, et al. "Latticenet: Fast point cloud segmentation using permutohedral lattices." *arXiv preprint arXiv:1912.05905* (2019).

Discretization-based methods: Comparison

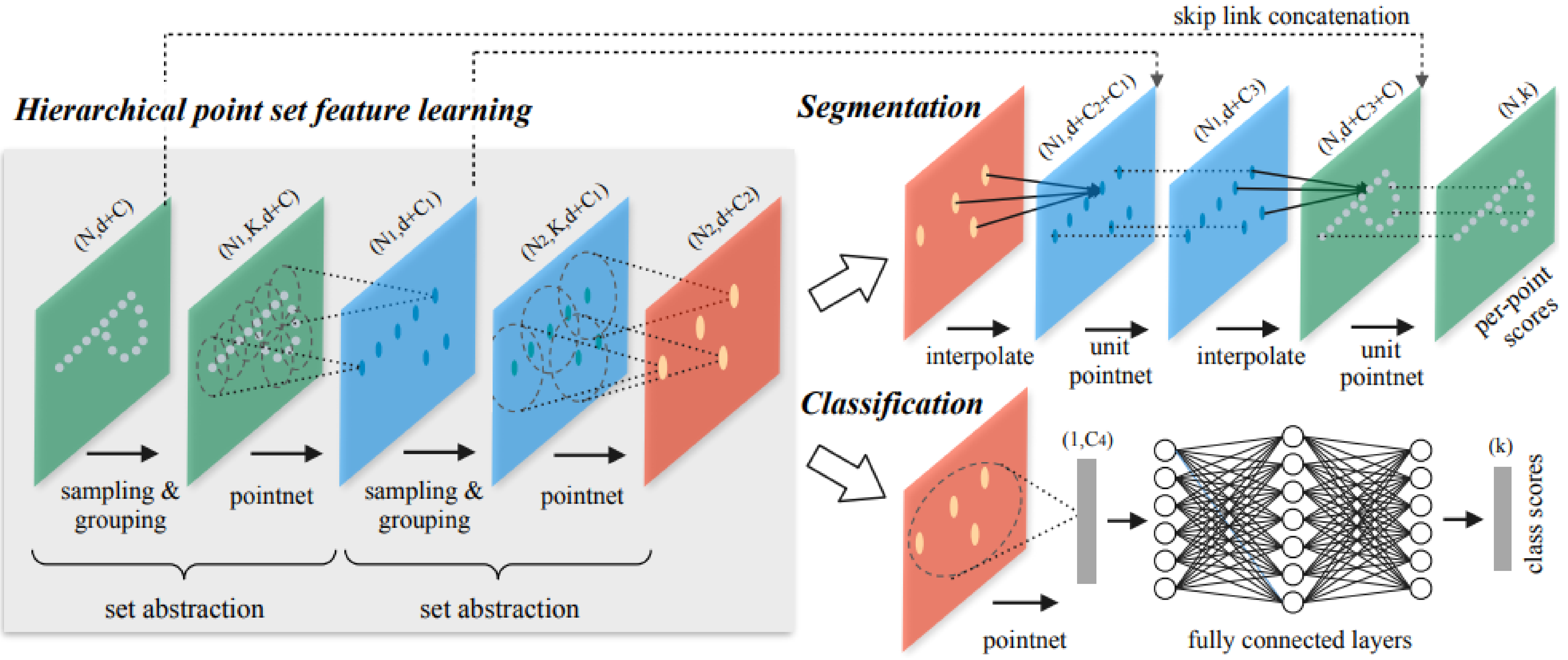
Method	Year	Dataset	Performance			Contribution
			OA	mAcc	mIoU	
VoxNet [55]	2015	ModelNet10	-	92.0%	-	The first method to process raw point clouds using voxelization
		ModelNet40	85.9%	83.0%	-	
SEGCloud [58]	2015	ShapeNet Part	-	-	79.4%	Combining 3DFCNN with fine representation using trilinear interpolation and conditional random field
		ScanNet	73.0%	-	-	
		S3DIS	-	57.4%	48.9%	
		Semantic3D	88.1%	73.1%	61.3%	
		KITTI	-	49.5%	36.8%	
OctNet [59]	2017	ModelNet10	90.0%	-	-	Divide the space into nonuniform voxels using unbalanced octrees
		ModelNet40	83.8%	-	-	
O-CNN [60]	2017	ModelNet40	90.2%	-	-	Making 3D-CNN feasible for high-resolution voxels
		ShapeNet Part	-	-	85.9%	
SPLATNet [56]	2018	ShapeNet Part	-	83.7%	-	Hierarchical and spatially aware feature learning
VV-Net [61]	2019	ShapeNet Part	-	-	87.4%	Using the radial basis function to compute the localized continuous representation within each voxel
		S3DIS	87.8%	-	78.2%	
LatticeNet [57]	2020	ShapeNet Part	-	83.9%	-	Proposing a novel slicing operator for computational efficiency
		ScanNet	-	-	64.0%	
		SemanticKITTI	-	-	52.9%	
PCSCNet [62]	2022	nuScenes	-	-	72.0%	Reducing the voxel discretization error
		SemanticKITTI	-	-	62.7%	
SIEV-Net [63]	2022	KITTI	-	-	62.6%	Effectively reduces loss of height information

Point-based methods: MLP



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.

Point-based methods: Neighboring Feature Pooling (1a)



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).

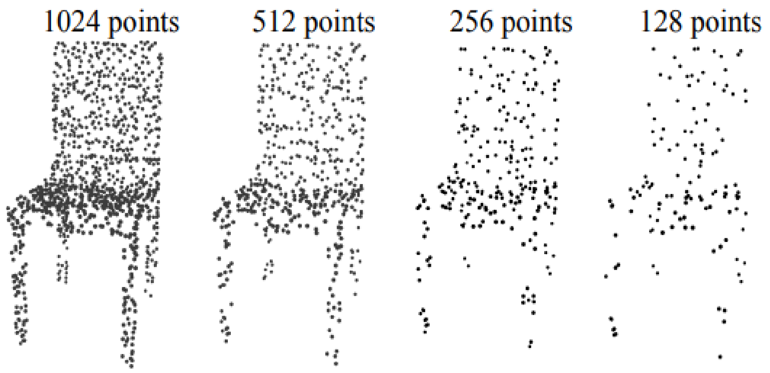
Point-based methods: Neighboring Feature Pooling (1b)

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

Table 1: MNIST digit classification.

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

Table 2: ModelNet40 shape classification.



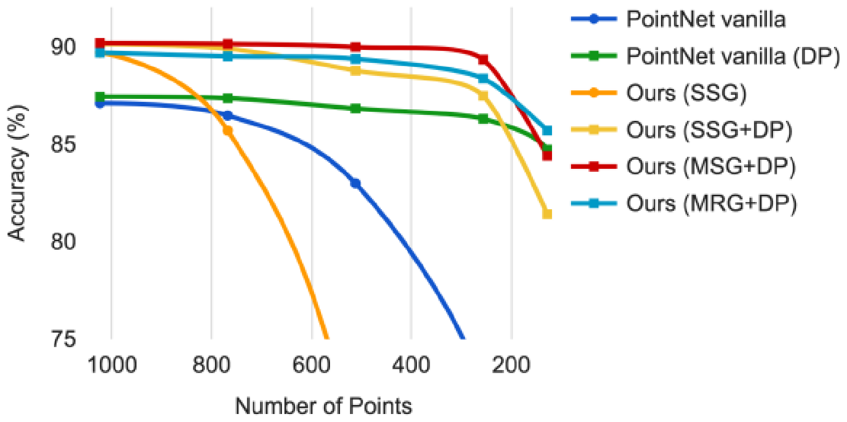


Figure 4: Left: Point cloud with random point dropout. Right: Curve showing advantage of our density adaptive strategy in dealing with non-uniform density. DP means random input dropout during training; otherwise training is on uniformly dense points. See Sec.3.3 for details.

Point-based methods: Neighboring Feature Pooling (2a)

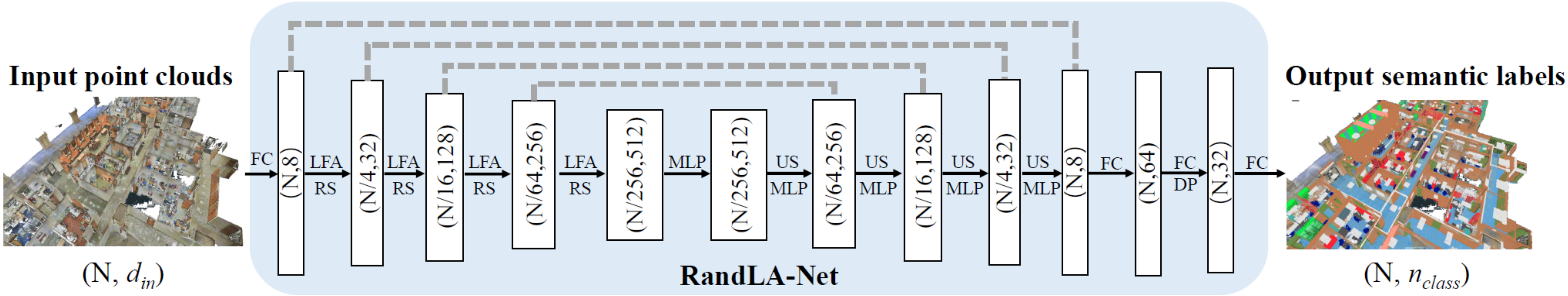
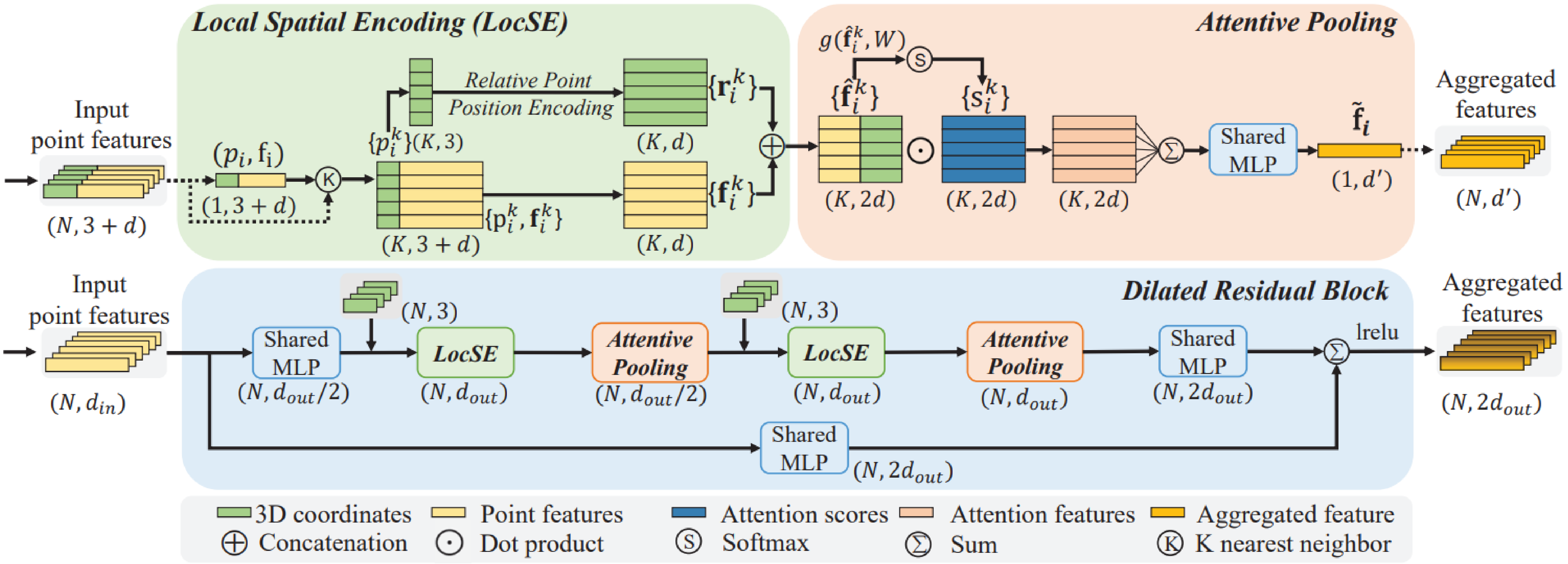


Figure 7. The detailed architecture of our RandLA-Net. (N, D) represents the number of points and feature dimension respectively. FC: Fully Connected layer, LFA: Local Feature Aggregation, RS: Random Sampling, MLP: shared Multi-Layer Perceptron, US: Up-sampling, DP: Dropout.

Hu, Qingyong, et al. "Randla-net: Efficient semantic segmentation of large-scale point clouds." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

Point-based methods: Neighboring Feature Pooling (2b)



Point-based methods: Neighboring Feature Pooling (2c)

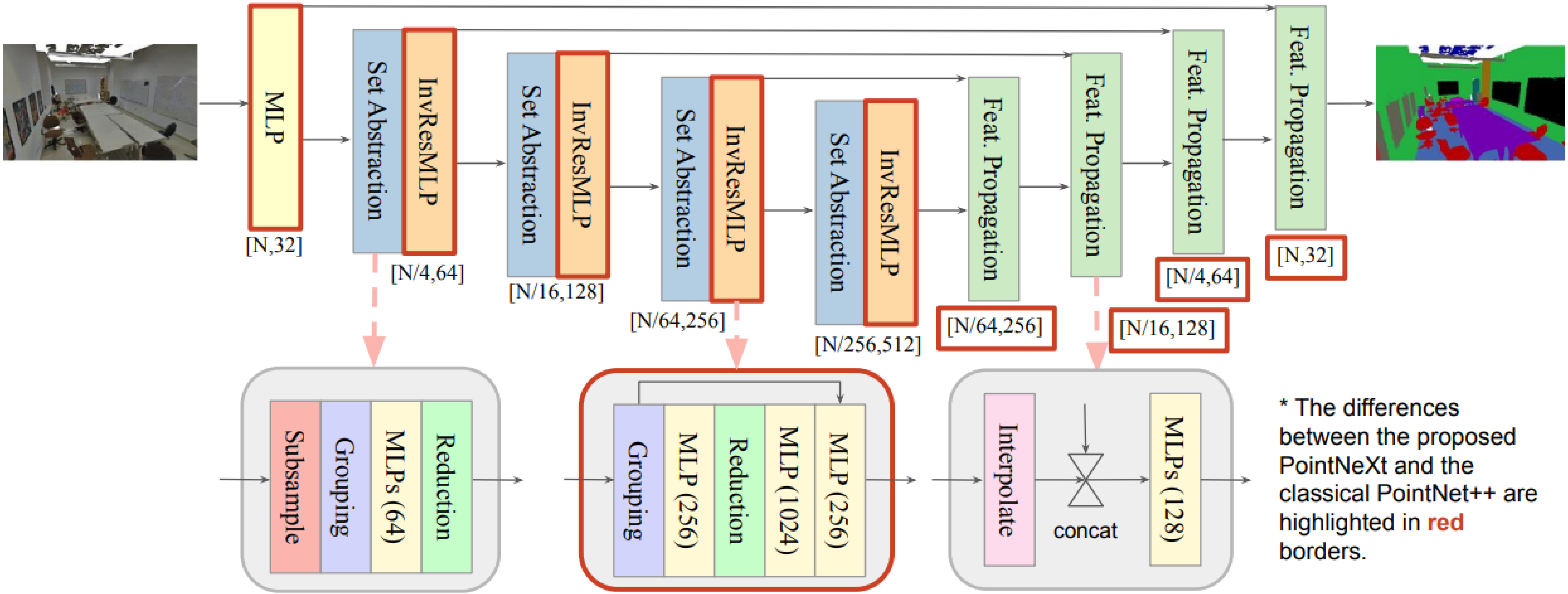
	mIoU (%)	OA (%)	man-made.	natural.	high veg.	low veg.	buildings	hard scape	scanning art.	cars
SnapNet_ [4]	59.1	88.6	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
SEGCloud [52]	61.3	88.1	83.9	66.0	86.0	40.5	91.1	30.9	27.5	64.3
RF_MSSF [53]	62.7	90.3	87.6	80.3	81.8	36.4	92.2	24.1	42.6	56.6
MSDeepVoxNet [46]	65.3	88.4	83.0	67.2	83.8	36.7	92.4	31.3	50.0	78.2
ShellNet [69]	69.3	93.2	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
GACNet [56]	70.8	91.9	86.4	77.7	**88.5**	**60.6**	94.2	37.3	43.5	77.8
SPG [26]	73.2	94.0	**97.4**	**92.6**	87.9	44.0	83.2	31.0	63.5	76.2
KPConv [54]	74.6	92.9	90.9	82.2	84.2	47.9	94.9	40.0	**77.3**	**79.7**
RandLA-Net (Ours)	**77.4**	**94.8**	95.6	91.4	86.6	51.5	**95.7**	**51.5**	69.8	76.8

Table 2. Quantitative results of different approaches on Semantic3D (reduced-8) [17]. Only the recent published approaches are compared. Accessed on 31 March 2020.

Methods	Size	mIoU(%)	Params(M)	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic-sign
PointNet [43]	50K pts	14.6	3	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7
SPG [26]		17.4	**0.25**	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8
SPLATNet [49]		18.4	0.8	64.6	39.1	0.4	0.0	58.3	58.2	0.0	0.0	0.0	0.0	71.1	9.9	19.3	0.0	0.0	0.0	23.1	5.6	0.0
PointNet++ [44]		20.1	6	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
TangentConv [51]		40.9	0.4	83.9	63.9	33.4	15.4	83.4	90.8	15.2	2.7	16.5	12.1	79.5	49.3	58.1	23.0	28.4	**8.1**	49.0	35.8	28.5
SqueezeSeg [58]	64*2048 pixels	29.5	1	85.4	54.3	26.9	4.5	57.4	68.8	3.3	16.0	4.1	3.6	60.0	24.3	53.7	12.9	13.1	0.9	29.0	17.5	24.5
SqueezeSegV2 [59]		39.7	1	88.6	67.6	45.8	17.7	73.7	81.8	13.4	18.5	17.9	14.0	71.8	35.8	60.2	20.1	25.1	3.9	41.1	20.2	36.3
DarkNet21Seg [3]		47.4	25	91.4	74.0	57.0	26.4	81.9	85.4	18.6	**26.2**	26.5	15.6	77.6	48.4	63.6	31.8	33.6	4.0	52.3	36.0	50.0
DarkNet53Seg [3]		49.9	50	**91.8**	74.6	64.8	**27.9**	84.1	86.4	25.5	24.5	32.7	22.6	78.3	50.1	64.0	36.2	33.6	4.7	55.0	38.9	52.2
RangeNet53++ [40]		52.2	50	**91.8**	**75.2**	**65.0**	27.8	**87.4**	91.4	25.7	25.7	**34.4**	23.0	80.5	55.1	64.6	38.3	38.8	4.8	**58.6**	47.9	**55.9**
RandLA-Net (Ours)	50K pts	**53.9**	1.24	90.7	73.7	60.3	20.4	86.9	**94.2**	**40.1**	26.0	25.8	**38.9**	**81.4**	**61.3**	**66.8**	**49.2**	**48.2**	7.2	56.3	**49.2**	47.7

Table 3. Quantitative results of different approaches on SemanticKITTI [3]. Only the recent published methods are compared and all scores are obtained from the online single scan evaluation track. Accessed on 31 March 2020.

Point-based methods: Neighboring Feature Pooling (3)



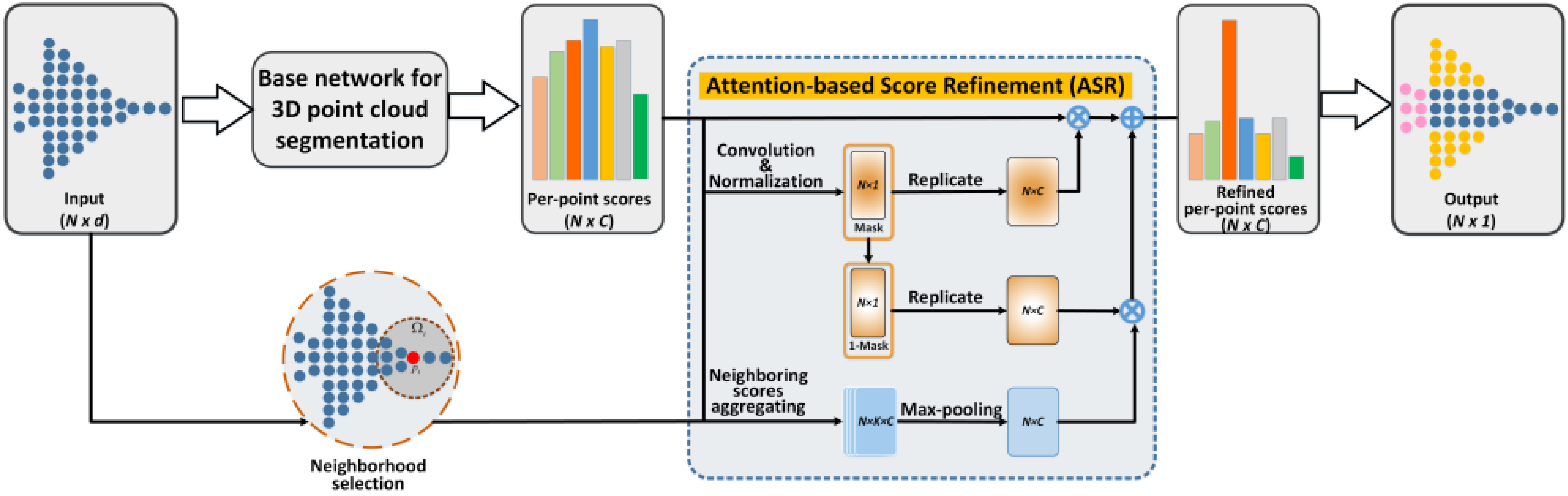
* The differences between the proposed PointNeXt and the classical PointNet++ are highlighted in **red** borders.

Qian, Guocheng, et al. "Pointnext: Revisiting pointnet++ with improved training and scaling strategies." *Advances in Neural Information Processing Systems* 35 (2022): 23192-23204.

Point-based methods: MLP / NFP Comparison

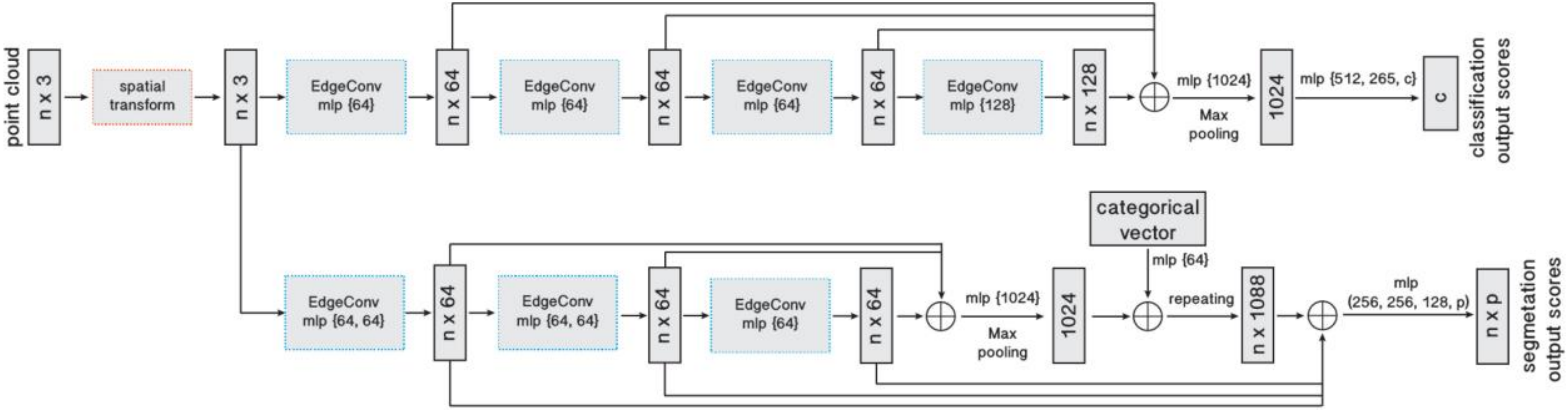
Method	Year	Dataset	Performance			Contribution
			OA	mAcc	mIoU	
PointNet++ [65]	2017	ModelNet40	90.7%	-	-	Improvements to PointNet and design of hierarchical network architecture
		ShapeNet Part	-	-	85.1%	
		ScanNet	84.5%	-	34.3%	
		S3DIS	81.0%	-	54.5%	
SO-Net [68]	2018	ModelNet10	94.1%	-	-	SOM for modeling the spatial distribution of points
		ModelNet40	90.8%	-	-	
		ShapeNet	-	-	84.6%	
PointSIFT [66]	2018	ScanNet	86.2%	-	-	Integration of multidirectional features using orientation-encoding convolution
		S3DIS	88.7%	-	70.2%	
PointWeb [67]	2019	ModelNet40	92.3%	89.4%	-	Proposing an adaptive feature adjustment module for interactive feature exploitation
		S3DIS	86.9%	66.6%	60.3%	
ShellNet [69]	2019	ScanNet	85.2%	-	-	Proposing an efficient point cloud processing network using statistics from concentric spherical shells
		S3DIS	87.1%		66.8%	
		Semantic3D	93.2%	-	69.4%	
RandLA-Net [71]	2020	Semantic3D	94.8%	-	77.4%	Proposing a lightweight network that exploits large receptive fields and keeps geometric details through LFAM
		SemanticKITTI	-	-	53.9%	
PointASNL [70]	2020	ModelNet10	95.9%	-	-	Proposing a local–nonlocal module with strong noise robustness
		ModelNet40	93.2%	-	-	
		ScanNet	-	-	63.0%	
		S3DIS	-	-	68.7%	
PointMLP [72]	2022	ModelNet40	94.1%	91.5%	-	A pure residual MLP network
		ScanObjectNN	86.1%	84.4%	-	

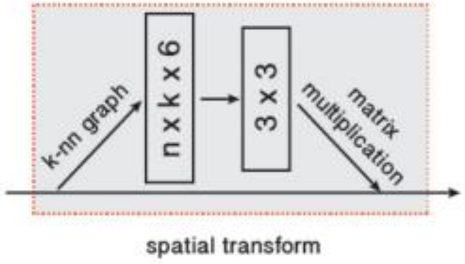
Point-based methods: Attention-based aggregation

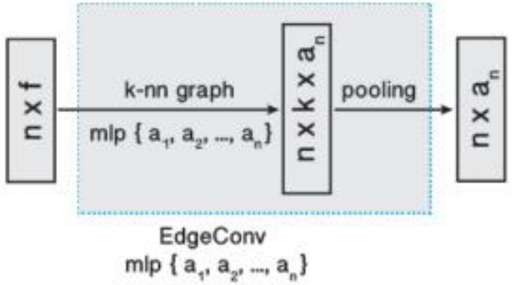


Zhao, Chenxi, et al. "Pooling scores of neighboring points for improved 3D point cloud segmentation." *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2019.

Point-based methods: Local-Global concatenation

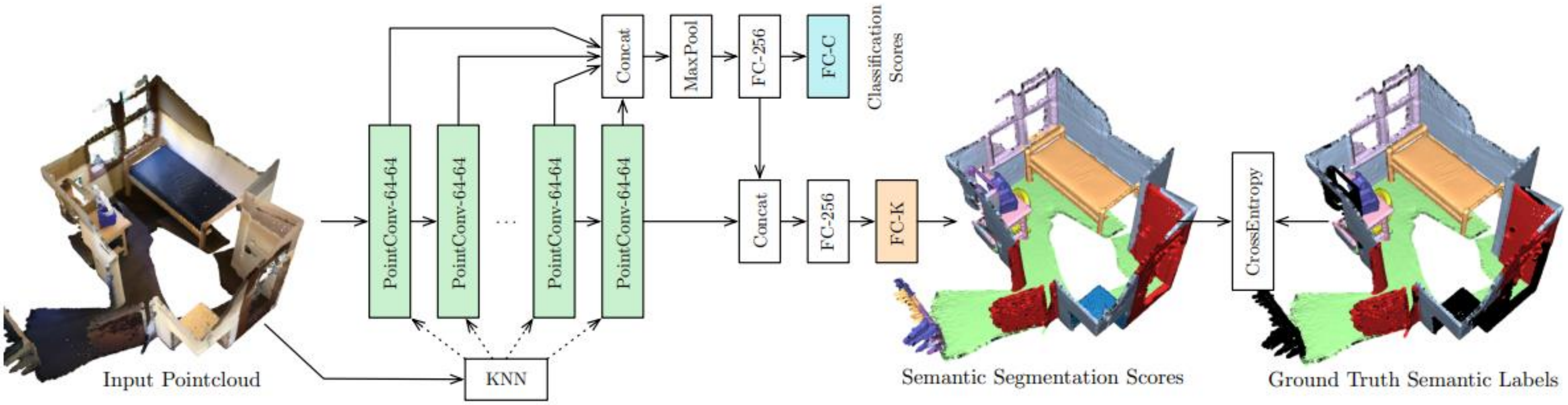






Wang, Yue, et al. "Dynamic graph cnn for learning on point clouds." *ACM Transactions on Graphics (tog)* 38.5 (2019): 1-12.

Point-based methods: Point convolution (a)



Engelmann, Francis, Theodora Kontogianni, and Bastian Leibe. "Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds." *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

Point-based methods: Point convolution (b)

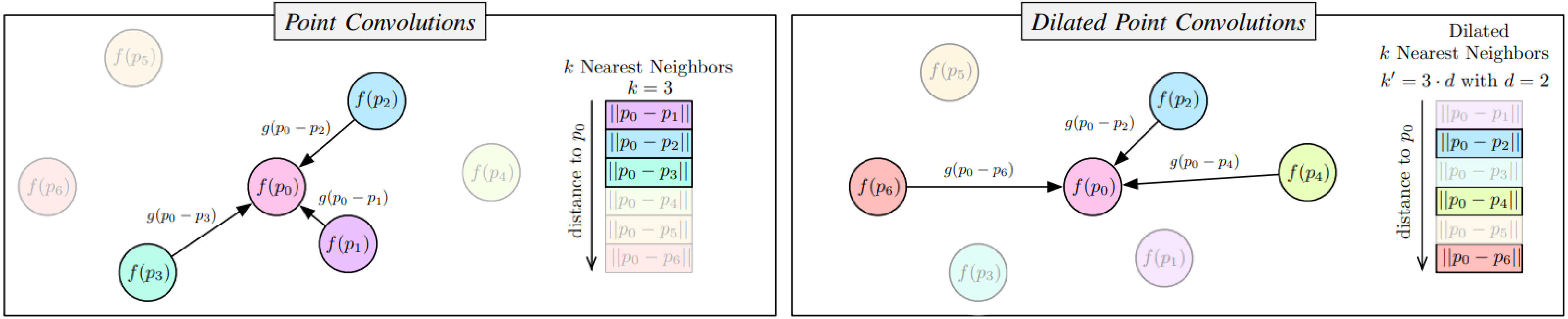
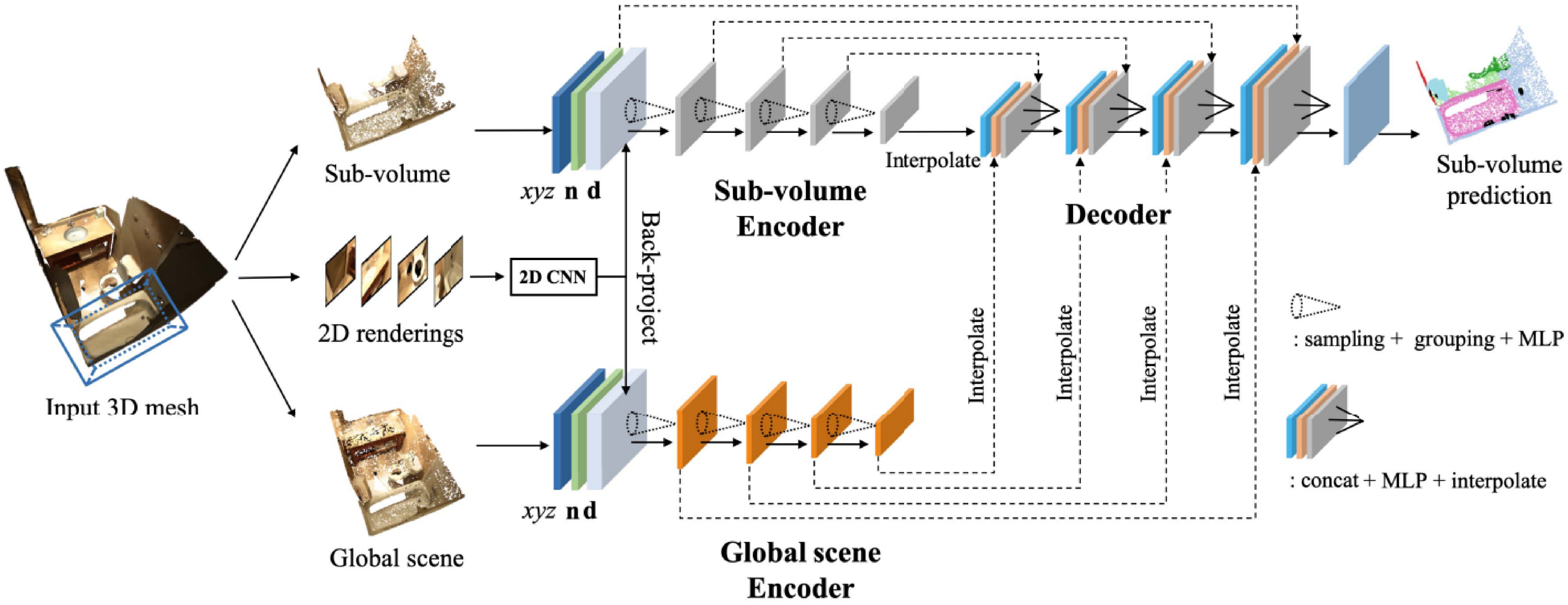


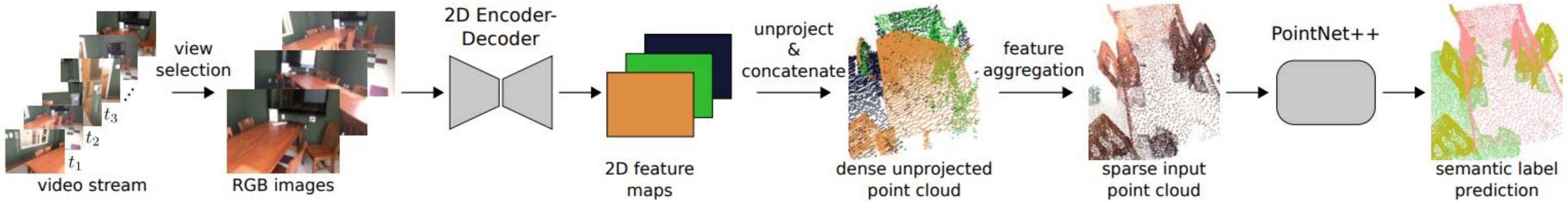
Fig. 2. *(Left)* **Point Convolutions.** Schematic illustration of point convolutions. The continuous feature function $f(\cdot)$ assigns a feature value to continuous point positions p. *(Right)* **Dilated Point Convolutions.** We propose *dilated* point convolutions as an elegant mechanism to significantly increase the receptive field of point convolutions resulting in a notable boost in performance at almost no additional computational cost (see Table IV). Instead of computing the kernel weights $g(\cdot)$ over the k nearest neighbors, we propose to compute the kernel weights over a *dilated* neighborhood obtained by computing the sorted $k \cdot d$ nearest neighbors and preserving only every d-th point.

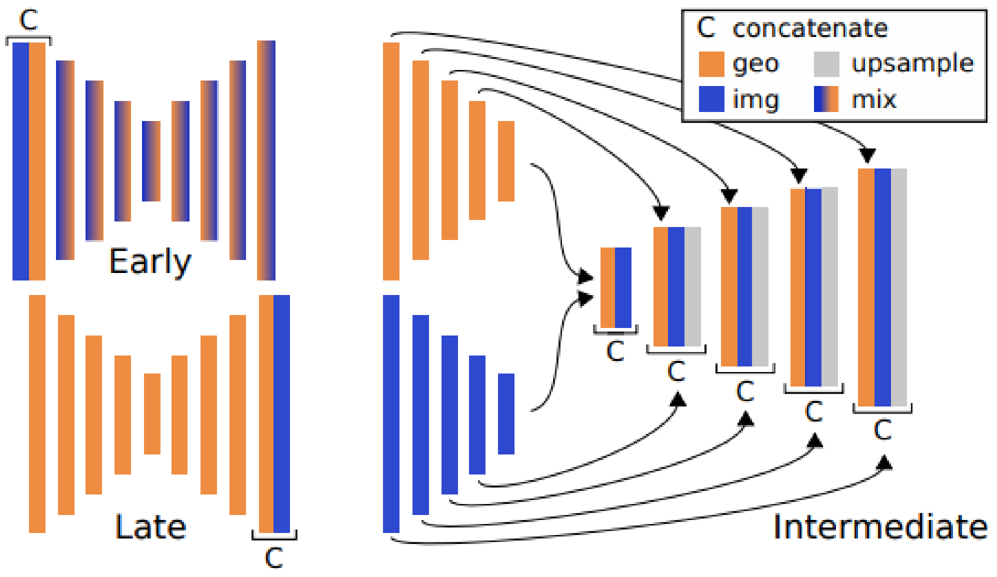
Hybrid-based methods (1)



Chiang, Hung-Yueh, et al. "A unified point-based framework for 3d segmentation." *2019 International Conference on 3D Vision (3DV)*. IEEE, 2019.

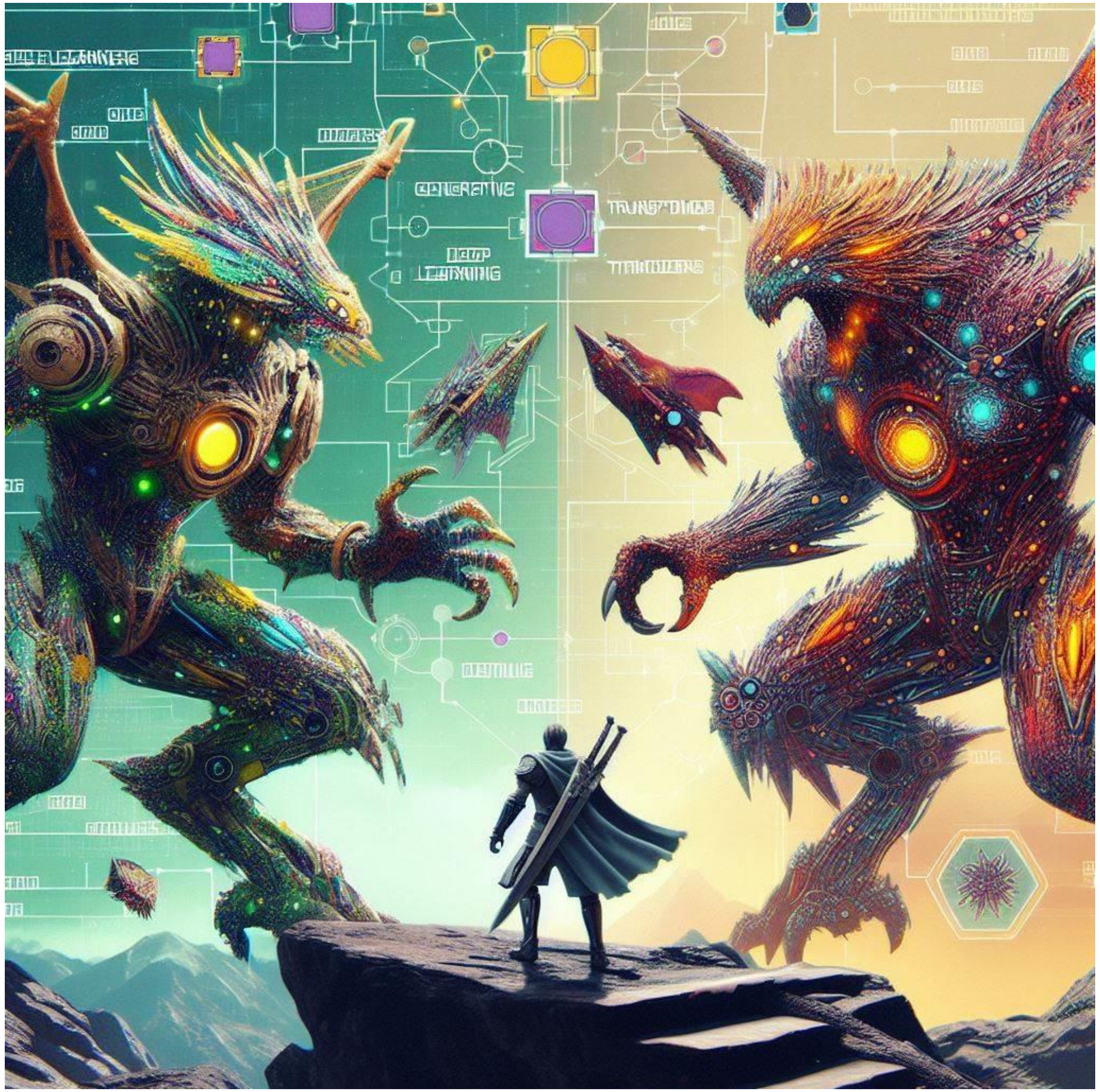
Hybrid-based methods (2)1





Jaritz, Maximilian, Jiayuan Gu, and Hao Su. "Multi-view pointnet for 3d scene understanding." *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019.

A brave new world



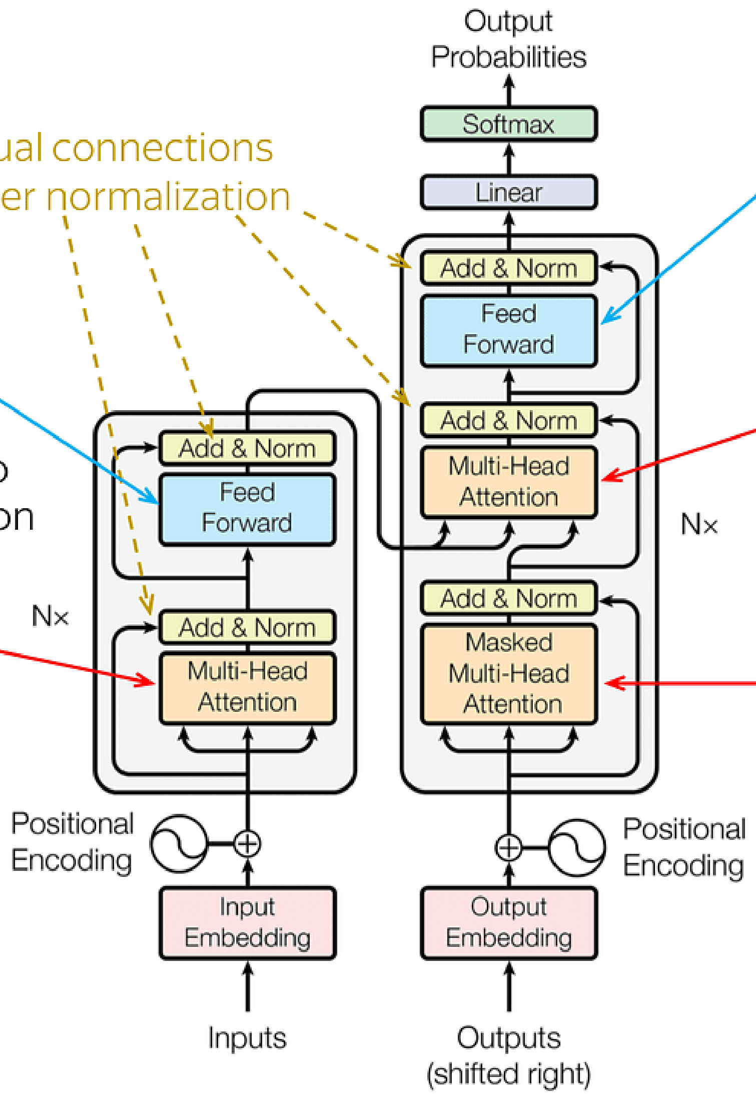
Transformers

Residual connections
and layer normalization

Feed-forward network:
after taking information from
other tokens, take a moment to
think and process this information

Encoder self-attention:
tokens look at each other

queries, keys, values
are computed from
encoder states



Feed-forward network:
after taking information from
other tokens, take a moment to
think and process this information

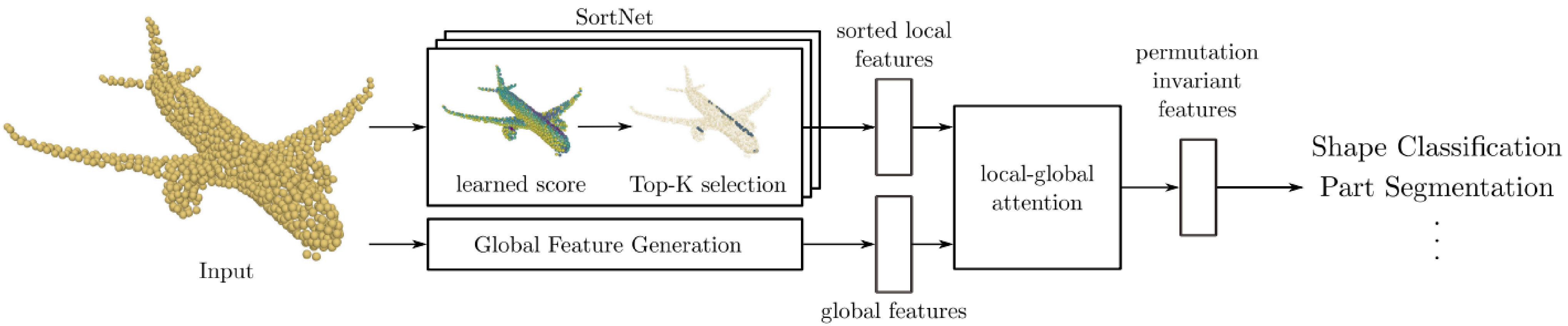
Decoder-encoder attention:
target token looks at the source

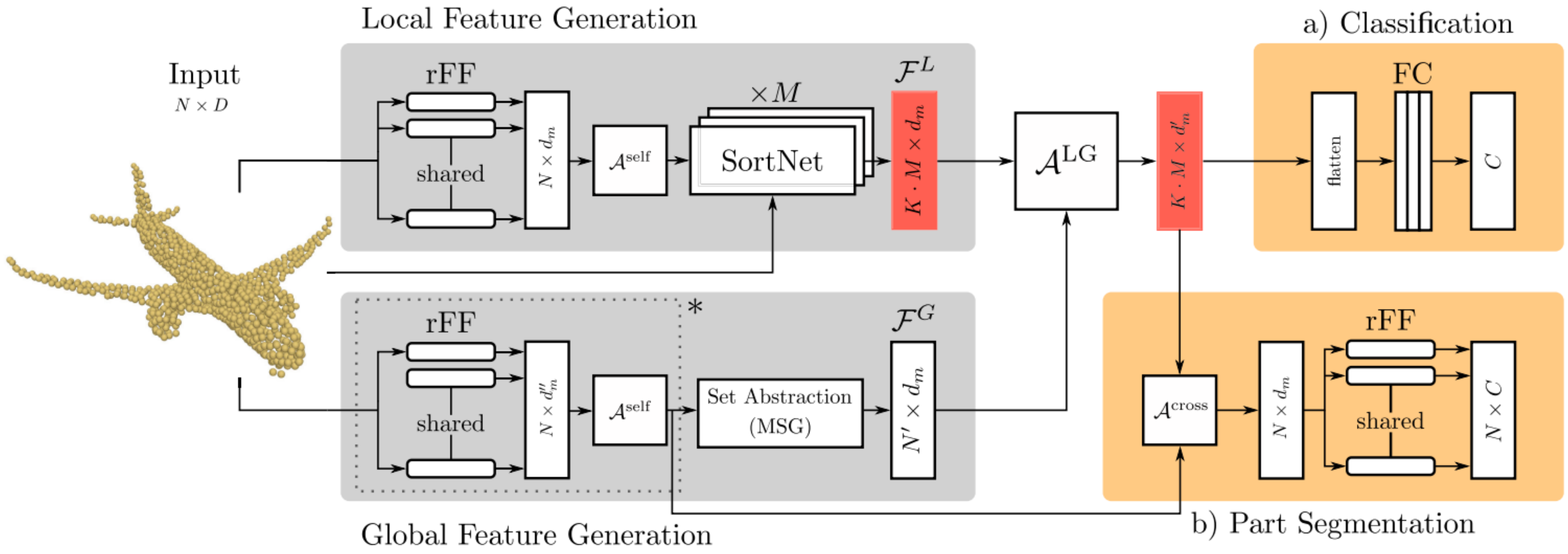
queries – from decoder states; keys
and values from encoder states

Decoder self-attention (masked):
tokens look at the previous tokens

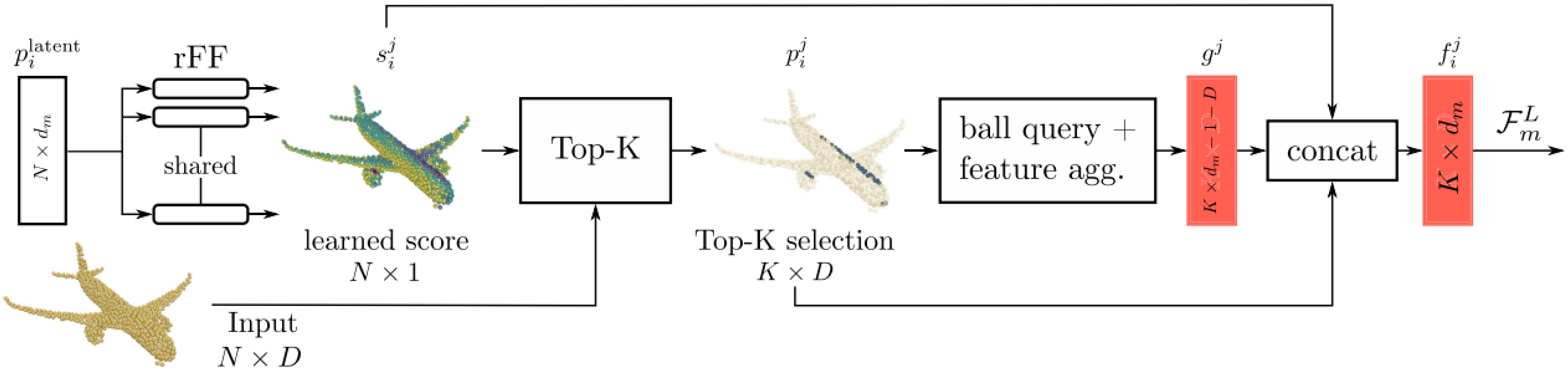
queries, keys, values are computed
from decoder states

Transformers – Point Transformer



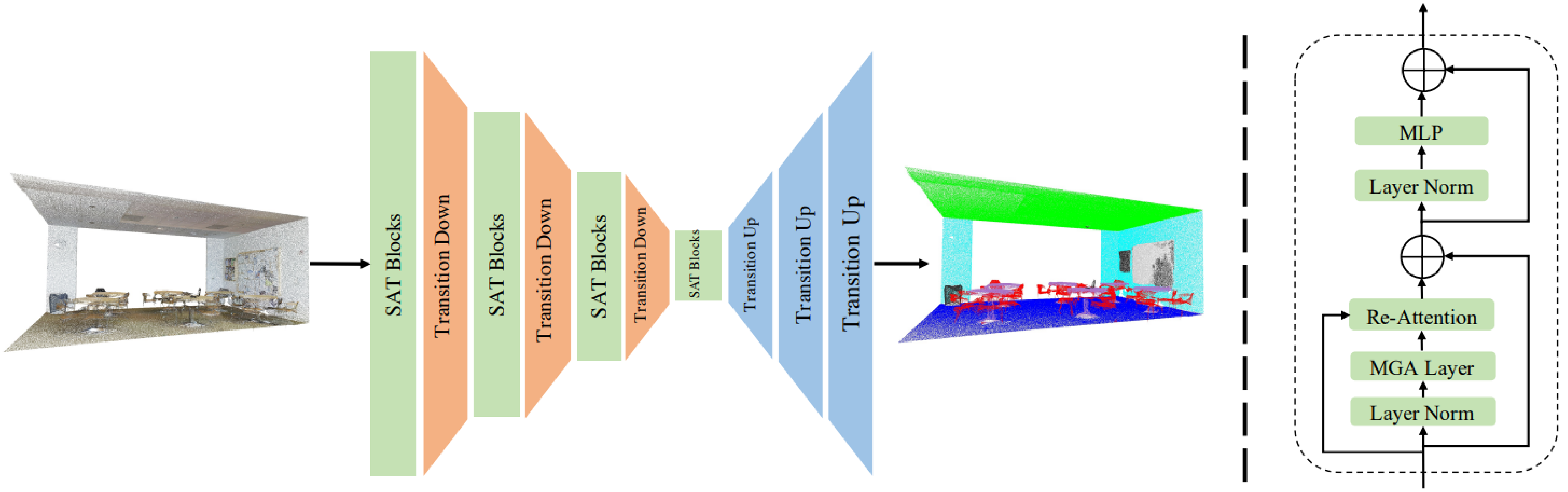


Transformers – Point Transformer



Engel, Nico, Vasileios Belagiannis, and Klaus Dietmayer. "Point transformer." *IEEE access* 9 (2021): 134826-134840.

Transformers - SAT

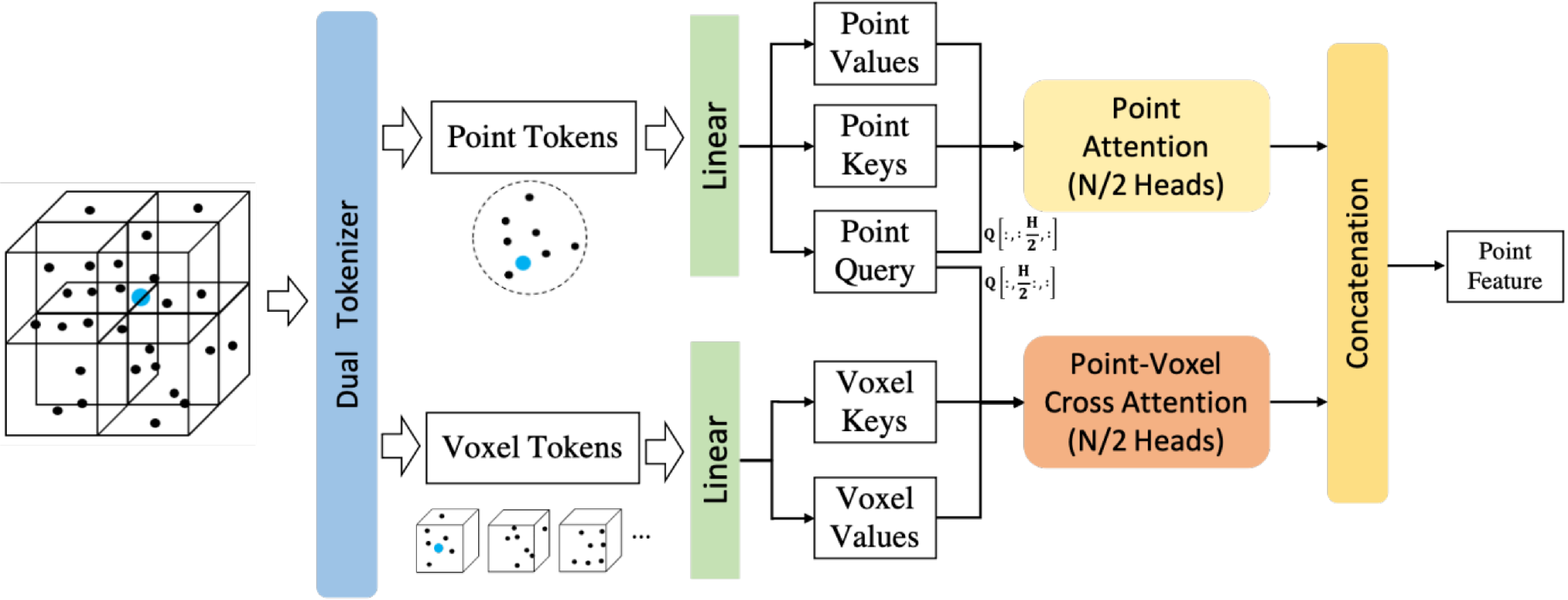


(a) Overview

(b) Size-Aware Transformer Block

Zhou, Junjie, et al. "SAT: Size-Aware Transformer for 3D Point Cloud Semantic Segmentation." *arXiv preprint arXiv:2301.06869* (2023).

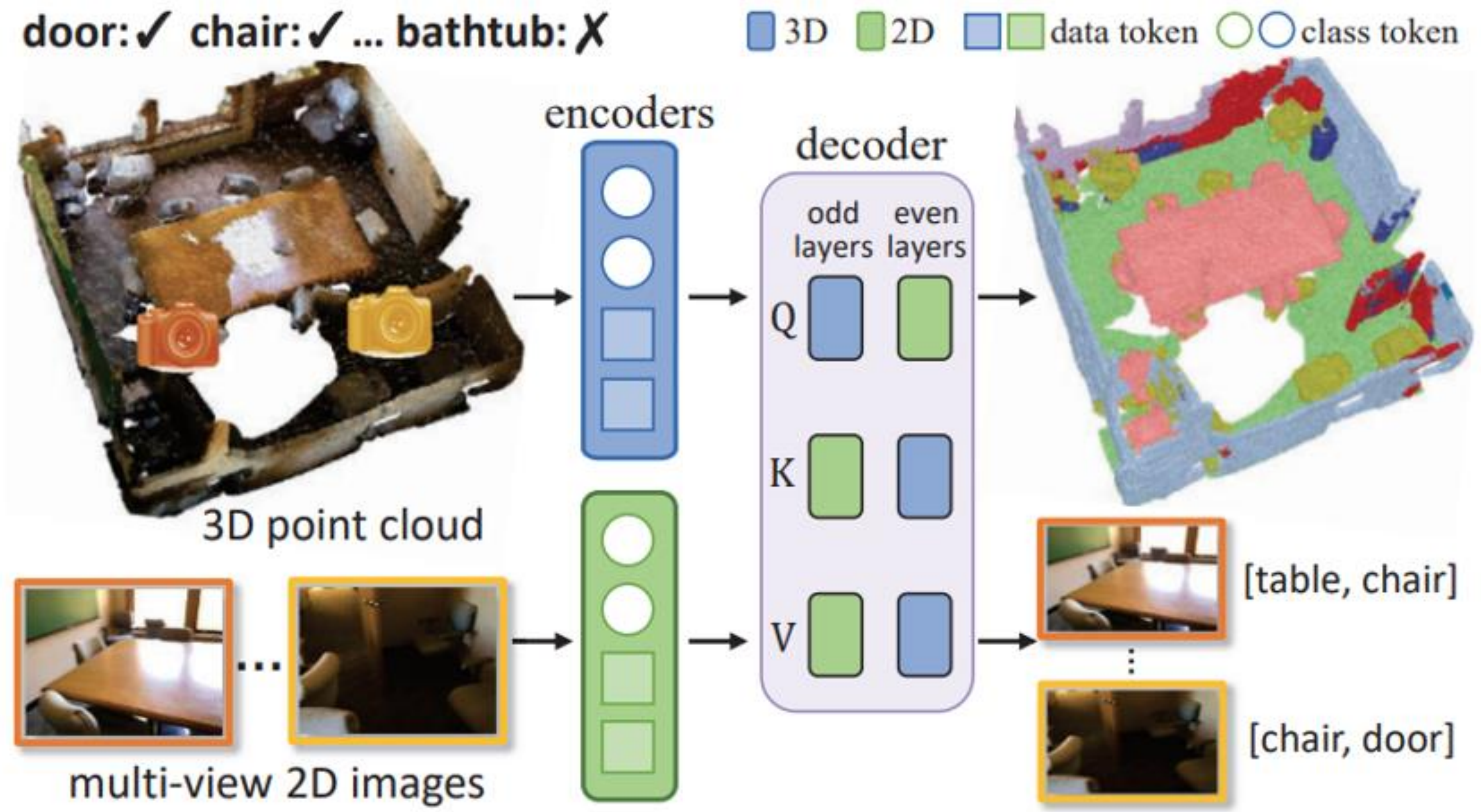
Transformers - SAT



Transformers: Comparison

Method	Year	Dataset	Performance			Contribution
			OA	mAcc	mIoU	
PAT [94]	2019	ModelNet40	91.7%	-	-	Pioneering Transformer-based processing of point clouds
		S3DIS	-	-	64.28%	
PCT [91]	2021	ModelNet40	93.2%	-	-	Proposing a coordinate-based embedding module and an offset attention module
		S3DIS	-	67.7%	61.33%	
Point Transformer [92] (Zhao et al.)	2021	ModelNet40	93.7%	90.6%	-	Facilitating interactions between local feature vectors through residual transformer blocks
		S3DIS	90.2%	81.9%	73.5%	
		ShapeNet Part	-	-	86.6%	
Point Transformer [93] (Engel et al.)	2021	ModelNet40	92.8%	-	-	Proposing a multihead attention network
		ShapeNet	-	-	85.9%	
MLMST [95]	2021	ModelNet10	95.5%	-	-	Proposing a multilevel multiscale Transformer
		ModelNet40	92.9%	-	-	
		ShapeNet Part	-	-	86.4%	
		S3DIS	-	-	62.9%	
DTNet [96]	2021	ModelNet40	92.9%	90.4%	-	Proposing a novel dual-point cloud Transformer architecture
		ShapeNet Part	-	-	85.6%	
Stratified Transformer [97]	2022	ShapeNet Part	-	-	86.6%	Adaptive contextual relative position encoding and point embedding effective learning of long-range contexts
		ScanNet	-	-	73.7%	
SAT [98]	2023	ScanNet	-	-	74.2%	Proposing a multigranular attention scheme and a reattention module
		S3DIS	-	78.8%	72.6%	

Transformers – Towards a Multimodal Approach

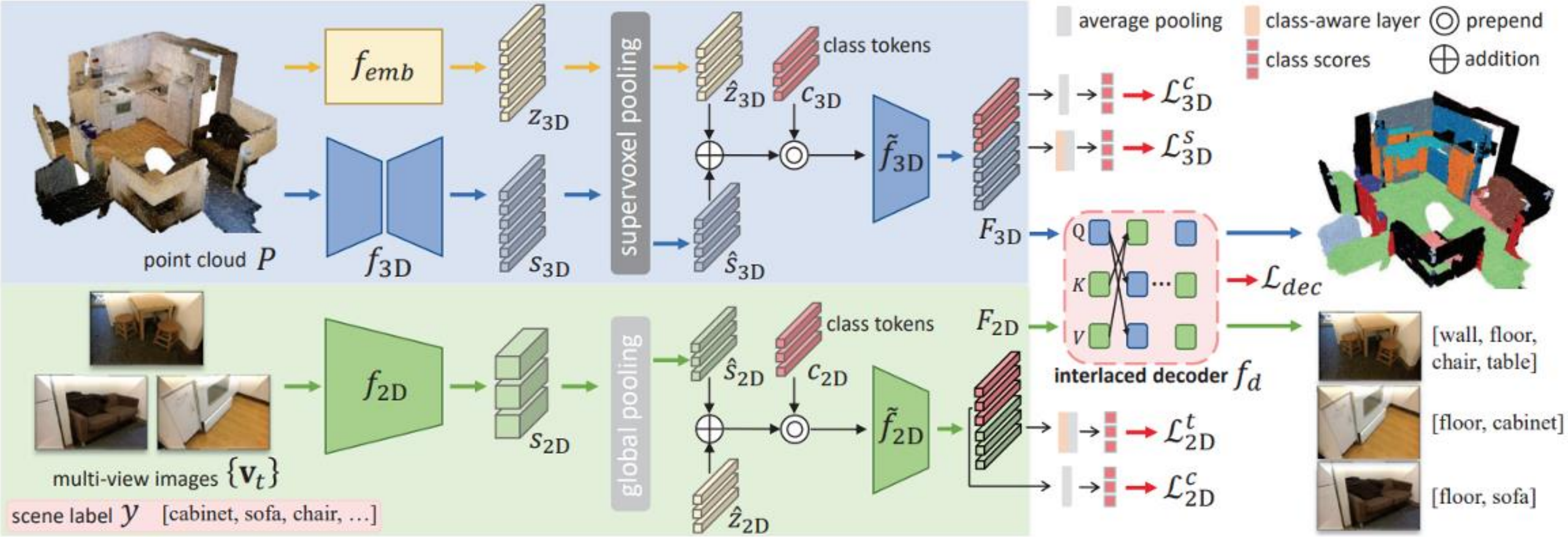


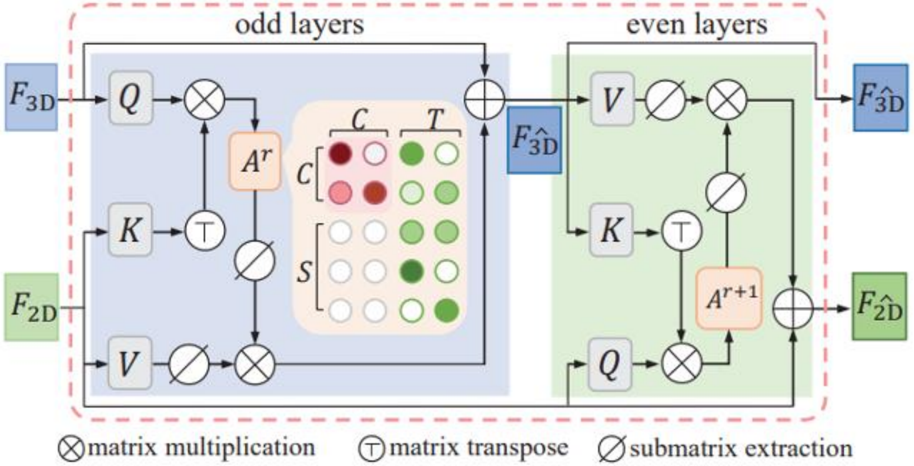
Yang, Cheng-Kun, et al. "2D-3D Interlaced Transformer for Point Cloud Segmentation with Scene-Level Supervision." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

Multimodal Interlaced Transformer (MIT)

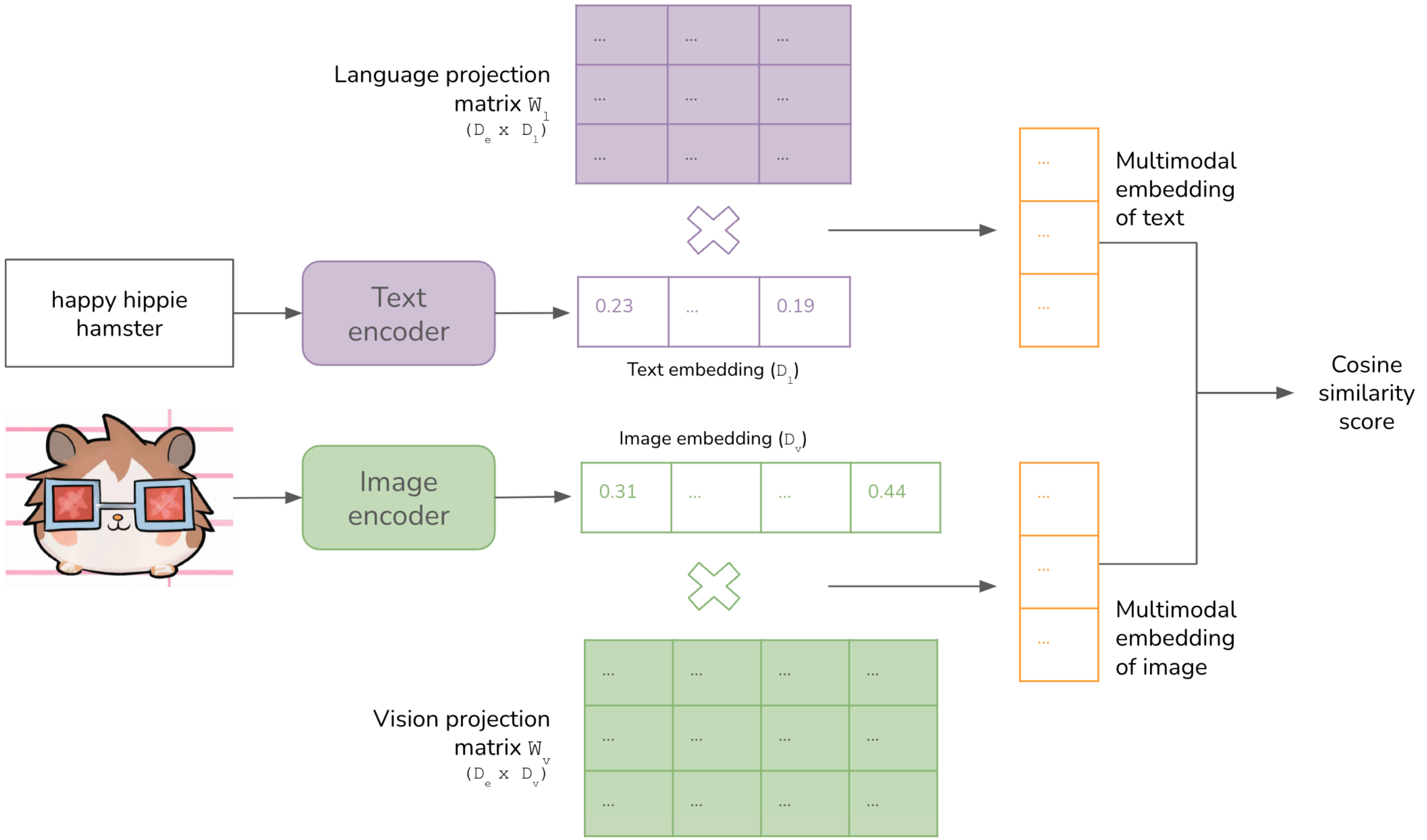
MinkUNet

Resnet 50





Large Multimodal Models (LMM)



At a high level, a multimodal system consists of the following components.

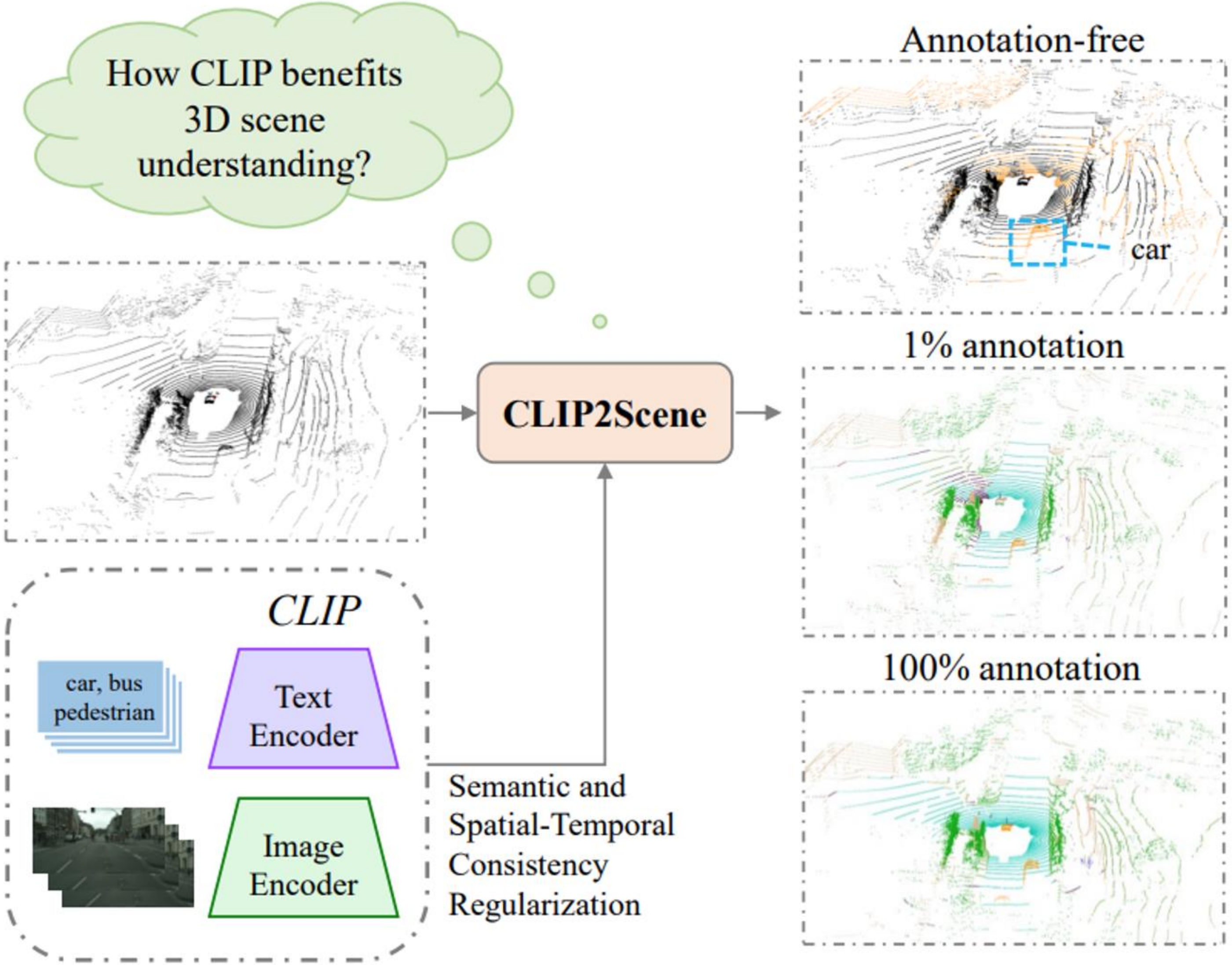
1. An **encoder** for each data modality to generate the embeddings for data of that modality.
2. A way to **align embeddings** of different modalities into the same **multimodal embedding space**.
3. [Generative models only]
   A **language model to generate text responses**. Since inputs can contain both text and visuals, new techniques need to be developed to allow the language model to condition its responses on not just text, but also visuals.

LMM classification

GENERATIVE MODELS

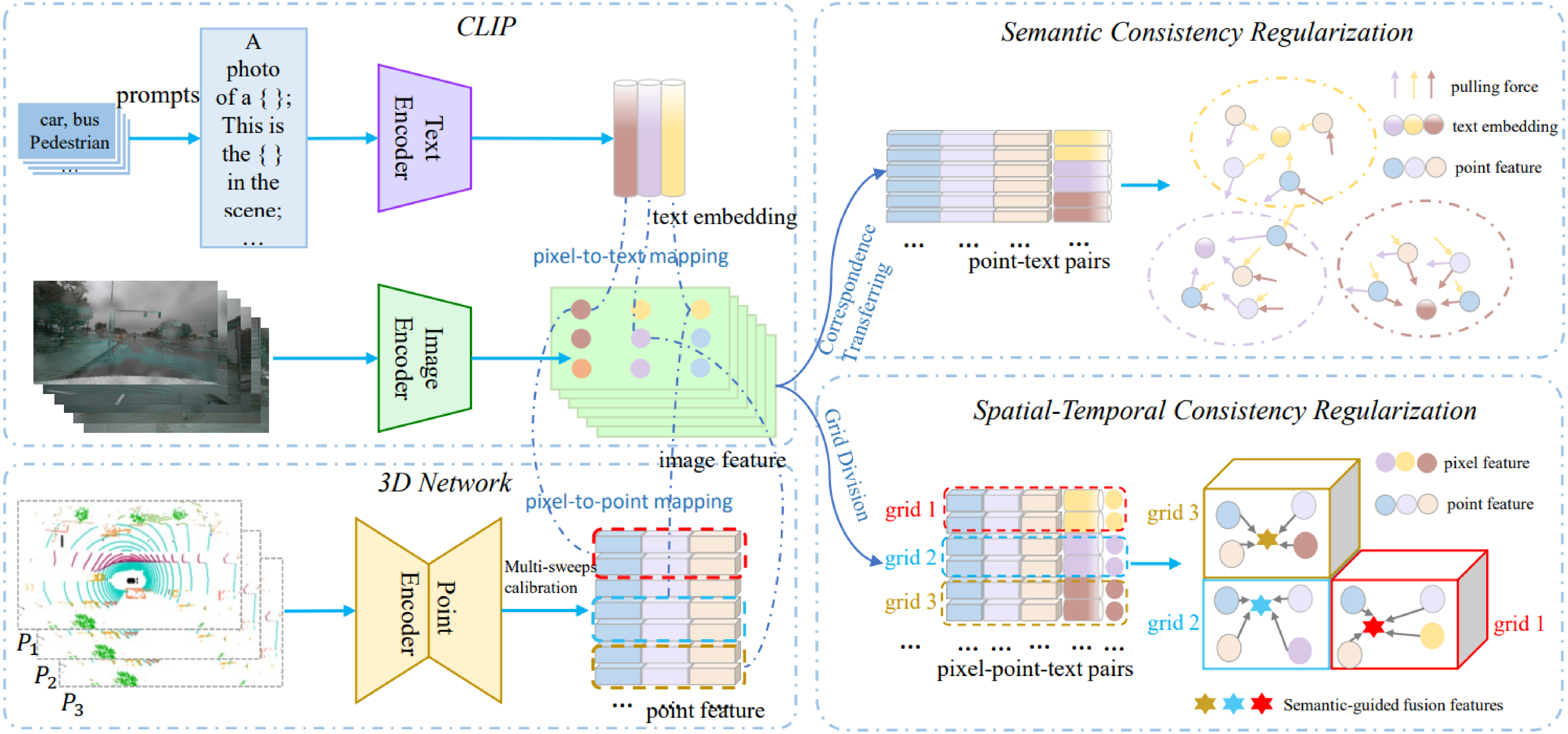
SCENE UNDERSTANDING MODELS

CLIP2SCENE



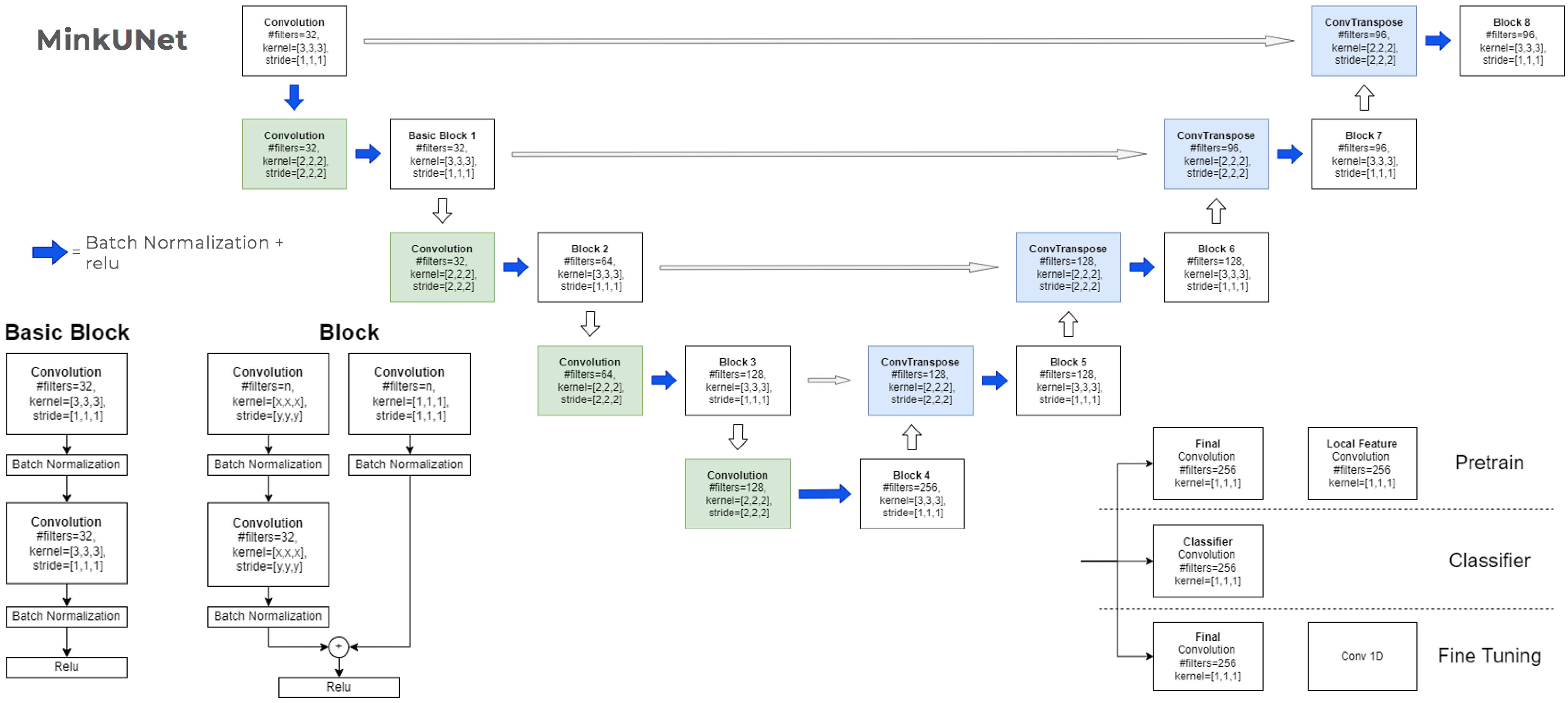
Chen, Runnan, et al. "CLIP2Scene: Towards Label-efficient 3D Scene Understanding by CLIP." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

CLIP2SCENE



CLIP
car, bus
Pedestrian
prompts
A photo of a { }; This is the { } in the scene; ...
Text Encoder
text embedding
pixel-to-text mapping
Image Encoder
image feature
3D Network
P_1
P_2
P_3
Point Encoder
Multi-sweeps calibration
pixel-to-point mapping
point feature
Correspondence Transferring
Grid Division
Semantic Consistency Regularization
point-text pairs
pulling force
text embedding
point feature
Spatial-Temporal Consistency Regularization
grid 1
grid 2
grid 3
pixel-point-text pairs
pixel feature
point feature
Semantic-guided fusion features

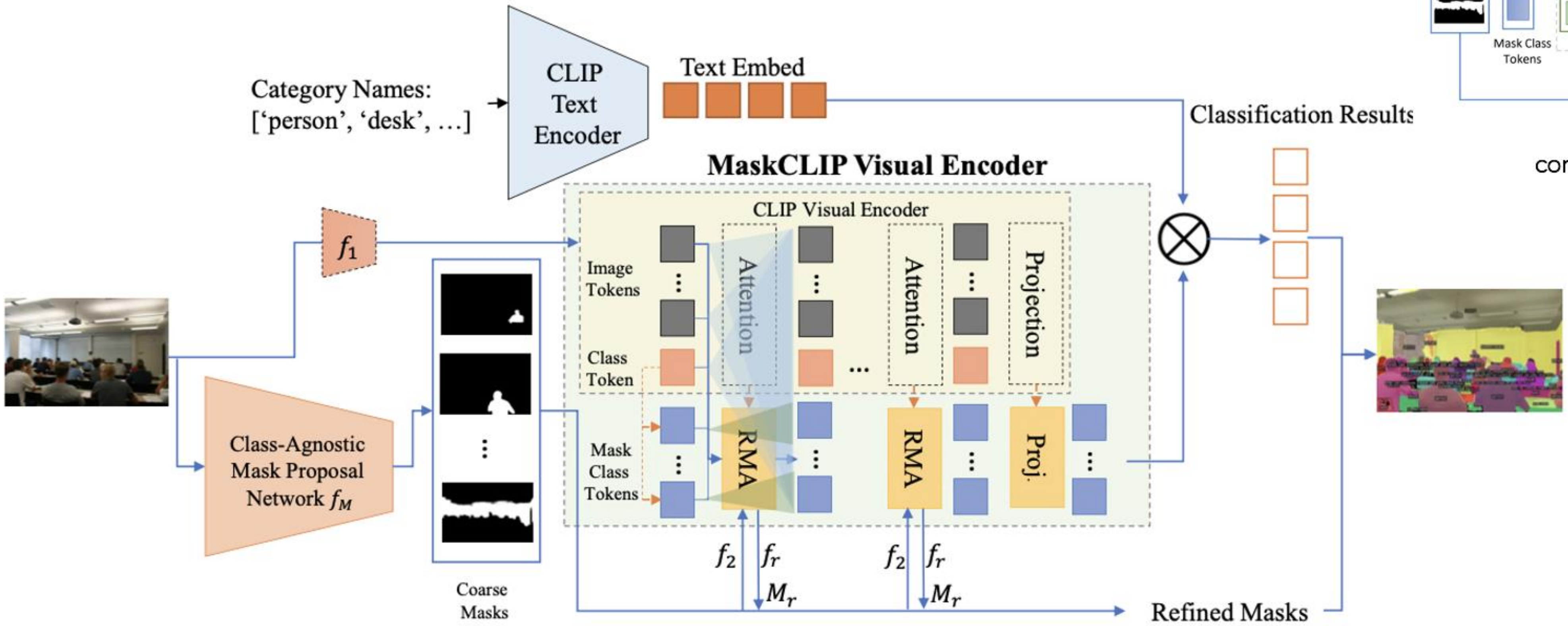
CLIP2SCENE – 3D Feature Extractor



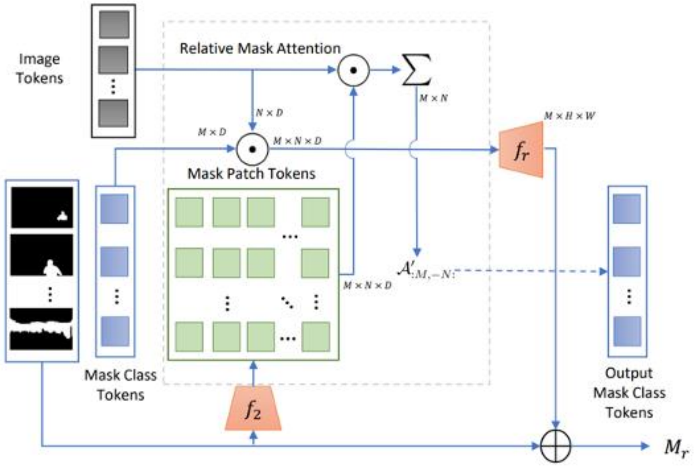
CLIP2SCENE – 2D Feature Extractor

MaskCLIP

Uses 12 encoder MaskCLIP layers



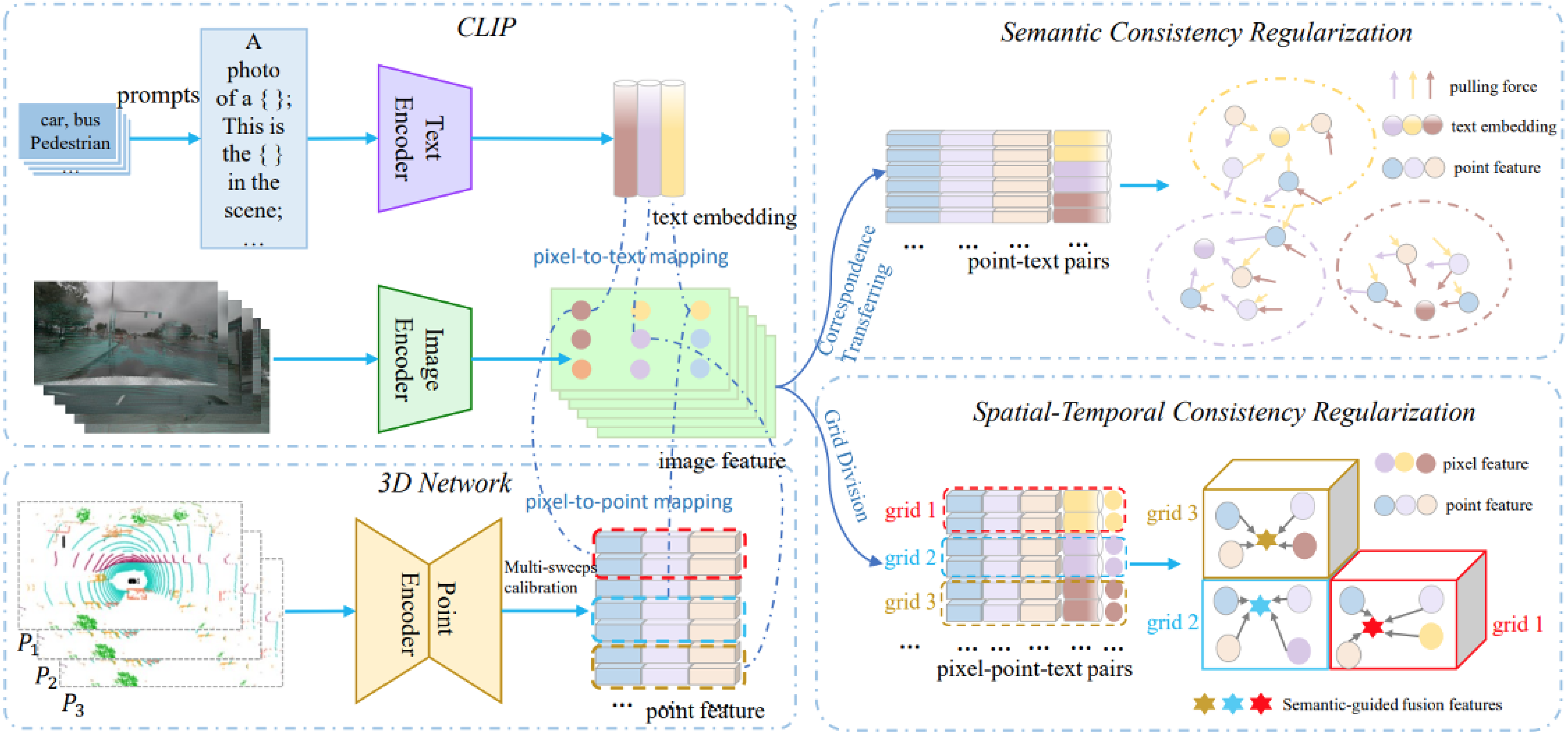
Relative Mask Attention



Adds another attention matrix computed using the Image Tokens and the Mask Patch Tokens

CLIP2SCENE – Training strategy

- The 2 regularization are in practice losses
- Every 10 training steps, there is a probability of changing loss



Losses:

Semantic = crossentropy (pairing_points , prediction)

Spatial = mean (1 - Cosine (image_features, points_features))

CLIP2SCENE – Evaluation

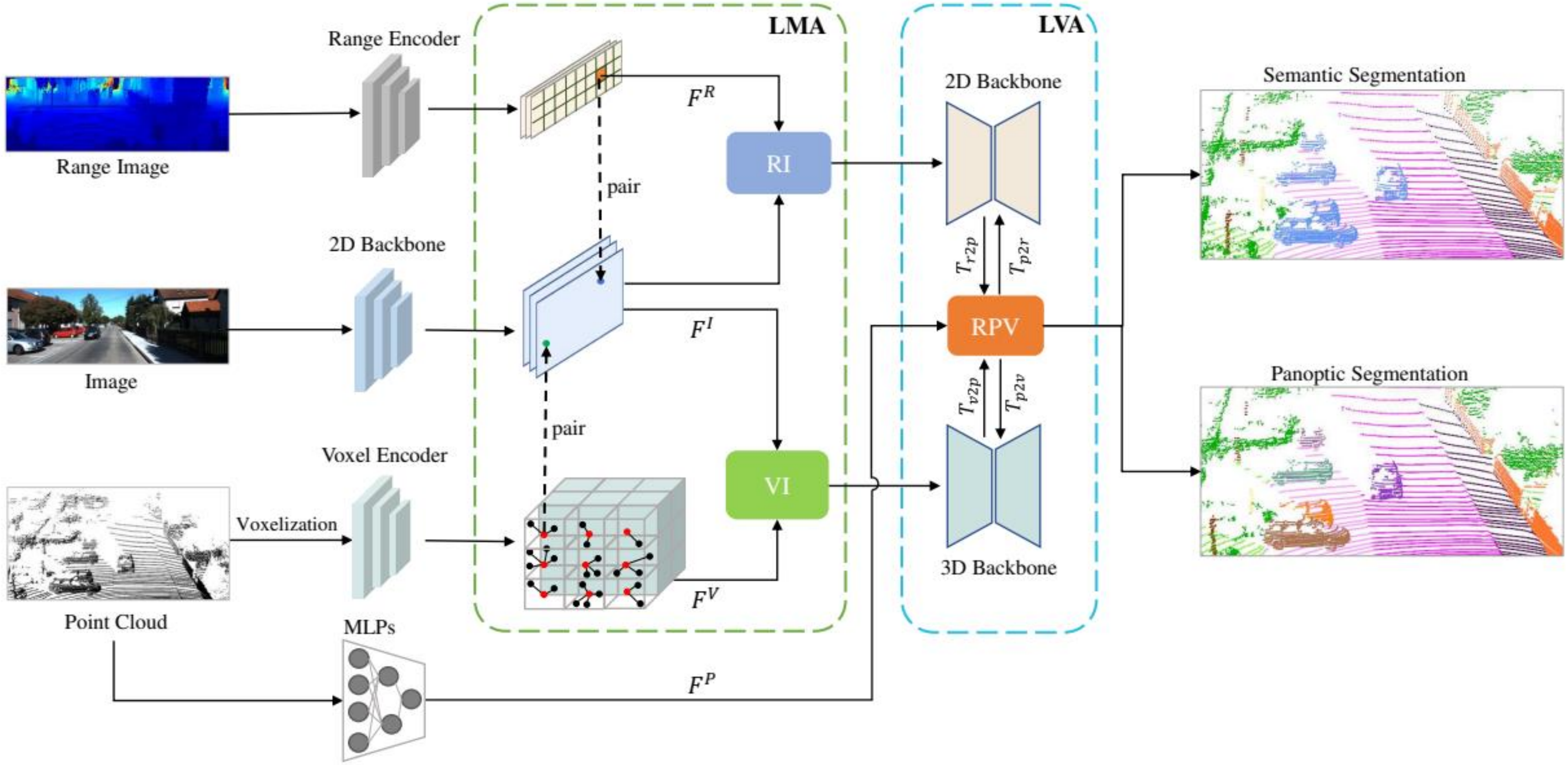
Table 1. Comparisons (mIoU) among self-supervised methods on the nuScenes [24], SemanticKITTI [3], and ScanNet [20] *val* sets.

Initialization	nuScenes		SemanticKITTI		ScanNet	
	1%	100%	1%	100%	5%	100%
Random	42.2	69.1	32.5	52.1	46.1	63.3
PPKT [44]	48.0	70.1	39.1	53.1	47.5	64.2
SLidR [51]	48.2	70.4	39.6	54.3	47.9	64.9
PointContrast [55]	47.2	69.2	37.1	52.3	47.6	64.5
CLIP2Scene	**56.3**	**71.5**	**42.6**	**55.0**	**48.4**	**65.1**

Table 2. Annotation-free 3D semantic segmentation performance (mIoU) on the nuScenes [24] and ScanNet [20] *val* sets.

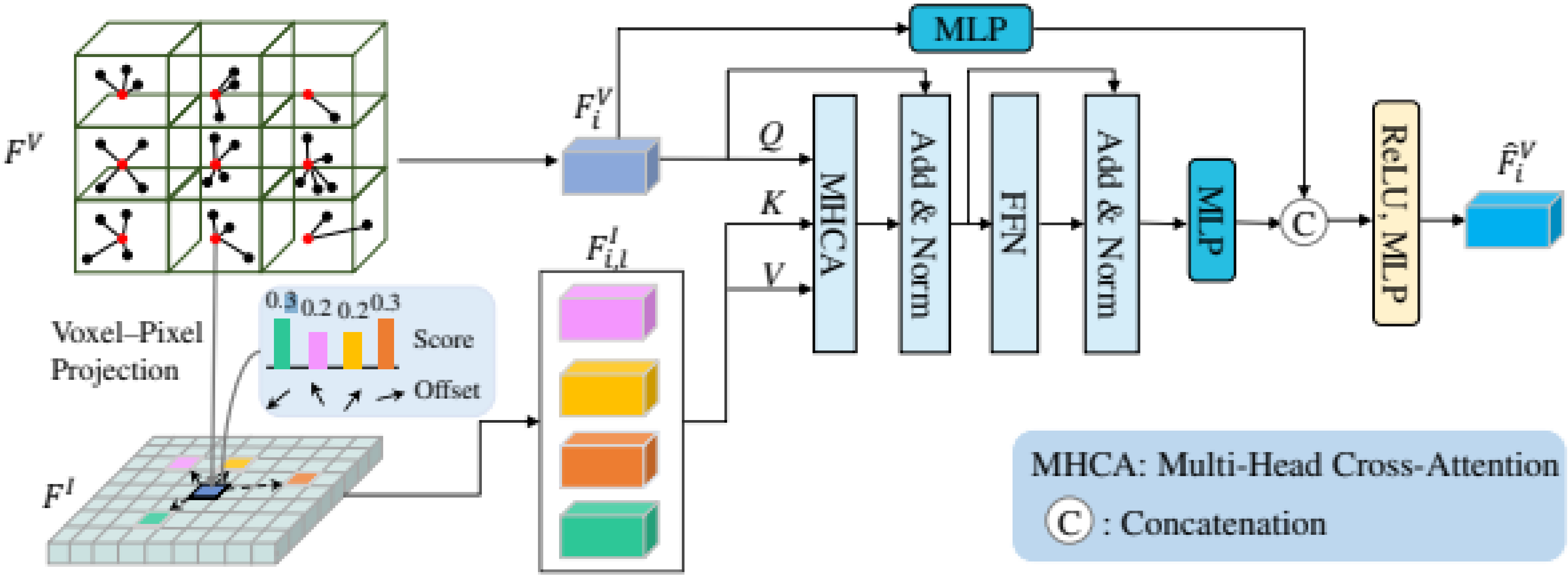
Method	nuScenes	ScanNet
CLIP2Scene	20.80	25.08

UniSeg

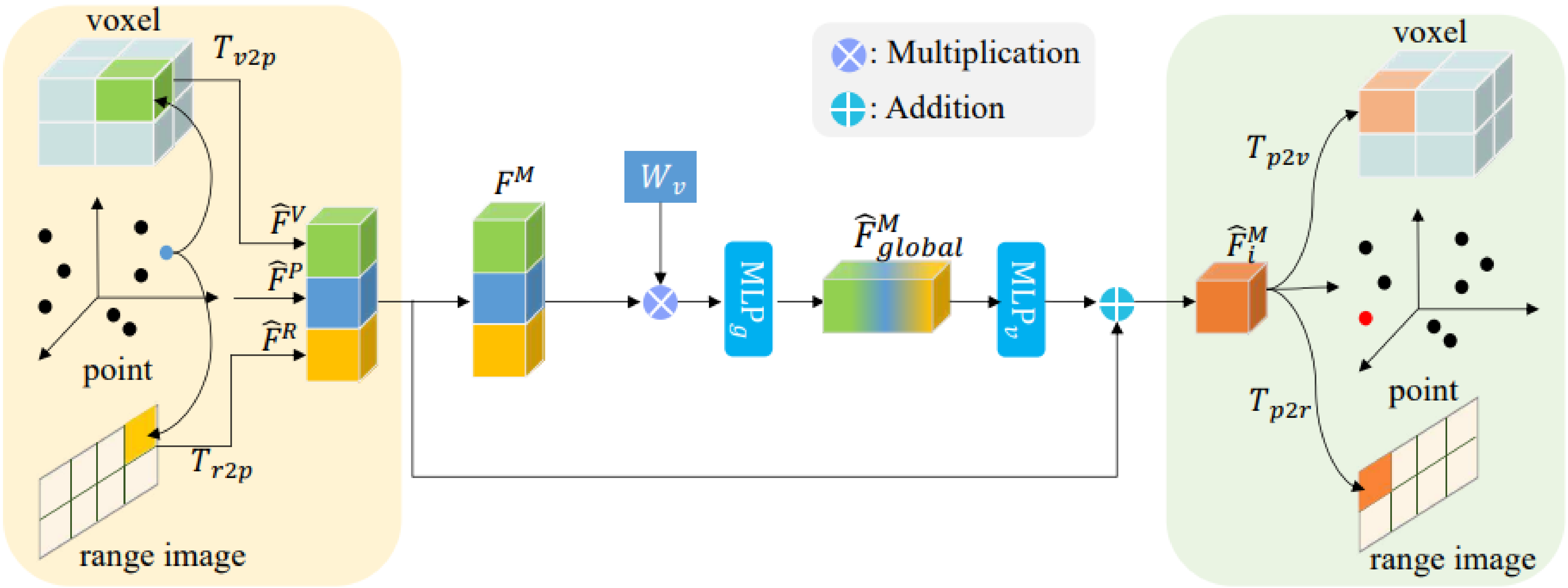


Liu, Youquan, et al. "Uniseg: A unified multi-modal lidar segmentation network and the openpcseg codebase." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

UniSeg – Learnable Cross-Modal Association (LMA)



UniSeg – Learnable Cross-View Association (LVA)



UniSeg - Evaluation

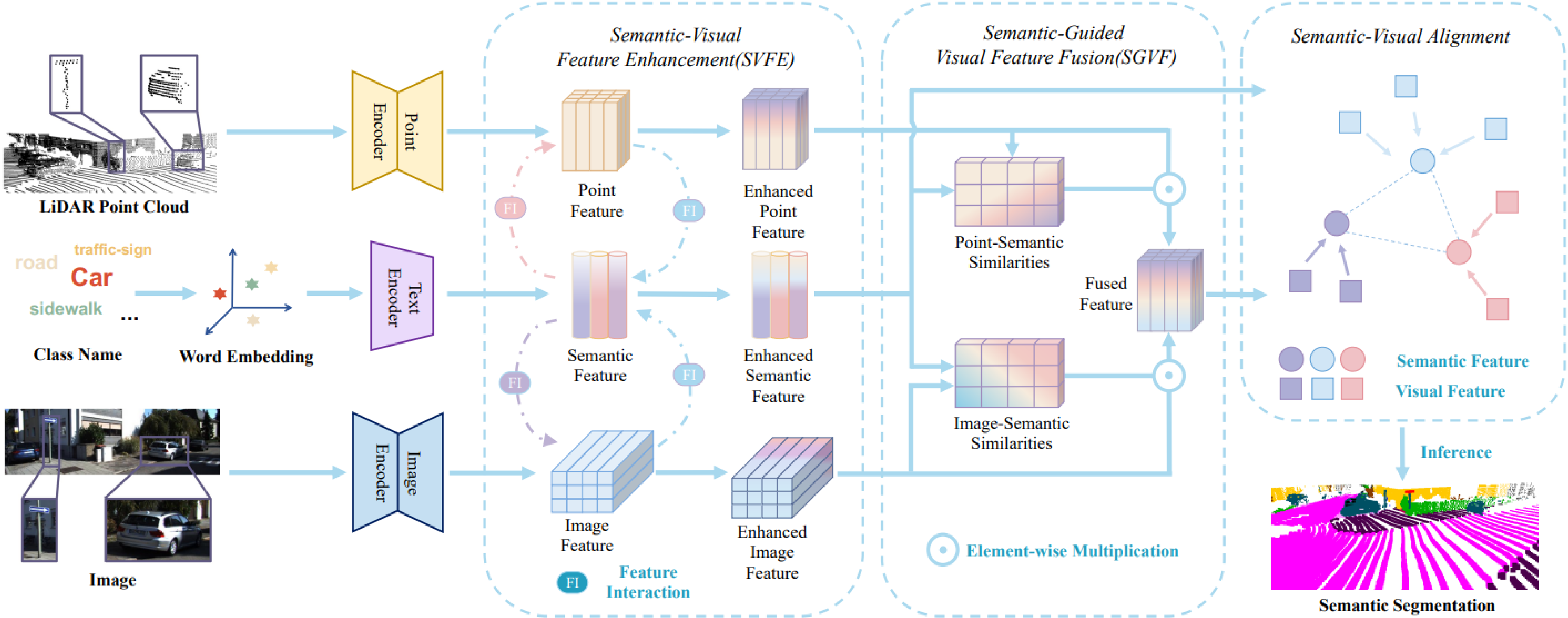
Table 2: Quantitative results of UniSeg and SoTA LiDAR semantic segmentation methods on the SemanticKITTI *test* set.

Method	mIoU	car	bicy	moto	truc	o.veh	ped	b.list	m.list	road	park	walk	o.gro	build	fenc	veg	trun	terr	pole	sign
AMVNet [33]	65.3	96.2	59.9	54.2	48.8	45.7	71.0	65.7	11.0	90.1	71.0	75.8	32.4	92.4	69.1	85.6	71.7	69.6	62.7	67.2
JS3C-Net [51]	66.0	95.8	59.3	52.9	54.3	46.0	69.5	65.4	39.9	88.9	61.9	72.1	31.9	92.5	70.8	84.5	69.8	67.9	60.7	68.7
SPVNAS [43]	66.4	97.3	51.5	50.8	59.8	58.8	65.7	65.2	43.7	90.2	67.6	75.2	16.9	91.3	65.9	86.1	73.4	71.0	64.2	66.9
Cylinder3D [62]	68.9	97.1	67.6	63.8	50.8	58.5	73.7	69.2	48.0	92.2	65.0	77.0	32.3	90.7	66.5	85.6	72.5	69.8	62.4	66.2
AF2S3Net [9]	69.7	94.5	65.4	**86.8**	39.2	41.1	**80.7**	80.4	**74.3**	91.3	68.8	72.5	**53.5**	87.9	63.2	70.2	68.5	53.7	61.5	71.0
RPVNet [48]	70.3	97.6	68.4	68.7	44.2	61.1	75.9	74.4	73.4	**93.4**	70.3	**80.7**	33.3	**93.5**	72.1	86.5	75.1	71.7	64.8	61.4
SDSeg3D [29]	70.4	97.4	58.7	54.2	54.9	65.2	70.2	74.4	52.2	90.9	69.4	76.7	41.9	93.2	71.1	86.1	74.3	71.1	65.4	70.6
GASN [54]	70.7	96.9	65.8	58.0	59.3	61.0	80.4	**82.7**	46.3	89.8	66.2	74.6	30.1	92.3	69.6	87.3	73.0	72.5	66.1	**71.6**
PVKD [20]	71.2	97.0	67.9	69.3	53.5	60.2	75.1	73.5	50.5	91.8	70.9	77.5	41.0	92.4	69.4	86.5	73.8	71.9	64.9	65.8
2DPASS [52]	72.9	97.0	63.6	63.4	61.1	61.5	77.9	81.3	74.1	89.7	67.4	74.7	40.0	**93.5**	**72.9**	86.2	73.9	71.0	65.0	70.4
RangeFormer [24]	73.3	96.7	69.4	73.7	59.9	66.2	78.1	75.9	58.1	92.4	73.0	78.8	42.4	92.3	70.1	86.6	73.3	72.8	66.4	66.6
UniSeg (Ours)	**75.2**	**97.9**	**71.9**	75.2	**63.6**	**74.1**	78.9	74.8	60.6	92.6	**74.0**	79.5	46.1	93.4	72.7	**87.5**	**76.3**	**73.1**	**68.3**	68.5

Table 3: Quantitative results of UniSeg and SoTA LiDAR semantic segmentation methods on the nuScenes *test* set.

Method	mIoU	barr	bicy	bus	car	const	motor	ped	cone	trail	truck	driv	other	walk	terr	made	veg
PMF [63]	77.0	82.0	40.0	81.0	88.0	64.0	79.0	80.0	76.0	81.0	67.0	97.0	68.0	78.0	74.0	90.0	88.0
Cylinder3D [62]	77.2	82.8	29.8	84.3	89.4	63.0	79.3	77.2	73.4	84.6	69.1	97.7	70.2	80.3	75.5	90.4	87.6
AMVNet [33]	77.3	80.6	32.0	81.7	88.9	67.1	84.3	76.1	73.5	84.9	67.3	97.5	67.4	79.4	75.5	91.5	88.7
SPVCNN [43]	77.4	80.0	30.0	91.9	90.8	64.7	79.0	75.6	70.9	81.0	74.6	97.4	69.2	80.0	76.1	89.3	87.1
AF2S3Net [9]	78.3	78.9	52.2	89.9	84.2	77.4	74.3	77.3	72.0	83.9	73.8	97.1	66.5	77.5	74.0	87.7	86.8
2D3DNet [17]	80.0	83.0	59.4	88.0	85.1	63.7	84.4	82.0	76.0	84.8	71.9	96.9	67.4	79.8	76.0	**92.1**	89.2
GASN [54]	80.4	85.5	43.2	90.5	**92.1**	64.7	86.0	83.0	73.3	83.9	75.8	97.0	71.0	**81.0**	**77.7**	91.6	**90.2**
2DPASS [52]	80.8	81.7	55.3	92.0	91.8	73.3	86.5	78.5	72.5	84.7	75.5	97.6	69.1	79.9	75.5	90.2	88.0
LidarMultiNet [53]	81.4	80.4	48.4	**94.3**	90.0	71.5	87.2	**85.2**	**80.4**	**86.9**	74.8	**97.8**	67.3	80.7	76.5	**92.1**	89.6
UniSeg (Ours)	**83.5**	**85.9**	**71.2**	92.1	91.6	**80.5**	**88.0**	80.9	76.0	86.3	**76.7**	97.7	**71.8**	80.7	76.7	91.3	88.8

Zero-shot point cloud segmentation



Lu, Yuhang, et al. "See more and know more: Zero-shot point cloud segmentation via multi-modal visual data." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

POLITECNICO
MILANO 1863

Deep Learning in 3D for Robotics

- Cooperative 3D Point Clouds Perception -

M. Matteucci (matteo.matteucci@polimi.it) and L. Cazzella (lorenzo.cazzella@polimi.it)

Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

Beyond single-vehicle perception

V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction

Tsun-Hsuan Wang[1], Sivabalan Manivasagam[1,2(✉)], Ming Liang[1], Bin
Wenyuan Zeng[1,2], and Raquel Urtasun[1,2]

[1] UberATG, Pittsburgh, USA
[2] University of Toronto, Toronto, Canada
{tsunhsuan.wang,manivasagam,ming.liang,byang,wenyuan,urtasun}@u

Abstract. In this paper, we explore the use of vehicle-to-vehicle (V2
communication to improve the perception and motion forecasting p
formance of self-driving vehicles. By intelligently aggregating the inf
mation received from multiple nearby vehicles, we can observe the sa
scene from different viewpoints. This allows us to see through occlusi
and detect actors at long range, where the observations are very spa
or non-existent. We also show that our approach of sending compres
deep feature map activations achieves high accuracy while satisfying co
munication bandwidth requirements.

Keywords: Autonomous driving · Object detection · Motion foreca

1 Introduction

While a world densely populated with self-driving vehicles (SDVs) m
futuristic, these vehicles will one day soon be the norm. They will prov
cheaper and less congested transportation solutions for everyone, every
core component of self-driving vehicles is their ability to perceive the wo
sensor data, the SDV needs to reason about the scene in 3D, identify
agents, and forecast how their futures might play out. These tasks are c
referred to as perception and motion forecasting. Both strong percep
motion forecasting are critical for the SDV to plan and maneuver throu
to get from one point to another safely.

A Survey of Collaborative Machine Learning Using 5G Vehicular Communications

Salvador V. Balkus, Honggang Wang, *Fellow, IEEE*, Brian D. Cornet,
Chinmay Mahabal, *Graduate Student Member, IEEE*, Hieu Ngo, and Hua Fang

***Abstract*—By enabling autonomous vehicles (AVs) to share data while driving, 5G vehicular communications allow AVs to collaborate on solving common autonomous driving tasks. AVs often rely on machine learning models to perform such tasks; as such, collaboration requires leveraging vehicular communications to improve the performance of machine learning algorithms. This paper provides a comprehensive literature survey of the intersection between machine learning for autonomous driving and vehicular communications. Throughout the paper, we explain how vehicle-to-vehicle (V2V) and vehicle-to-everything (V2X) communications are used to improve machine learning in AVs, answering five major questions regarding such systems. These questions include: 1) How can AVs effectively transmit data wirelessly on the road? 2) How do AVs manage the shared data? 3) How do AVs use shared data to improve their perception of the environment? 4) How do AVs use shared data to drive more safely and efficiently? and 5) How can AVs protect the privacy of shared data and prevent cyberattacks? We also summarize data sources that may support research in this area and discuss the future research potential surrounding these five questions.**

***Index Terms*—Vehicular communications, machine learning.**

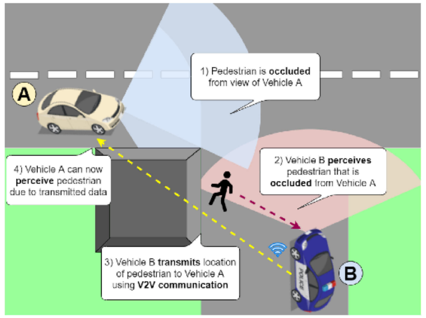


Fig. 1. Depiction of occlusion. The pedestrian is occluded from the top left vehicle by the building, but the lower right vehicle can detect the pedestrian and communicate this information to the top left car.

I. Introduction

As AUTONOMOUS vehicles (AVs) enter the commercial market and advance towards full autonomy, more AVs will be present on the world's roadways [1]. Today, AVs rely on sensors including cameras and LiDAR to monitor the road in order to drive safely and efficiently [2]. However, if other AVs are also present on the road, the vehicles can send data between each other in a process called vehicle-to-vehicle (V2V) communication.

V2V communication allows AVs to share information in

referred to as vehicle-to-infrastructure (V2I) communication. The term vehicle-to-everything (V2X) encompasses both V2V and V2I paradigms.

As AVs become more ubiquitous, V2V and V2I communications can provide improvements to common autonomous driving tasks. They will also serve to connect AVs to the Internet of Things as a whole, allowing for an interconnected world (Fig. 2). 5G communication technologies are largely considered the future of V2V communications due to their

2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

OPV2V: An Open Benchmark Dataset and Fusion Pipeline for Perception with Vehicle-to-Vehicle Communication

Xiang[1]*, Xin Xia[1], Xu Han[1], Jinlong Li[2], Jiaqi Ma[1]

information could be raw data, intermediate features, single CAV's detection output, and metadata e.g., timestamps and poses. Despite the big potential in this field, it is still in its infancy. One of the major barriers is the lack of a large open-source dataset. Unlike the single vehicle's perception area where multiple large-scale public datasets exist [12], [13], [14], most of the current V2V perception algorithms conduct experiments based on their customized data [15], [16], [17]. These datasets are either too small in scale and variance or they are not publicly available. Consequently, there is no large-scale dataset suitable for benchmarking distinct V2V perception algorithms, and such deficiency will preclude further progress in this research field.

To address this gap, we present OPV2V, the first large-scale Open Dataset for Perception with V2V communication. By utilizing a cooperative driving co-simulation framework named OpenCDA [18] and CARLA simulator [19], we collect 73 divergent scenes with a various number of connected vehicles to cover challenging driving situations like severe occlusions. To narrow down the gap between the simulation and real-world traffic, we further build a digital town of Culver City, Los Angeles with the same road topology and spawn dynamic agents that mimic the realistic traffic flow on it. Data samples are shown in Fig. 1 and Fig. 2. We benchmark several state-of-the-art 3D object detection algorithms combined with different multi-vehicle fusion strategies. On top of that, we propose an Attentive Intermediate Fusion pipeline to better capture interactions between connected agents within the network. Our experiments show that the proposed pipeline can efficiently reduce the bandwidth requirements while achieving state-of-the-art performance.

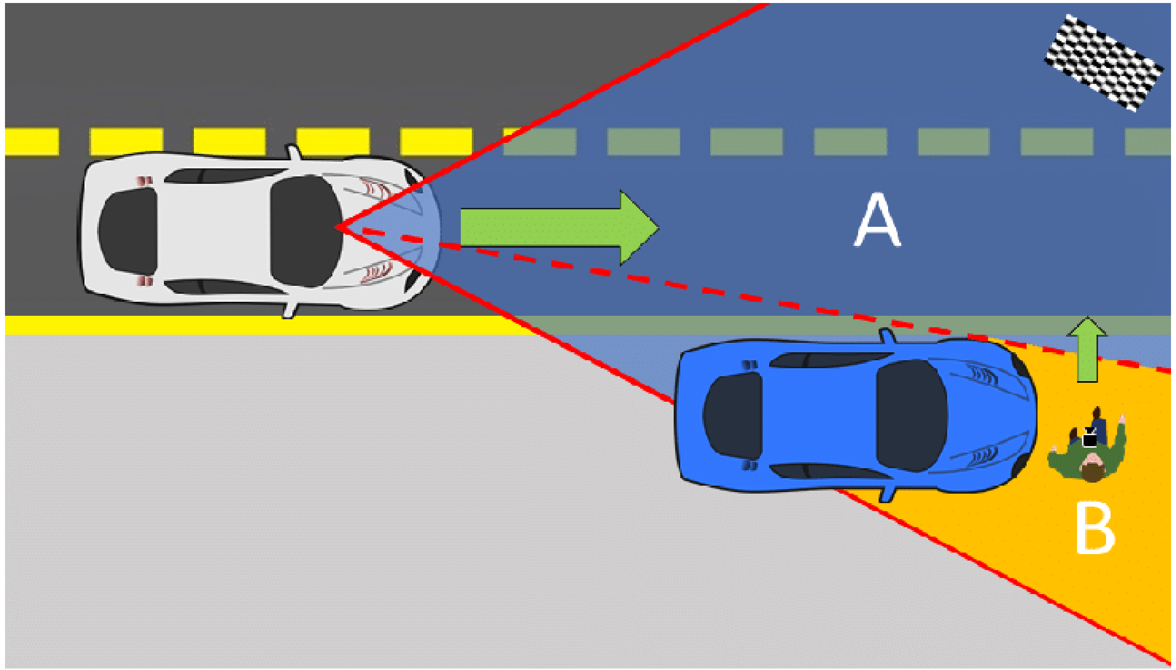
II. Related Work

Vehicle-to-Vehicle Perception: V2V perception methods can be divided into three categories: early fusion, late fusion, and intermediate fusion. Early fusion methods [11] share raw data with CAVs within the communication range, and the ego vehicle will predict the objects based on the aggregated data. These methods preserve the complete sensor measurements but require large bandwidth and are hard to operate in real time [15]. In contrast, late fusion methods transmit the detection outputs and fuse received proposals into a consistent prediction. Following this idea, Rauch et al. [20] propose a Car2X-based perception module to jointly align

Beyond single-vehicle perception

Single-vehicle perception comes with some intrinsic notable limitations:

- Observations can be limited by occlusions, restricted sensor field of view and sensor resolution.
- Perception robustness is affected by sensor errors that can derive from adverse weather or hardware failures.



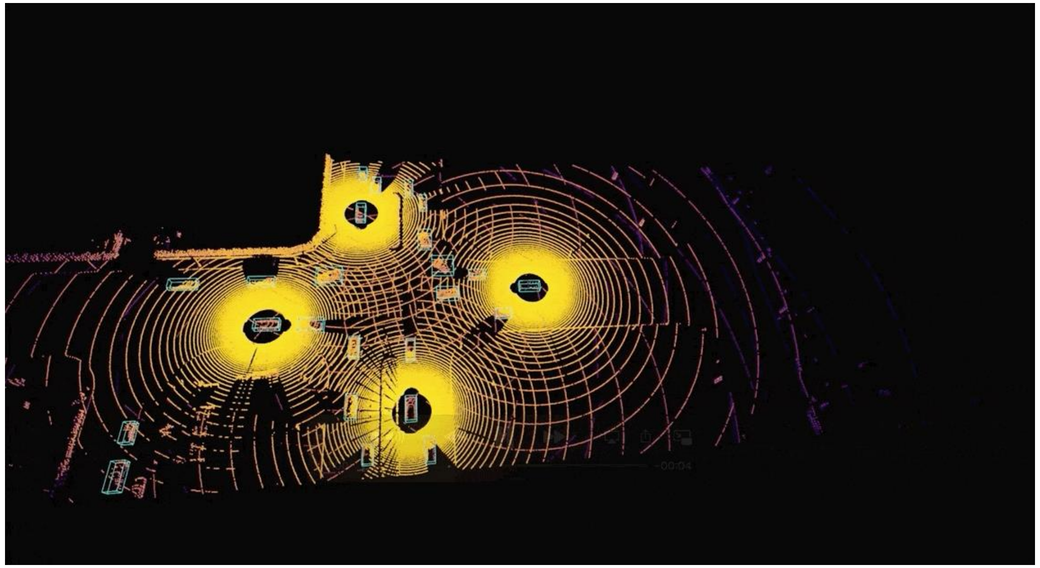
Wang, D., Fu, W., Song, Q., & Zhou, J. (2022). Potential risk assessment for safe driving of autonomous vehicles under occluded vision. *Scientific reports*, *12*(1), 4981.

Image from Palffy, A., Kooij, J. F., & Gavrila, D. M. (2019, June). Occlusion aware sensor fusion for early crossing pedestrian detection. In *2019 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1768-1774). IEEE.

What is cooperative perception?

Cooperative perception has emerged to address the single-vehicle perception limitations by means of interactions among collaborating agents.

- Aims: enhance road safety and user experience trough increased perception quality and robustness.



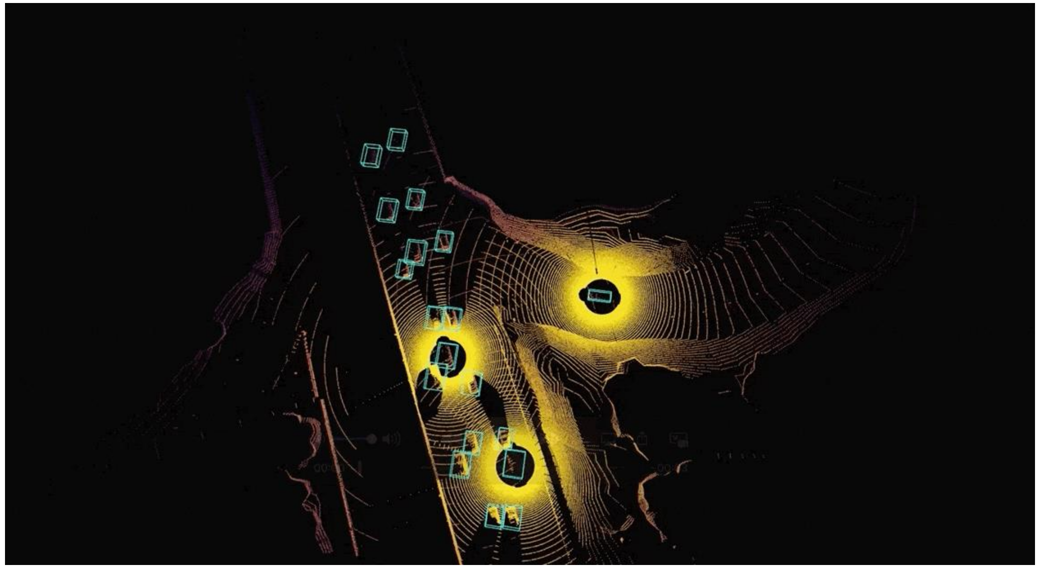


Image from https://mobility-lab.seas.ucla.edu/opv2v/

The Grand Cooperative Driving Challenge 2011

The Grand Cooperative Driving Challenge (GCDC) 2011:

- Aim: **support and accelerate the introduction of cooperative and automated vehicles** through a driving challenge.
- 9 international teams.
- Challenge: perform collaborative platooning to save fuel, improve safety and throughput.

Vehicle platooning: close and coordinated following mechanism of vehicles without any mechanical linkage while mantaining a safe distance, to reduce carbon footprint and traffic congestion, and enhance road safety.

Lauer, M. (2011). Grand cooperative driving challenge 2011 [its events]. *IEEE Intelligent Transportation Systems Magazine*, *3*(3), 38-40.

The Grand Cooperative Driving Challenge 2016

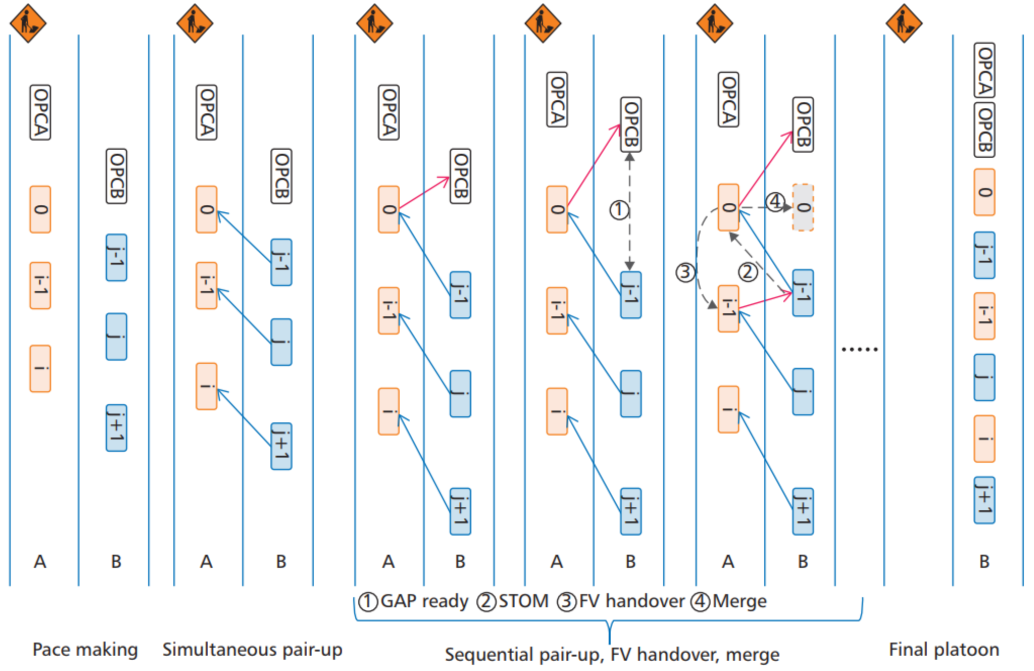
The Grand Cooperative Driving Challenge (GCDC) 2016

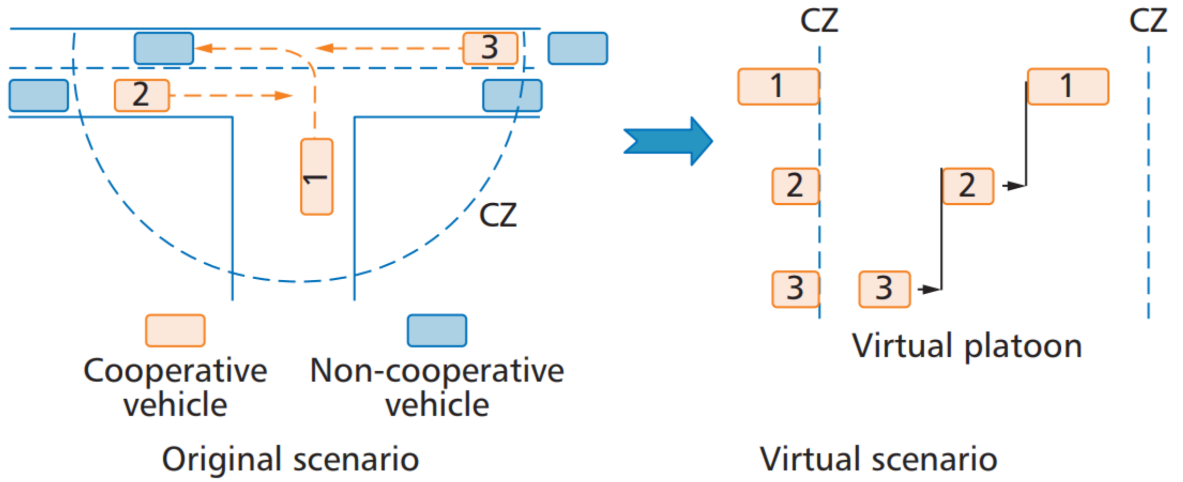
- AIM: to further **boost the introduction of cooperative automated vehicles** by means of wireless communications.
- Three scenarios requiring close cooperation among teams through wireless communication:
  - Cooperative platoon merge;
  - Cooperative intersection passing;
  - Passage of an emergency vehicle.

Englund, C., Chen, L., Ploeg, J., Semsar-Kazerooni, E., Voronov, A., Bengtsson, H. H., & Didoff, J. (2016). The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications*, *23*(4), 146-152.

The Grand Cooperative Driving Challenge 2016

GCDC 2016 challenges:





Cooperative platoon merge: two platoons driving on a motorway must merge into one platoon due to an upcoming construction site.

Cooperative intersection passing: vehicle 1 transmits its intention to turn left. The cooperative vehicles's goal is to facilitate intersection passing for vehicle 1.

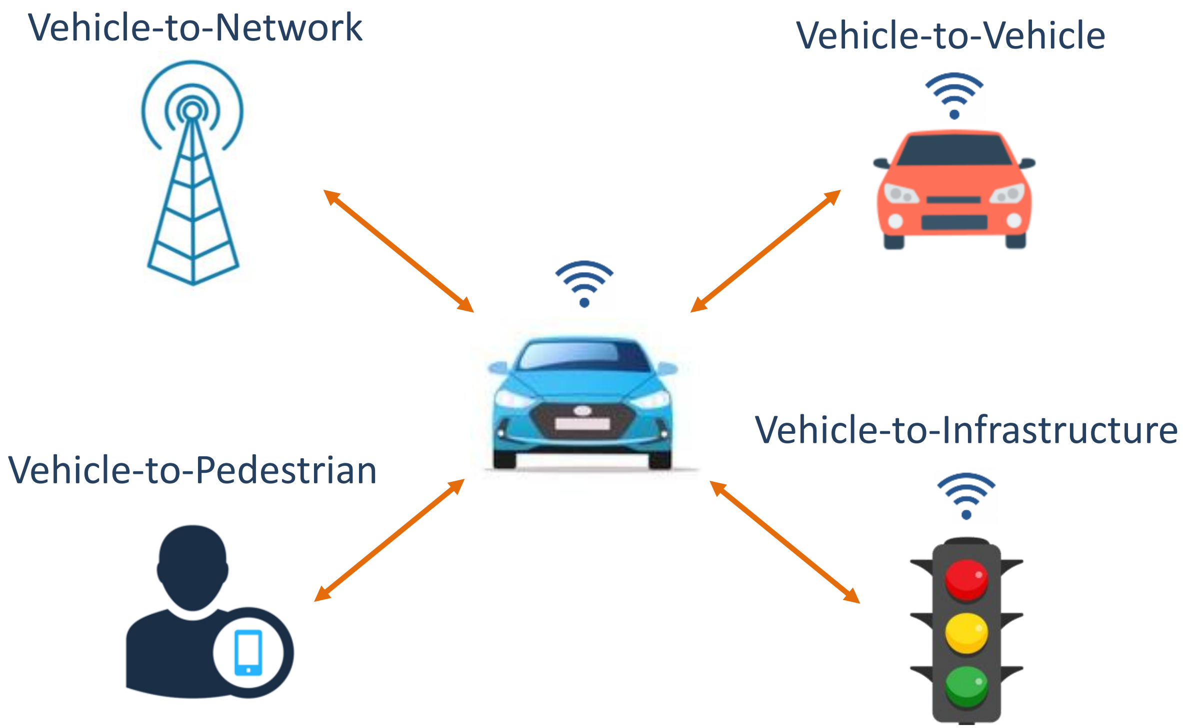
Englund, C., Chen, L., Ploeg, J., Semsar-Kazerooni, E., Voronov, A., Bengtsson, H. H., & Didoff, J. (2016). The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles. *IEEE Wireless Communications*, *23*(4), 146-152.

Enabling cooperative perception with wireless communications

Cooperative perception can currently be enabled by 5th Generation (5G) Cellular Vehicle-to-Everything (C-V2X) communications, including:

- Vehicle-to-Vehicle (V2V)
- Vehicle-to-Infrastructure (V2I)
- Vehicle-to-Network (V2N)
- Vehicle-to-Pedestrian (V2P)

In the cooperative automotive framework, the connected agents are usually referred to as CAVs (Connected Autonomous Vehicles).



5GAA Automotive Association: https://5gaa.org/; 3GPP: https://www.3gpp.org/

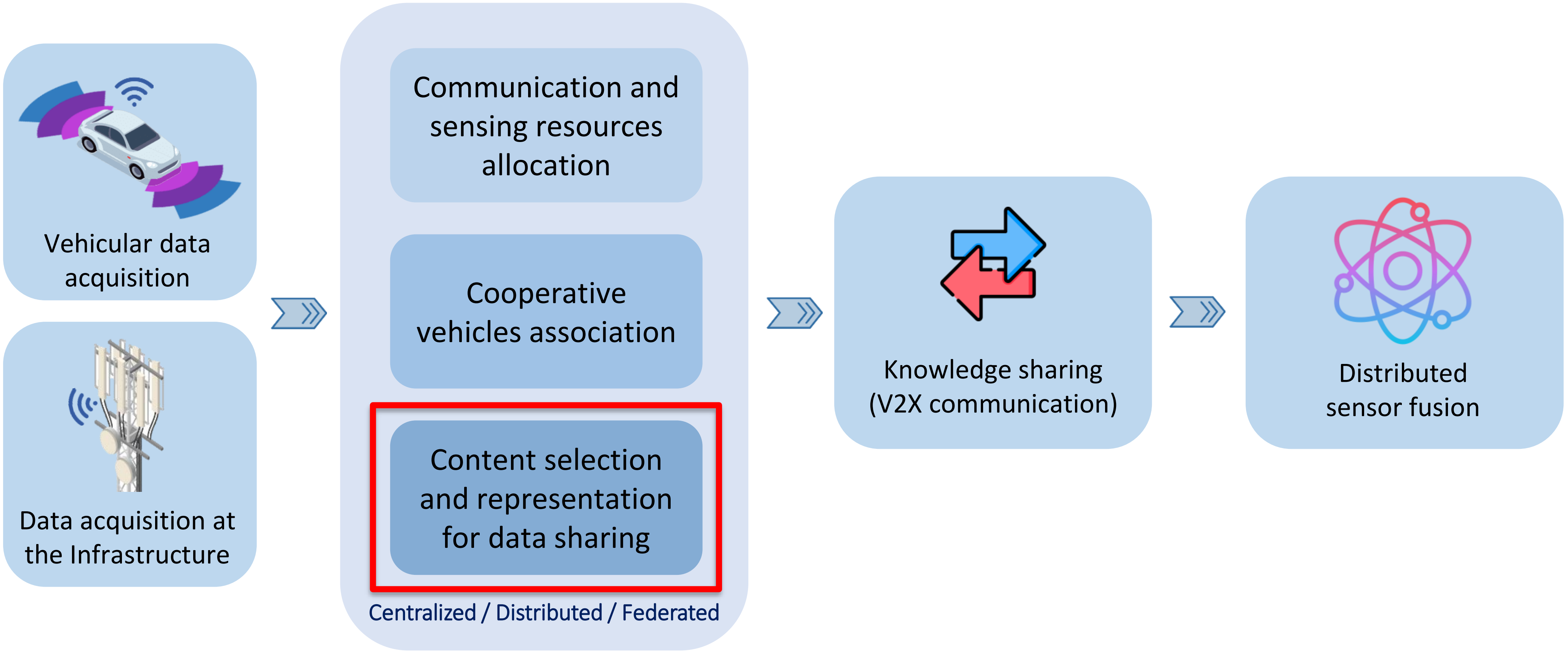
5GAA. White Paper C-V2X Use Cases: Methodology, Examples and Service Level Requirements. https://5gaa.org/content/uploads/2019/07/5GAA_191906_WP_CV2X_UCs_v1-3-1.pdf

Open challenges in V2X for cooperative perception

- Which point selection and representation strategies can be devised to cope with limited communication resources?
- How can vehicles work together to solve security issues ensuring that V2V communications are secure?
- How can V2X communications ensure that messages arrive fast enough to inform the AV's decision-making system?
- What assumptions on the CAV sensor data can be made in a dynamic vehicular environment?
- What are the scalability limits of cooperative perception and how do they impact on the coordination of the driving movements of a large number of CAVs?

Balkus, S. V., Wang, H., Cornet, B. D., Mahabal, C., Ngo, H., & Fang, H. (2022). A survey of collaborative machine learning using 5G vehicular communications. *IEEE Communications Surveys & Tutorials*, *24*(2), 1280-1303.

The cooperative perception problem(s)



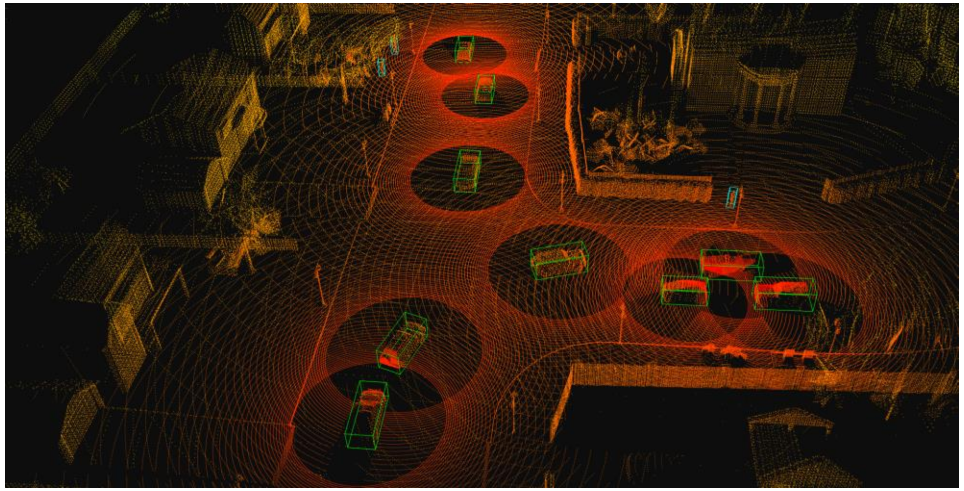
We will focus on point clouds representation for data sharing

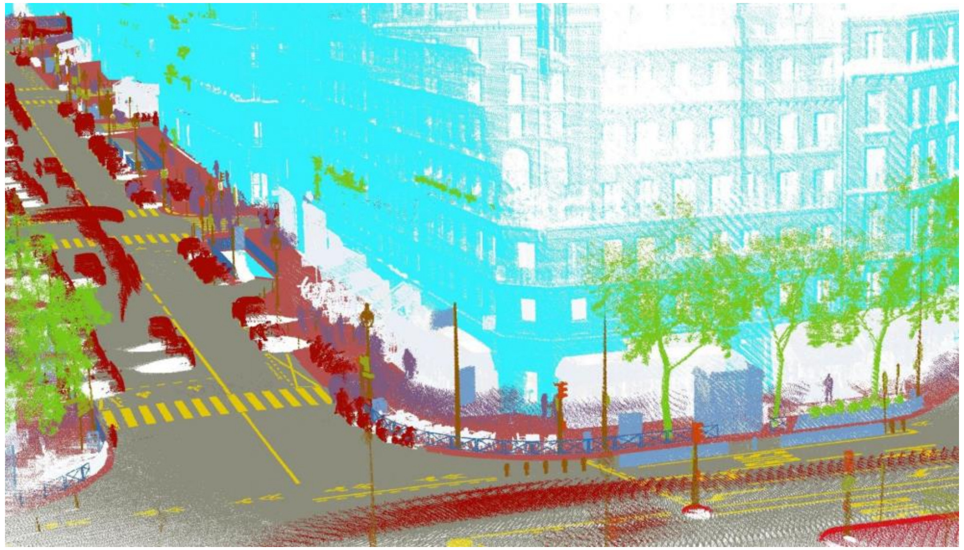
Why is point cloud cooperative perception useful?

Among the main point cloud processing downstream tasks to which cooperative perception is beneficial are:

- 3D object detection
- 3D object tracking
- Semantic and instance point cloud segmentation
- Map generation
- Localization

We will focus on 3D object detection.



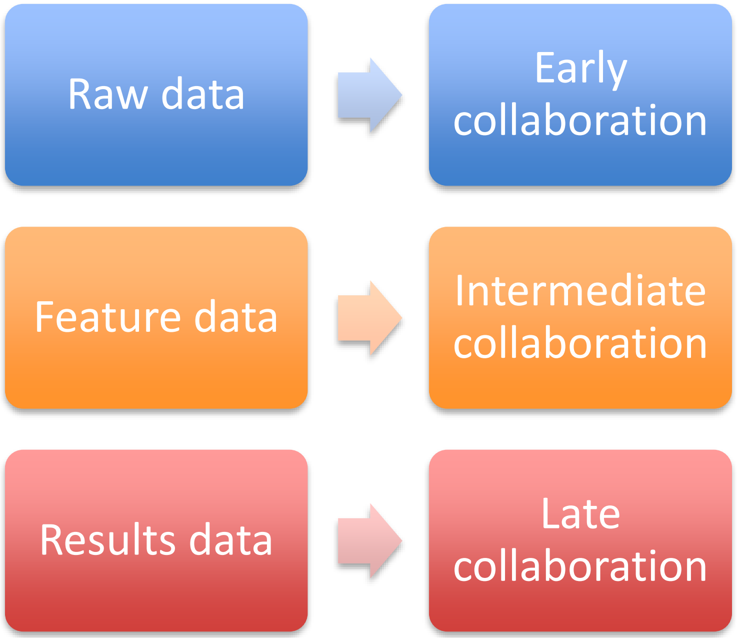


From raw data… to results

Typically, three types of perception data are generated from heterogenous perception nodes:

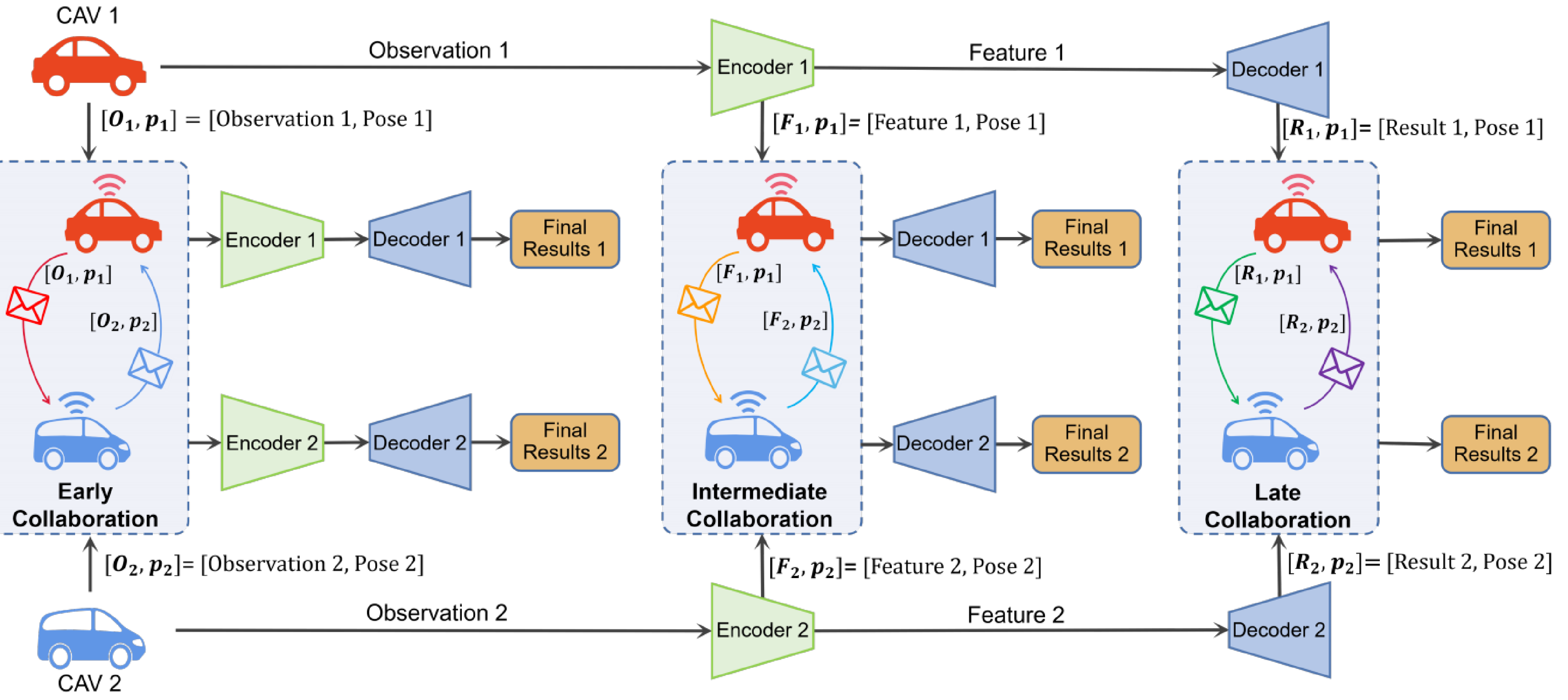
- Raw sensor data (e.g., camera RGB images or LiDAR point cloud data);
- Feature data, containing meaningful features extracted by classic statistical methods or, usually, based on deep learning (e.g., through neural networks);
- Results data, containing the results of the semantic perception information (like the bounding boxes coordinates ofr a detected object or its classification).

A collaborative scheme among CAVs can be associated to each of the perception data types



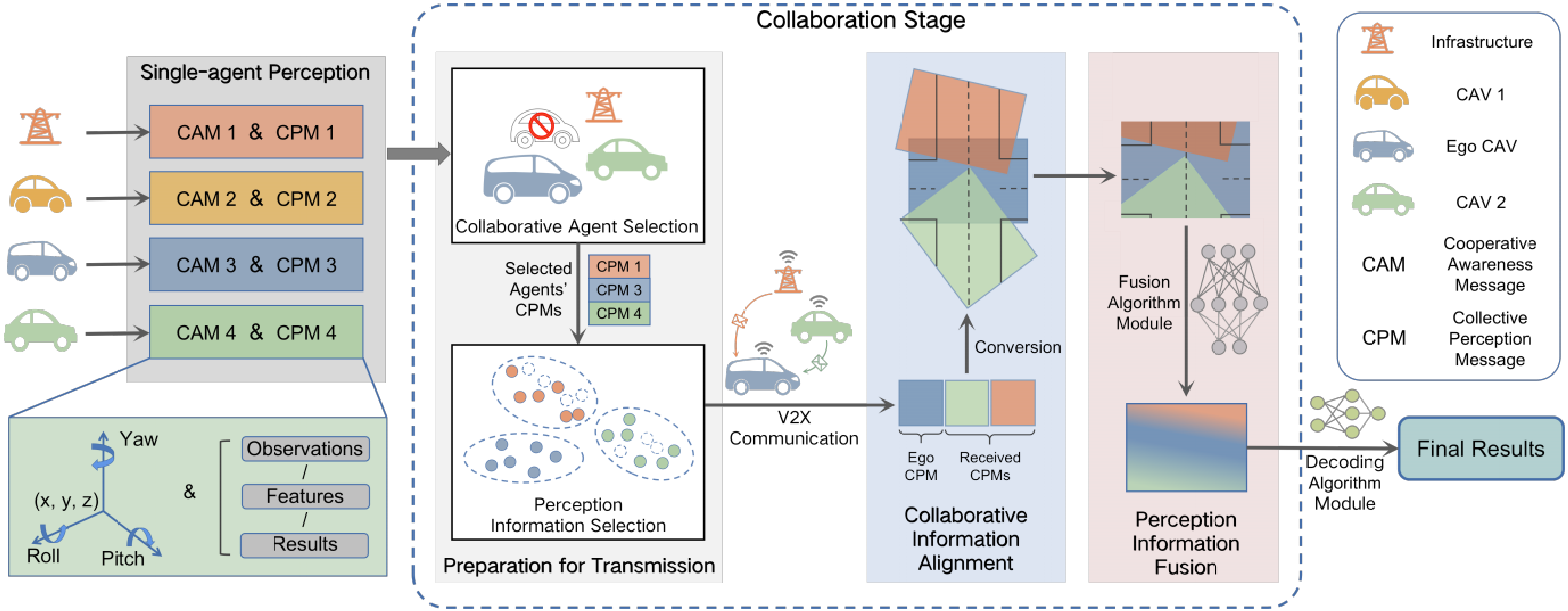
Bai, Z., Wu, G., Barth, M. J., Liu, Y., Sisbot, E. A., Oguchi, K., & Huang, Z. (2022). A survey and framework of cooperative perception: From heterogeneous singleton to hierarchical cooperation. *arXiv preprint arXiv:2208.10590*.

Vehicle collaboration schemes



Huang, T., Liu, J., Zhou, X., Nguyen, D. C., Azghadi, M. R., Xia, Y., & Sun, S. (2023). V2X cooperative perception for autonomous driving: Recent advances and challenges. *arXiv preprint arXiv:2310.03525*.

Vehicles collaboration pipeline



Huang, T., Liu, J., Zhou, X., Nguyen, D. C., Azghadi, M. R., Xia, Y., & Sun, S. (2023). V2X cooperative perception for autonomous driving: Recent advances and challenges. *arXiv preprint arXiv:2310.03525*.

Early collaboration (share the point clouds)

The CAVs share the collected raw sensor data at the pre-processing stage.

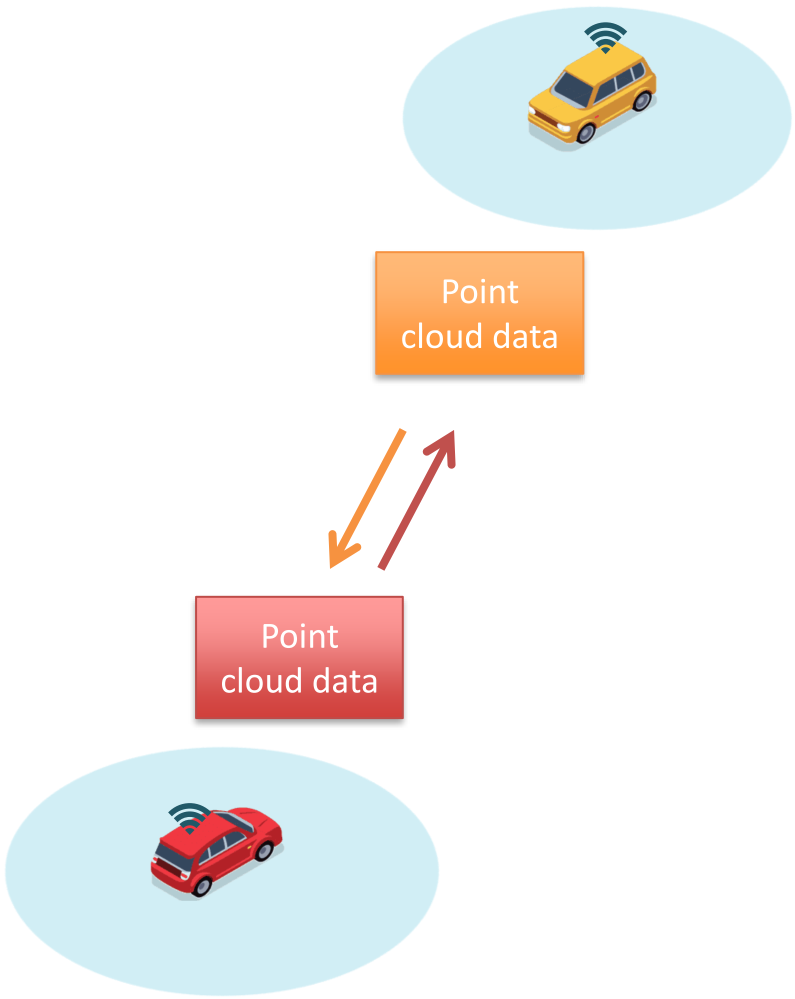
Pros:

- Raw data is shared and integrated to build a holistic view.
- Effectively copes with occlusions and long-range obstacles acquired in single-vehicle perception.

Cons:

- Low tolerance to noise and transmission delays.
- Constrained by the communication resources.

Example: Cooper (Chen et al.)



Chen, Q., Tang, S., Yang, Q., & Fu, S. (2019, July). Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 514-524). IEEE.

Intermediate collaboration (share the features)

The CAVs extract features from the acquired raw sensor data and share the features.

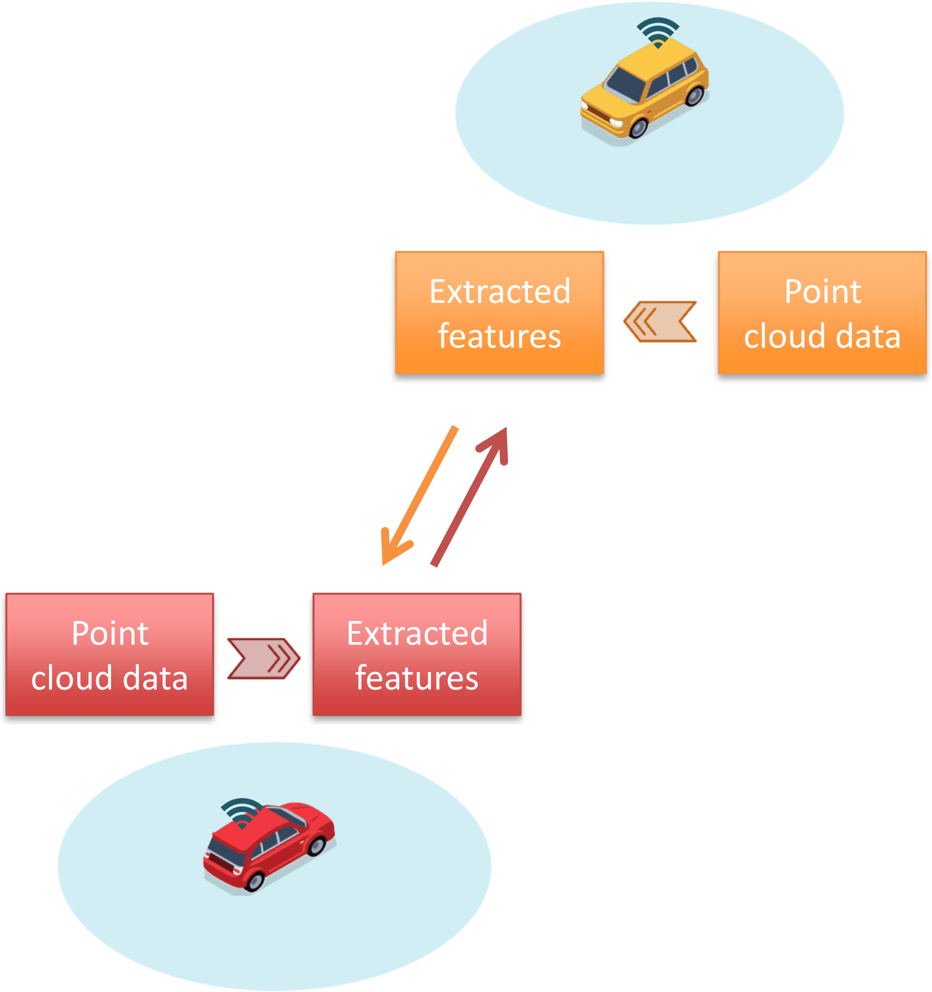
Pros:

- High tolerance to noise, transmission delays with respect to early collaboration.
- More robust to differences between nodes and sensor models.

Cons:

- Requires suitable model training.
- It is complex to find a systematic method for model design.

Examples: F-Cooper, V2VNet, OPV2V, Pillargrid



Bai, Z., Wu, G., Barth, M. J., Liu, Y., Sisbot, E. A., & Oguchi, K. (2022, October). Pillargrid: Deep learning-based cooperative perception for 3d object detection from onboard-roadside lidar. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1743-1749). IEEE.

Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M. H., & Ma, J. (2022, October). V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision* (pp. 107-124). Cham: Springer Nature Switzerland.

Late collaboration (share the results)

The CAVs process the perceived raw data and share the perception results.

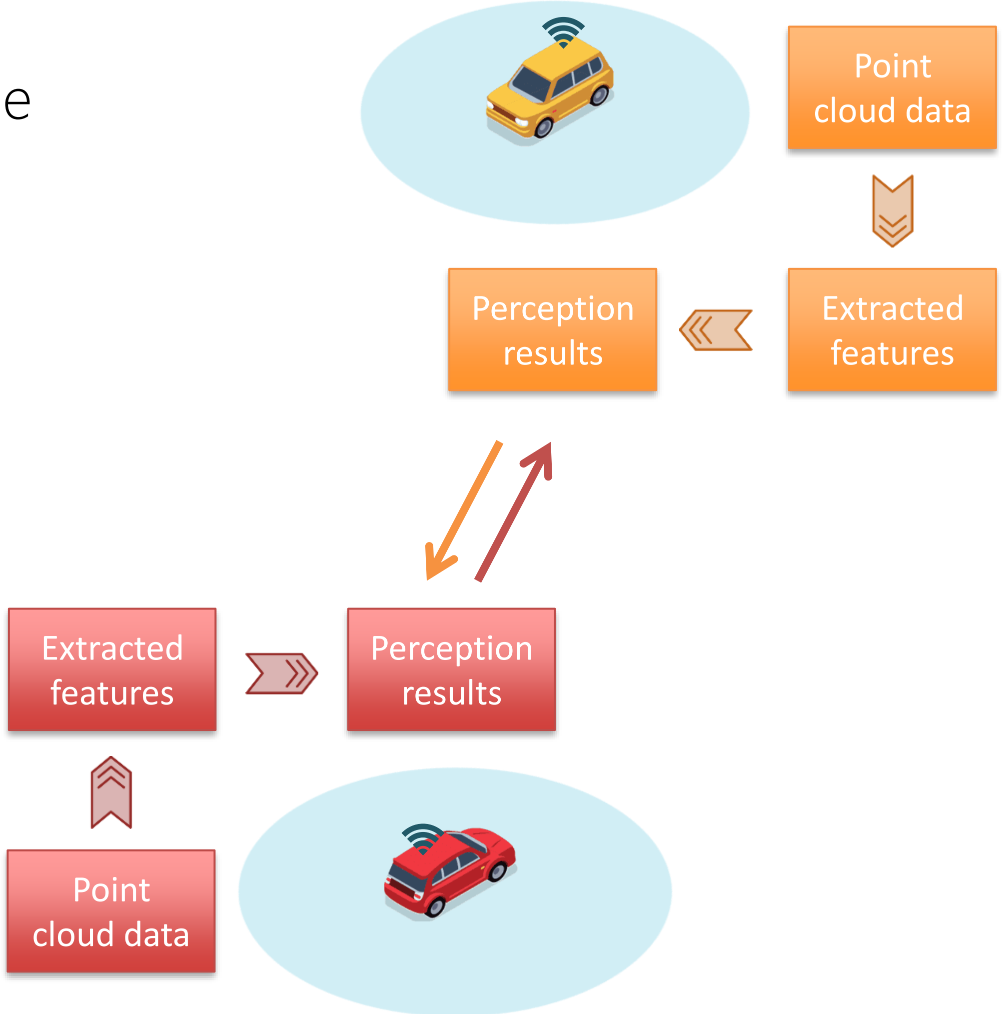
Pros:

- Easier to design and deploy in a real-world cooperative perception system.
- Can achieve better real-time performance.

Cons:

- Limited by wrong perception results or differences between the sources.
- Accuracy is usually lower with respect to early and intermediate collaboration.

Examples: Rauch et al., Zhang et al.



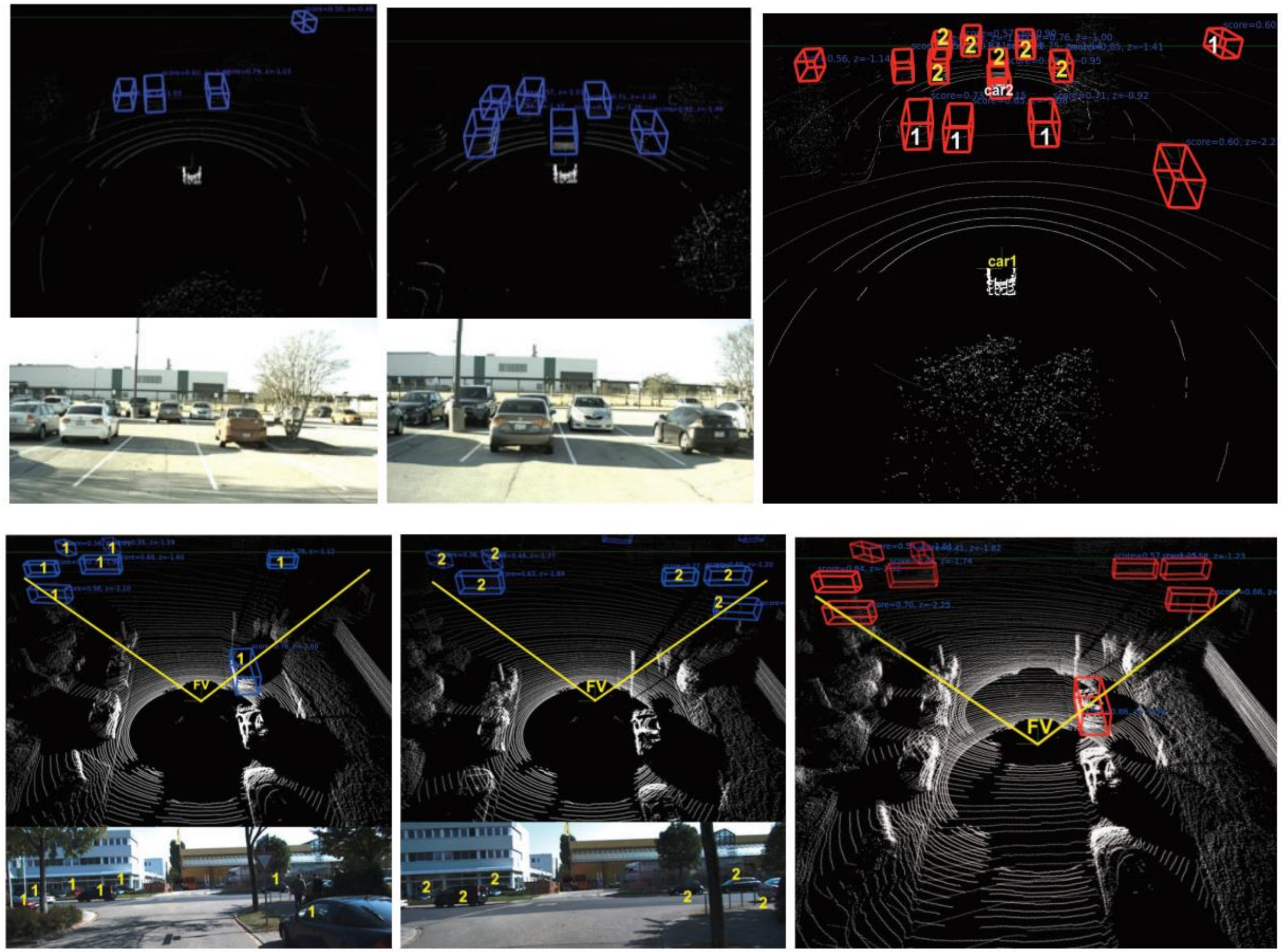
Rauch, A., Klanner, F., Rasshofer, R., & Dietmayer, K. (2012, June). Car2x-based perception in a high-level fusion architecture for cooperative perception systems. In *2012 IEEE Intelligent Vehicles Symposium* (pp. 270-275). IEEE.

Zhang, Z., Wang, S., Hong, Y., Zhou, L., & Hao, Q. (2021, May). Distributed dynamic map fusion via federated learning for intelligent networked vehicles. In *2021 IEEE International conference on Robotics and Automation (ICRA)* (pp. 953-959). IEEE.

Cooper - Cooperative Perception for CAVs on 3D point clouds

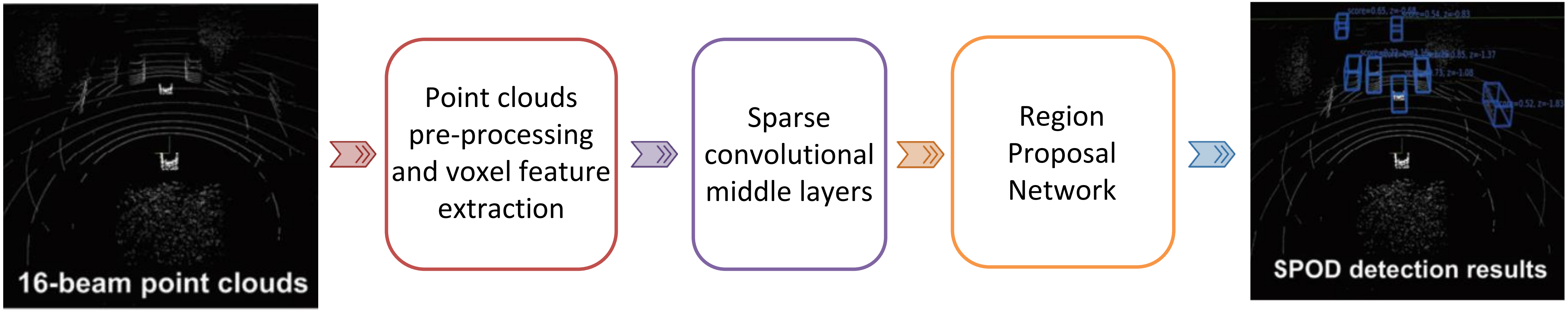
Cooper is an early collaboration system which aims to improve the detection performance on low-density point clouds.

- Introduces Sparse Point-cloud Object Detection (SPOD) method to increase object detection performance in low-density point clouds.
- The transmission of low-density point clouds (e.g., from 16-channels LiDARs) relaxes the communication bandwith requrements.
- The authors collect a real-world dataset (T&J dataset) explicitly designed to assess object detection in cooperative perception conditions.



Chen, Q., Tang, S., Yang, Q., & Fu, S. (2019, July). Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 514-524). IEEE.

Cooper - Sparse Point-cloud Object Detection



- Input 3D lidar points are represented by a tuple of cartesian coordinates and reflection value (x, y, z, r).
- In the pre-processing, point clouds are projected onto a sphere to generate a dense representation.
- Voxel-wise features are extracted by means of Voxelnet.
- Sparse convolutional middle layers are applied.
- The Region Proposal Network is built using the SSD object detection architecture.

Liu, Wei, et al. "Ssd: Single shot multibox detector." *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016.*

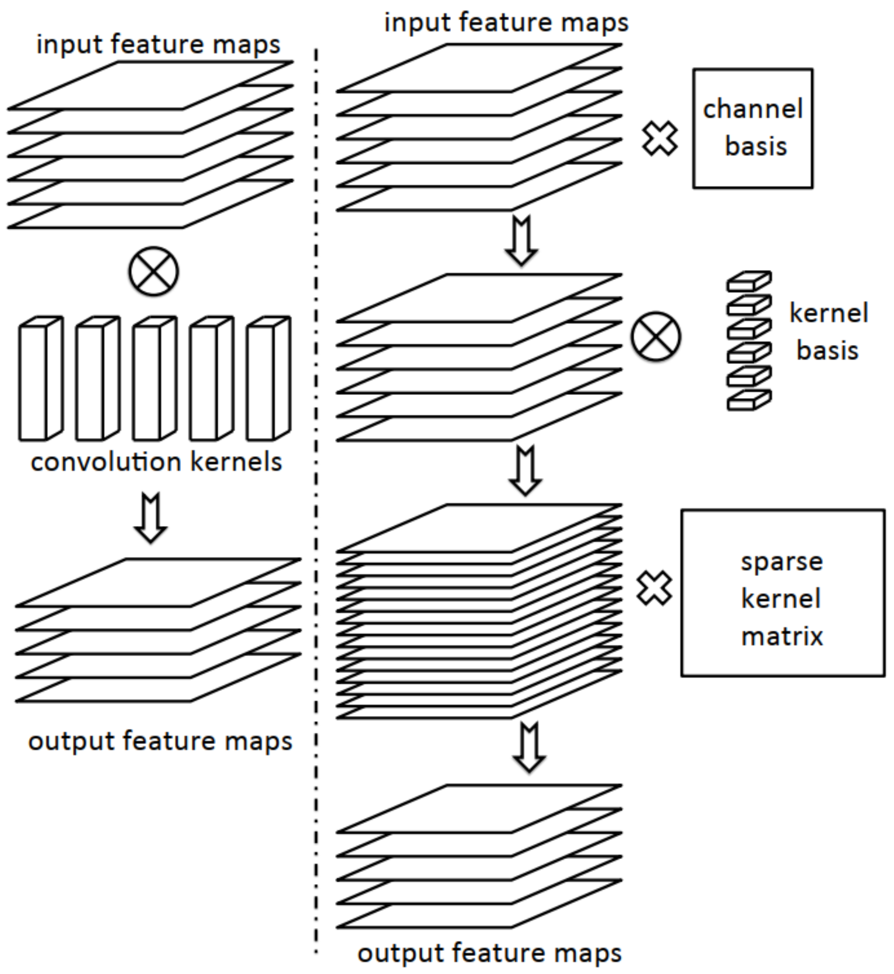
Zhou, Yin, and Oncel Tuzel. "Voxelnet: End-to-end learning for point cloud based 3d object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Chen, Q., Tang, S., Yang, Q., & Fu, S. (2019, July). Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 514-524). IEEE.

Cooper - Sparse Convolutional Neural Networks

Sparse Convolutional Neural Networks tackle the reduction of computational complexity in common CNN models

- Introduce **sparse decomposition** in the CNN filtering steps.
- Sparse decomposition can significantly cut down the cost of computation while maintaining accuracy.
- Each sparse convolutional layer can be performed with a few convolution kernels followed by a **sparse matrix multiplication**.



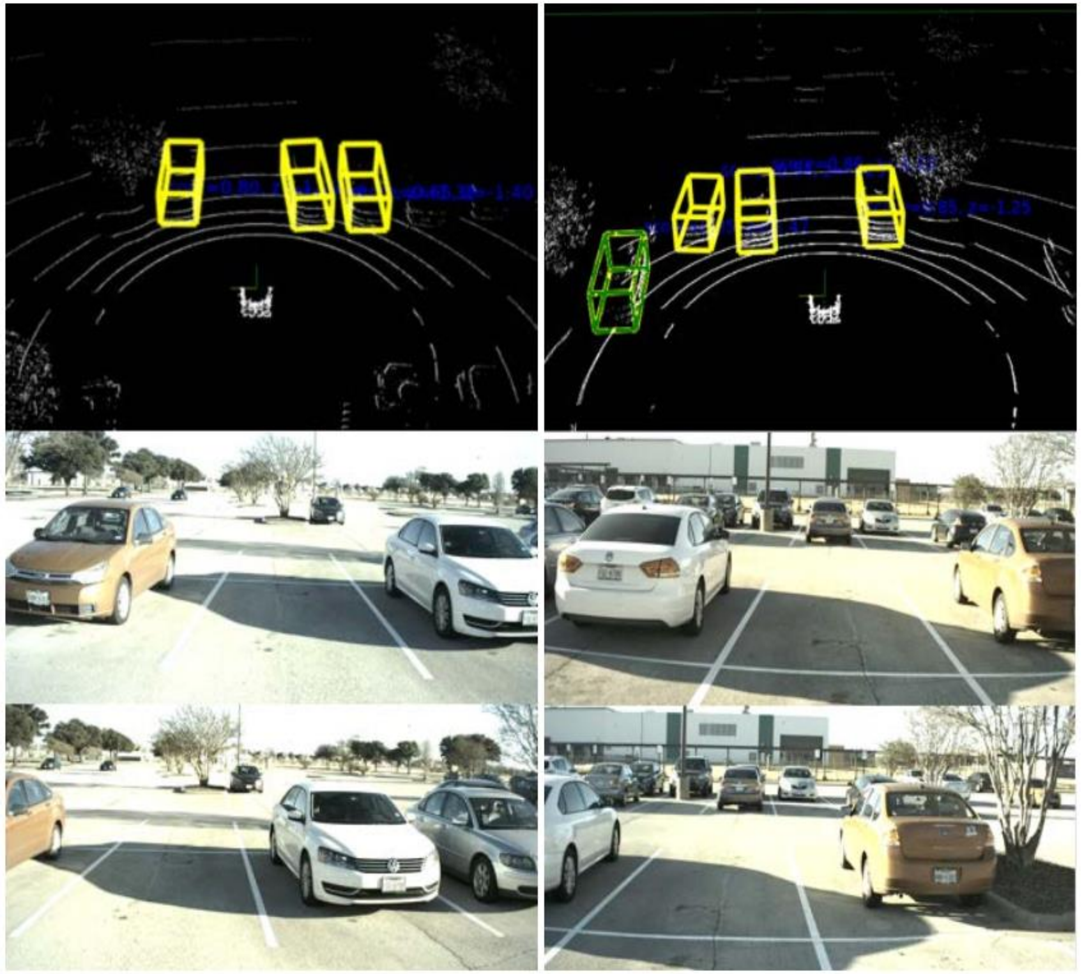
Liu, Baoyuan, et al. "Sparse convolutional neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Chen, Q., Tang, S., Yang, Q., & Fu, S. (2019, July). Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 514-524). IEEE.

F-Cooper – Feature-based cooperative perception

F-Cooper is an intermediate collaboration method introducing **feature-level data fusion**.

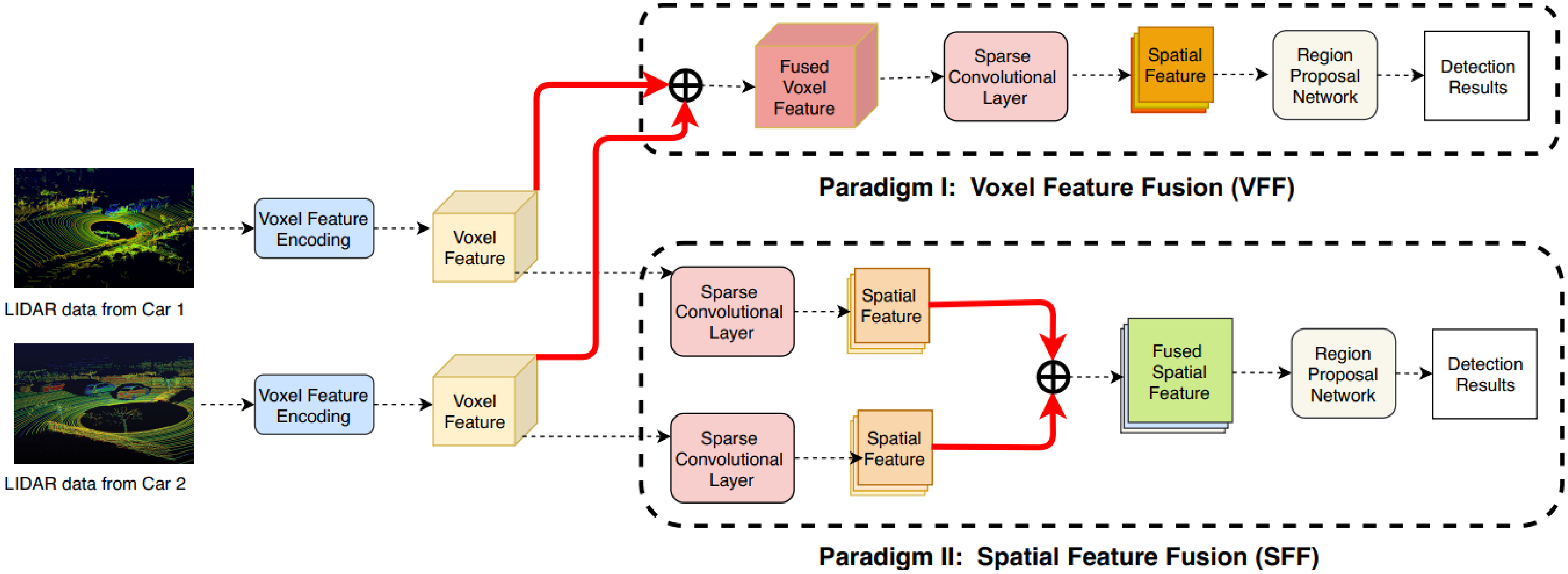
- Shows that feature fusion allows to achieve higher object detection performance.
- Achieves faster edge computing with a low communication delay (owing to the features smaller size w.r.t. the raw point cloud data).



Model code and dataset: https://github.com/Aug583/F-COOPER

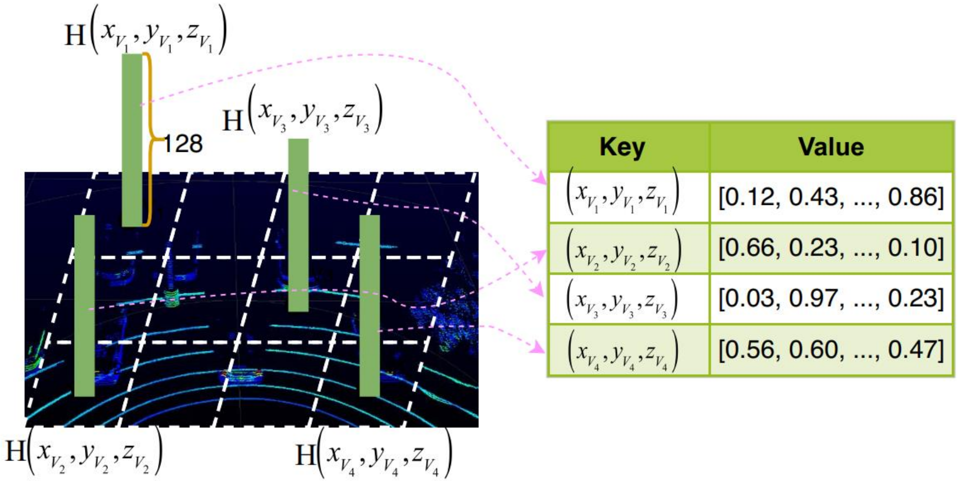
Chen, Qi, et al. "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds." *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. 2019.

F-Cooper – Architecture



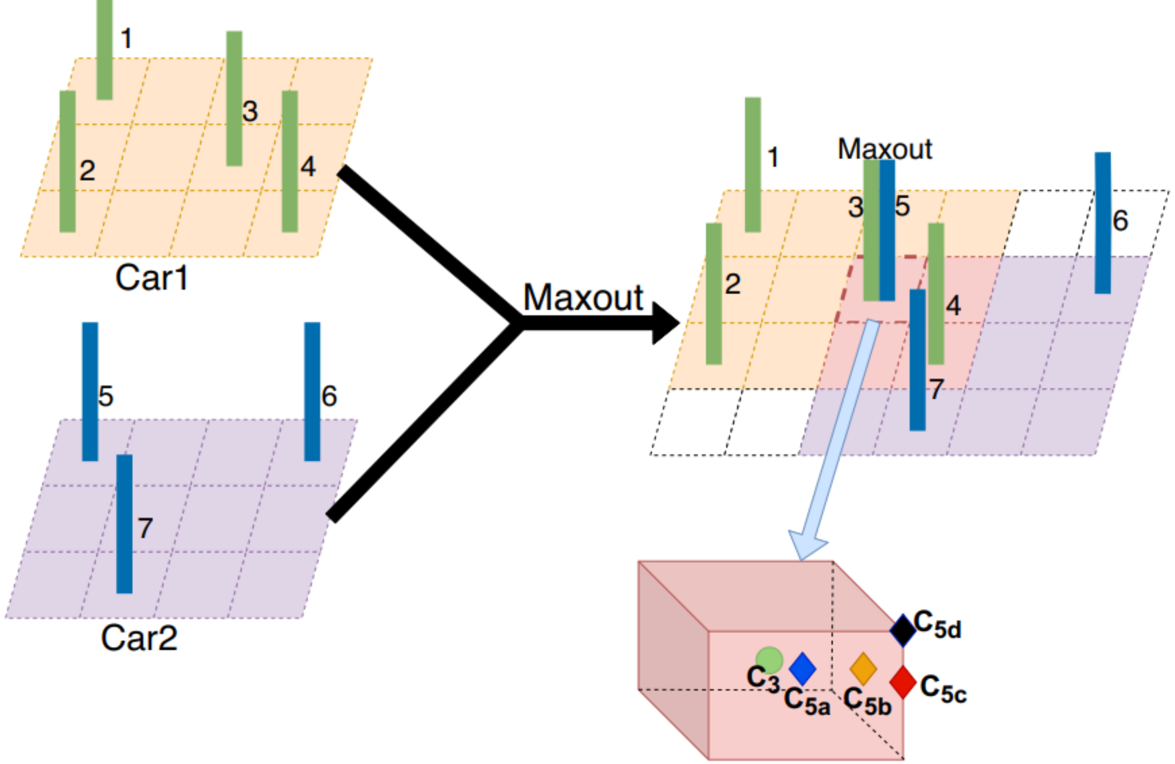
F-Cooper – Voxel Features fusion

Voxel features



- A feature is associated to each non-empty point cloud voxel.
- Voxels containing more than 35 points are randomly sampled.
- The points in a voxel are provided to the Voxel Feature Encoding (VFE) layer, which produces a 128-dimensional vector

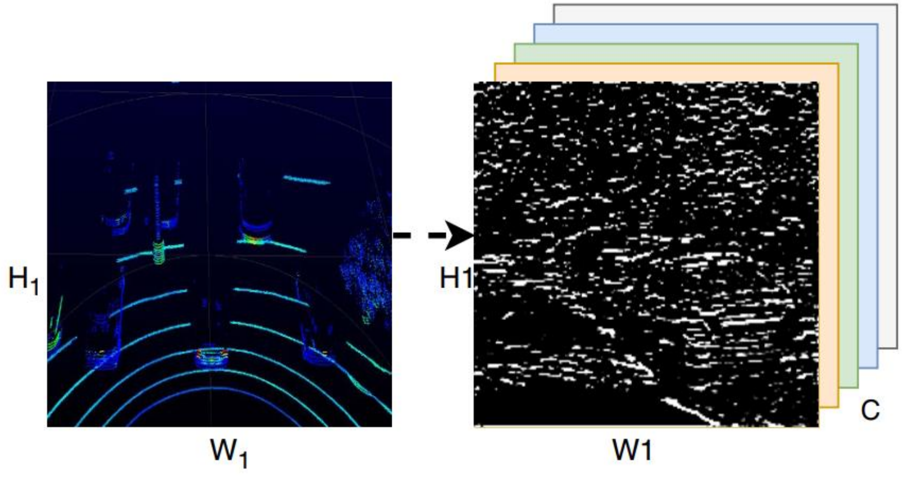
Voxel features fusion



Voxels sharing the same location are fused by max function.

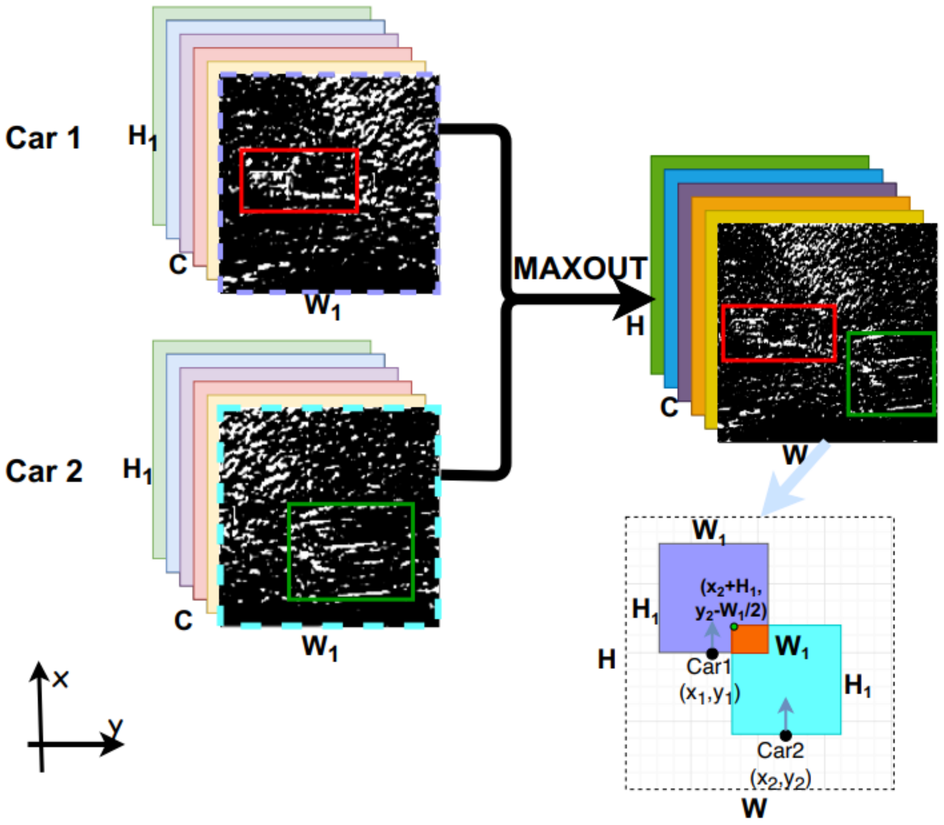
F-Cooper – Spatial Features fusion

Spatial feature maps



- The spatial feature maps are generated by a set of sparse convolutional layers.
- (H_1, W_1) is the size of the LiDAR bird-eye view.
- C is the number of output channels of the last sparse convolutional layer.

Spatial features fusion



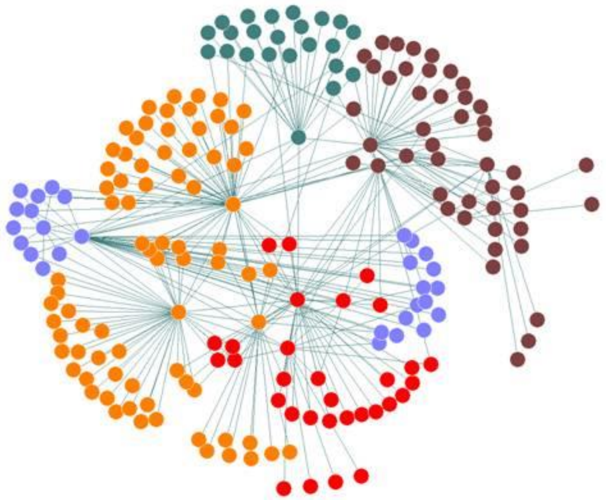
Spatial features are fused channel-wise using maxout.

F-Cooper – Results

(a) Car 1 (Receiver) (b) Car 2 (Sender) (c) Fusion Detection Result on Car 1

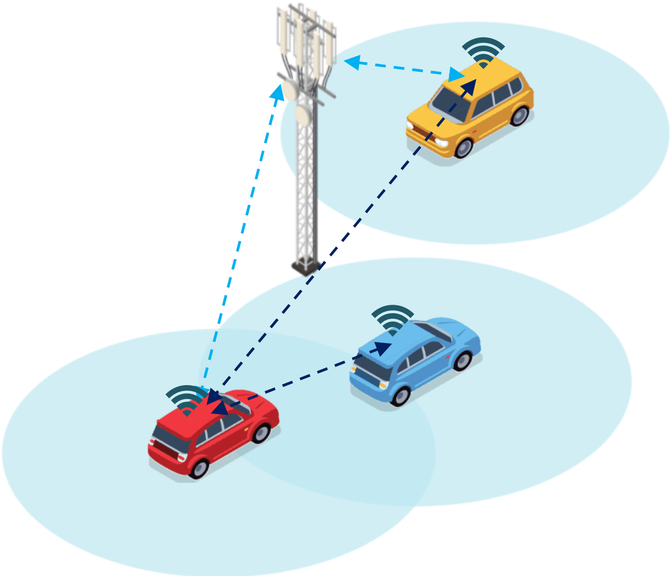
Chen, Qi, et al. "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds." *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. 2019.

Machine Learning on Graphs

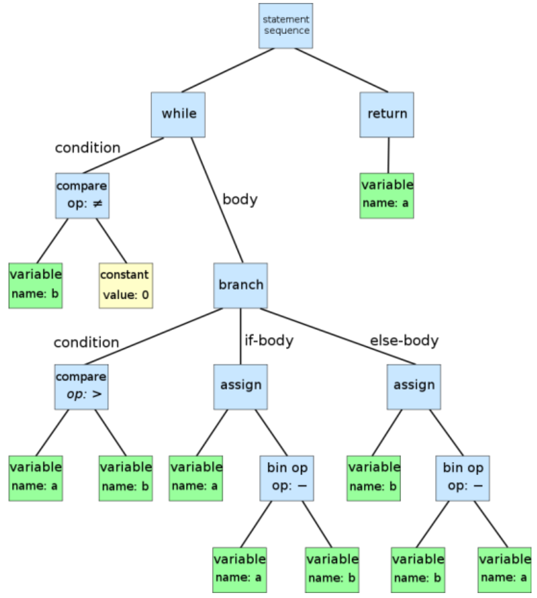


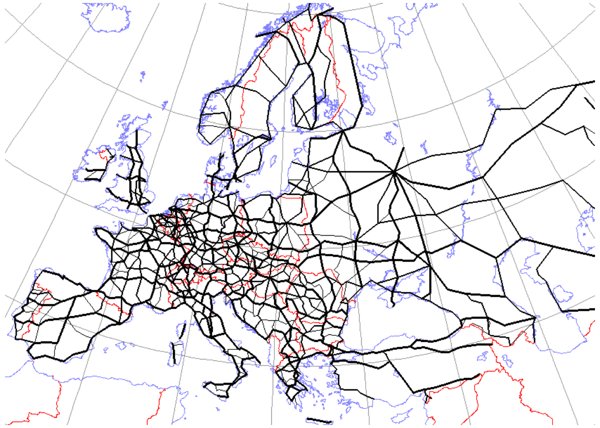
Social networks

V2V communication networks

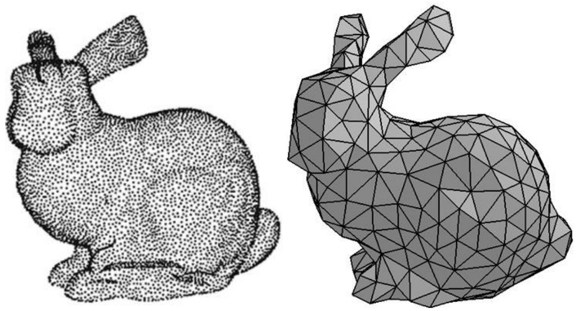


Code graphs

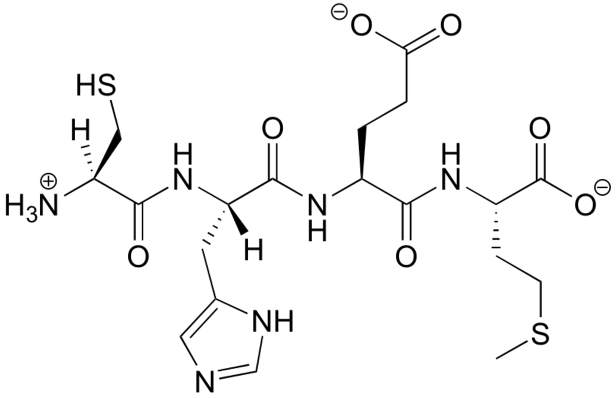




Road networks



3D data processing
(e.g., point clouds, meshes)

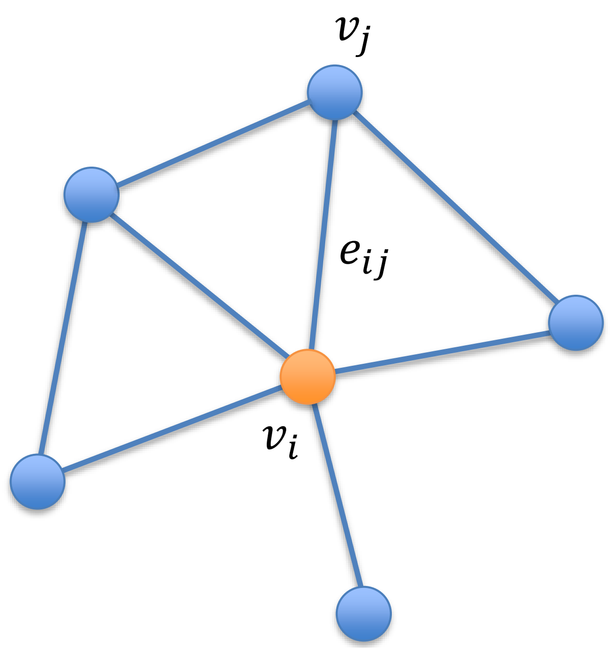


Drug design/
Molecular modelling

Graph structured data

A graph $G = (V, E)$ is represented by

- A set of nodes (or vertices) $v_i \in V$
- A set of edges $e_{ij} = (v_i, v_j) \in E$
- The neighborhood of a node v is the set of nodes directly connected to v: $N(v) = \{u \in V \mid (v, u) \in E\}$



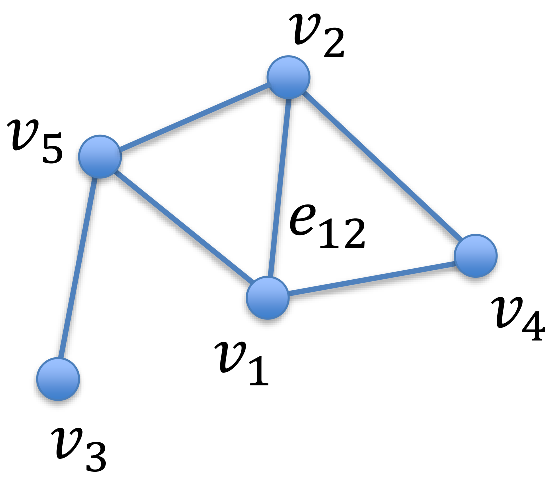
Directed graph: its edges are directed from one node to the other.
Undirected graph: a pair of edges with inverse direction is defined among all connected nodes.

Graph representation – Adjacency matrix

The adjacency matrix A of a graph $G = (V, E)$ with n nodes is an $n \times n$ matrix with:

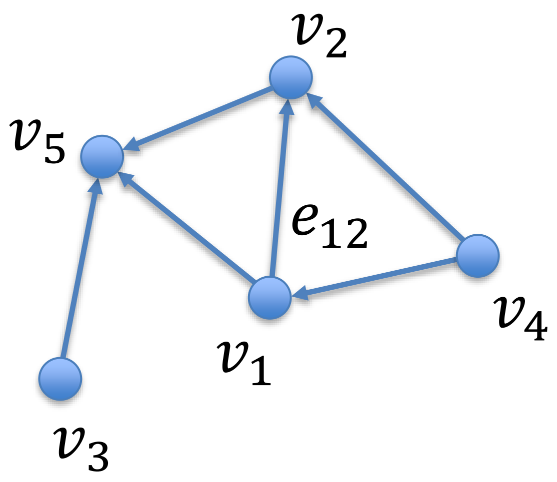
- $A_{ij} = 1$, if $e_{ij} \in E$
- $A_{ij} = 0$, otherwise

Undirected graph



$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Directed graph

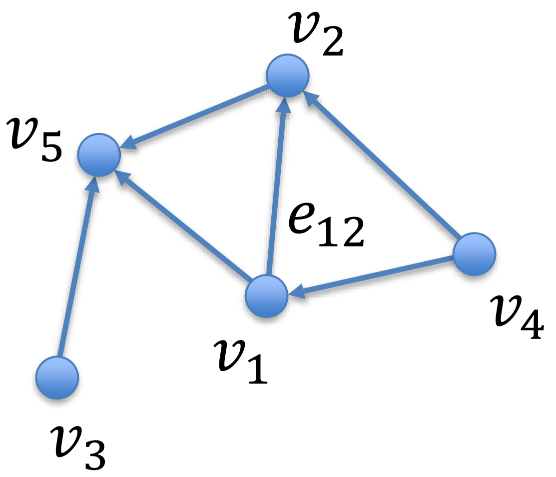


$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Graph representation – Adjacency list

The adjacency list reports for each node the list of nodes it is connected to

- It is more efficient for some applications, e.g., in large and sparse networks.
- It allows to retrieve all the neighbors in a single lookup.



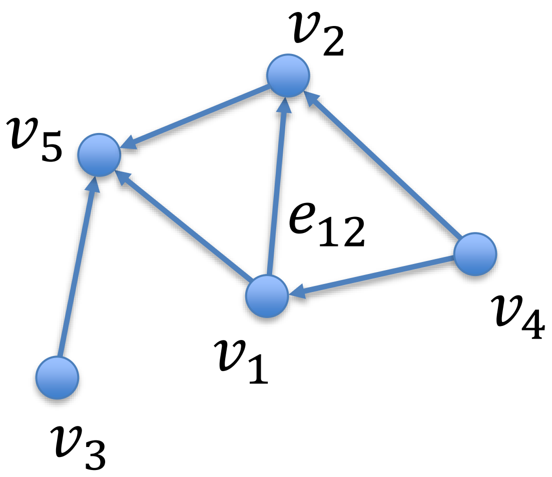
Adjacency list

1: [2, 5]
2: [5]
3: [5]
4: [1, 2]
5: []

Graph representation – Edge list

The edge list is the list of all the edges in the graph.

- It requires an additional step to retrieve the neighborhood of a node.
- It is more efficient for the message-passing interface.

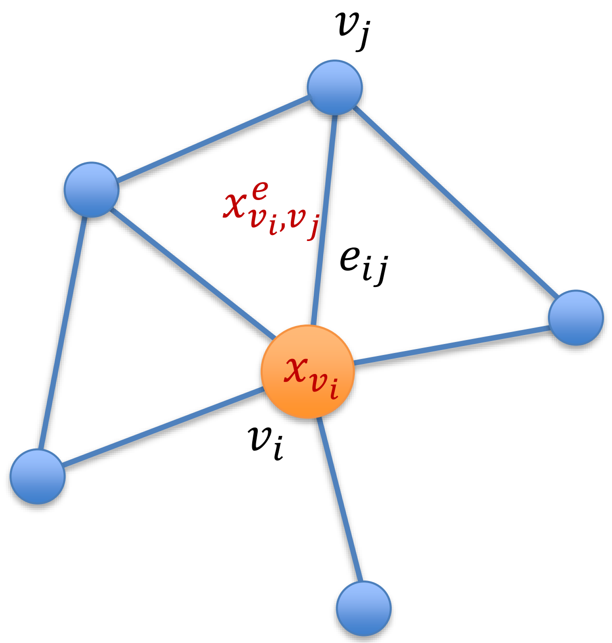


Edge list

(1, 2)
(1, 5)
(2, 5)
(3, 5)
(4, 1)
(4, 2)

Attributed graphs

We consider attributed graphs, where a feature vector can be associated to each node or to each edge.



Node features

$$x_{v_i} \in \mathbb{R}^d, \text{ for } v_i \in V$$

Edge features

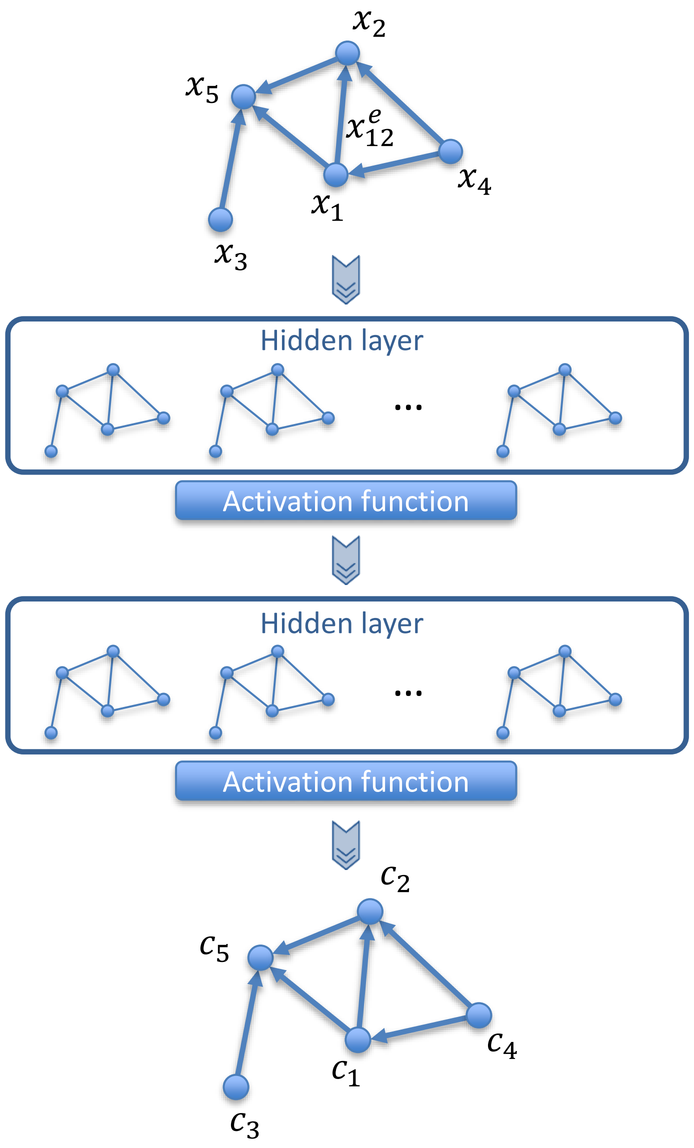
$$x^e_{v_i,v_j} \in \mathbb{R}^c, \text{ for } e_{ij} = (v_i, v_j) \in E$$

What is a Graph Neural Network?

A Graph Neural Network (GNN) is a neural network architecture suited to effectively process graph data.

From several domains, graph data comes with complex relationships and object interdependencies, posing challenges on existing ML algorithms.

GNNs exploit the potentials of deep learning processing while accounting for the features of graph data.



Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Graph Neural Networks (GNNs)

Recurent GNNs

Pioneer works on GNNs that inspired later research on Convolutional GNNs.

AIM: Learn node representations exploiting recurrent neural architectures.

Convolutional GNNs

Generalize the convolution operation from grid data to graph data.

AIM: Generate a nodes' representation aggregating its own features and neighbors' features

Graph autoencoders

Unsupervised learning frameworks.

AIM: Encode nodes/graphs into a latent vector space and reconstruct graph data from the latent encoding.

Spatial-Temporal GNNs

Consider spatial and temporal dependences at the same time.

AIM: Learn hidden patterns from spatial-temporal graphs.

Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

GNN downstream tasks

The outputs of a GNN can focus on different analytic tasks operating at different levels:

- Node level: outputs relate to node regression and node classification tasks.
- Edge level: outputs relate to edge classification and link prediction tasks.
- Graph level: outputs relate to the graph classification task.

Node level – E.g., node classification

c_i: vector of classification scores for node v_i

Edge level – E.g., link prediction

s_{ij}: link activation score for edge e_{ij}

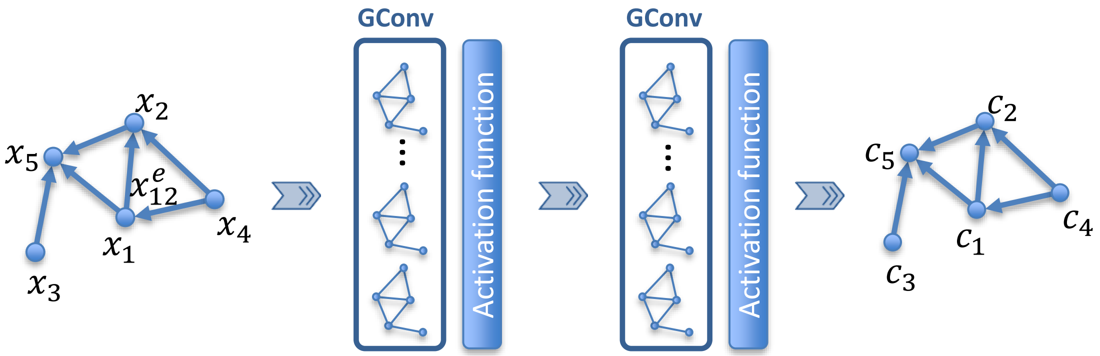
Graph level – E.g., graph classification

c: vector of classification scores for the input graph.

Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Convolutional GNNs

Convolutional GNNs (ConvGNNs) stack multiple graph convolutional layers to extract high-level node representations.



Spectral-based ConvGNNs

Define graph convolutions introducing filters from the point of view of graph signal processing.

(E.g., Spectral CNN, GCN, AGCN).

Spatial-based ConvGNNs

Define graph convolutions by information propagation (message passing), analogously to applying convolutions on images in conventional CNNs.

(E.g., MPNN, NN4G, DCNN, GraphSage, GAT).

Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Message Passing Neural Networks (MPNNs)

Spatial ConvGNNs treat convolutions as a message passing process, in which information can be passed from one node to the other along edges.

Neighbors' information aggregation

$$m_v^{(k)} = \sum_{u \in N(v)} M_k\left(h_v^{(k-1)}, h_u^{(k-1)}, x_{vu}^e\right)$$

In message-passing neural networks (MPNNs) a graph convolution operation is divided into:

- aggregation of the information from neighboring nodes;
- combination of the local node features with the aggregated neighbors' data.

$$h_v^{(k)} = U_k\left(h_v^{(k-1)}, m_v^{(k)}\right)$$

k is the layer index
$h_v^{(k)}$ is the hidden representation of node v
$h_v^{(0)} = x_v$, i.e., the input features of node v
$N(v)$ is the is of neighboring nodes of v
$M_k(\cdot)$ is a learnable message passing function
$U_k(\cdot)$ is a learnable update function

MPNN: Gilmer, Justin, et al. "Neural message passing for quantum chemistry." *International conference on machine learning*. PMLR, 2017.

GNNs – Permutation invariance and equivariance

For node-level tasks, the GNN output should respect the input order of the graph nodes. That is, the GNN must be an equivariant function with respect to input nodes permutations.

$$f(X, A) \in \mathbb{R}^{n \times d}$$

$$f(PX, PAP^T) = Pf(X, A)$$

$f(X, A)$: function representing the GNN
$X \in \mathbb{R}^{n \times d}$: nodes features matrix
$A \in \mathbb{R}^{n \times n}$: graph adjacency matrix
$P \in \mathbb{R}^{n \times n}$: arbitrary nodes permutation matrix

For graph-level tasks, the GNN output should not change if the input order of the graph nodes is different. That is, the GNN must be an invariant function with respect to input nodes permutations.

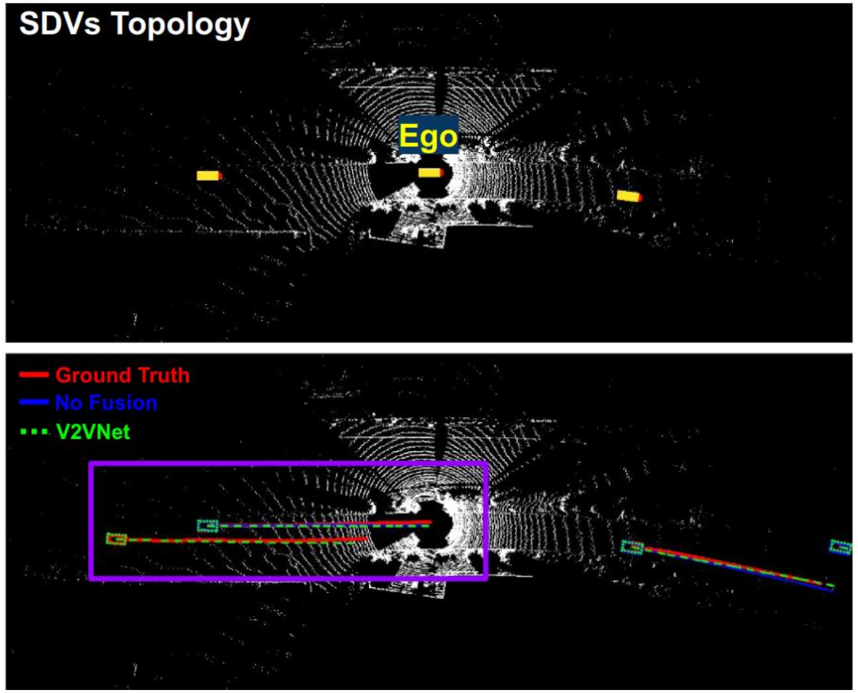
$$f(X, A) \in \mathbb{R}^{d}$$

$$f(PX, PAP^T) = f(X, A)$$

V2VNet - Joint perception and prediction in V2V communications

V2VNet is an intermediate collaboration method that improves the detection and motion-forecasting performance under V2V communication constraints by:

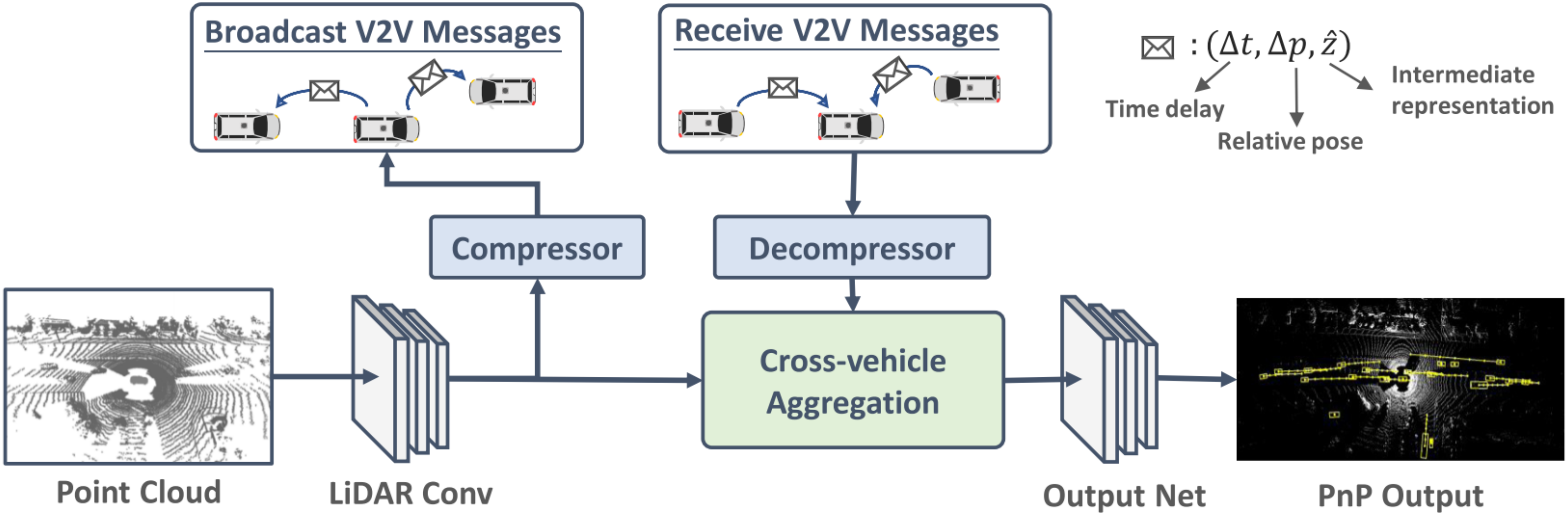
- Introducing a spatially aware GNN to intelligently combine the information received from the nearby CAVs.
- Integrating a variational compression algorithm to compress the intermediate representations to be shared.



The recently introduced approaches that perform joint detection and motion forecasting are named *perception and prediction* (P&P)

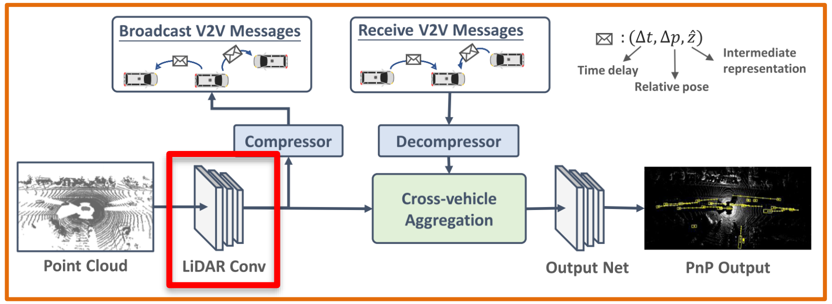
Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet - Architecture



Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – LiDAR Conv block



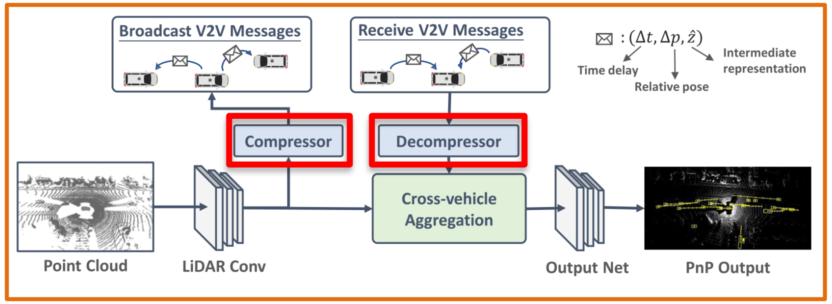
The LiDAR Conv block processes raw sensor data and creates a compressible intermediate representation.

- The past 5 LiDAR point cloud sweeps are voxelized (into 15.6 cm voxels).
- Several convolutional layers are applied.
- The output feature maps have dimensions $H \times W \times C$, where $H \times W$ is the scene range in BEV, and C is the number of feature channels.

3 conv. layers with 3 × 3 filters and strides of (2, 1, 2) produce a 4x downsampled feature map.

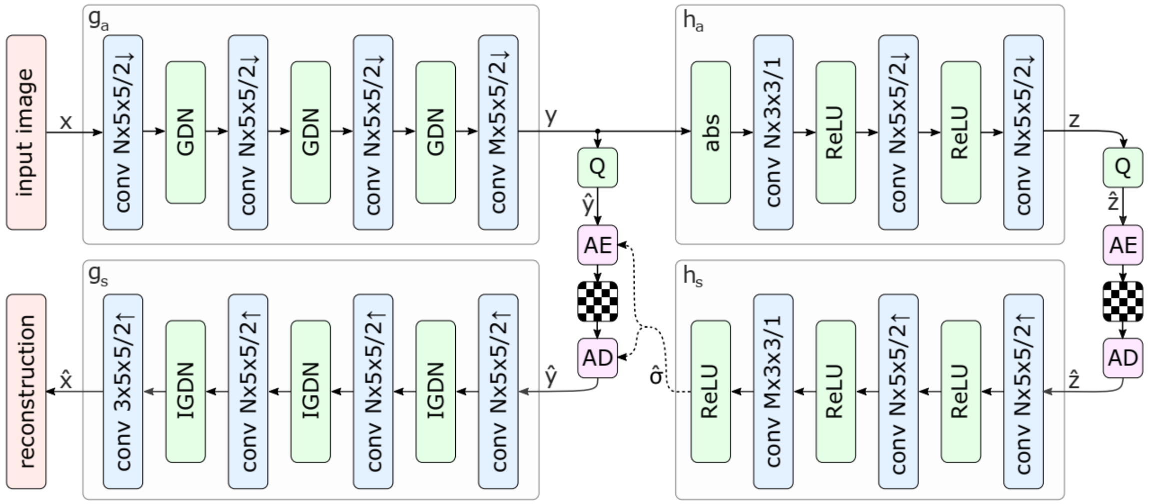
Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – Data compression



Data compression is achieved in V2VNet training a variational compression module by Ballé et al.

- The left side shows an image autoencoder architecture.
- The right side is an autoencoder implementing a hyperprior.
- The hyperprior allows to effectively capture spatial dependencies in the latent representation.



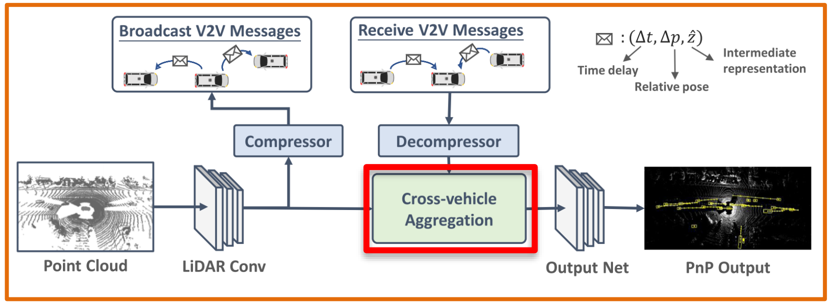
Conventional compression and hyperpriors

Using a VAE architecture, the entropy model given by Shannon cross-entropy corresponds to the prior of the latents. In turn, side information can be seen as a prior on the parameters of the entropy model, which makes it an hyperprior of the latents.

Ballé, Johannes, et al. "Variational image compression with a scale hyperprior." *arXiv preprint arXiv:1802.01436* (2018).

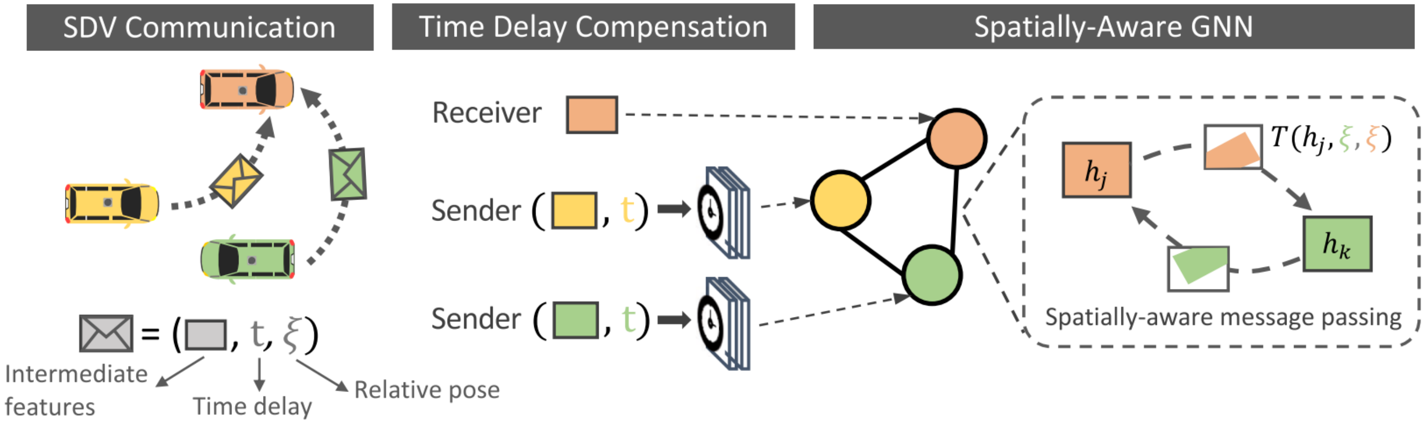
Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – Cross-vehicle Aggregation



The cross-vehicle aggregation module integrates the received information from other vehicles to produce an updated intermediate representation.

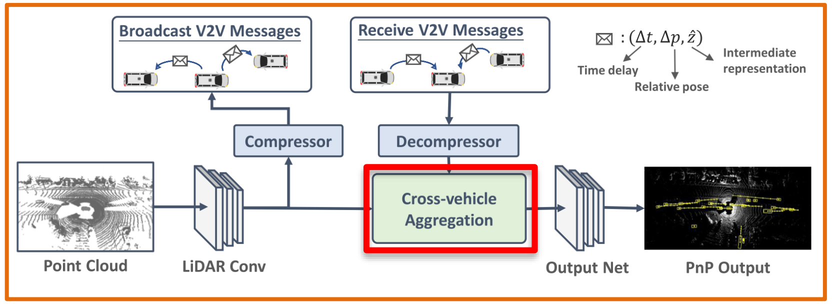
- This module has to handle data from CAVs located at different locations and seeing actors at different timestamps.
- The intermediate feature representations have to be spatially aware.



A spatially aware GNN is used to aggregate the data received from the nearby CAVs

Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – Spatially aware GNN



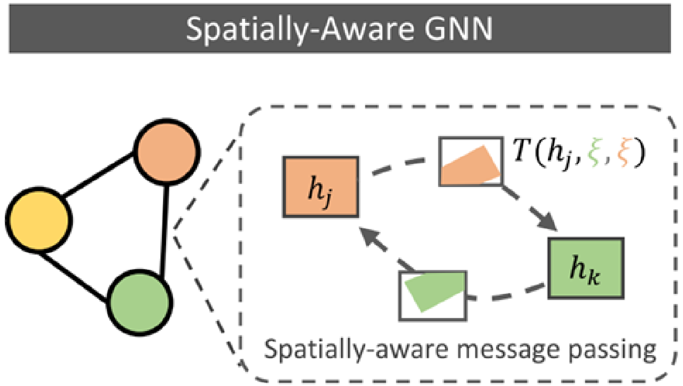
Each vehicle uses a fully-connected GNN as aggregation module.

- Each GNN node is the state representation of a connected CAV (including the CAV itself).
- Since the other CAVs are in the same local area, the node representations will have overlapping fields of view.
- Overlappings can be used to enhance the CAV's scene understanding.

Algorithm 1. Cross-vehicle Aggregation

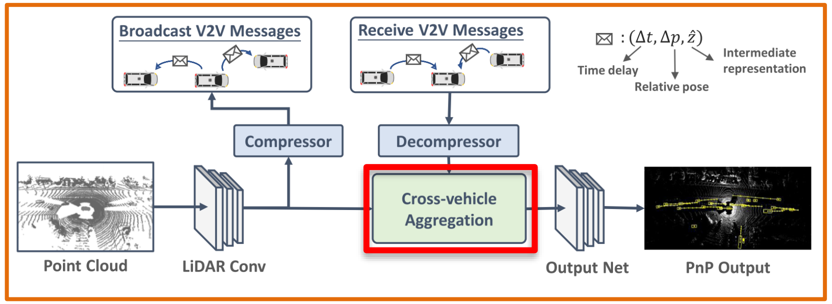
input: representation $\hat{z}_i$, relative pose Δp_i, and time delay $\Delta t_{i\to k}$ for each SDV i
for each vehicle i **do**
    $h_i^{(0)} = CNN(\hat{z}_i, \Delta t_{i\to k}) \parallel \mathbf{0}$ ▷ Compensate time delay, init. node state
end for
for l iterations **do**   ▷ Message passing
    for each vehicle i **do**   ▷ Processed in parallel
        $m_{i\to k}^{(l)} = CNN(T(h_i^{(l)}, \xi_{i\to k}), h_k^{(l)}) \cdot M_{i\to k}$ ▷ Spatially transform message
        $h_i^{(l+1)} = ConvGRU(h_i^{(l)}, \phi_M([\forall_{j\in N(i)}, m_{j\to i}^{(l)}]))$ ▷ Node state update
    end for
end for
$z_i^{(L)} = MLP(h_i^{(L)})$ ▷ Output updated intermediate representation

A GNN is a natural choice to handle dynamic graph topologies which arise in the V2V setting.



Schlichtkrull, Michael, et al. "Modeling relational data with graph convolutional networks." *The Semantic Web: 15th International Conference, ESWC 2018.*
Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – Spatially aware GNN



Spatial transformation message

$$m_{i \to k}^{(l)} = CNN(\underbrace{T}(h_i^{(l)}, \xi_{i \to k}), h_k^{(l)}) \cdot \underbrace{M_{i \to k}}$$

Spatial transformation and resampling of the feature state via bilinear interpolation.

Masking for non-overlapping areas between the fields of view

With this design, the message keeps spatial awareness.

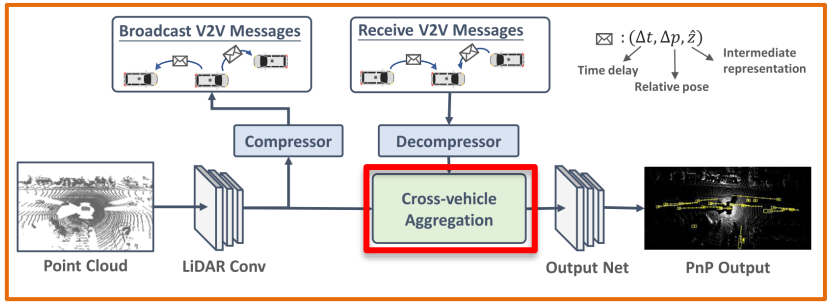
$\xi_{i \to k}$ is a spatial transformation that warps the intermediate state of the i-th node to send a GNN message to the k-th node.

The spatially aligned feature maps of both nodes are processed through a CNN.

A mask is applied to non-overlapping areas bewteen the nodes' fields of view.

Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – Spatially aware GNN



Node state update

$$m_{i\to k}^{(l)} = CNN(T(h_i^{(l)}, \xi_{i\to k}), h_k^{(l)}) \cdot M_{i\to k}$$

$$h_i^{(l+1)} = ConvGRU(h_i^{(l)}, \underbrace{\phi_M}_{\text{Function aggregating the received messages}}([\forall_{j\in \underbrace{N(i)}_{\text{Neighboring nodes}}}, m_{j\to i}^{(l)}]))$$

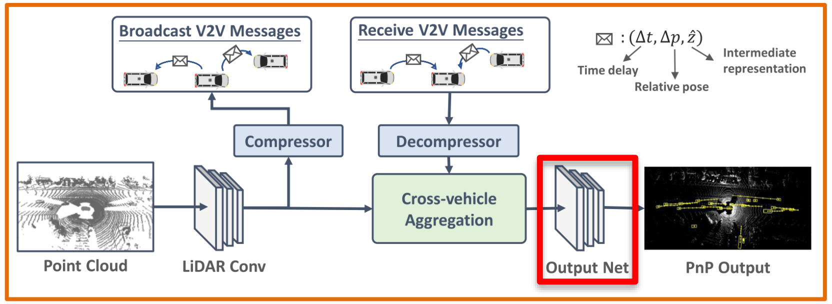
The gating mechanism enables information selection for the accumulated messages based on the current receiving node belief.

ϕ_M is a mask-aware permutation-invariant function aggregating the received messages.

The node state is updated using a convolutional Gated Recurrent Unit (ConvGRU).

Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – Output Network



- The output network consists in a 4 Inception-like convolutional blocks that efficiently capture multi-scale context.
- The resulting feature map is processed by two network branches to output object detection and motion forecasting estimates.

Object detection outputs

$$(\underbrace{x, y}_{\text{Position}}, \underbrace{w, h}_{\text{Size}}, \underbrace{\theta}_{\text{Orientation}})$$

Motion forecasting outputs

$$\underbrace{(x_t, y_t)}_{\text{Object location at future time step } t}$$

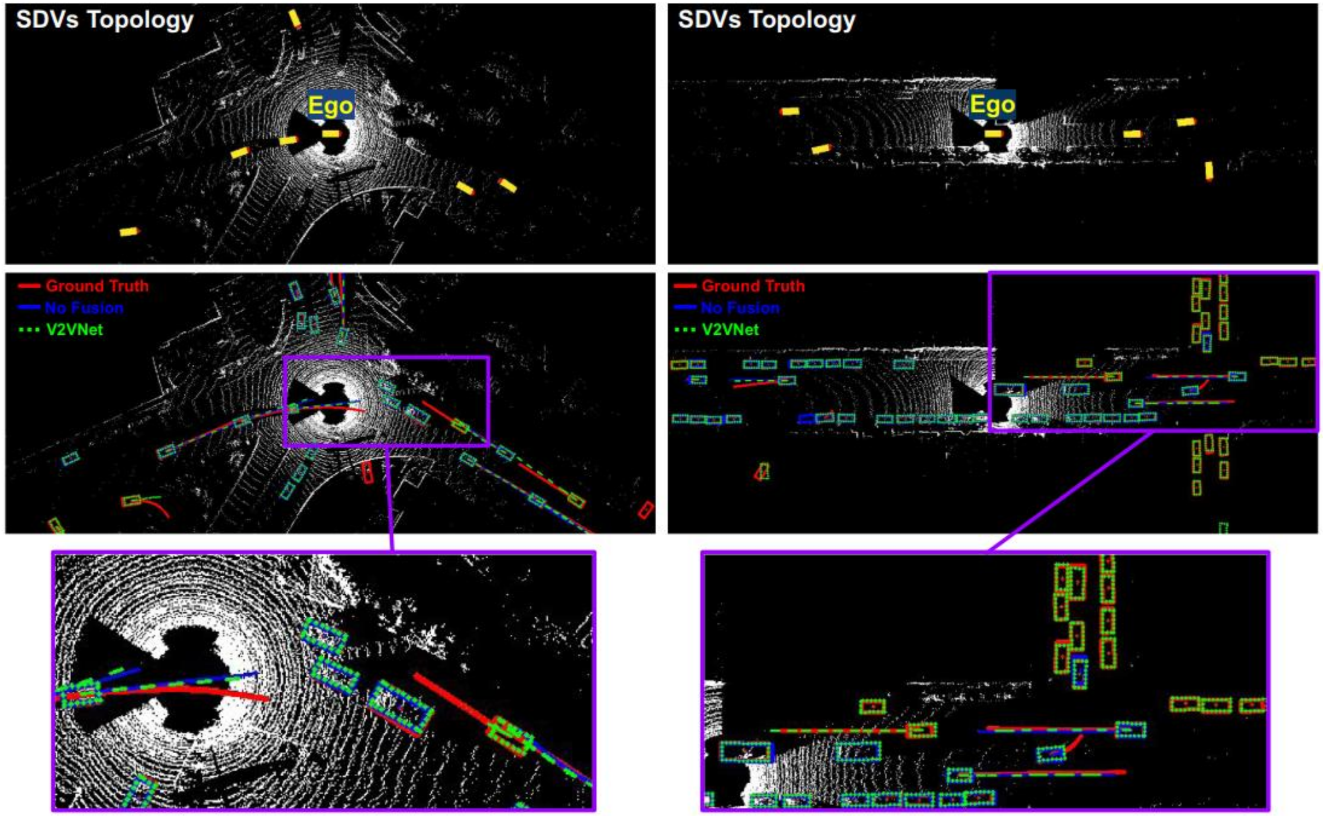
Inception blocks: Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet – Evaluation dataset

The **V2V-Sim** is a simulated large-scale V2V communication dataset.

- Based on the LiDARsim high-fidelity simulation system.
- Leverages traffic scenarios captured in the real-world ATG4D dataset.
- Composed by 51,200 total frames.
- 10 candidate vehicles per sample on average (max: 63, variance: 7).



Manivasagam, Sivabalan, et al. "Lidarsim: Realistic lidar simulation by leveraging the real world." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

Yang, Bin, Wenjie Luo, and Raquel Urtasun. "Pixor: Real-time 3d object detection from point clouds." *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018.

Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet - Results

3D object detection and tracking results on the V2V-Sim dataset

Method	AP@IoU ↑		ℓ_2 Error (m) ↓			TCR ↓
	0.5	0.7	1.0 s	2.0 s	3.0 s	$\tau = 0.01$
No Fusion	77.3	68.5	0.43	0.67	0.98	2.84
Output Fusion	90.8	86.3	**0.29**	**0.50**	0.80	3.00
LiDAR Fusion	92.2	88.5	**0.29**	**0.50**	0.79	2.31
V2VNet	**93.1**	**89.9**	**0.29**	**0.50**	**0.78**	**2.25**

ℓ_2 error is evaluated at recall 0.9 at different timestamps.

TCR: Trajectory Collision Rate
NMS: Non-maximum Suppression

Baselines

No Fusion: Single vehicle setting, without V2V communication.

Output Fusion (late collaboration): each vehicle sends post-processed outputs, i.e., bounding boxes with confidence scores, and predicted future trajectories after NMS.

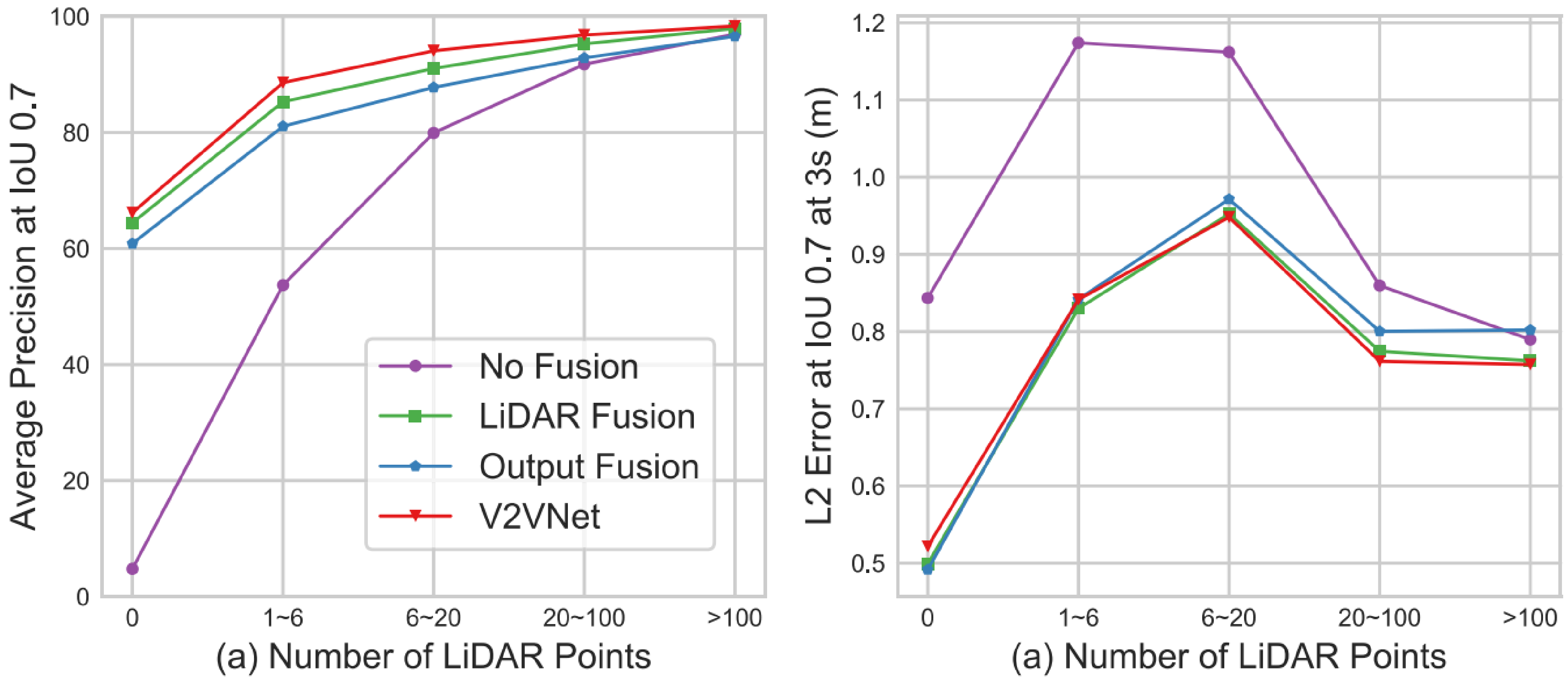
LiDAR Fusion (early collaboration): the raw LiDAR point clouds received from the other vehicles are referred to the receiver coordinate frame and direct aggregation is performed. Draco has been used to compress the LiDAR fusion messages.

Draco 3d data compression (2019) - https://github.com/google/draco

Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet - Results

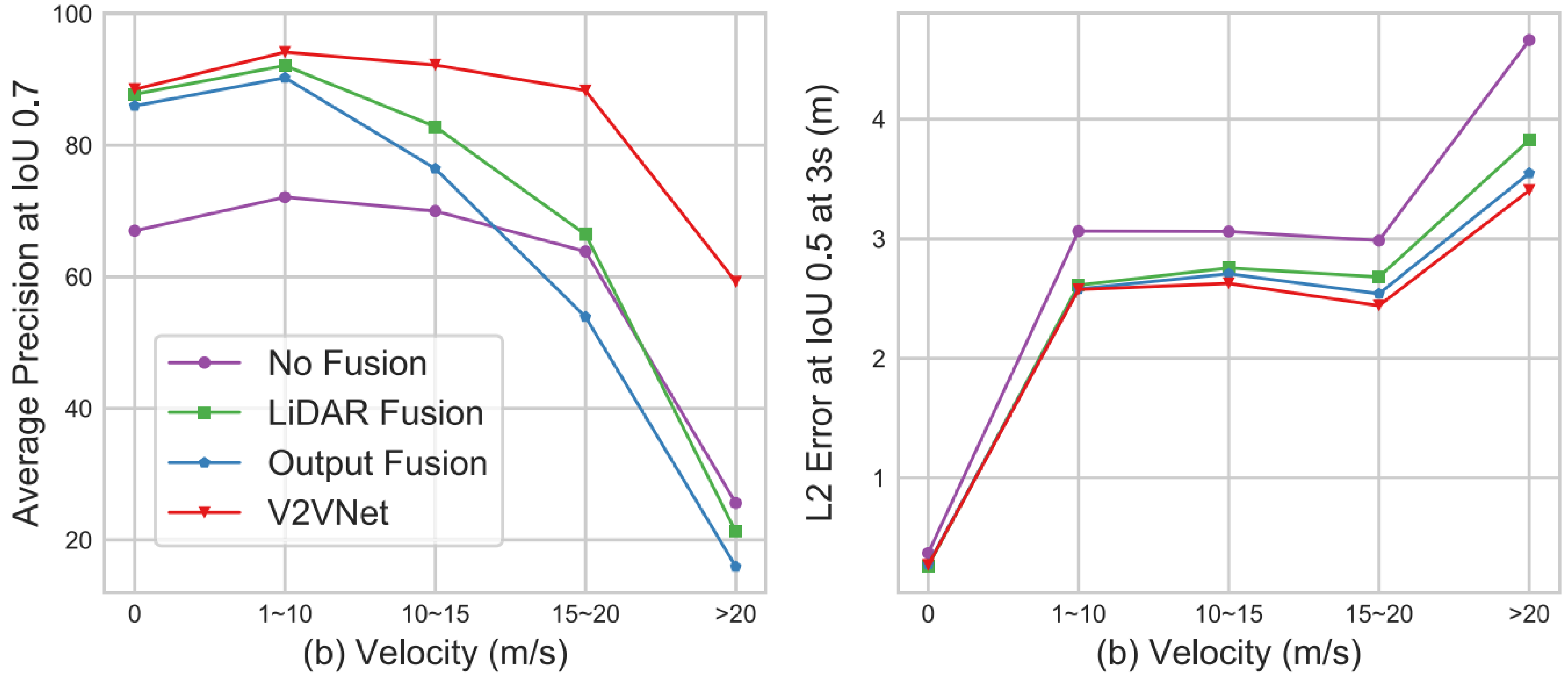
3D object detection results on the V2V-Sim dataset varying the number of LiDAR points



Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet - Results

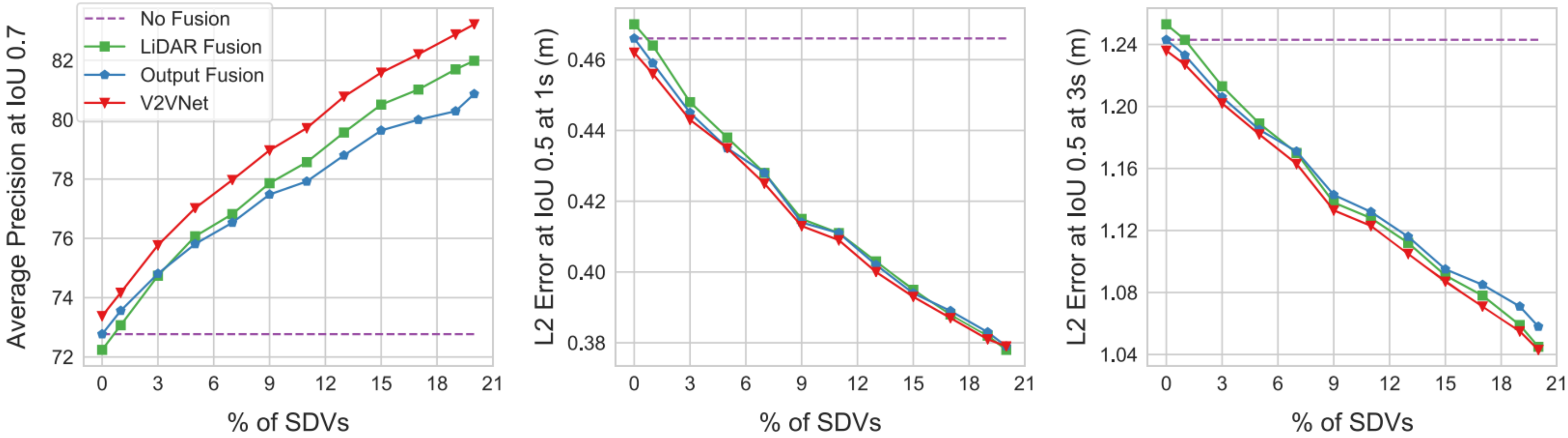
3D object detection results on the V2V-Sim dataset for different velocities



Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet - Results

3D object detection results on the V2V-Sim dataset for varying percentage of CAVs

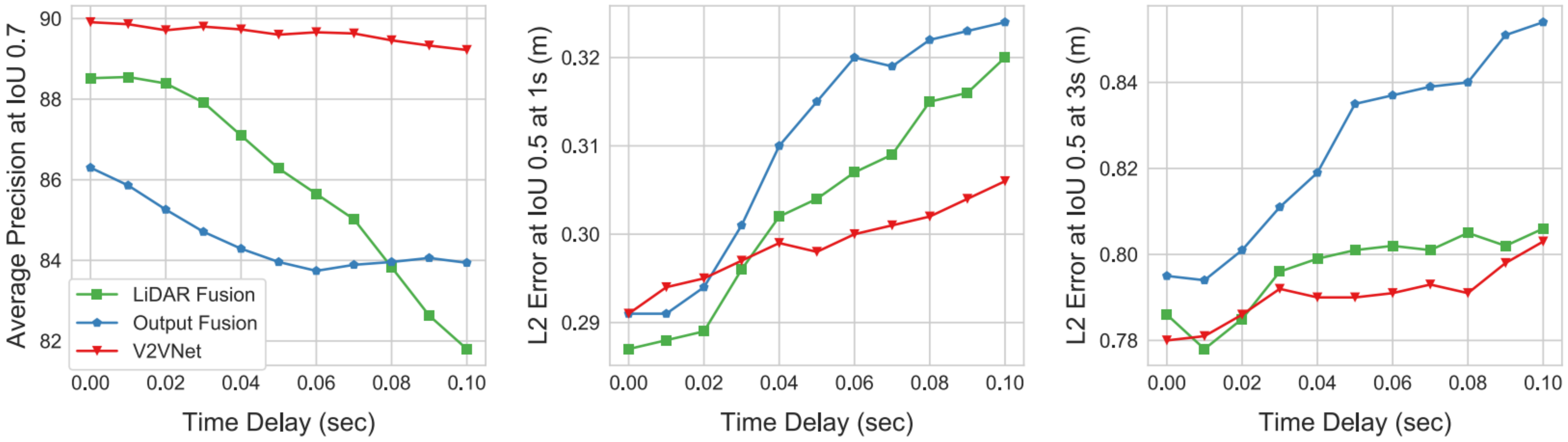


SDV: Self-driving vehicle (alternative definition to CAV used in the article)

Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet - Results

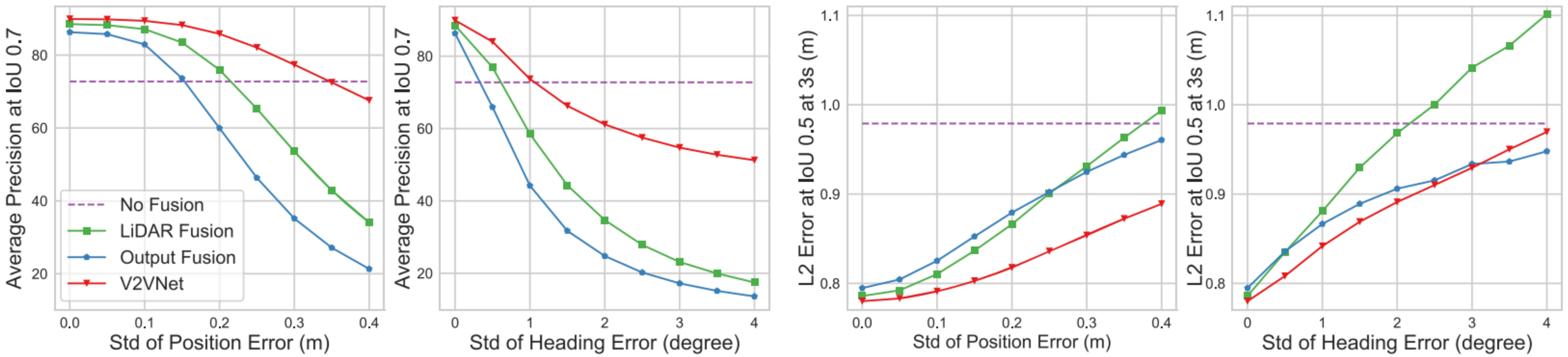
3D object detection results on the V2V-Sim dataset for different time delays in data exchange



Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

V2VNet - Results

3D object detection results on the V2V-Sim dataset for noisy vehicles' relative pose estimates

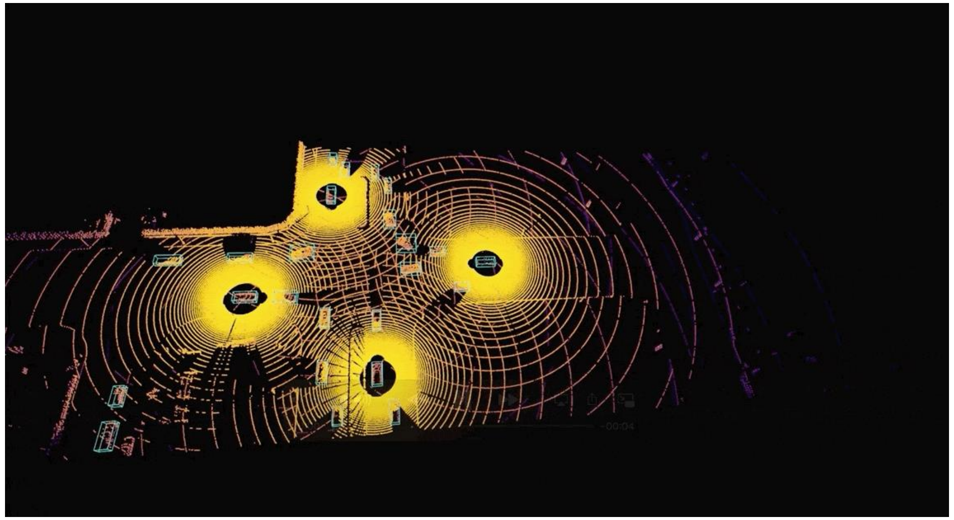


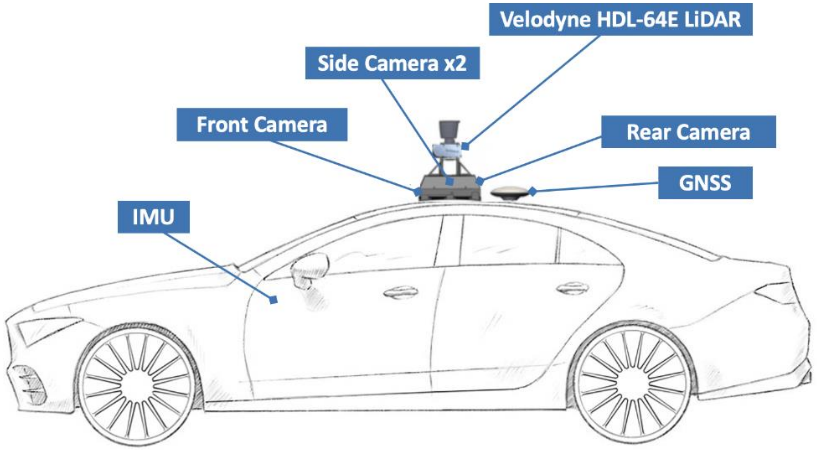
Wang, Tsun-Hsuan, et al. "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction." *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer International Publishing, 2020.

Benchmarks and datasets – OPV2V

OPV2V is a large-scale simulated dataset for perception with V2V communication

- based on OpenCDA and CARLA;
- aggregated sensor data from multi-connected CAVs;
- 73 scenes, 6 road types, 9 cities;
- 12K frames of LiDAR point clouds and RGB camera images, 230K annotated 3D bounding boxes;
- comprehensive benchmark with 4 LiDAR detectors and 4 different fusion strategies.





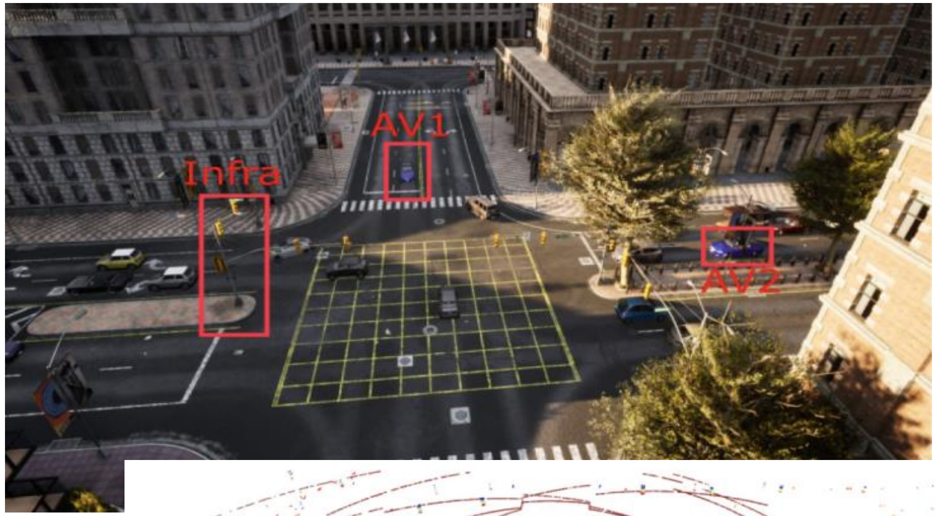
OPV2V: https://mobility-lab.seas.ucla.edu/opv2v/; OpenCDA: https://github.com/ucla-mobility/OpenCDA; CARLA: https://carla.org

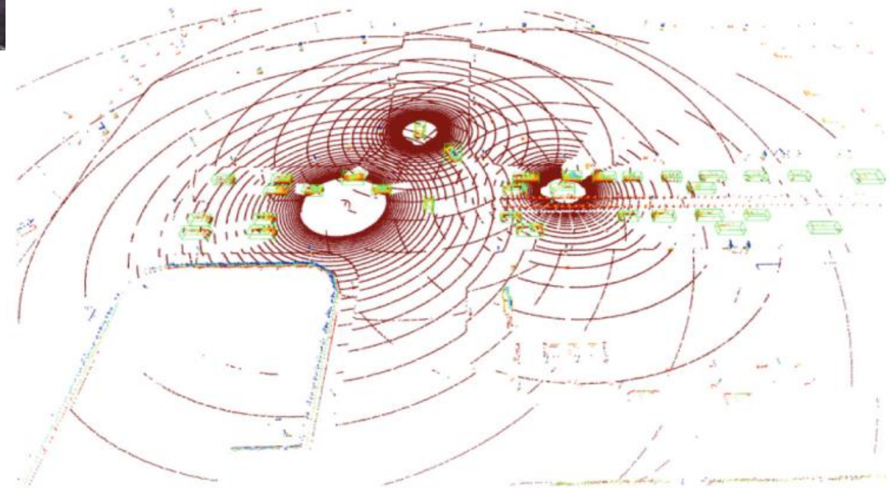
Xu, R., Xiang, H., Xia, X., Han, X., Li, J., & Ma, J. (2022, May). Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)* (pp. 2583-2589). IEEE.

Benchmarks and datasets - V2XSet

V2XSet is a large-scale simulated dataset for perception with V2X communication

- Based on OpenCDA and CARLA.
- Contains 11,447 frames.
- Explicitly considers real-world noises during V2X communication.
- Considers V2X communications (includes also the communication infrastructure), with respect to OPV2V, which restricts to V2V.



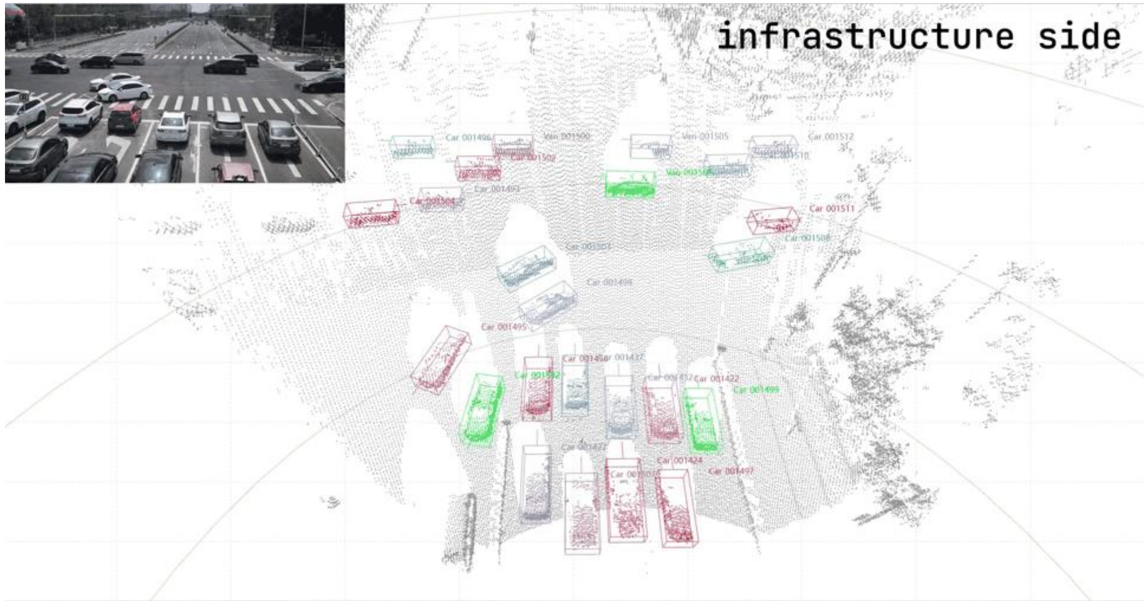


Dataset and model website: https://github.com/DerrickXuNu/v2x-vit; OpenCDA: https://github.com/ucla-mobility/OpenCDA; CARLA: https://carla.org

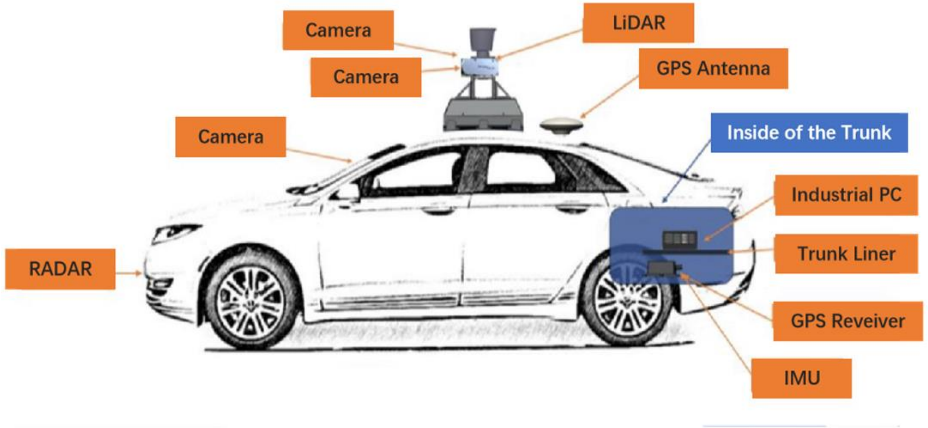
Xu, R., Xiang, H., Tu, Z., Xia, X., Yang, M. H., & Ma, J. (2022, October). V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision* (pp. 107-124). Cham: Springer Nature Switzerland.

Benchmarks and datasets - DAIR-V2X

DAIR-V2X is a multi-modal multi-view real-world dataset for V2I cooperative 3D object detection



- It comprises a total of 71,254 frames of image data and 71,254 frames of point cloud data;
- It is integrated with the OpenDAIR-V2X framework.



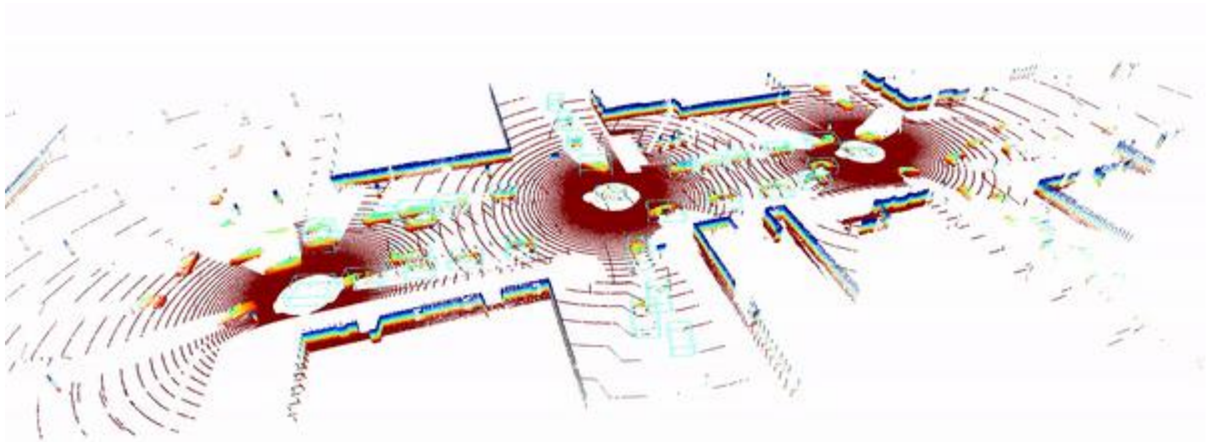
Dataset and framework websites: https://thudair.baai.ac.cn/index; https://github.com/AIR-THU/DAIR-V2X

Yu, H., Luo, Y., Shu, M., Huo, Y., Yang, Z., Shi, Y., ... & Nie, Z. (2022). DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 21361-21370).

Benchmarks and datasets - OpenCOOD

OpenCOOD is an open cooperative detection framework integrating state-of-the-art (SOTA) datasets and perception models.

- Provides an easy data API for both OPV2V and V2X-Set datasets.
- Includes multiple SOTA 3D detection backbones (e.g., PointPillar and VoxelNet)
- Integrates a wide variety of SOTA cooperative perception models.



Framework website: https://github.com/DerrickXuNu/OpenCOOD

Xu, R., Xiang, H., Xia, X., Han, X., Li, J., & Ma, J. (2022, May). Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)* (pp. 2583-2589). IEEE.

Beyond data sharing…

Who2com

(2020, Liu et al.)

Proposes a three-stage communication mechanism (request, match, and connect) in order to select the best matching agents for communication.

Liu, Y. C., Tian, J., Ma, C. Y., Glaser, N., Kuo, C. W., & Kira, Z. (2020, May). Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

When2com

(2020, Liu et al.)

Introduces a method to learn to construct the communication group and to decide when to share (without explicit supervision for such decisions).

Liu, Yen-Cheng, et al. "When2com: Multi-agent perception via communication graph grouping." *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2020.

Where2com

(2022, Hu et al.)

Defines a spatial-confidence-aware communication strategy by learning a spatial confidence map to identify the perceptually critical areas.

Hu, Yue, et al. "Where2comm: Communication-efficient collaborative perception via spatial confidence maps." *Advances in neural information processing systems* 35 (2022): 4874-4886.

How2com

(2023, Yang et al.)

Provides a collaborative perception framework that seeks a trade-off between perception performance and communication bandwidth.

Yang, Dingkang, et al. "How2comm: Communication-efficient and collaboration-pragmatic multi-agent perception." *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

Open challenges and future directions

- Test the methods performances on challenging scenes and corner cases (common datasets include only typical traffic situations).
- Generalizability of models trained on simulated data to real scenarios.
- Counteract possible a malicious and selfish behavior of an agent (e.g., an agent collaborating solely to reduce its costs while causing detriment to the other nodes).
- Exploit multi-sensor data through multi-modal data sharing.
- Integrated sensing and communication for cooperative perception.
- Privacy preserving cooperative perception.

Huang, T., Liu, J., Zhou, X., Nguyen, D. C., Azghadi, M. R., Xia, Y., ... & Sun, S. (2023). V2X cooperative perception for autonomous driving: Recent advances and challenges. *arXiv preprint arXiv:2310.03525*.

Deep Learning in 3D for Robotics

- Robot Localization (without GNSS) -

M. Matteucci (matteo.matteucci@polimi.it) and D. Cattaneo daniele.cattaneo@disco.unimib.it

Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

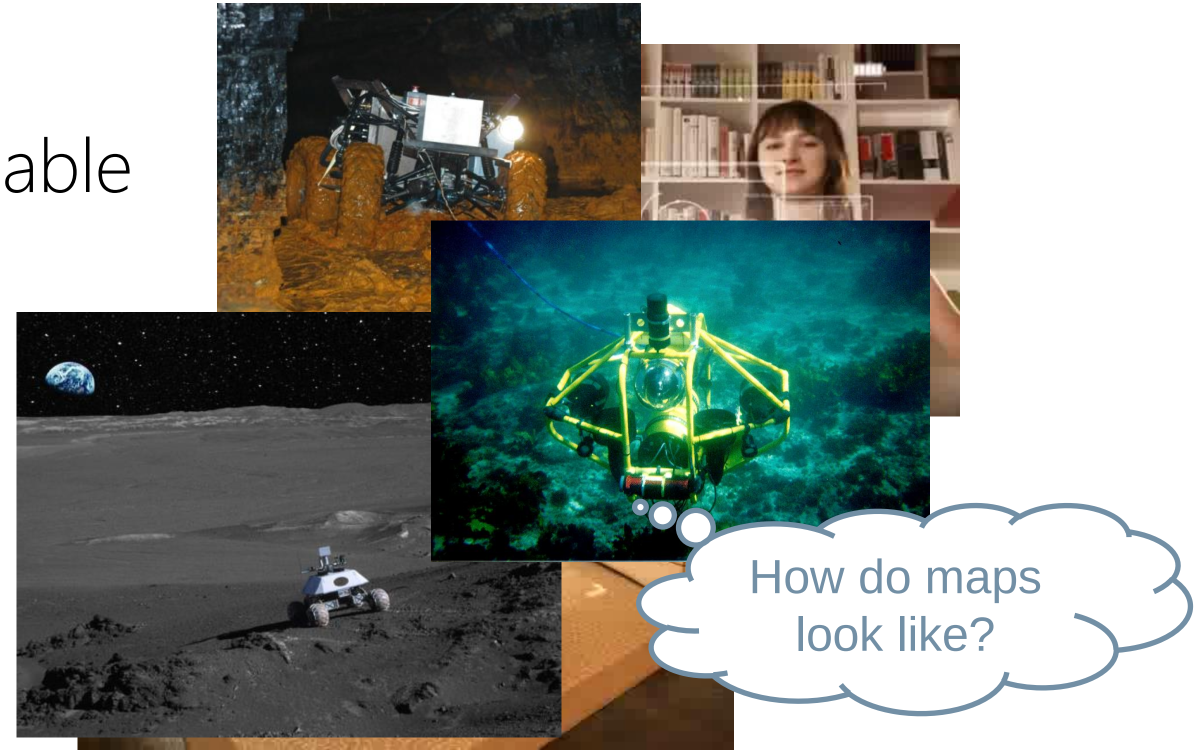
Where Am I?

To perform their tasks autonomous robots and unmanned vehicles need

- To know where they are (e.g., Global Positioning System)
- To know the environment map (e.g., Geographical Institutes Maps)

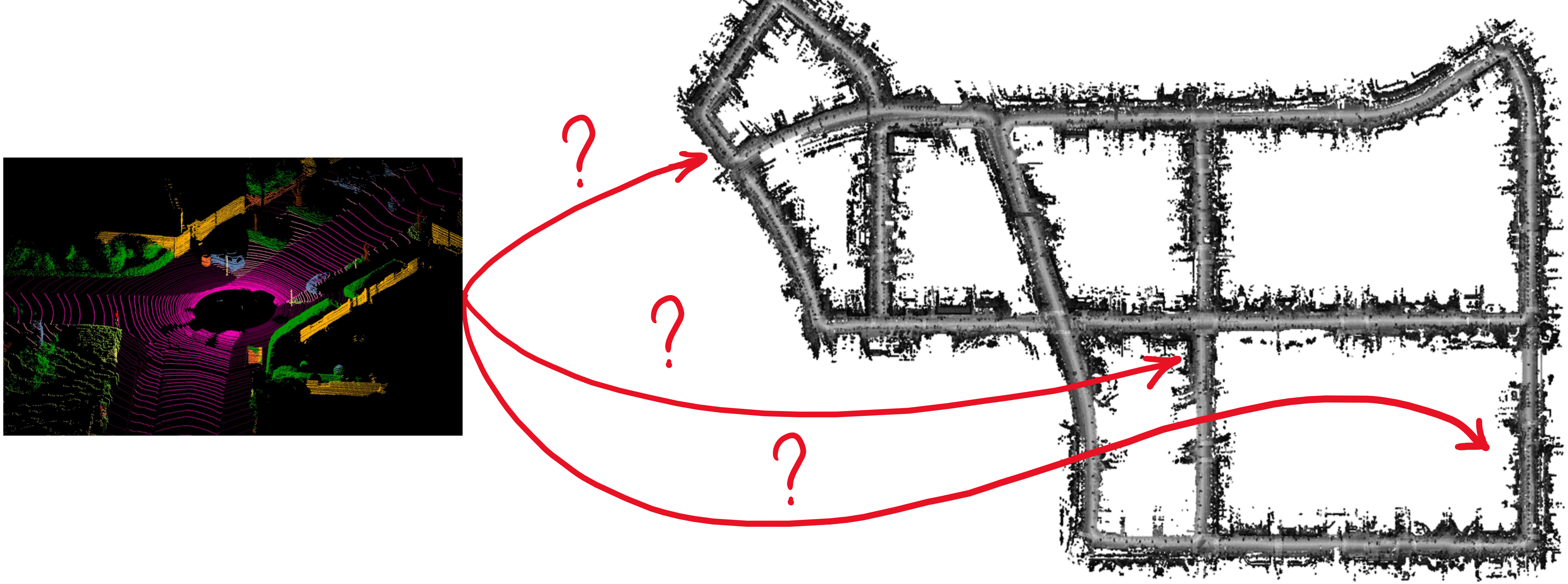
These are not always possible or reliable

- GNSS are not always reliable/available
- Not all places have been mapped
- Environment changes dynamically
- Maps need to be updated



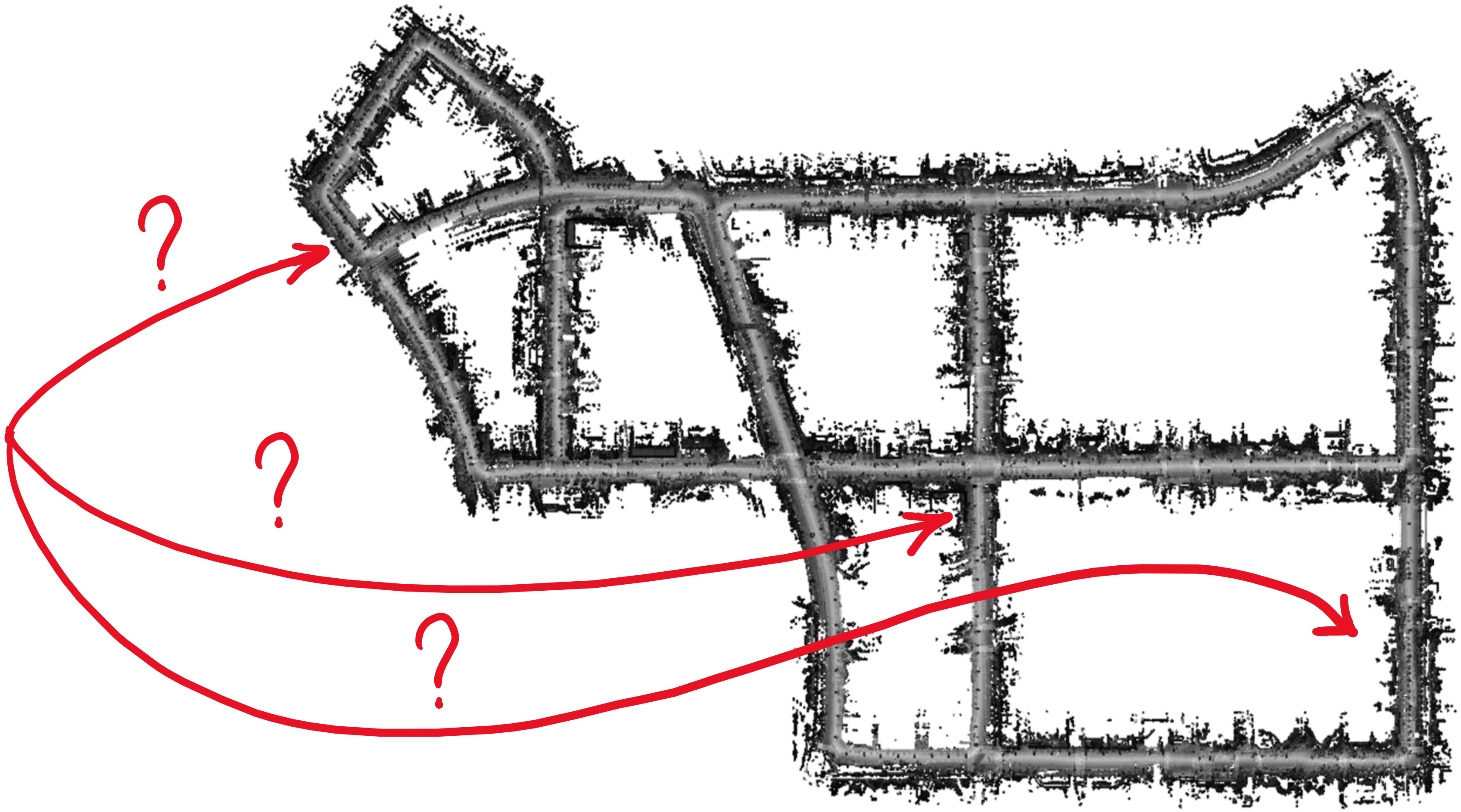
Localization without GNSS

Problem: getting a coarse global localization estimate in LiDAR maps when GNSSs are unavailable?



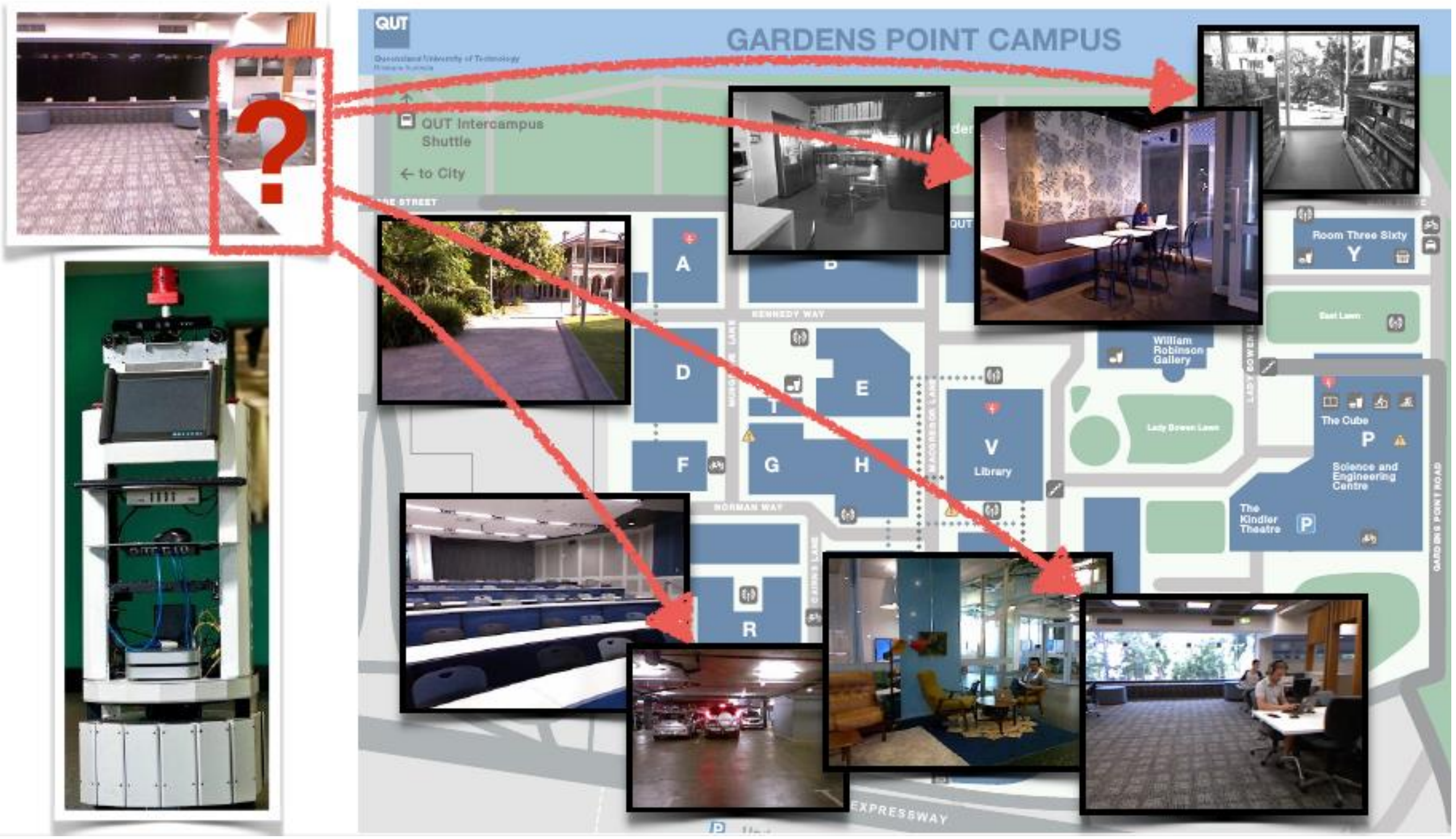
Localization without GNSS

This ca be framed as a classical place recognition task …



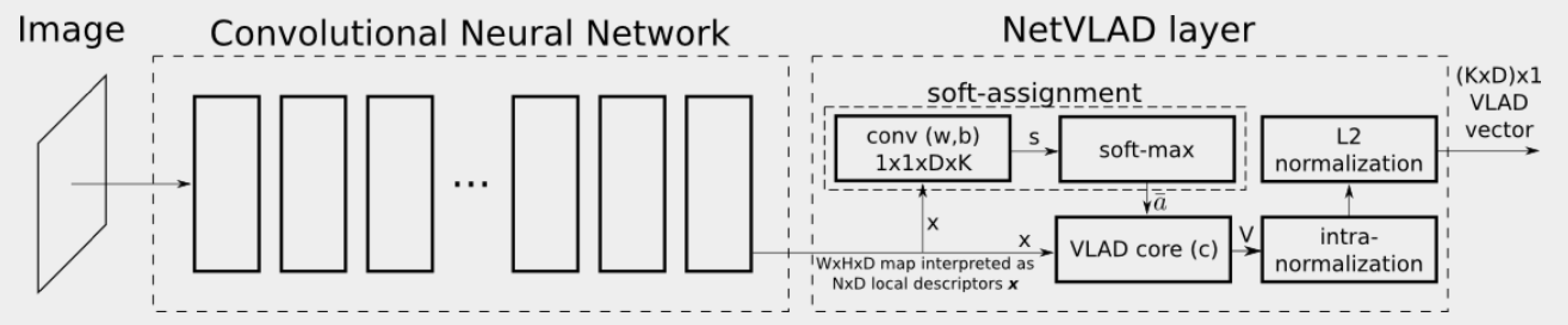
Place recognition

Given a query image find the corresponding one in a (geo-referenced) database of images



Place Recognition - Introduction

State of the art approaches use CNNs.

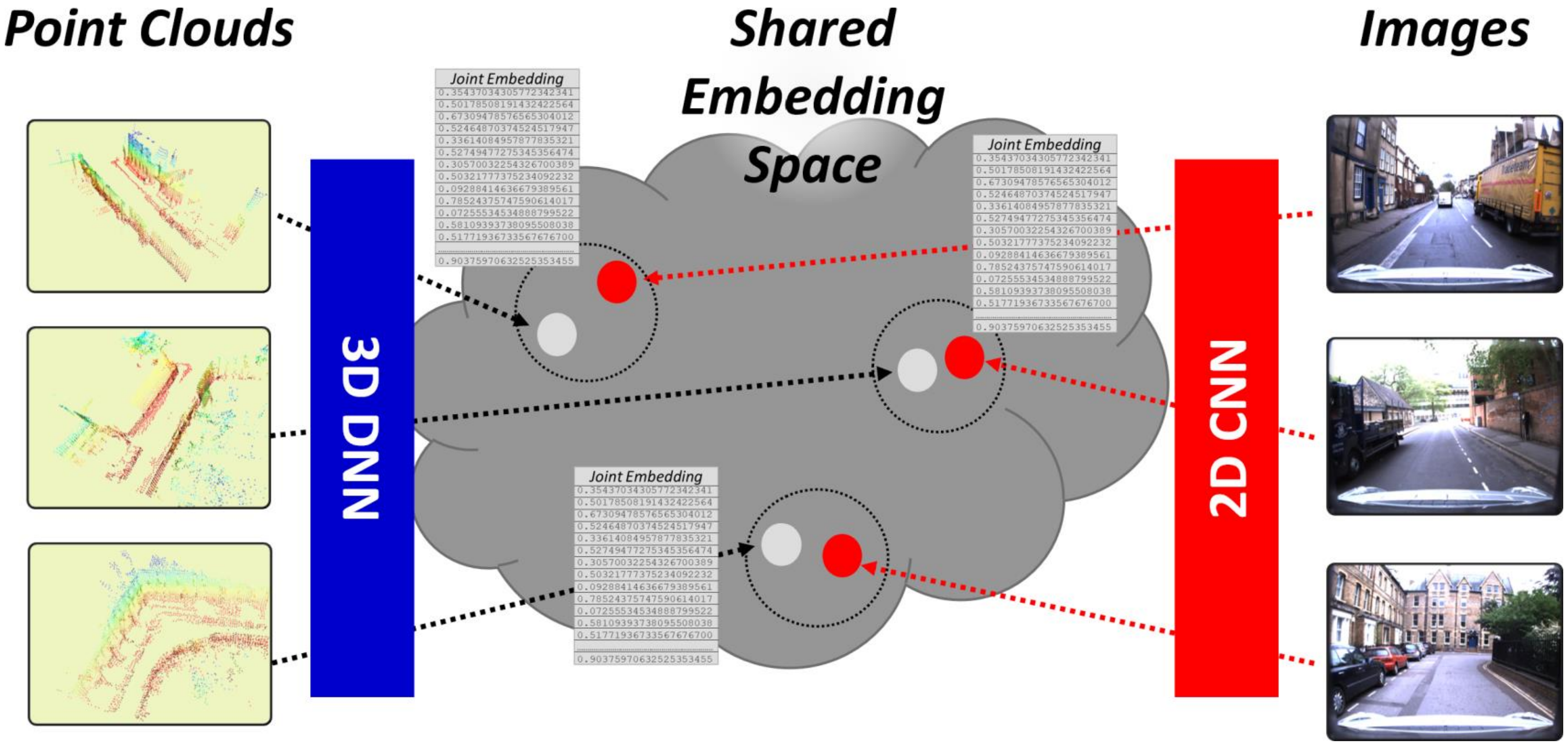


CNN architecture with the NetVLAD layer

(a) Mobile phone query (b) Retrieved image of same place

Global localization in LiDAR-maps via 2D-3D embedding space

Joint training of a 3D-CNN and a 2D-CNN in such a way that point clouds and images from the same place have similar embedding vectors



D. Cattaneo, M. Vaghi, S. Fontana, A. L. Ballardini, D. G. Sorrenti: Global visual localization in LiDAR-maps through shared 2D-3D embedding space. ICRA 2020: 4365-4371

Global localization in LiDAR-maps via 2D-3D embedding space

3D Feature Extractor:

- Pointnet
- Pointnet++
- <u>SECOND</u>
- EdgeConv

Triplet Selection:

- Offline Mining
- <u>Online Mining</u>
  - Hard negative
  - Semi-Hard negative
  - Random Negative

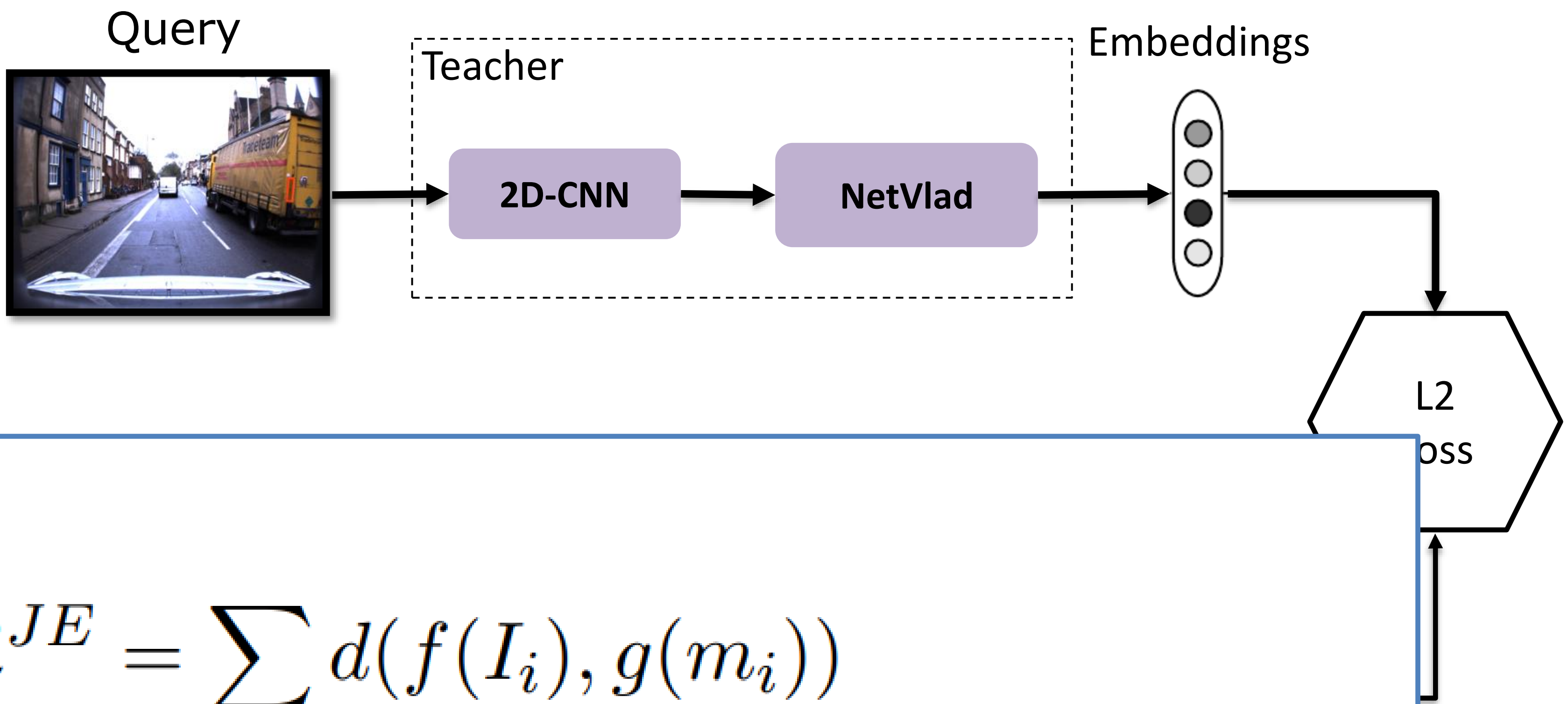
Loss Function:

- <u>Triplet</u>
- Contrastive
- Npair
- Lifted Structured Embedding
- Learning by Association

Training method:

- <u>Teacher / Student</u>
- Joint Training

Knowledge Distillation

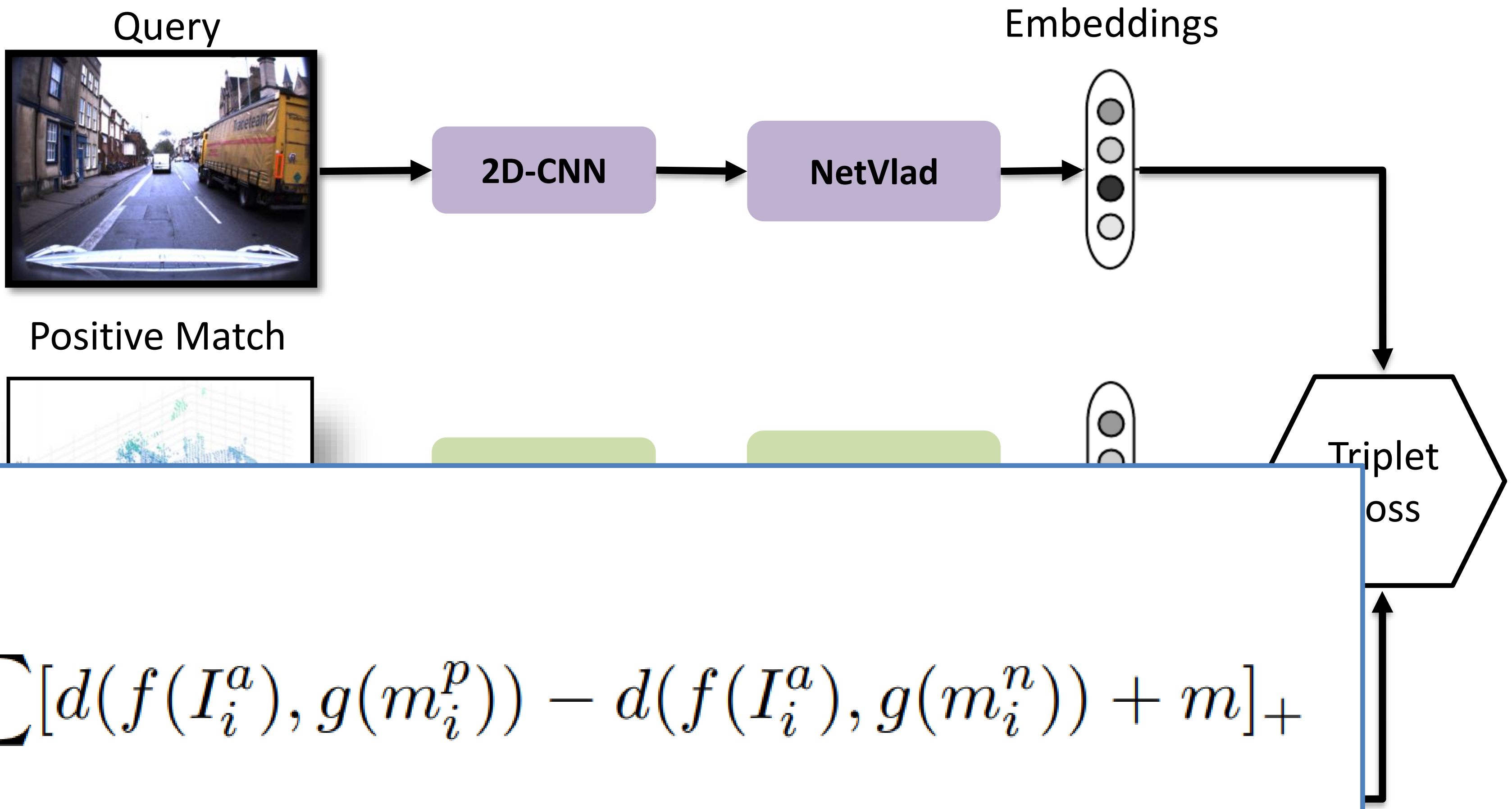


- Initially a CNN model learns to perform place recognition with images

- Then a
emula
descrip

$$\mathcal{L}^{JE} = \sum_i d(f(I_i), g(m_i))$$

Joint Training - triplets

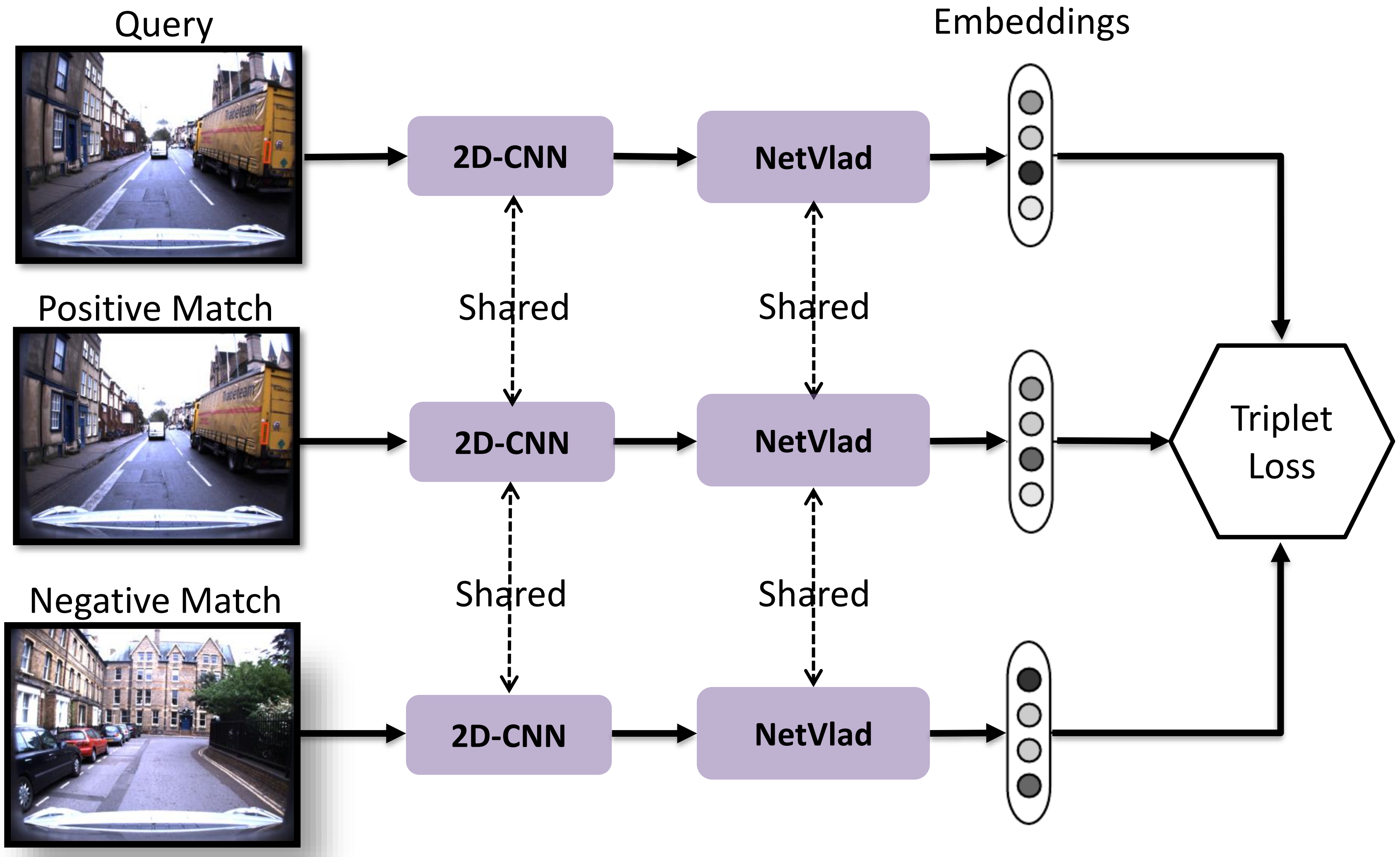


The **triplet** technique consider a positive and a negative sample with res

$$\mathcal{L}_{trp}^{2D\text{-to-}3D} = \sum_i [d(f(I_i^a), g(m_i^p)) - d(f(I_i^a), g(m_i^n)) + m]_+$$

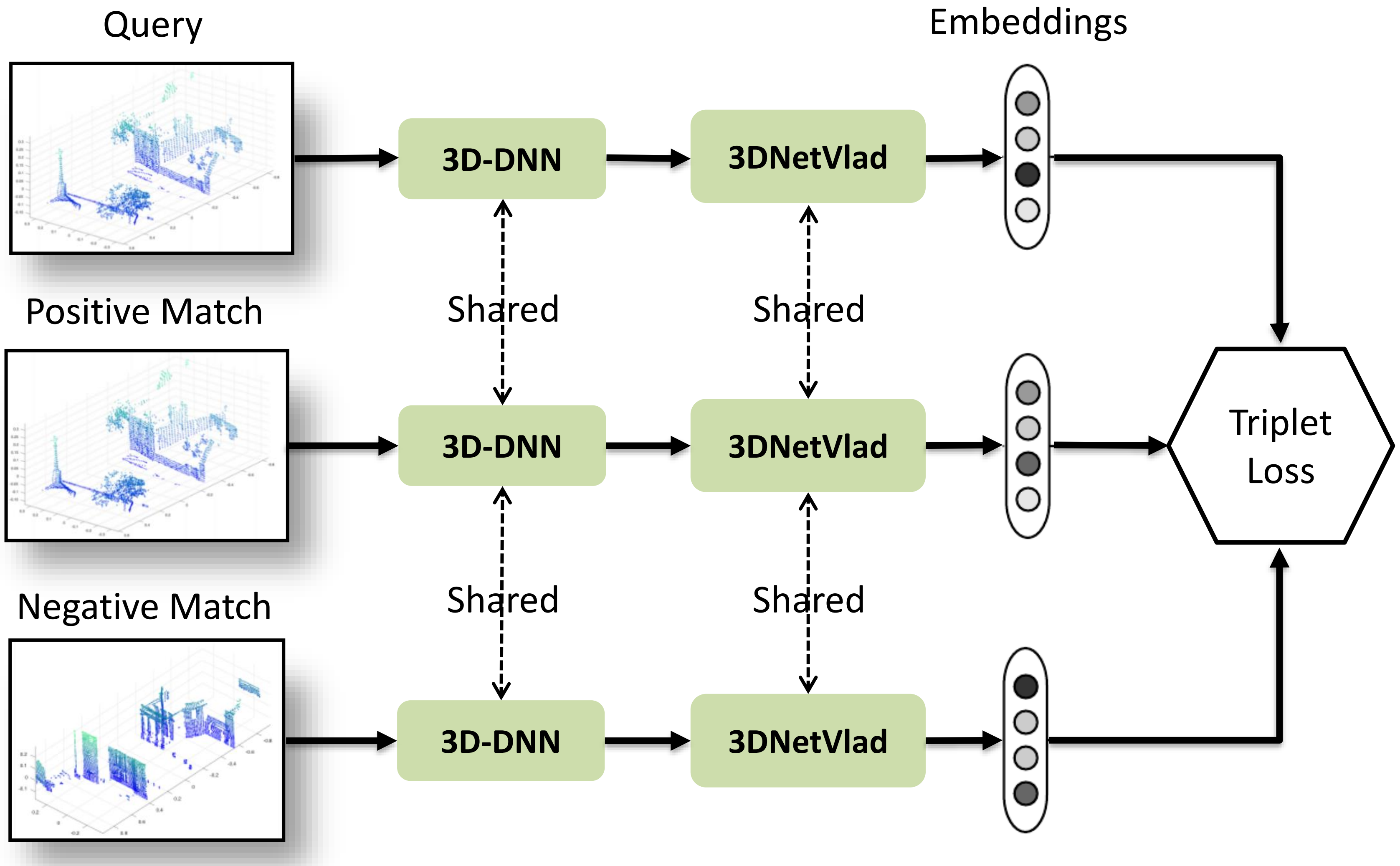
Joint Training - triplets

The **triplet** technique consider a positive and a negative sample with respect to a query



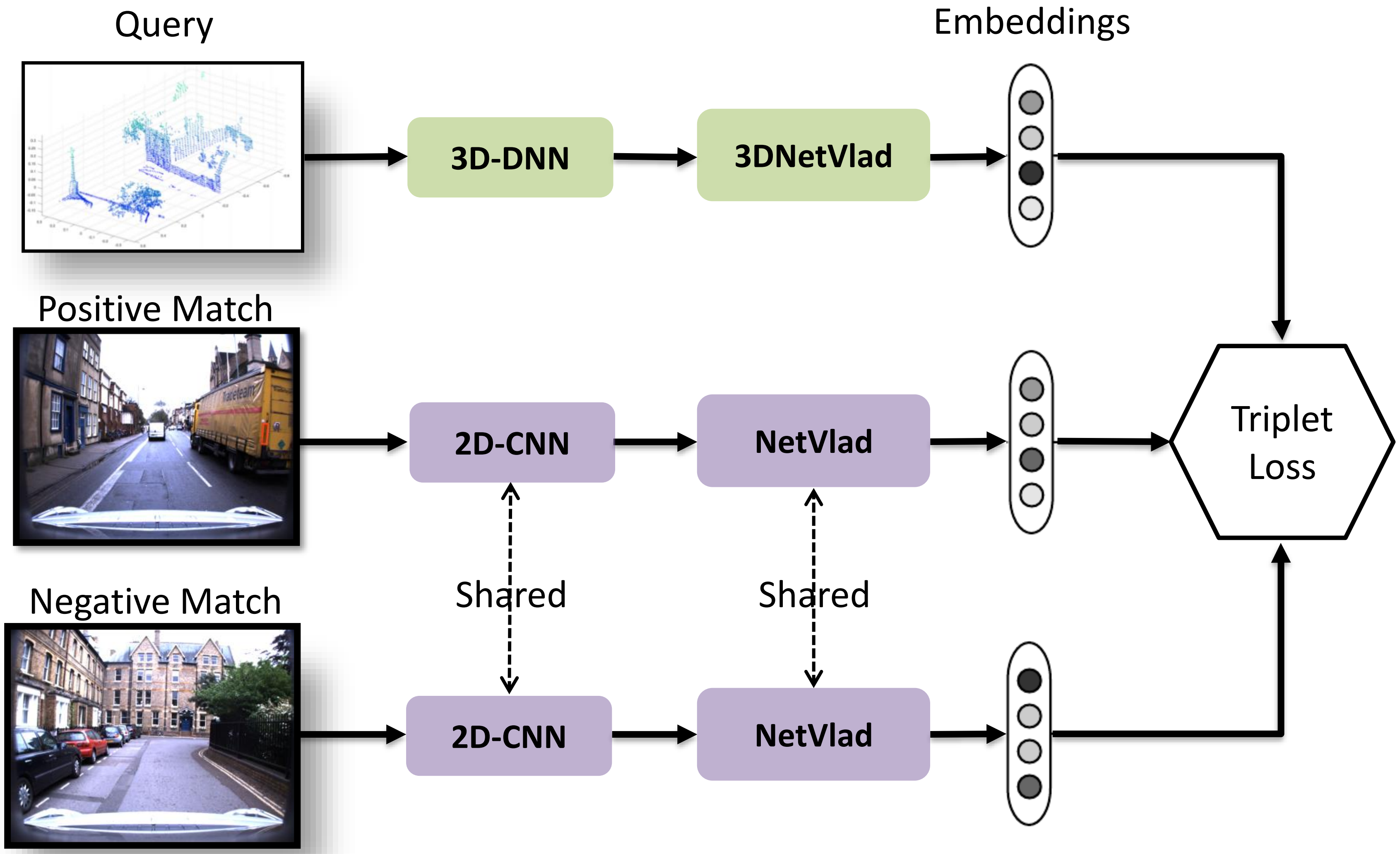
Joint Training - triplets

The **triplet** technique consider a positive and a negative sample with respect to a query



Joint Training - triplets

The **triplet** technique consider a positive and a negative sample with respect to a query



Joint Training - Loss

$$\mathcal{L}_{trp}^{2D\text{-to-}2D} = \sum_i [d(f(I_i^a), f(I_i^p)) - d(f(I_i^a), f(I_i^n)) + m]_+$$

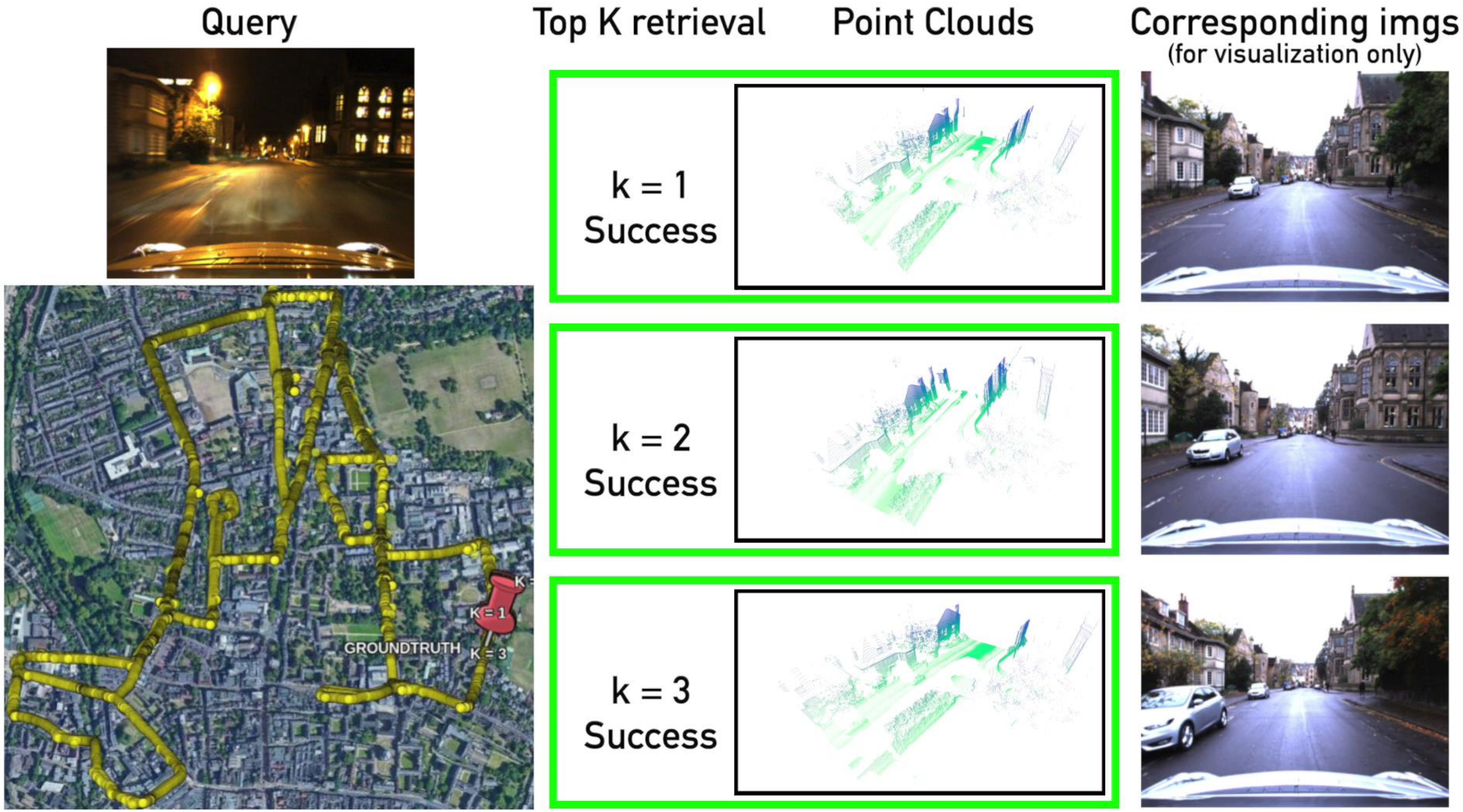
$$\mathcal{L}_{trp}^{3D\text{-to-}2D} = \sum_i [d(g(m_i^a), g(m_i^p)) - d(g(m_i^a), g(m_i^n)) + m]_+$$

$$\mathcal{L}_{trp}^{2D\text{-to-}3D} = \sum_i [d(f(I_i^a), g(m_i^p)) - d(f(I_i^a), g(m_i^n)) + m]_+$$

$$\mathcal{L}_{trp}^{3D\text{-to-}2D} = \sum_i [d(g(m_i^a), f(I_i^p)) - d(g(m_i^a), f(I_i^n)) + m]_+$$

$$\mathcal{L}_{total} = \lambda_1(\mathcal{L}_{trp}^{2D\text{-to-}2D} + \mathcal{L}_{trp}^{3D\text{-to-}3D}) + \lambda_2(\mathcal{L}_{trp}^{2D\text{-to-}3D} + \mathcal{L}_{trp}^{3D\text{-to-}2D}) + \lambda_3\mathcal{L}^{JE}$$

Global localization results



Quantitative results

PLACE RECOGNITION

	Database 2D	Database 3D
Query 2D	97.03 %	78.01 %
Query 3D	73.00 %	98.39 %

3D Place recognition - comparison

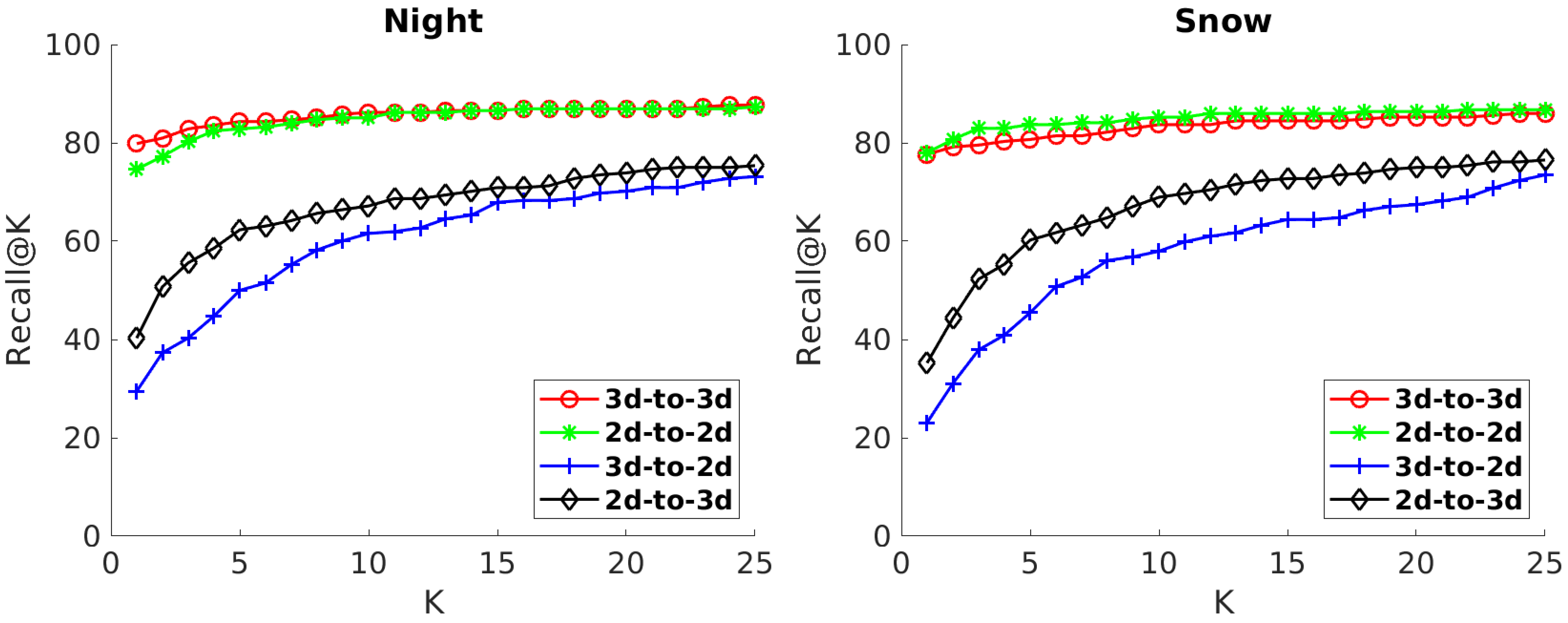
PLACE RECOGNITION

	Database 2D	Database 3D
Query 2D	97.03 %	78.01 %
Query 3D	73.00 %	98.39 %

	Recall@1%	Recall@1
3D-2D	**93.24%**	**87.56%**
PNVlad [1]	80.09%	63.33%
PCAN [2]	86.40%	70.72%

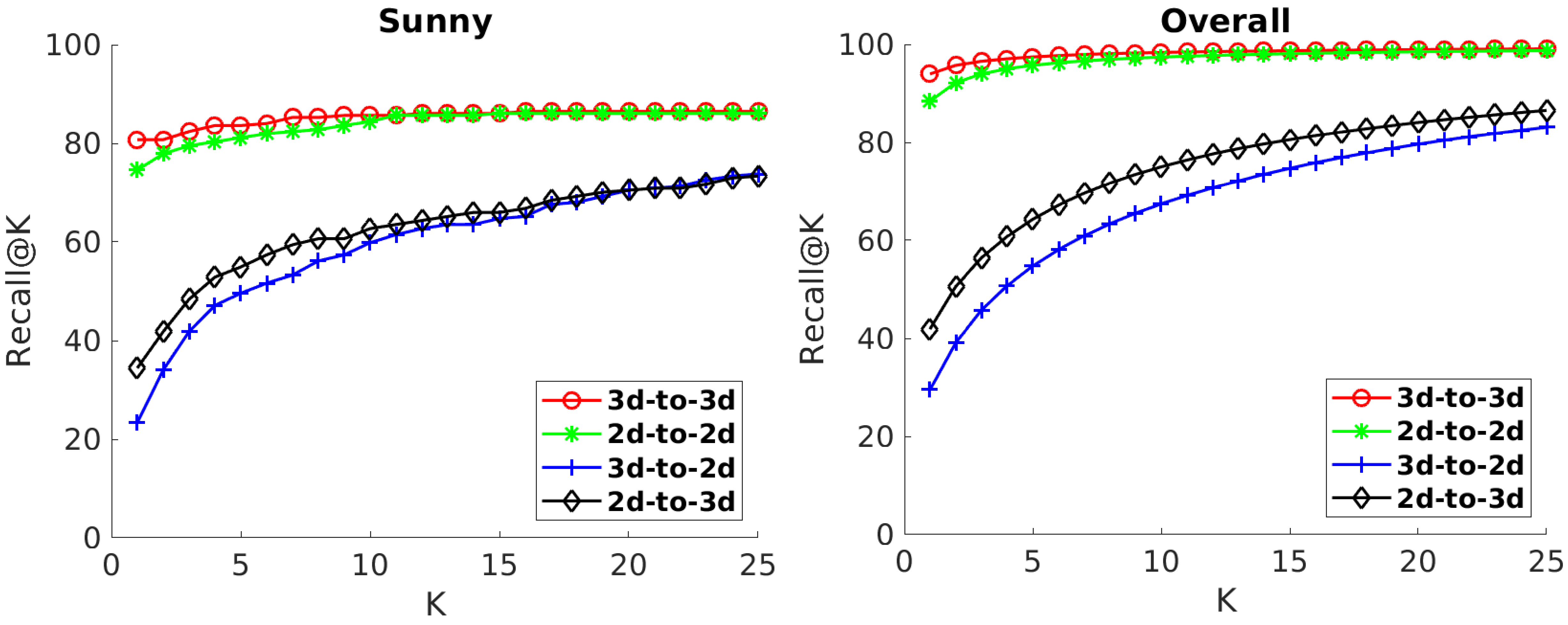
[1] Mikaela Angelina Uy and Gim Hee Lee. «Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition.», CVPR, 2018
[2] Wenxiao Zhang and Chunxia Xiao, «PCAN: 3D Attention Map Learning Using Contextual Information for Point Cloud Based Retrieval", CVPR 2019

2D-3D graphs



Night
Snow
100
80
60
40
20
0
Recall@K
0
5
10
15
20
25
K
3d-to-3d
2d-to-2d
3d-to-2d
2d-to-3d

2D-3D graphs



Sunny
Overall
Recall@K
K
100
80
60
40
20
0
5
10
15
20
25
3d-to-3d
2d-to-2d
3d-to-2d
2d-to-3d

POLITECNICO MILANO 1863

Advanced Deep Learning for 3D Spatial Data

- Deep Learning in 3D for Robotics (a.k.a. too much for 4 hours) -

Prof Matteo Matteucci (matteo.matteucci@polimi.it)

Artificial Intelligence and Robotics Laboratory
Politecnico di Milano

AIRLAB
ARTIFICIAL INTELLIGENCE AND ROBOTICS LAB