Segmentation of LiDAR scans using Cylindrical 3D Convolution Network

Zubin Bhuyan
zubin_bhuyan@student.uml.edu

Zhang Zhang
zhang_zhang@student.uml.edu

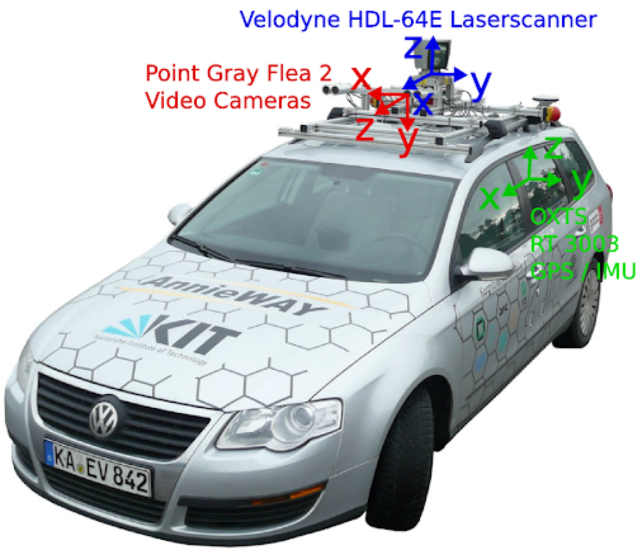
December 15, 2021

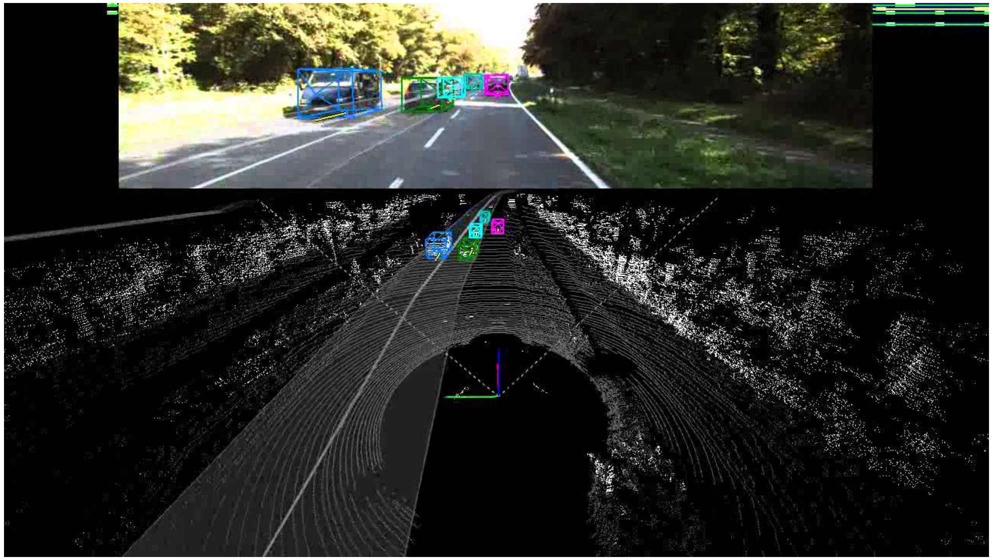
Outline

- Introduction
  - Semantic segmentation of LiDAR scans
- Existing approaches
- Proposed Method
  - Transforming 3D to Cylindrical
  - Proposed framework
- Evaluation & Performance

Introduction

- Semantic scene understanding is essential for many applications.
- Especially for autonomous driving vehicles.
  - Additional depth information.
  - RGB camera, LiDAR, RADAR.
- Much more challenging than detecting 2D image segmentations





Introduction

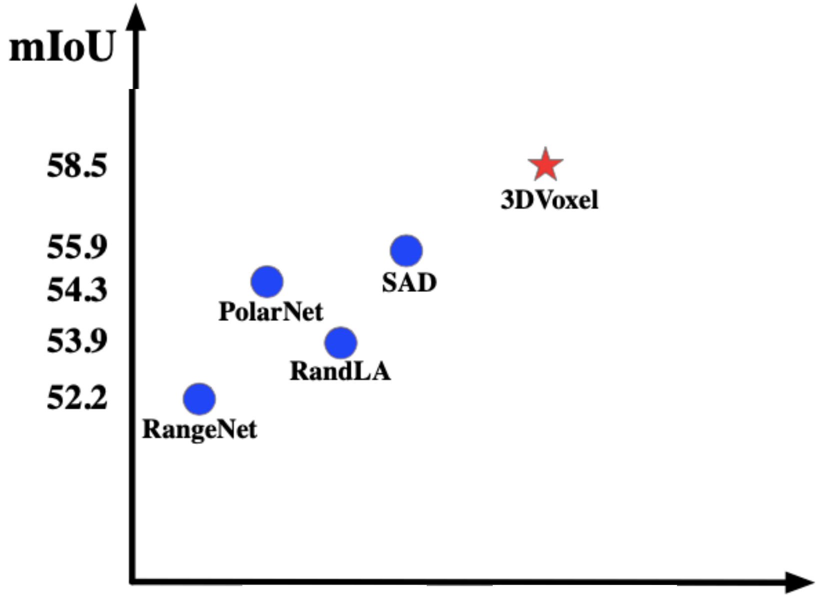
- Publicly available data and benchmarks have allowed for rapid advancement of the field.
- LiDAR allows accurate measurement for even small objects.

- Semantic segmentation of LiDAR scans is currently one of the most researched areas.
  - Improved scene understanding.
  - Better performance of autonomous driving vehicles in unmapped areas, especially urban setting.

Existing approaches: LiDAR data

- Transform data from 3D to 2D space.
  - Range/depth image based approaches.
  - Bird's-eye-view image based.

- Partition data into 3D cube spaces and apply 3D convolutions.
  - 3D-Voxel.
- Improvement in outdoor scene is minimal.



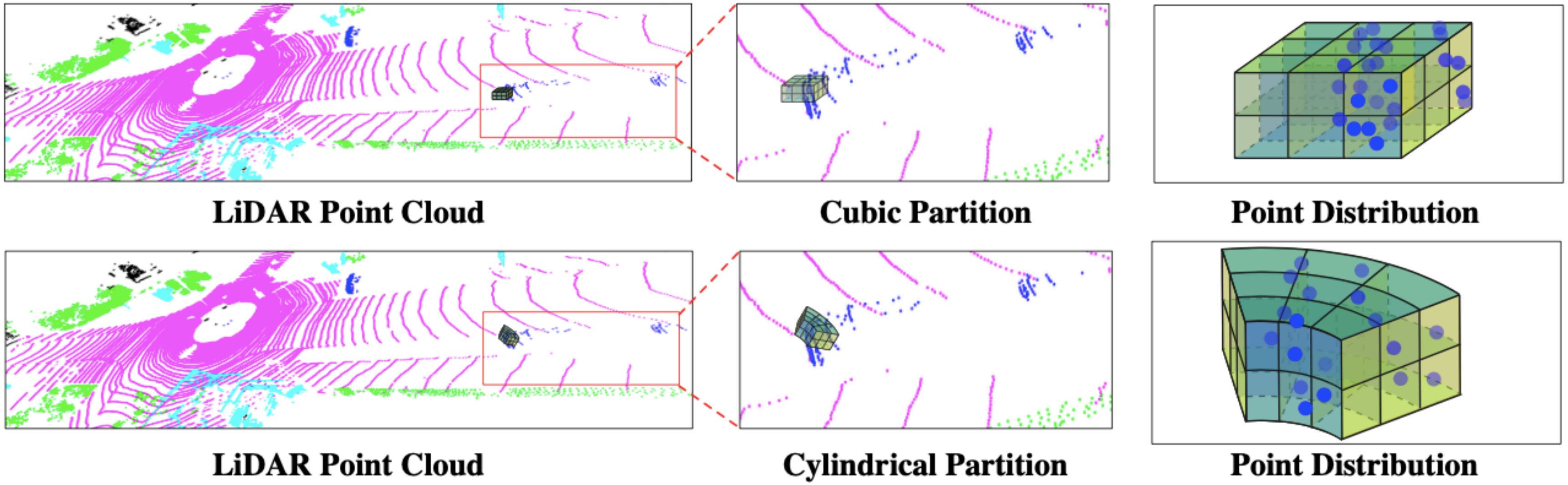
Motivation

- Two issues with point cloud data of LiDAR:
    - Sparsity and varying density.
- Most approaches assume uniform distribution and split the space in uniform cubes.
- This results in incorrect representation and later affects the 3D convolution process.

Motivation

- Two issues with point cloud data of LiDAR:
  - Sparsity and varying density.
- Most approaches assume uniform distribution and split the space in uniform cubes.
- This results in incorrect representation and later affects the 3D convolution process.
- Zhu, et al. in *Cylindrical and Asymmetrical 3D Convolution Networks*, uses 3D cylindrical partitions.

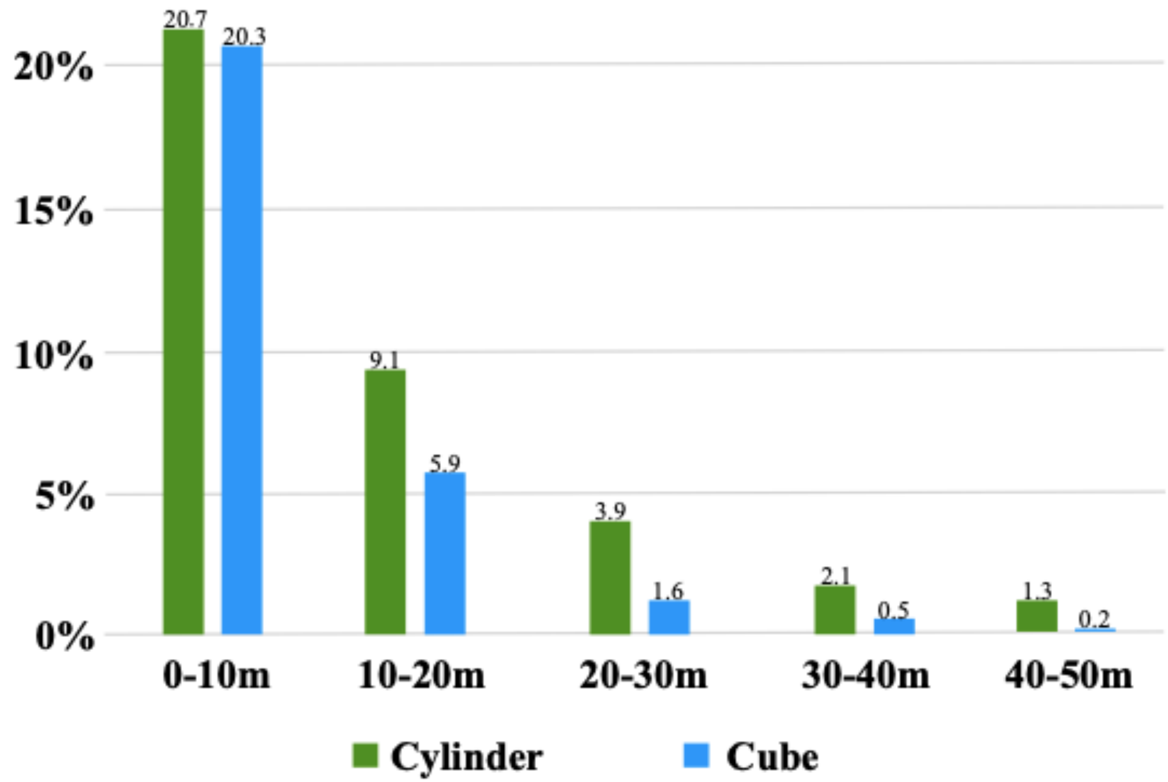
Motivation

- Zhu, et al. in *Cylindrical and Asymmetrical 3D Convolution Networks*.
- *Farther the distance lesser is the point density.*
- 

  - comparison of a standard cubic partition and a cylindrical partition.

Motivation

- Zhu, et al. in *Cylindrical and Asymmetrical 3D Convolution Networks*.



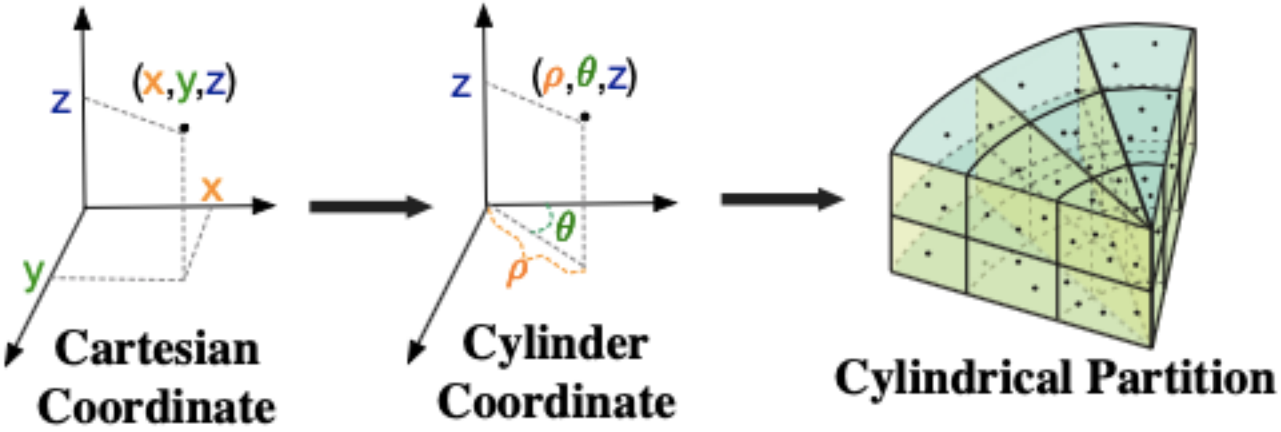
Proportion of non-empty cells at different distances between cylindrical and cubic partition.

Proposed Method

- Transforming 3D euclidian location to cylinder coordinate:
  - $(x, y, z) \rightarrow (\rho, \theta, z)$

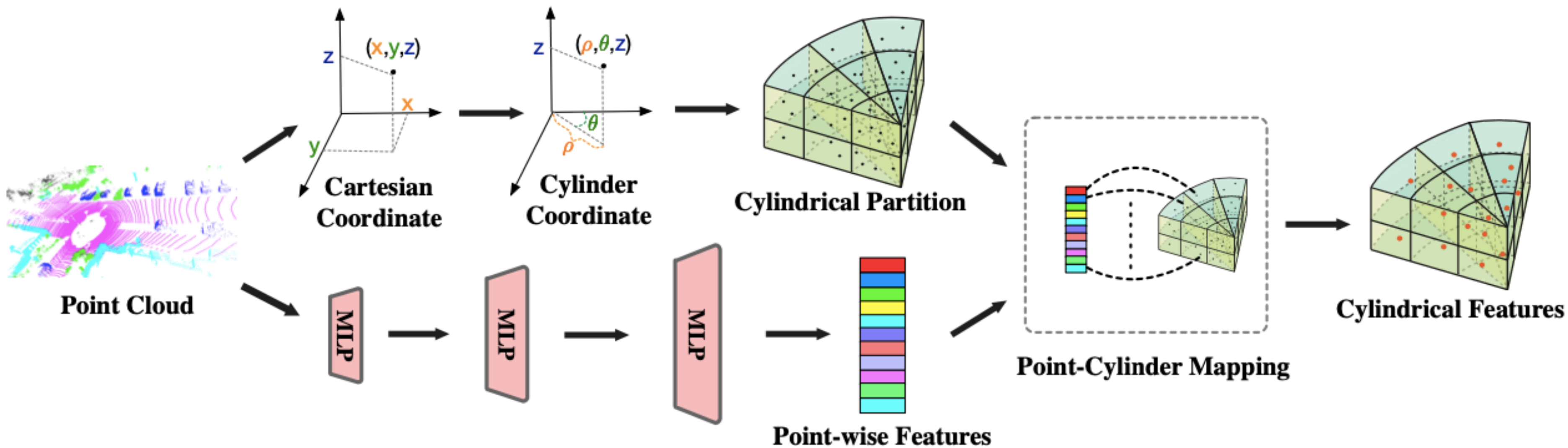
  Then cylindrical partition performs the split on these three dimensions.

-



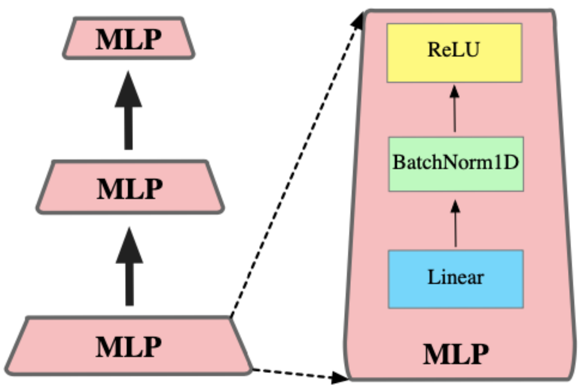
Proposed Method

- Pipeline of the cylindrical partition:



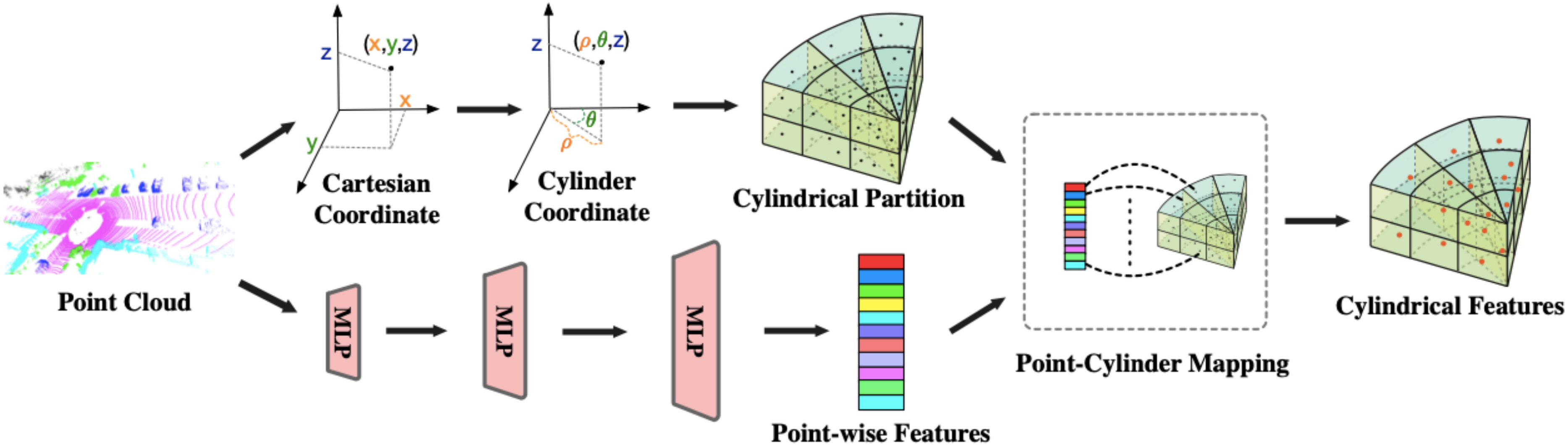
Per-point Multi-Layer Perceptions(MLPs) to extract point features, but it lacks local context modeling.

Detailed workflow of MLPs in Cylindrical Partition and Point-wise Refinement Module

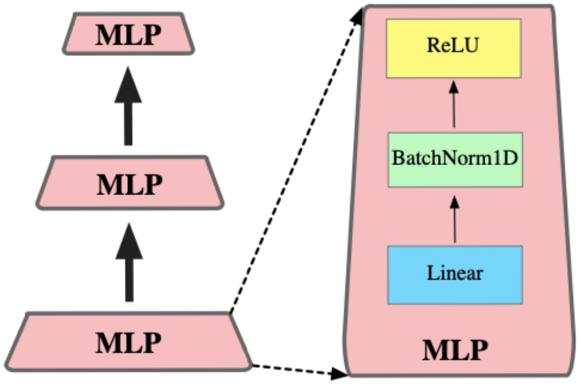


Proposed Method

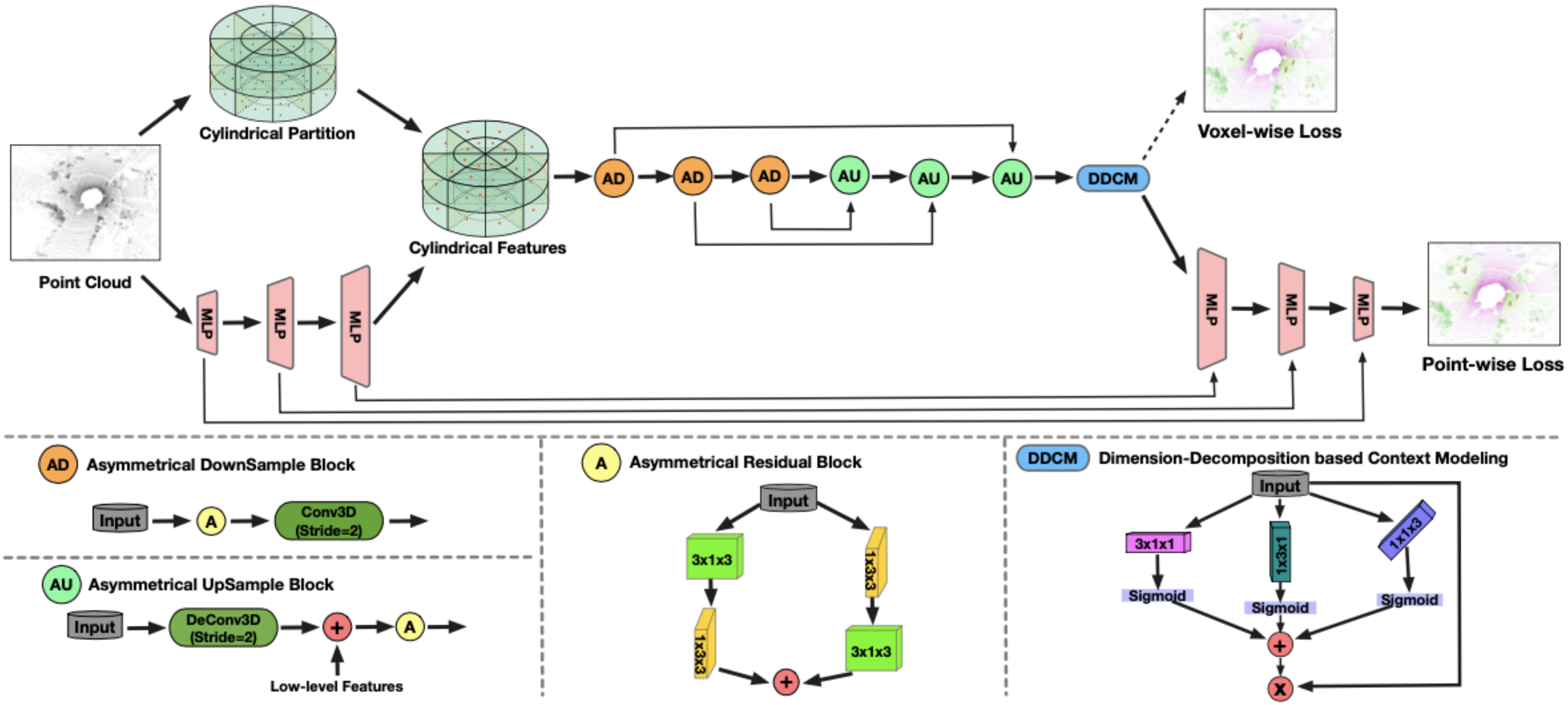
- Pipeline of the cylindrical partition:



Detailed workflow of MLPs in Cylindrical Partition and Point-wise Refinement Module



Overall framework



Asymmetrical residual block strengthens the horizontal and vertical kernels

[Ding et al., ACNet: Strengthening the kernel skeletons for powerful CNN via asymmetric convolution.]

Proposed method

- We further improve the model by implementing our sparse convolution checker which was missing in the official implementation.
- Currently our implementation does not show improvement over the SOTA, but it can perform at par with other models.

Experiments and results

- Semantic Kitti dataset
- Implementation was done in PyTorch.

Methods	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic
TangentConv [34]	35.9	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
Darknet53 [2]	49.9	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
RandLA-Net [16]	50.3	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
RangeNet++ [27]	52.2	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
PolarNet [51]	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
SqueezeSegv3 [45]	55.9	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
Salsanext [10]	59.5	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	62.1
KPConv [36]	58.8	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	**95.0**	64.2	84.8	69.2	69.1	56.4	47.4
FusionNet [48]	61.3	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8	68.8	77.1	30.8	92.5	**69.4**	84.5	69.8	68.5	60.4	**66.5**
KPRNet [19]	63.1	95.5	54.1	47.9	23.6	42.6	65.9	65.0	16.5	**93.2**	**73.9**	**80.6**	30.2	91.7	68.4	**85.7**	69.8	71.2	58.7	64.1
TORANDONet [13]	63.1	94.2	55.7	48.1	40.0	38.2	63.6	60.1	34.9	89.7	66.3	74.5	28.7	91.3	65.6	85.6	67.0	**71.5**	58.0	65.9
Ours	61.29	96.26	46.2	59.1	44.3	40	64.91	47.87	10.41	93.2	40.71	78.48	7.9	91.8	57.3	88.3	63.75	75.78	63.43	50.48
Cylinder3D	**67.8**	**97.1**	**67.6**	**64.0**	**59.0**	**58.6**	**73.9**	**67.9**	**36.0**	91.4	65.1	75.5	**32.3**	91.0	66.5	85.4	**71.8**	68.5	**62.6**	65.6

Future work

- We plan to continue to work on this project.
  - Improve training optimizations.
  - Test with hyper parameters.
  - Test with nuscenes dataset.

Thank you!