面向大规模场景识别和环境分析的三维点云深度学习研究

LPD-Series：

- 3D Point Cloud Learning for Large-Scale Place Recognition
- Sequence Loop Closure Detection for Self-driving Vehicles

索传哲

Contents

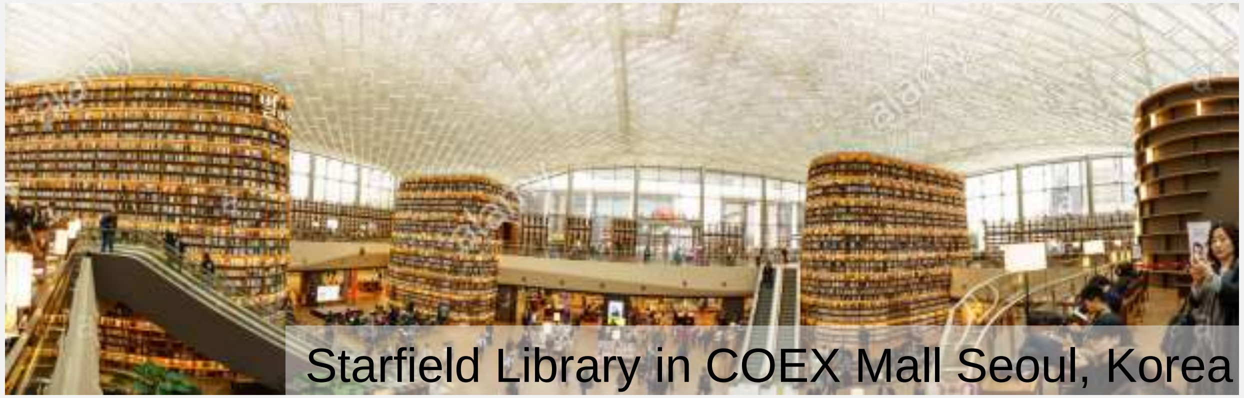
01

Introduction to Large-scale Scene Research

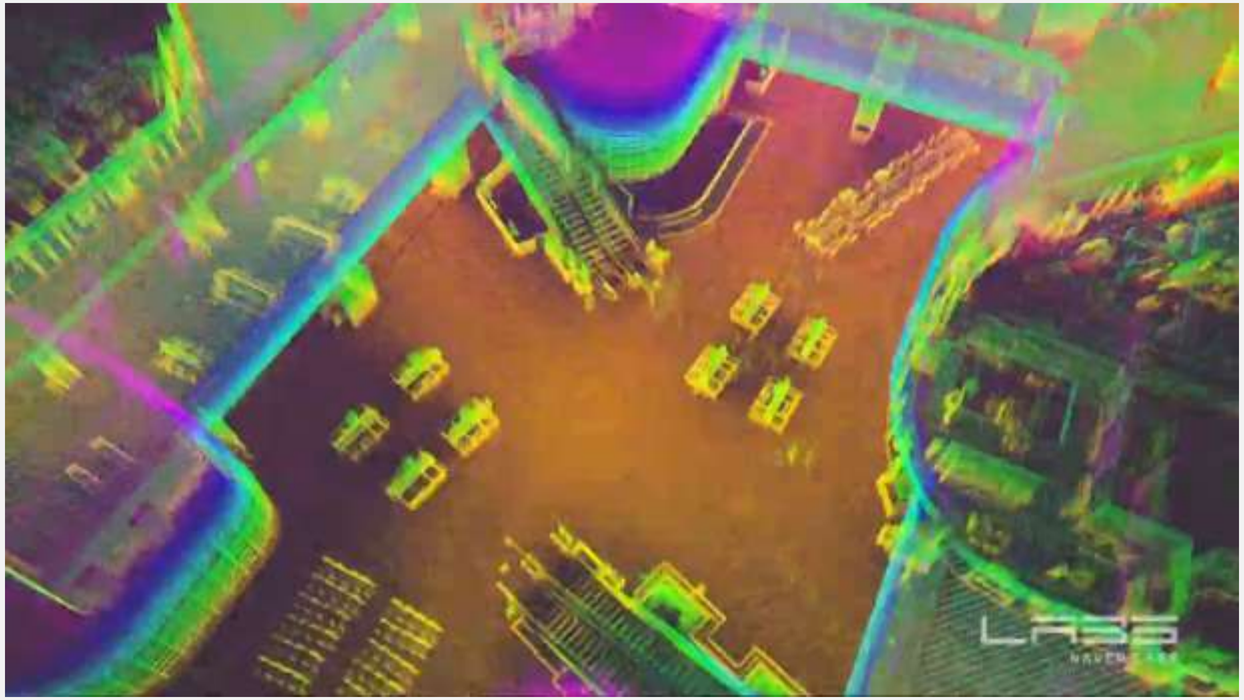
Contents，Background and Related works

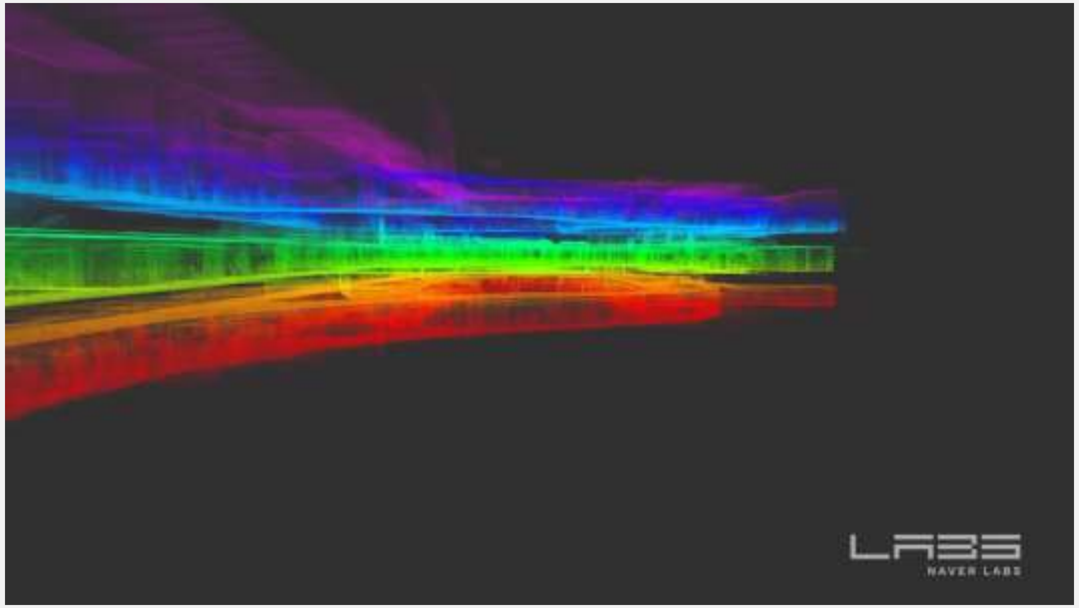
Introduction

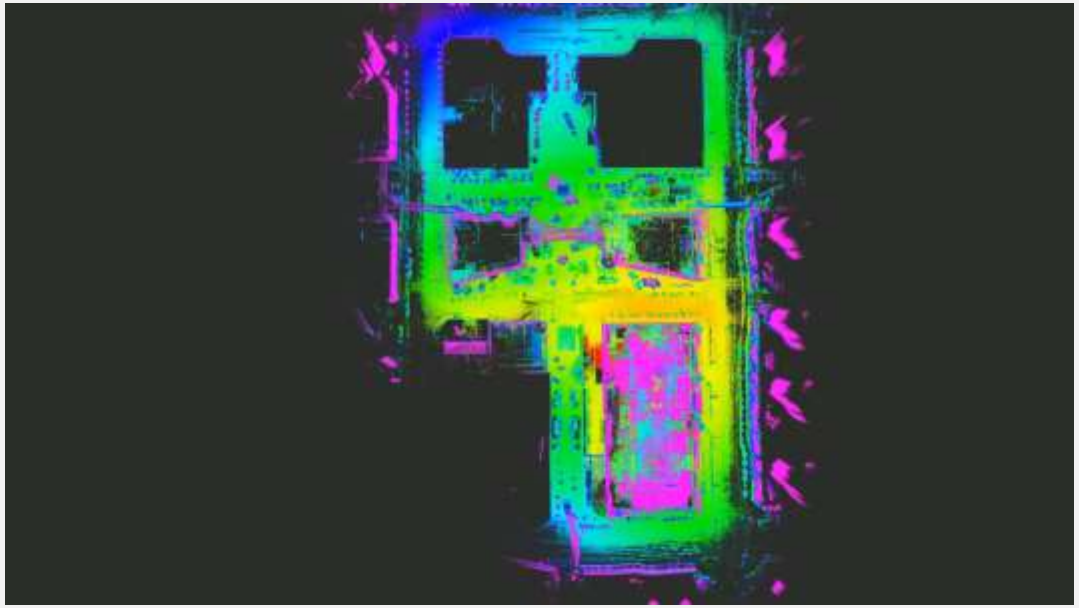
3D Point Cloud World



Starfield Library in COEX Mall Seoul, Korea







Introduction

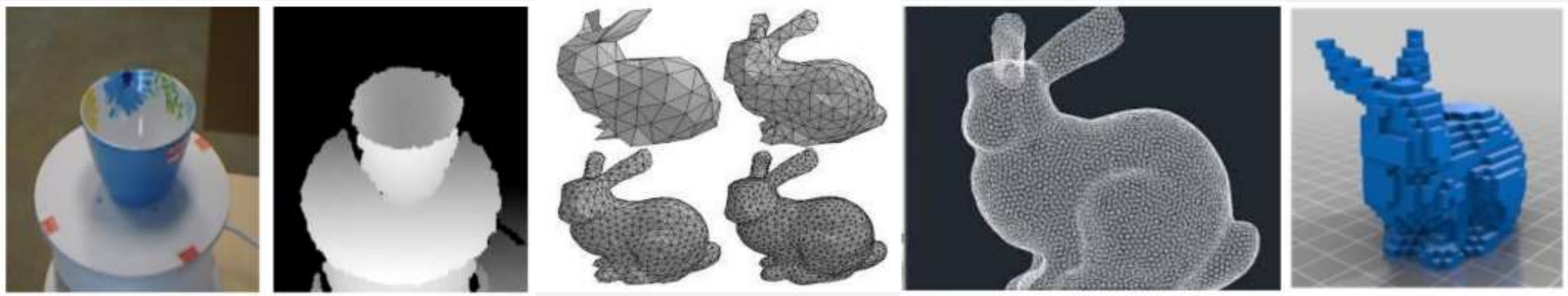
➢ 3D data representation format:

- RGB-D image
- Mesh
- PointCloud
- Voxel

➢ Academic community: very active from 2015:
Large 3D dataset:
ShapeNet (Stanford), ModelNet (Princeton)

➢ Industry community: broad applications

- Robotics
- Autonomous driving
- Virtual reality



Xie, J. (n.d.). Deep Learning Based 3D Shape Representation.

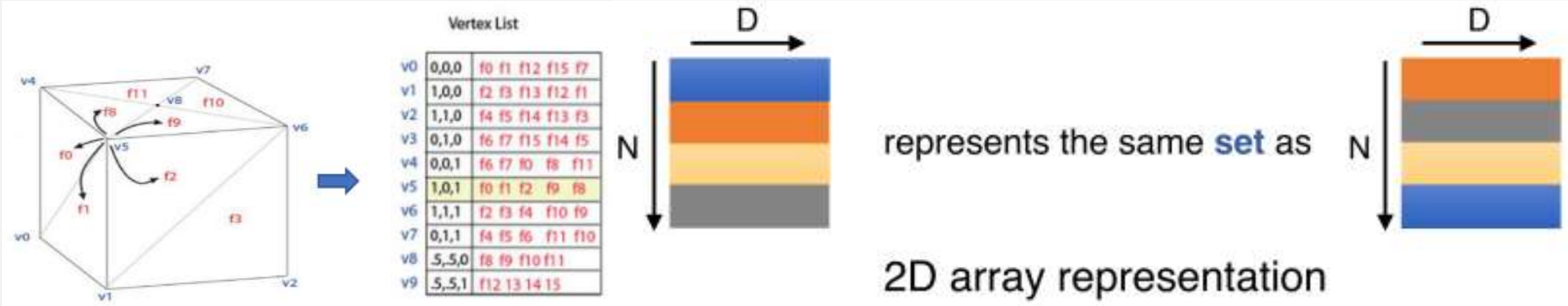
Introduction

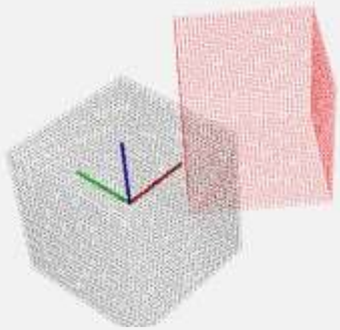
- ➢ Challenges in 3D deep learning:
- 3D model: geometric structure information ;
- 2D image: pixel value

- 3D model: irregular data structure;
- 2D image: regular data structure

- Unordered
- Invariance under transformations

- ➢ Large deformations of 3D shapes
- ➢ Large structure variations of 3Dshapes
- ➢ Partial models of 3D shapes



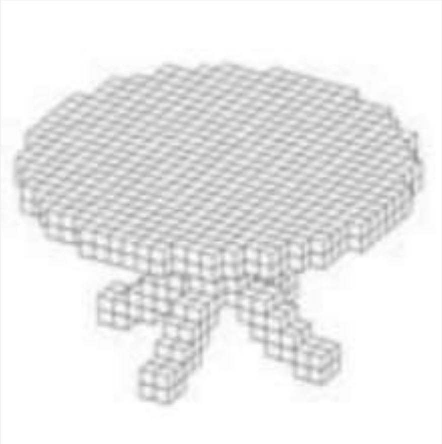


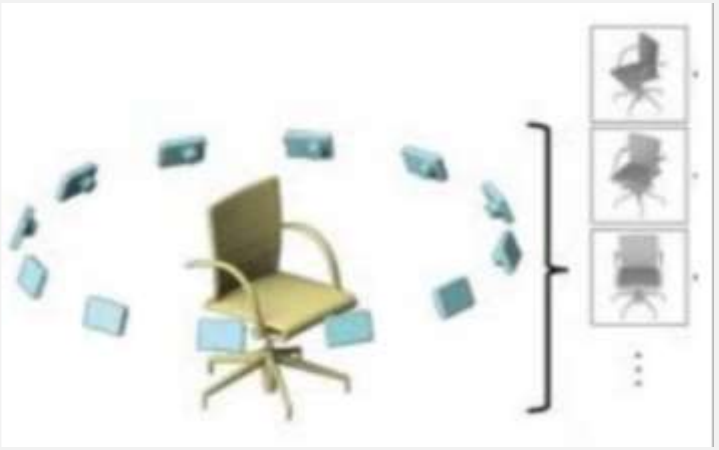
Xie, J. (n.d.). Deep Learning Based 3D Shape Representation.

Introduction

- Deep learning based 3D shape feature:
  - Pointcloud
  - Diffusion geometry
  - Voxelization
  - Projection (Multi-views)

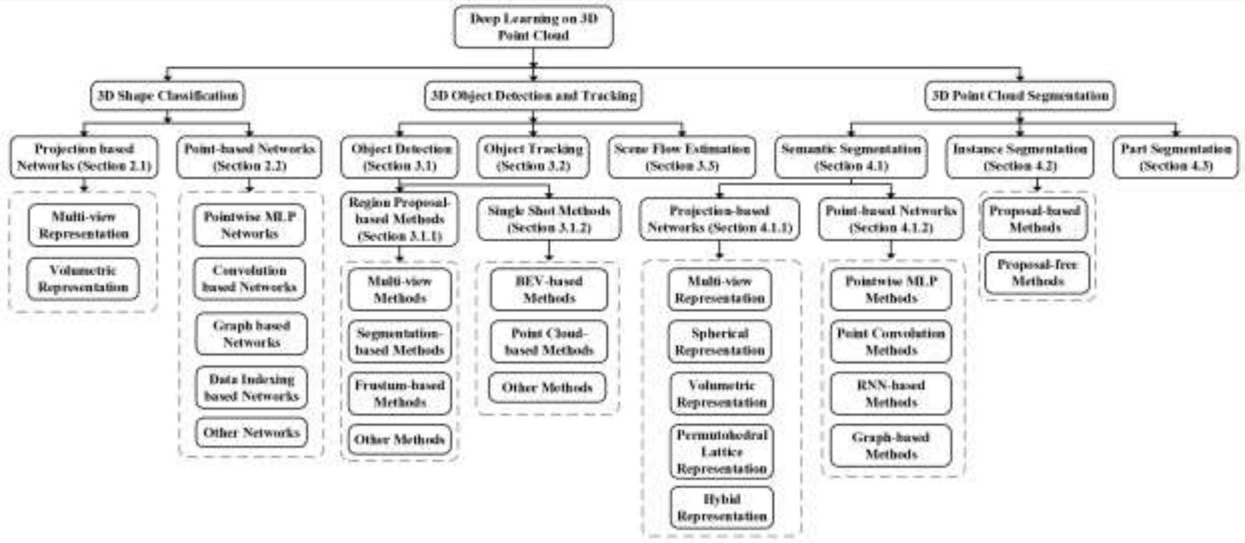
Conversion	Deep Net
Voxelization	3D CNN
Projection/Rendering	2D CNN
Feature extraction	MLP&Fully Connected



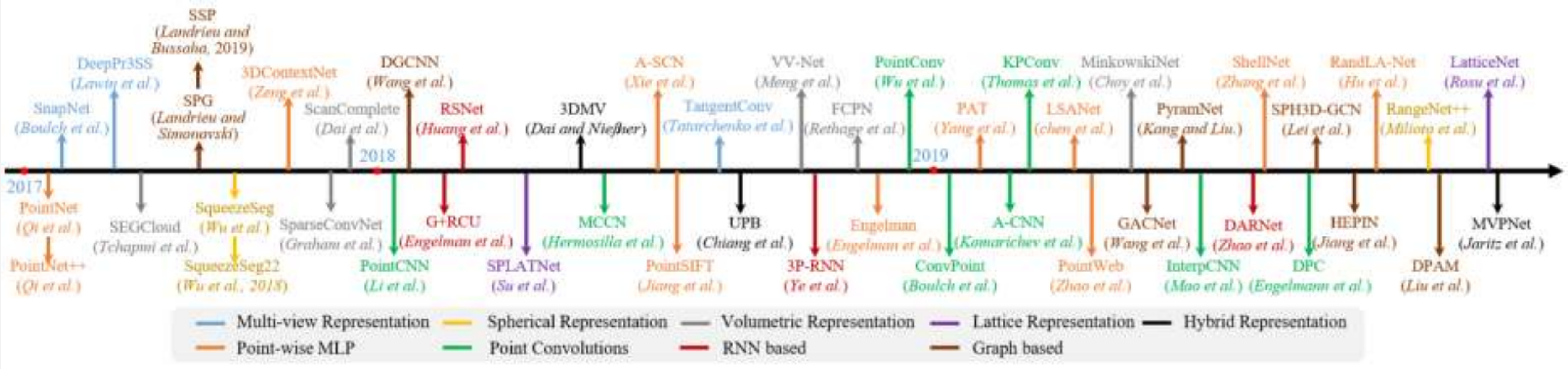


Xie, J. (n.d.). Deep Learning Based 3D Shape Representation.

Introduction



A taxonomy of deep learning methods for 3D point clouds



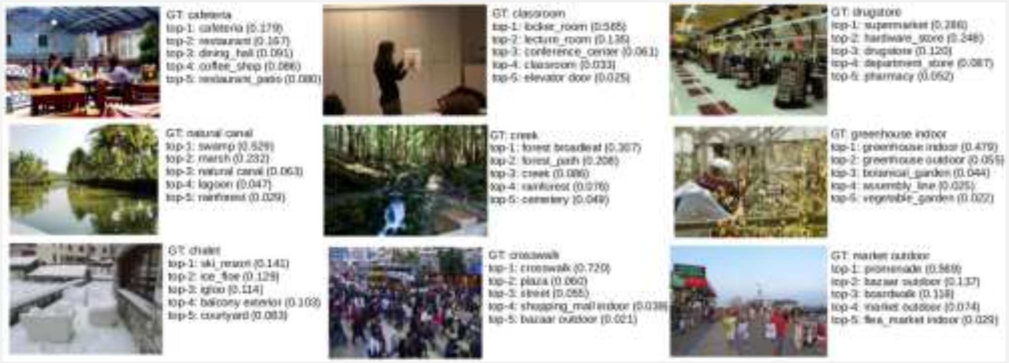
Milestone of Point Cloud Learning

Y. Guo, H. Wang, Q. Hu, et al. Deep Learning for 3D Point Clouds: A Survey. arXiv preprint arXiv:1912.12033, 2019.

Introduction

- What’ s Place Recognition?
- What can Place Recognition do?

(Places 365)

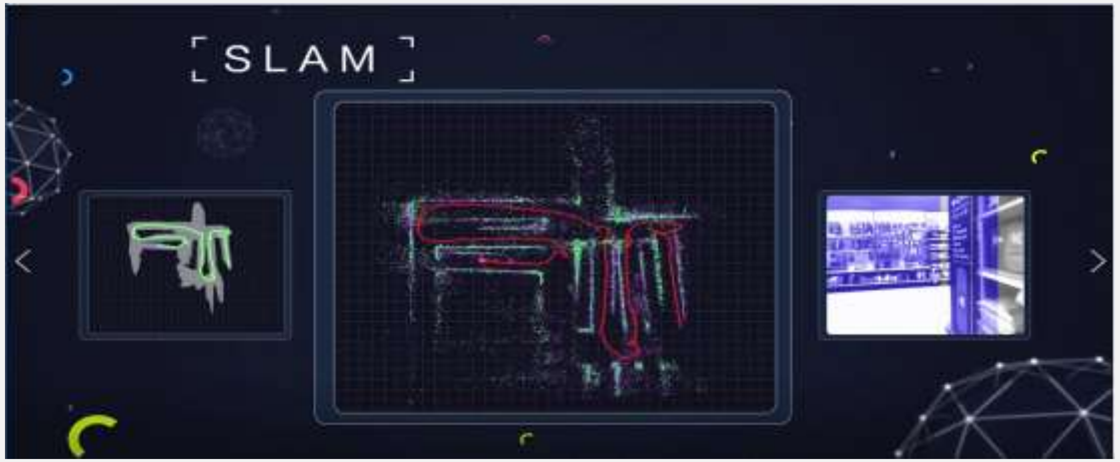


Classification

Retrieval

Introduction

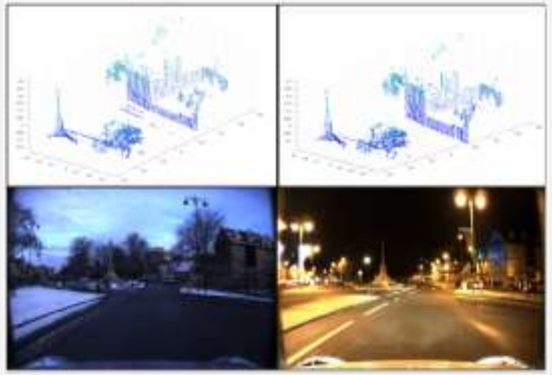
- What’ s Place Recognition?
- What can Place Recognition do?



(Baidu)

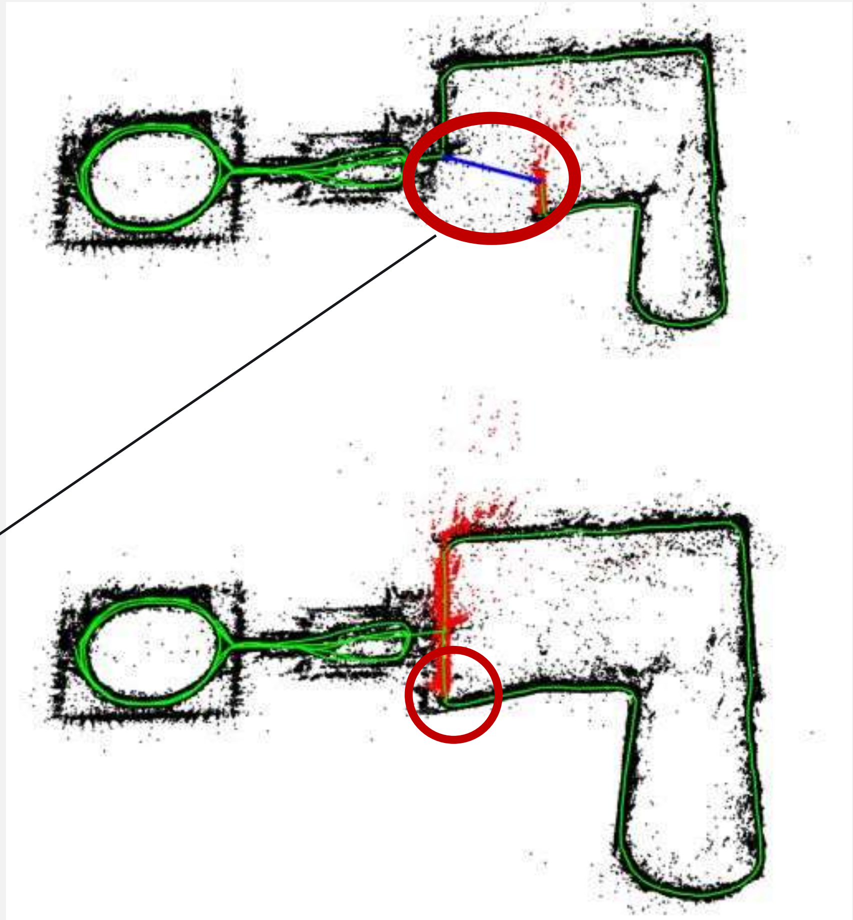
(NetVLAD)

Image



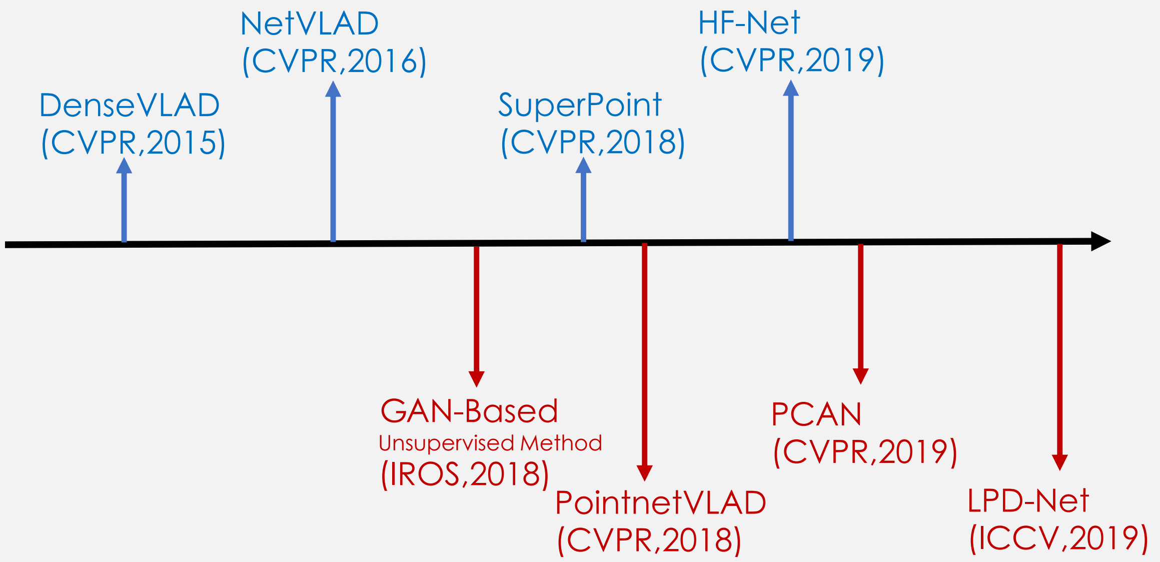
(PointnetVLAD)

Point cloud



(ORB-SLAM)

Related Works



Milestone of Point Cloud Learning for Place Recognition

Related Works

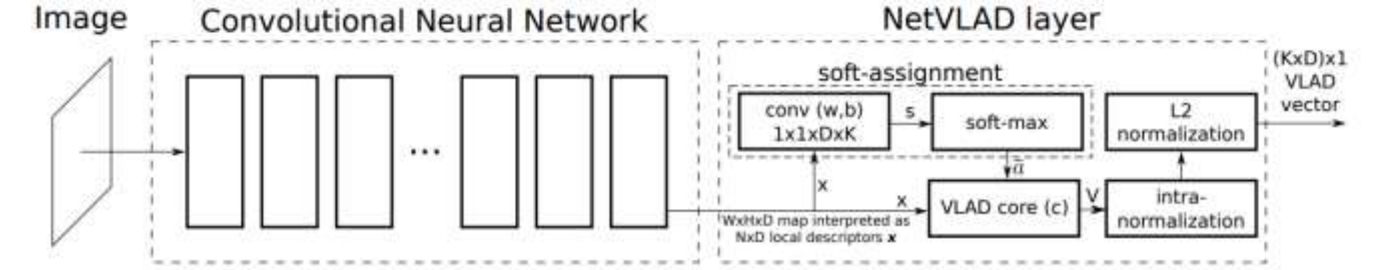
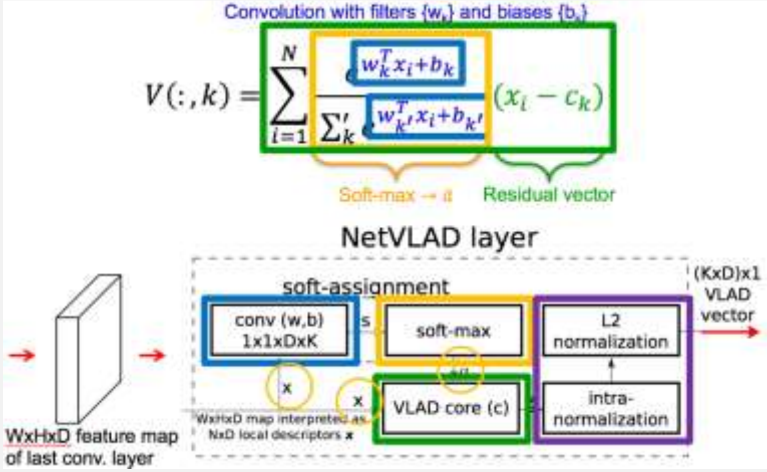


Figure 2. **CNN architecture with the NetVLAD layer.** The layer can be implemented using standard CNN layers (convolutions, softmax, L2-normalization) and one easy-to-implement aggregation layer to perform aggregation in equation (4) ("VLAD core"), joined up in a directed acyclic graph. Parameters are shown in brackets.



NetVLAD

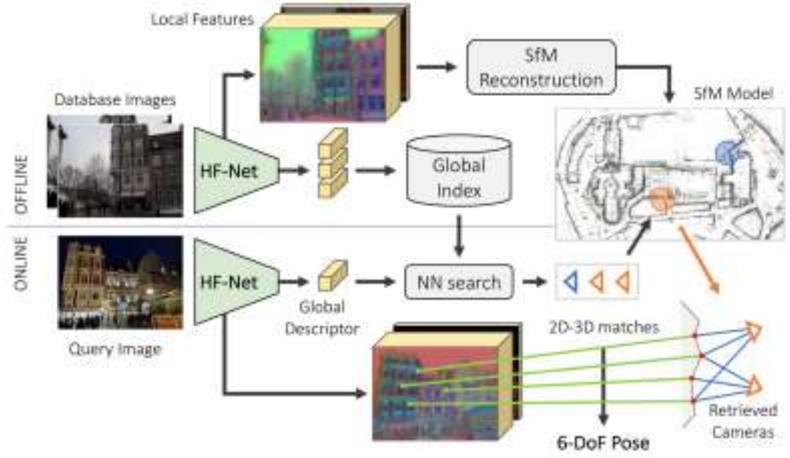


Figure 2. **The hierarchical localization with HF-Net** is significantly simpler than concurrent approaches [43, 51], yet more robust, accurate, and efficient.

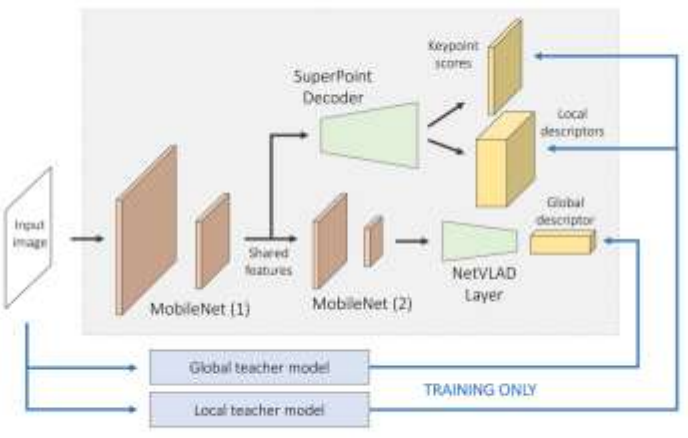
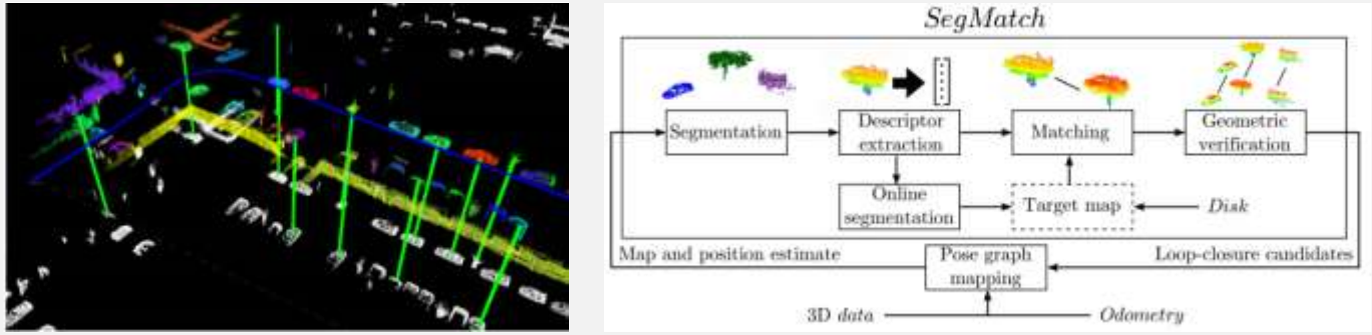


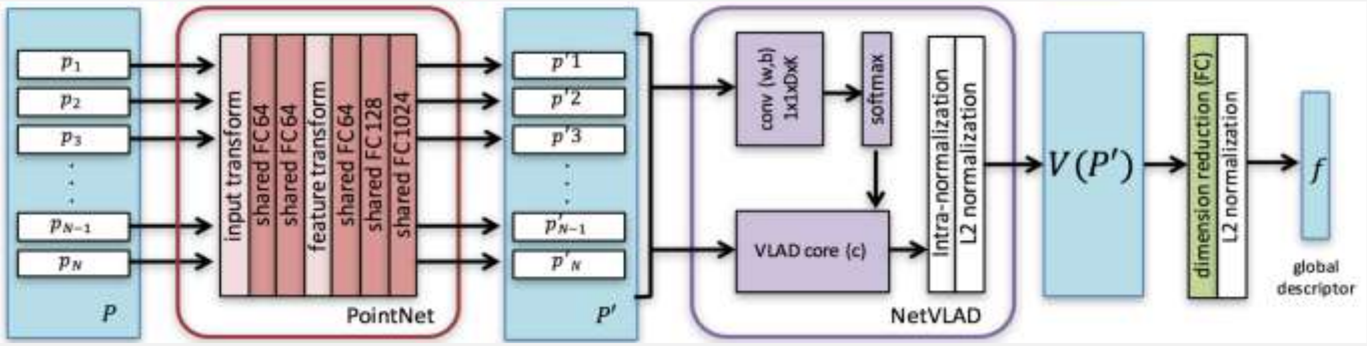
Figure 3. **HF-Net** generates three outputs from a single image: a global descriptor, a map of keypoint detection scores, and dense keypoint descriptors. All three heads are trained jointly with multi-task distillation from different teacher networks.

HF-Net

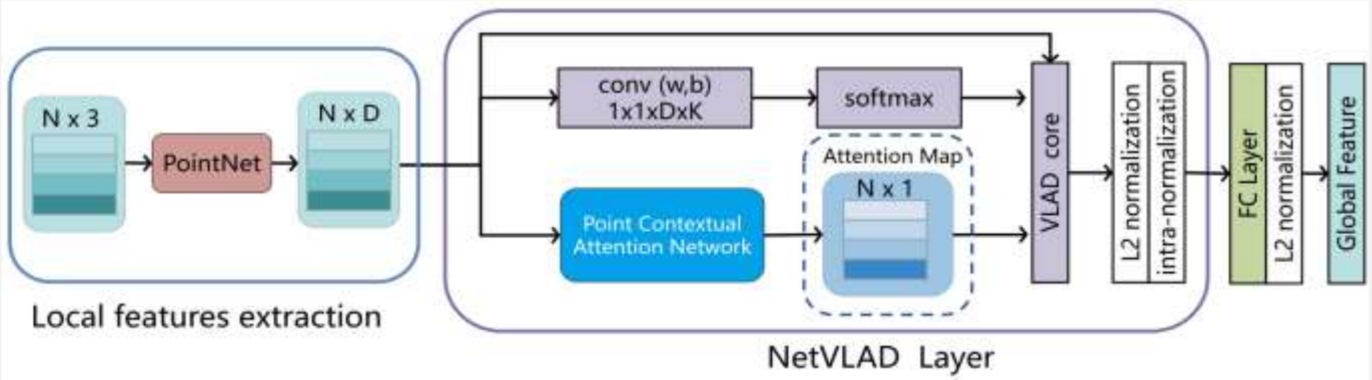
Related Works



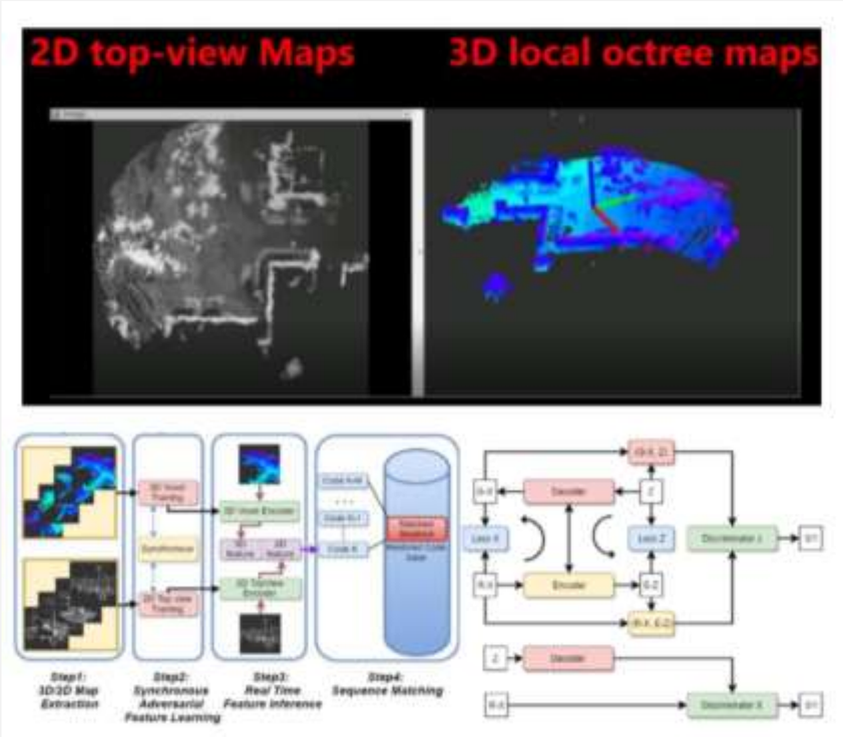
SegMatch & SegMap



PointnetVLAD



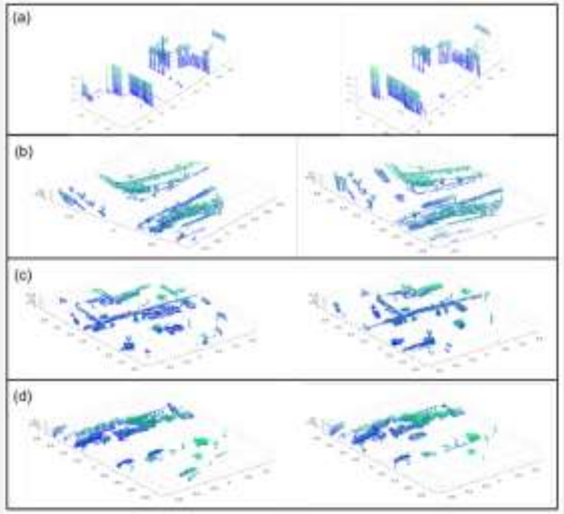
PCAN



Peng Yin. Unsupervised Method[1]

[1]P. Yin, L. Xu, Z. Liu, L. Li, H. Salman, Y. He, W. Xu, H. Wang, and H. Choset, "Stabilize an unsupervised feature learning for lidar-based place recognition," in Proc. IEEE/RSJ Int. Conf. Intell. Robots and Sys., pp. 1162-1167, 2018.

Related Works



PointnetVLAD

	PCAN	PN_VLAD	PN_MAX	PN_STD
Oxford	**83.81**	81.01	73.44	46.52
U.S.	**79.05**	77.83	64.64	61.12
R.A.	**71.18**	69.75	51.92	49.07
B.D.	**66.82**	65.30	54.74	53.02

Table 2. Baseline results showing the average recall (%) at top 1% for each of the models.

	Ave recall @1%		Ave recall @1	
	PCAN	PN_VLAD	PCAN	PN_VLAD
Oxford	**86.40**	80.70	**70.72**	63.33
U.S.	94.07	**94.45**	83.69	**86.06**
R.A.	92.27	**93.07**	82.26	**82.65**
B.D.	**87.00**	86.48	**80.31**	80.11

Table 3. Refined network results showing the average recall (%) at top 1% and at top 1 after training on Oxford, U.S. and R.A. for each of the models.

PCAN

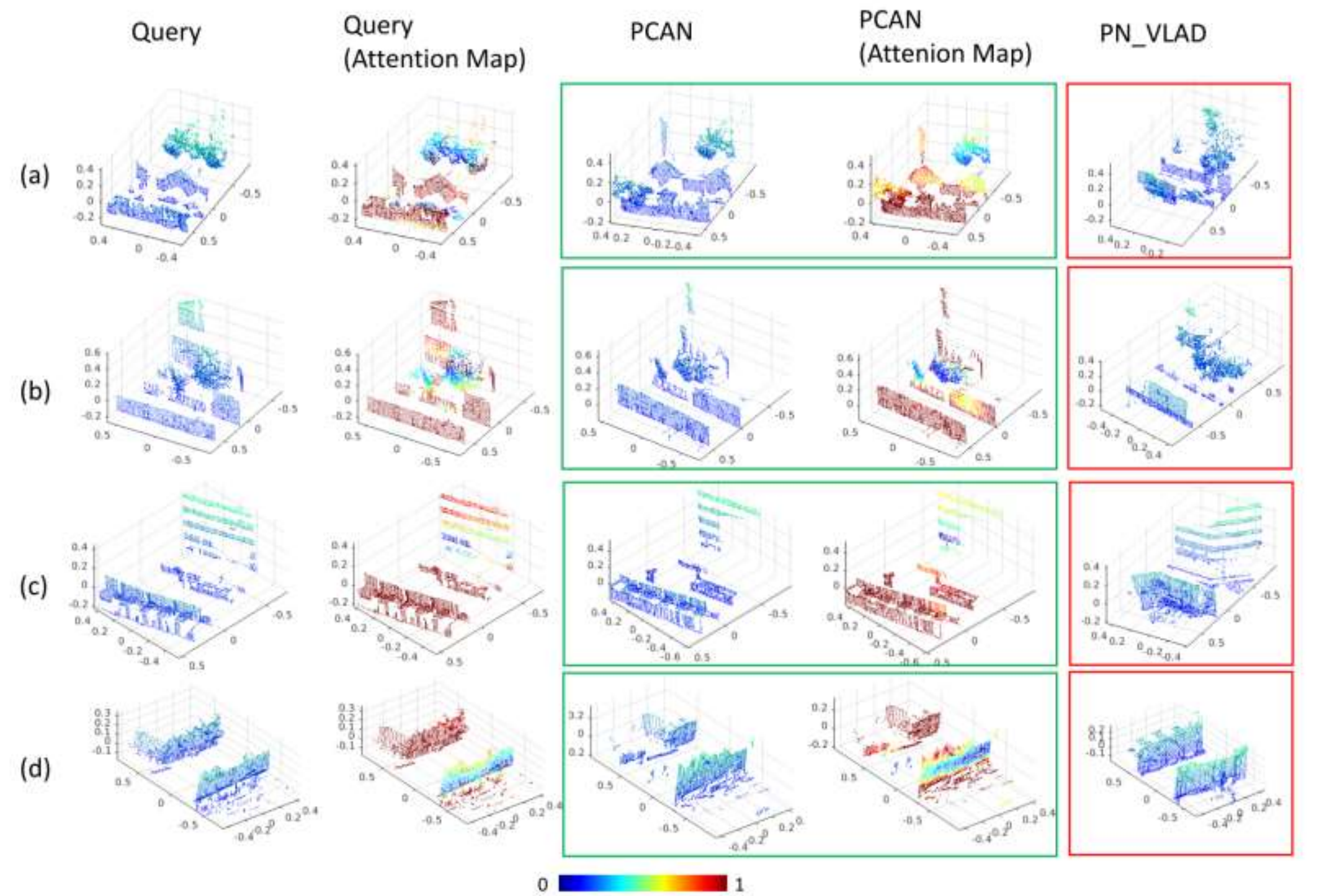


Figure 5. Example retrieval results of our network on Oxford datasets. From left to right: query point cloud, attention map of the query point cloud, the top retrieved point cloud using PCAN, the top retrieved point cloud using PointNetVLAD. Green and red borders indicate correct and incorrect retrieved results, respectively.

(PCAN,CVPR2019)

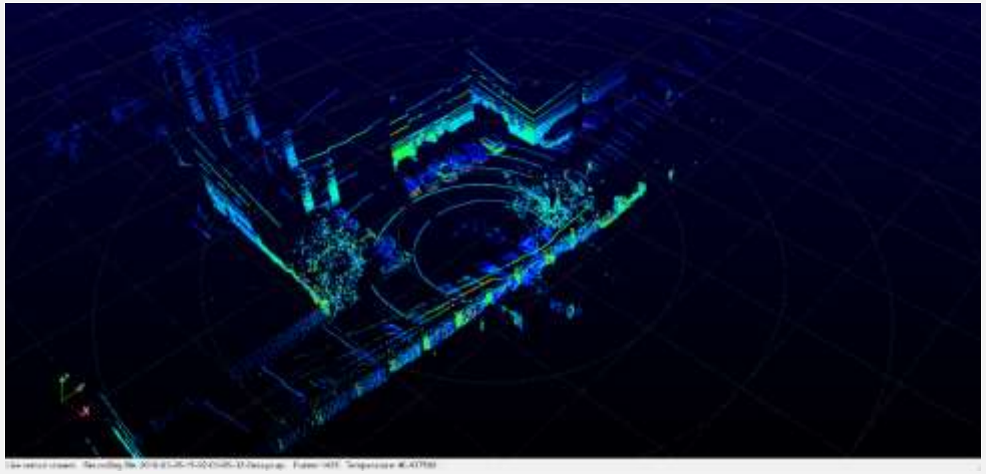
02

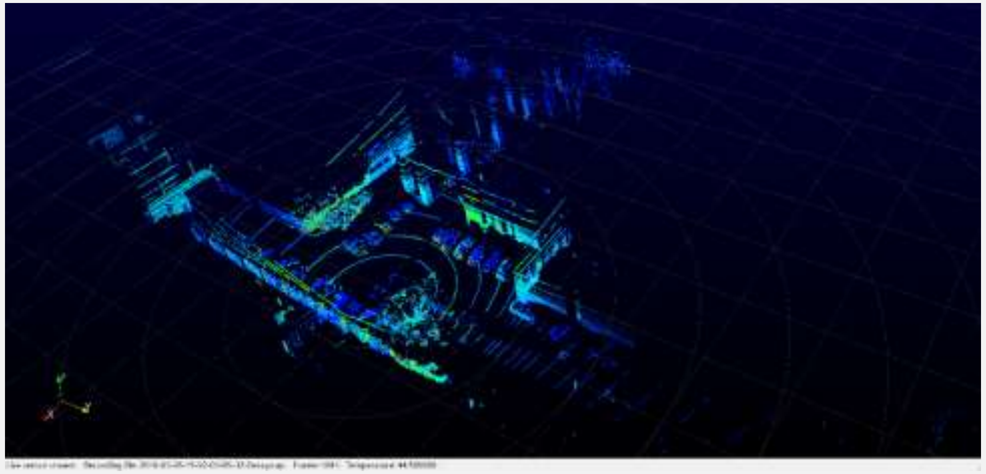
Large-scale Place Description Series

LPD-Net for place recognition

SeqLPD for loop closure detection in SLAM

LPD-Series





ICCV2019:

LPD-Net: 3D Point Cloud Learning for Large-Scale Place Recognition and Environment Analysis

Zhe Liu[1]*, Shunbo Zhou[1], Chuanzhe Suo[1], Yingtian Liu[1], Peng Yin[3], Hesheng Wang[2], Yun-Hui Liu[1]
[1]The Chinese University of Hong Kong [2]Shanghai Jiao Tong University
[3]Carnegie Mellon University

IROS2019:

SeqLPD: Sequence Matching Enhanced Loop-Closure Detection Based on Large-Scale Point Cloud Description for Self-Driving Vehicles

Zhe Liu, Chuanzhe Suo, Shunbo Zhou, Fan Xu, Huanshu Wei, Wen Chen, Hesheng Wang, Xinwu Liang, and Yun-Hui Liu

- Arxiv Paper:

ICCV: *https://arxiv.org/pdf/1812.07050.pdf*
IROS: *https://arxiv.org/pdf/1904.13030.pdf*

- Github:

LPD-Net: *https://github.com/Suoivy/LPD-net*

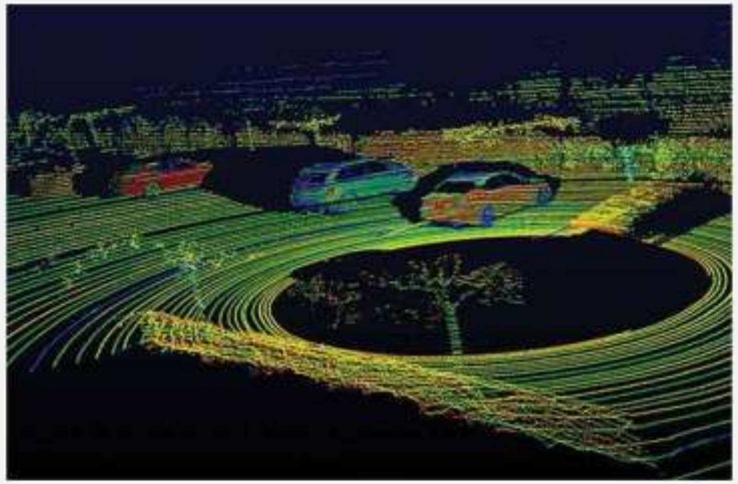
Motivation

Background:

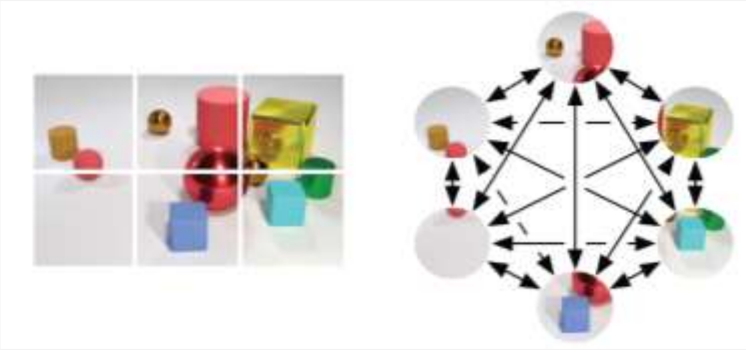
Large-scale place recognition and loop-closure detection is of great importance in robotic / self-driving applications for obtaining accurate locations and building drift-free globally consistent maps

Challenges:

- Large-scale place description approach from raw 3D point cloud
- Effective global feature extraction network, which is robust to dynamic environments, discriminative in similar places and generalizable under different applications
- Effective loop closure detection approach which is robust to unstable GPS signals and large odometer accumulate errors, for large-scale environment with long-term routes
- Ensuring real time performance and practical applicability

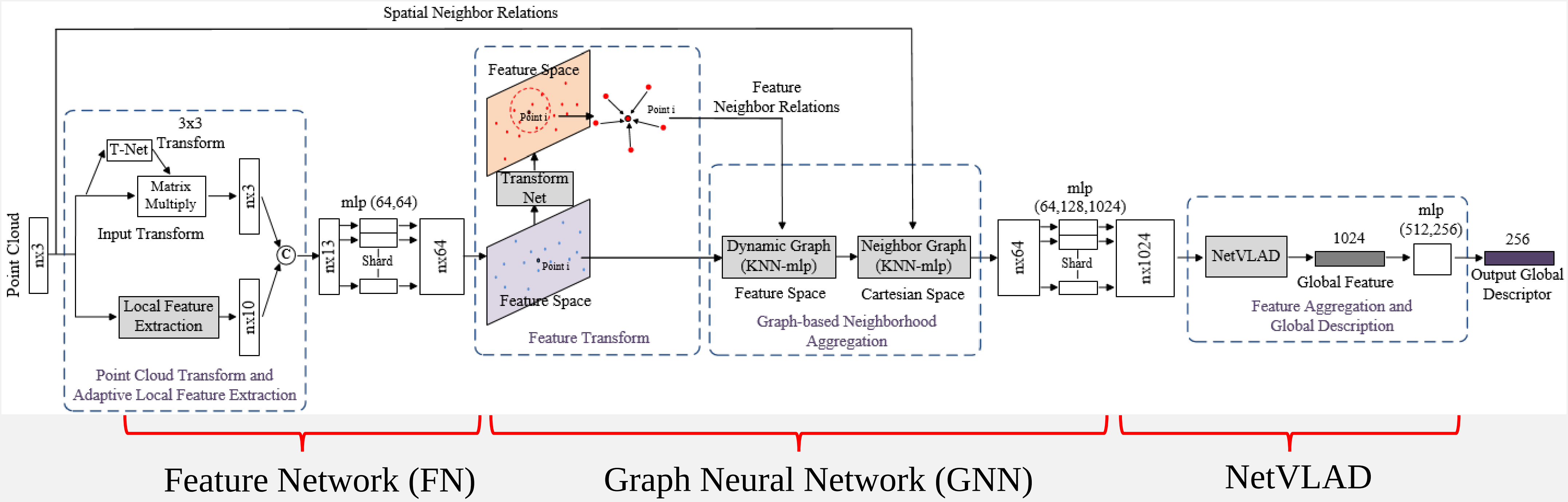


Key idea:
Efficient features
Distribution of features



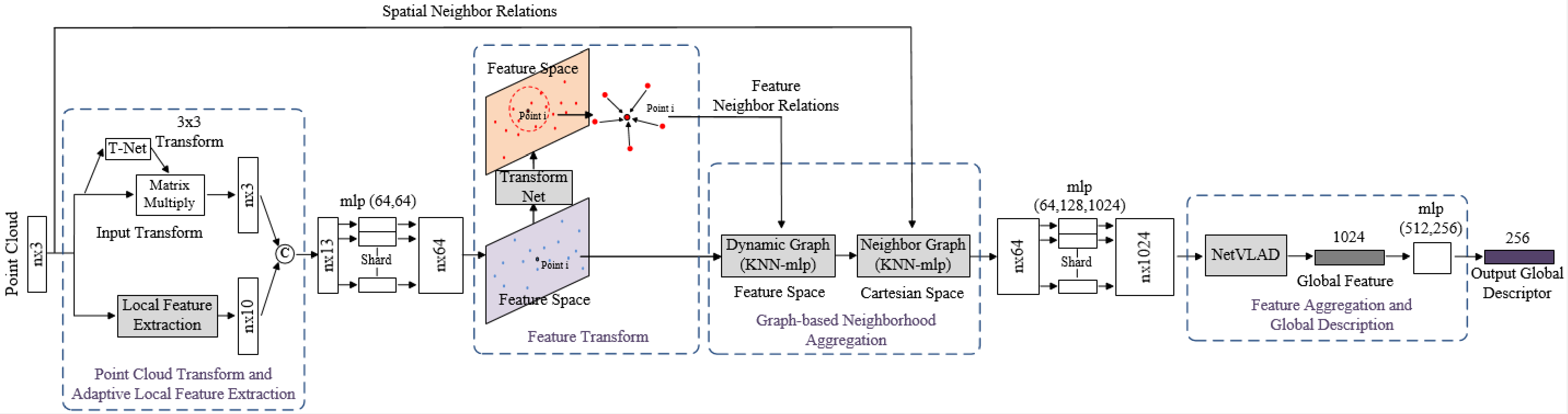
LPD-Series

- Objective: discriminative and generalizable global descriptors



LPD-Series

- Objective: discriminative and generalizable global descriptors



Feature Network (FN)

- Select optimal neighbor size (entropy minimization)
- Local features suitable for self-driving
- Extract the local structures (with mlp-based feature network)

Neighborhood size adaption

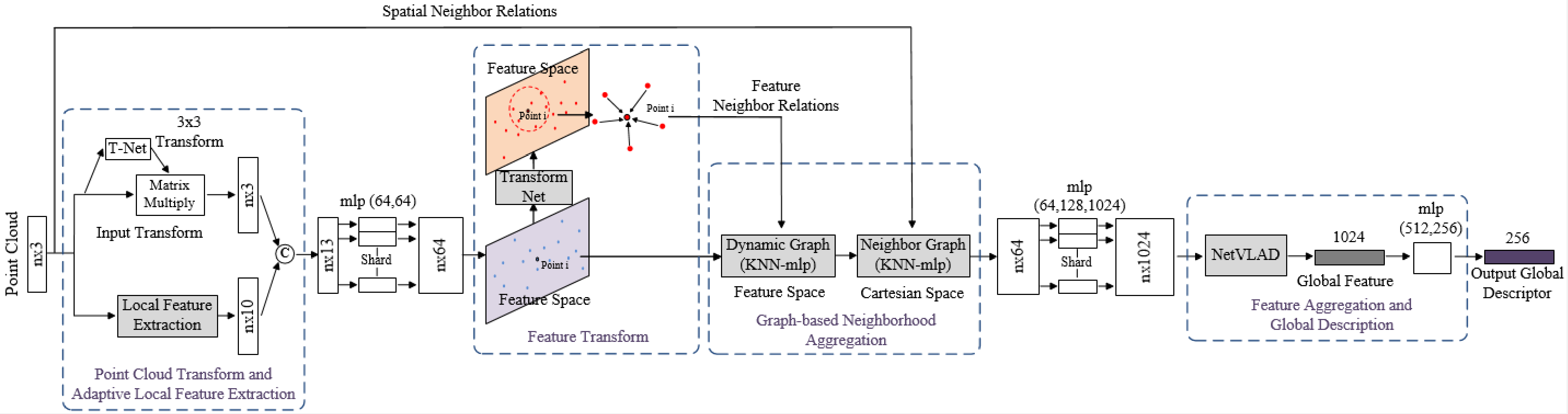
$$E_i = -L_i \ln L_i - P_i \ln P_i - S_i \ln S_i$$

$$L_i = \frac{\lambda_1^i - \lambda_2^i}{\lambda_1^i}, \; P_i = \frac{\lambda_2^i - \lambda_3^i}{\lambda_1^i} \text{ and } S_i = \frac{\lambda_3^i}{\lambda_1^i}$$

$$k_{opt}^i = \arg\min_k E_i(k)$$

LPD-Series

- Objective: discriminative and generalizable global descriptors



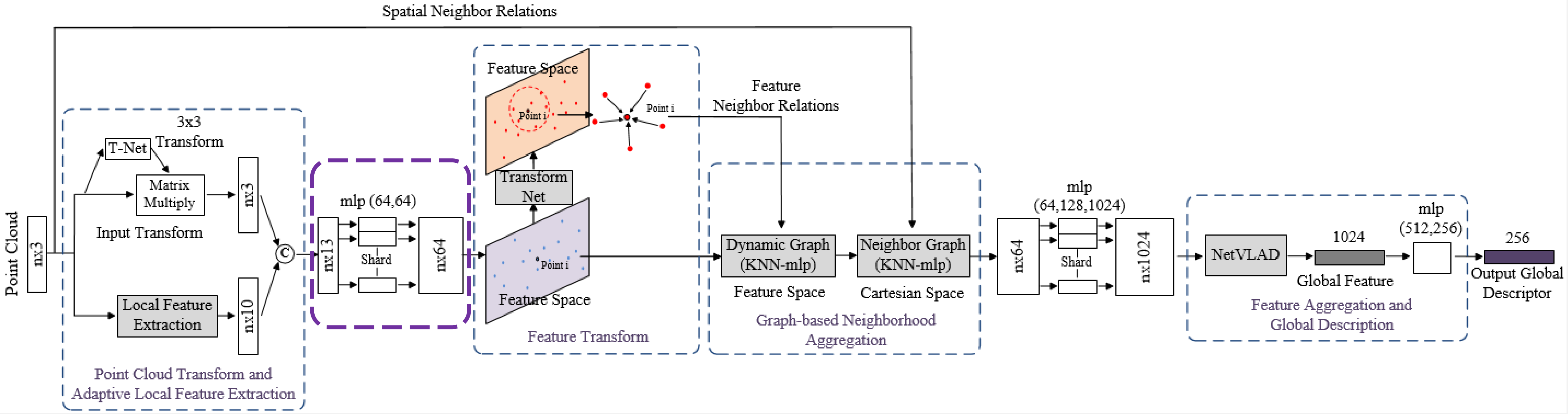
Feature Network (FN)

- Select optimal neighbor size (entropy minimization)
- Local features suitable for self-driving
- Extract the local structures (with mlp-based feature network)

- Change of curvature: $C_i = \frac{\lambda_3^i}{\sum_{j=1}^3 \lambda_j^i}$
- Omni-variance: $O_i = \frac{\sqrt[3]{\prod_{j=1}^3 \lambda_j^i}}{\sum_{j=1}^3 \lambda_j^i}$
- Vertical component of normal vector: V_i
- Height variance: $\sigma Z_{i,var}$
- Maximum height difference: $\Delta Z_{i,\max}$
- Linearity: $L_i = \frac{\lambda_1^i - \lambda_2^i}{\lambda_1^i}$
- Eigenvalue-entropy: $A_i = -\sum_{j=1}^3 (\lambda_j^i \ln \lambda_j^i)$
- Local point density: $D_i = \frac{k_{opt}^i}{\frac{4}{3}\prod_{j=1}^3 \lambda_j^i}$
- 2D linearity: $L_{i,2D} = \frac{\lambda_{2D,2}^i}{\lambda_{2D,1}^i}$
- 2D scattering: $S_{i,2D} = \lambda_{2D,1}^i + \lambda_{2D,2}^i$

LPD-Series

- Objective: discriminative and generalizable global descriptors

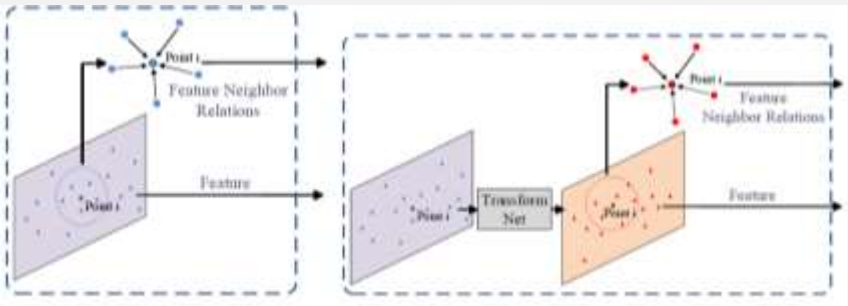


Feature Network (FN)

- Select optimal neighbor size (entropy minimization)
- Local features suitable for self-driving
- Extract the local structures (with mlp-based feature network)

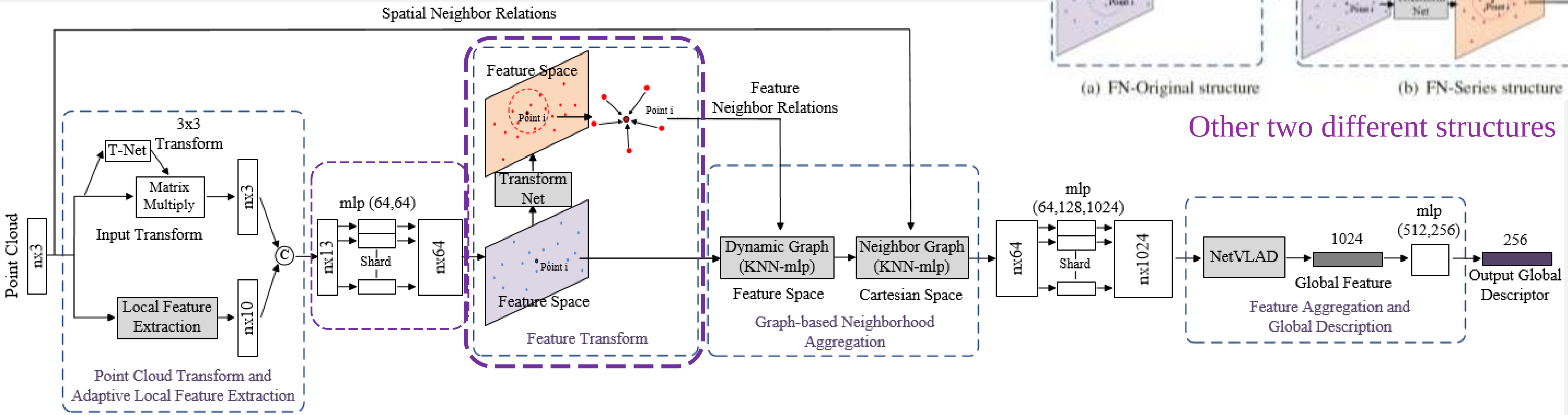
LPD-Series

- Objective: discriminative and generalizable global descriptors



(a) FN-Original structure
(b) FN-Series structure

Other two different structures



Feature Network (FN)

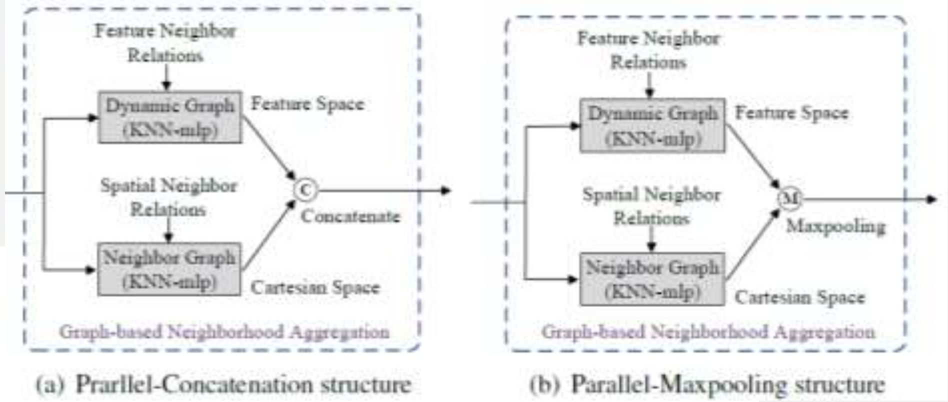
- Select optimal neighbor size (entropy minimization)
- Local features suitable for self-driving
- Extract the local structures (with mlp-based feature network)

Graph Neural Network (GNN)

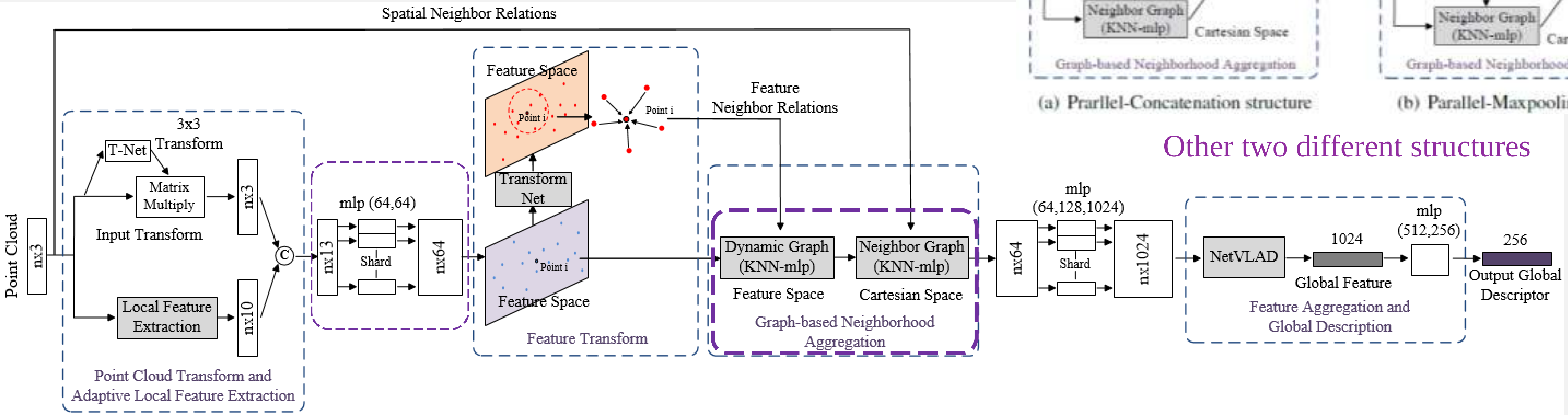
- Feature transform (invariance under different viewpoint)
- Feature aggregation
  - In Feature Space: achieve multi-scale feature learning
  - In Cartesian Space: learn geometrical distribution information of similar semantic structures
- Dynamic Graph + kNN

LPD-Series

- Objective: discriminative and generalizable global descriptors



Other two different structures



Feature Network (FN)

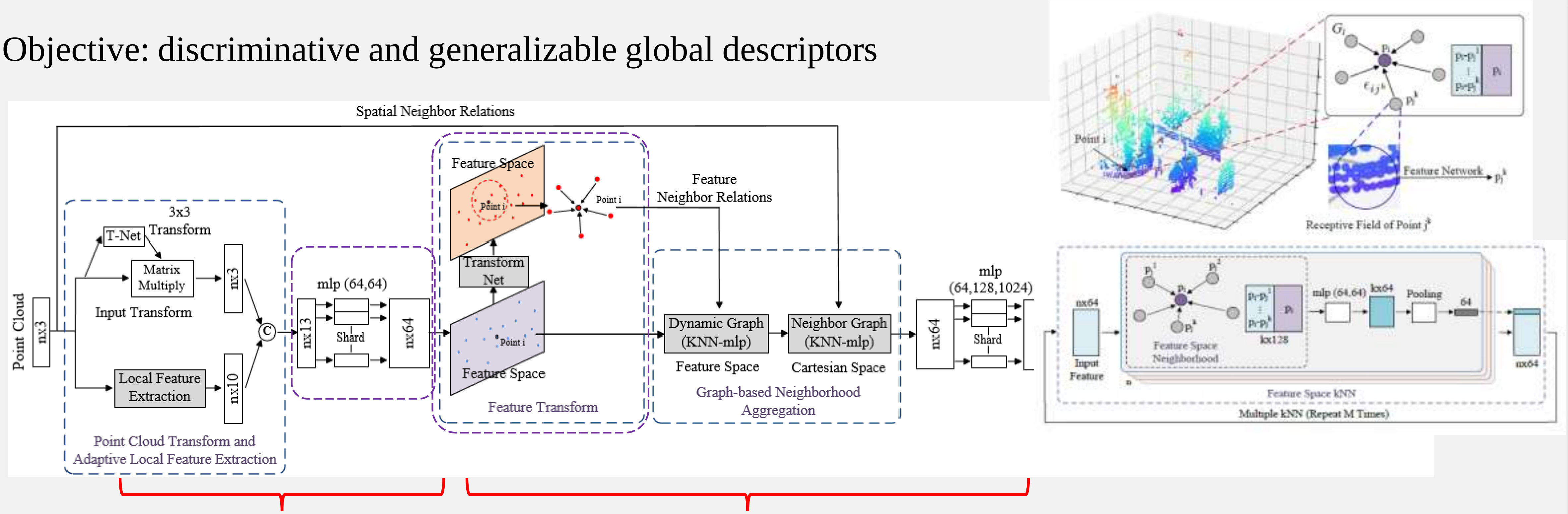
- Select optimal neighbor size (entropy minimization)
- Local features suitable for self-driving
- Extract the local structures (with mlp-based feature network)

Graph Neural Network (GNN)

- Feature transform (invariance under different viewpoint)
- Feature aggregation
  - In Feature Space: achieve multi-scale feature learning
  - In Cartesian Space: learn geometrical distribution information of similar semantic structures
- Dynamic Graph + kNN

LPD-Series

- Objective: discriminative and generalizable global descriptors



Feature Network (FN)

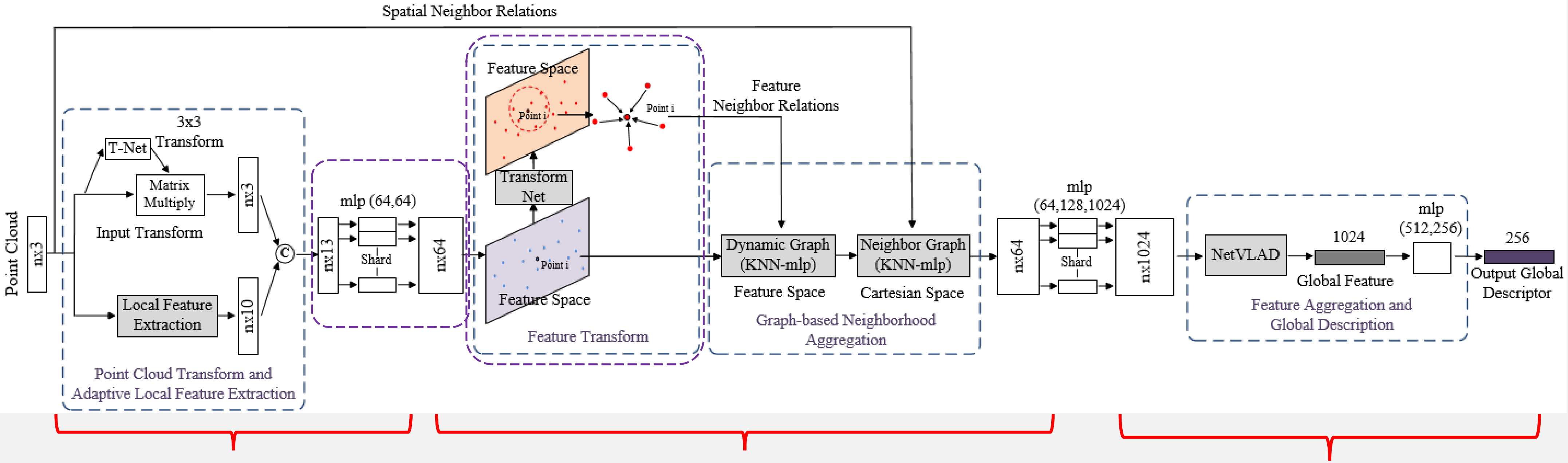
- Select optimal neighbor size (entropy minimization)
- Local features suitable for self-driving
- Extract the local structures (with mlp-based feature network)

Graph Neural Network (GNN)

- Feature transform (invariance under different viewpoint)
- Feature aggregation
  - In Feature Space: achieve multi-scale feature learning
  - In Cartesian Space: learn geometrical distribution information of similar semantic structures
- Dynamic Graph + kNN

LPD-Series

- Objective: discriminative and generalizable global descriptors



Feature Network (FN)

- Select optimal neighbor size (entropy minimization)
- Local features suitable for self-driving
- Extract the local structures (with mlp-based feature network)

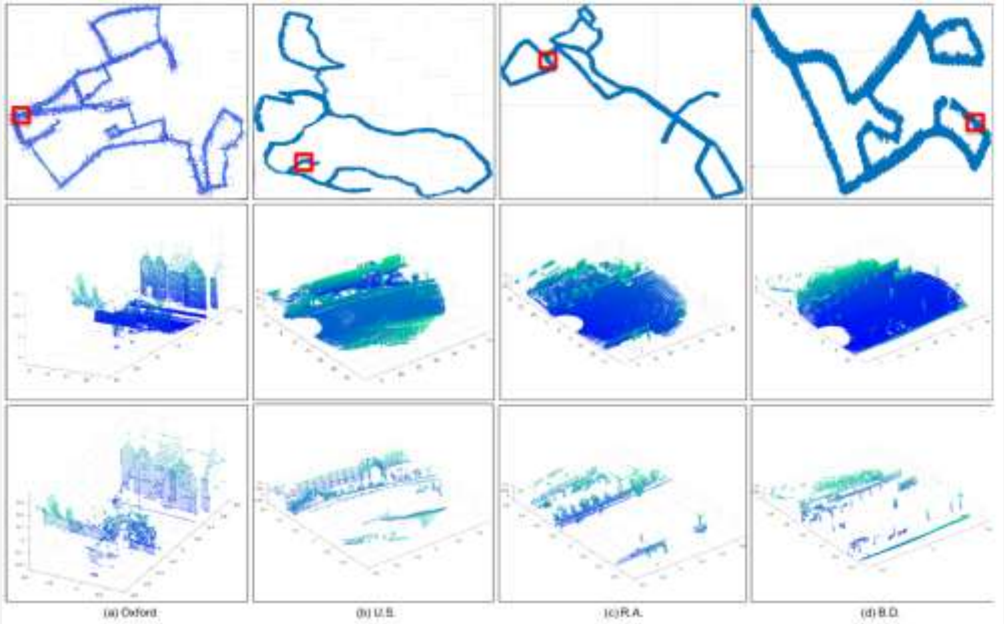
Graph Neural Network (GNN)

- Feature transform (invariance under different viewpoint)
- Feature aggregation
  - In Feature Space: achieve multi-scale feature learning
  - In Cartesian Space: learn geometrical distribution information of similar semantic structures
- Dynamic Graph + kNN

- Aggregate local features
- Generate global descriptor

LPD-Series

Loss and Dataset



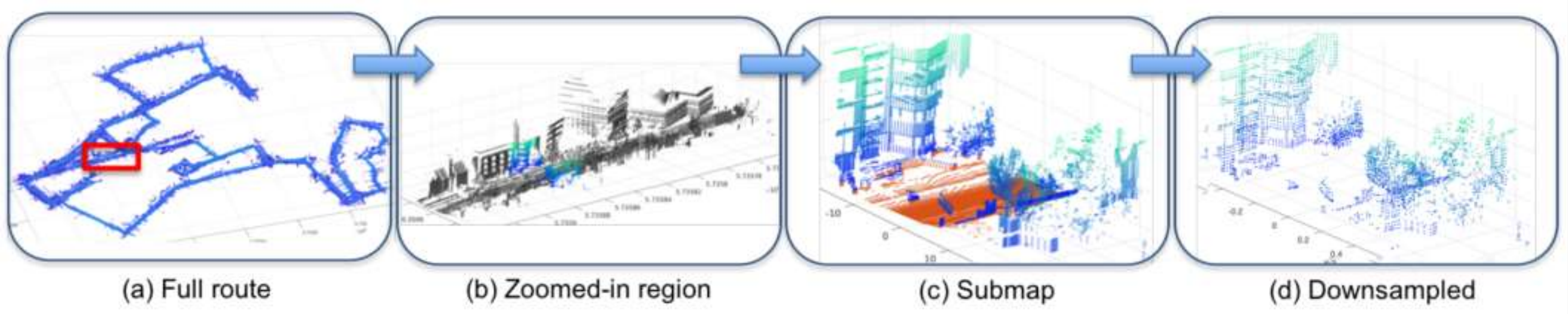
Loss Function:

lazy quadruplet loss:

$$\mathcal{L}_{lazyQuad}(\mathcal{T}, P_{neg^*}) = \max_j([\alpha + \delta_{pos} - \delta_{neg_j}]_+) + \max_k([\beta + \delta_{pos} - \delta_{neg_k^*}]_+)$$

$$\mathcal{T} = (P_a, P_{pos}, \{P_{neg}\}) \qquad \delta_{pos} = d(f(P_a), f(P_{pos}))$$

$$\delta_{neg_j} = d(f(P_a), f(P_{neg_j})) \quad \delta_{neg_k^*} = d(f(P_{neg^*}), f(P_{neg_k}))$$



LPD-Series

Performance

Table 1. Where our work fits into the literature.

	Local features	Feature space aggregation	Cartesian space aggregation	Feature distribution	Large-scale scene
PointNet [3]					
PointNet++ [2]			✓		
PointNetVLAD [5]		✓			✓
DGCNN [6]		✓			
KCNet [4]		✓			
RWTH-Net [1]		✓	✓		✓
Proposed	✓	✓	✓	✓	✓

[1] F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe. Know what your neighbors do: 3d semantic segmentation of point clouds. In *Proceedings of the IEEE European Conference on Computer Vision Workshops*, 2018.

[2] C. R. Qi, L. Li, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Conference on Neural Information Processing Systems*, 2018.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] Y. Shen, C. Feng, Y. Yang, and D. Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[5] M. A. Uy and G. H. Lee. Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. In *arXiv preprint arXiv:1801.07829v1*, 2018.

LPD-Series

Performance

	Ave recall @1%	Ave recall @1
PN STD	46.52	31.87
PN MAX	73.87	54.16
PN-VLAD baseline	81.01	62.76
PN-VLAD refine	80.71	63.33
NN-VLAD (our)	79.21	61.96
FN-VLAD (our)	89.77	75.79
FN-NG-VLAD (our)	90.38	77.74
FN-DG-VLAD (our)	91.44	80.14
FN-PM-VLAD (our)	91.20	78.77
FN-PC-VLAD (our)	92.27	81.41
FN-SF-VLAD (our)	**94.92**	**86.28**

Comparisons on Oxford RoboCar dataset

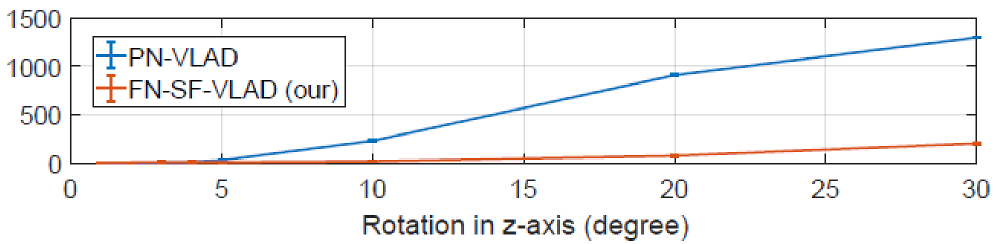
	Our[1]	PointNetVLAD[2]	PointNetVLAD[3]
Oxford	94.92	80.31	80.09
U.S.	96.00	72.63	90.10
R.A.	90.46	60.27	93.07
B.D.	89.14	65.30	86.49

Comparisons on Indoor dataset

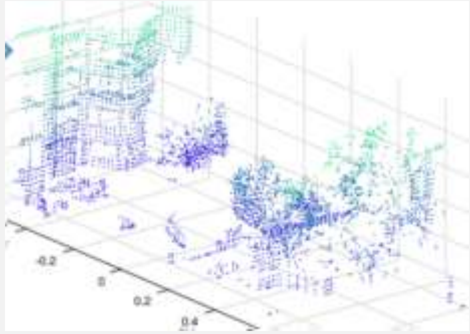
We have a 13% increase in retrieval performance at the cost of an average of 10.5ms added to per frame

	Parameters	FLOPs	Runtime per frame
FN-SF-VLAD (our)	1.981M	749M	23.58ms
FN-PM-VLAD (our)	1.981M	749M	29.23ms
FN-PC-VLAD (our)	1.981M	753M	27.03ms
PN-VLAD	1.978M	411M	13.09ms

Comparisons on computational and memory complexity



The number of place recognition mistakes under different input point cloud rotations (also with 10% white noise)



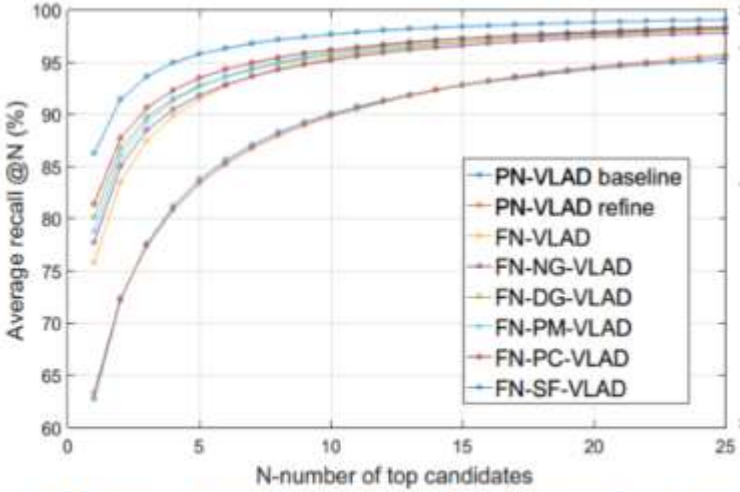
The Oxford RoboCar Dataset

Our method is more robust to point cloud rotations and white noises

LPD-Series

Performance

Performance:



(Oxford Robotcar)	Ave recall @1%	Ave recall @1
PN STD	46.52	31.87
PN MAX	73.87	54.16
PN-VLAD baseline*	81.01	62.76
PN-VLAD refine*	80.71	63.33
NN-VLAD (our)	79.21	61.96
FN-VLAD (our)	89.77	75.79
FN-NG-VLAD (our)	90.38	77.74
FN-DG-VLAD (our)	91.44	80.14
FN-PM-VLAD (our)	91.20	78.77
FN-PC-VLAD (our)	92.27	81.41
FN-SF-VLAD (our)	**94.92**	**86.28**

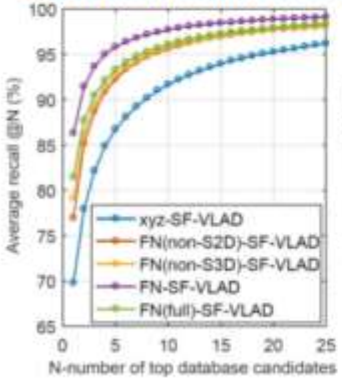
*This result is obtained by using their open-source programs.

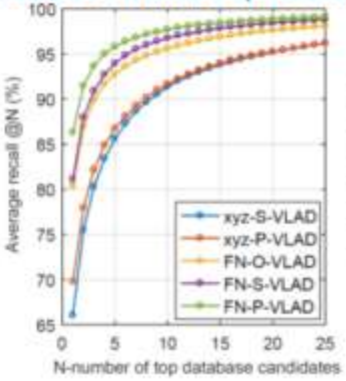
(Indoor Datasets)	U.S.	R.A.	B.D.
PN-VLAD baseline	72.63	60.27	65.30
PN-VLAD refine	90.10	**93.07**	86.49
FN-SF-VLAD (our)	**96.00**	90.46	**89.14**

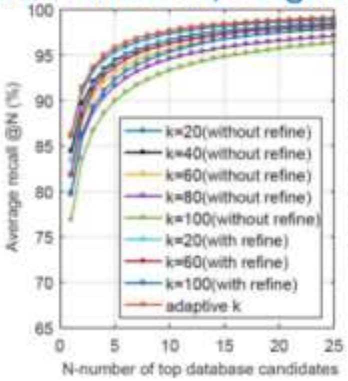
	Parameters	FLOPs	Runtime per frame
PN-VLAD baseline	1.978M	411M	13.09ms
FN-PM-VLAD (our)	1.981M	749M	29.23ms
FN-PC-VLAD (our)	1.981M	753M	27.03ms
FN-SF-VLAD (our)	1.981M	749M	23.58ms

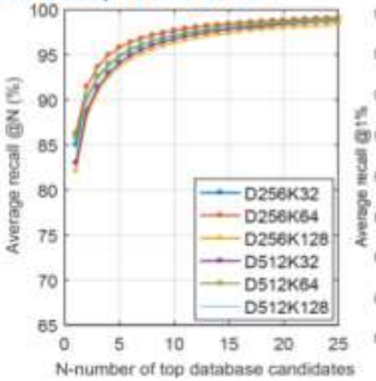
FLOPs: required floting-point operations.

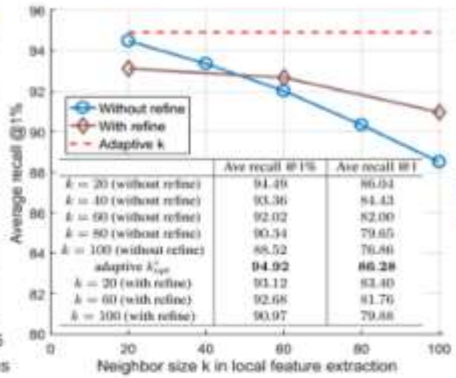
Ablation studies on Local Features/Feature Transform/Neighbor Size/NetVLAD:











Local Feature	Ave recall @1%	Ave recall @1
xyz-SF-VLAD	84.74	69.75
FN(non-F_{2D})-SF-VLAD	90.76	76.94
FN(non-F_{3D})-SF-VLAD	91.23	79.11
FN-SF-VLAD	**94.92**	**86.28**
FN(full)-SF-VLAD	92.03	81.45

Feature Transform	Ave recall @1%	Ave recall @1
xyz-Series-VLAD	83.22	66.01
xyz-Parallel-VLAD	84.74	69.75
FN-Original-VLAD (O)	91.53	80.29
FN-Series-VLAD (S)	92.60	81.09
FN-Parallel-VLAD (P)	**94.92**	**86.28**

LPD-Series

- Performance

Table 1. Comparisons with vision-based methods (Ave recall @1 with different GPS location bounds: 3m/5m/10m/15m).

	Our	HF-Net[2]	NV [9]	NV+SP
dawn	65.1 / 79.7 / 86.5 / 88.4	45.3 / 71.2 / 81.0 / 84.7	50.9 / 80.1 / 85.5 / 88.4	43.7 / 67.7 / 82.2 / 88.6
dusk	64.7 / 79.9 / 87.3 / 89.8	54.1 / 85.8 / 92.6 / 93.9	54.1 / 88.6 / 96.2 / 97.7	45.0 / 63.4 / 86.5 / 92.6
overcast_summer	63.5 / 79.7 / 85.3 / 86.8	55.5 / 78.8 / 83.2 / 84.7	68.9 / 92.2 / 95.2 / 96.8	48.8 / 68.7 / 84.9 / 92.7
overcast_winter	45.6 / 73.8 / 79.2 / 81.0	31.3 / 75.4 / 86.9 / 89.5	29.7 / 81.0 / 94.9 / 96.7	27.2 / 60.0 / 86.7 / 93.8
night-rain[1]	20.1 / 32.8 / 40.6 / 44.6	2.7 / 6.6 / 10.5 / 11.4	5.7 / 14.3 / 19.5 / 22.3	9.3 / 18.6 / 25.0 / 28.4
sun	74.1 / 82.3 / 87.8 / 89.4	54.6 / 68.3 / 75.7 / 81.7	70.0 / 82.4 / 87.6 / 89.3	48.0 / 64.3 / 84.8 / 92.4
night	63.2 / 77.3 / 83.1 / 84.5	2.1 / 3.9 / 7.1 / 7.3	9.4 / 17.1 / 23.7 / 26.9	11.2 / 19.2 / 29.0 / 33.6

[1] In this case, the point cloud is reconstructed using a 2D LiDAR and VO. The inaccuracy of VO causes the point cloud to be distorted, hence resulting in a reduced result. But we can still observe that our method significantly outperforms other approaches.

Laser wins in Rain and Night datasets, vision wins in Sun, Overcast and Dust datasets, they are similar in Dawn dataset

Method	day	night	Last updated
Visual Localization Using Sparse Semantic 3D Map	71.8 / 91.5 / 96.8	40.8 / 63.3 / 80.6	June 2, 2019, 4 a.m.
Hierarchical-Localization NetVLAD+SuperPoint	80.5 / 87.4 / 94.2	42.9 / 62.2 / 76.5	June 1, 2019, 9:50 a.m.
Hierarchical-Localization (multi-camera when available)	80.5 / 87.4 / 94.2	42.9 / 62.2 / 76.5	June 2, 2019, 6:55 a.m.
DenseVLAD & D2-Net (top-20)	80.1 / 88.0 / 93.4	39.8 / 55.1 / 74.5	May 29, 2019, 7:48 a.m.
Asymmetric Hypercolumn Matching	47.8 / 72.2 / 91.3	30.6 / 53.1 / 78.6	June 5, 2019, 7:28 p.m.
R2D2 10k keypoints	0.0 / 0.0 / 0.0	45.9 / 66.3 / 88.8	June 17, 2019, 9:56 a.m.
CityScaleLocalization	52.3 / 80.0 / 94.3	24.5 / 33.7 / 49.0	May 10, 2019, 2:22 p.m.
DELF - new model	0.0 / 0.0 / 0.0	39.8 / 61.2 / 85.7	June 12, 2019, 3:52 p.m.
NetVLAD+SuperPoint top 10 (baseline)	33.9 / 48.8 / 78.5	25.5 / 45.9 / 80.6	June 1, 2019, 8:59 p.m.

Long-term Visual Localization Benchmark on Oxford RoboCar Dataset (ETH)

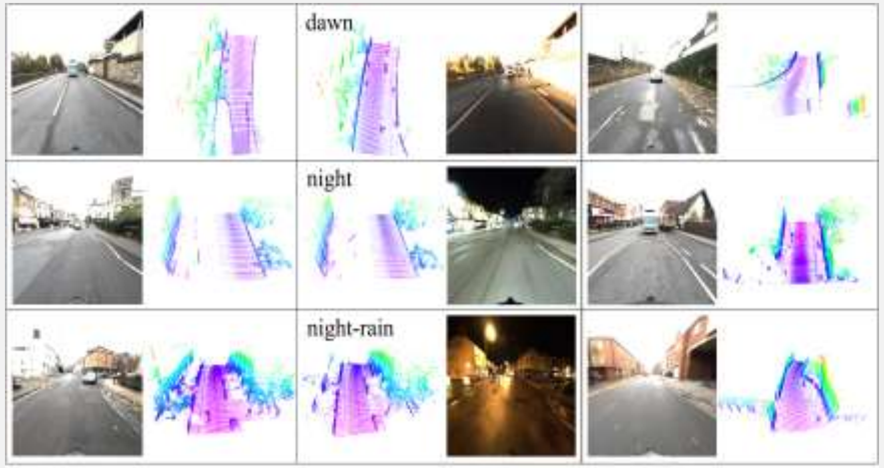
LPD-Series

Performance

Table 1. Comparisons with vision-based methods (Ave recall @1 with different GPS location bounds: 3m/5m/10m/15m).

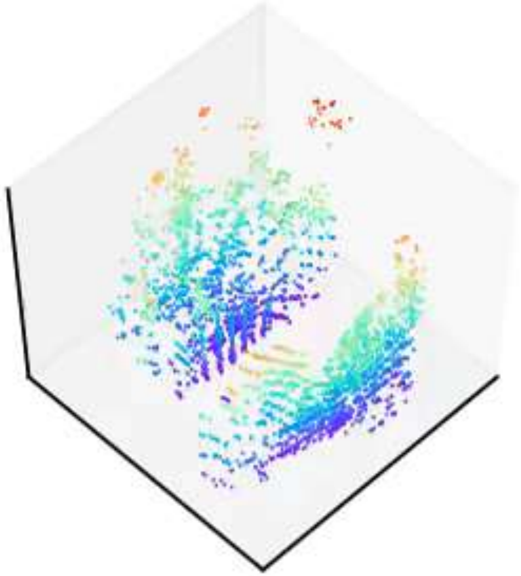
	Our	HF-Net[2]	NV [9]	NV+SP
dawn	65.1/79.7/86.5/88.4	45.3/71.2/81.0/84.7	50.9/80.1/85.5/88.4	43.7/67.7/82.2/88.6
dusk	64.7/79.9/87.3/89.8	54.1/85.8/92.6/93.9	54.1/88.6/96.2/97.7	45.0/63.4/86.5/92.6
overcast_summer	63.5/79.7/85.3/86.8	55.5/78.8/83.2/84.7	68.9/92.2/95.2/96.8	48.8/68.7/84.9/92.7
overcast_winter	45.6/73.8/79.2/81.0	31.3/75.4/86.9/89.5	29.7/81.0/94.9/96.7	27.2/60.0/86.7/93.8
night-rain[1]	20.1/32.8/40.6/44.6	2.7/6.6/10.5/11.4	5.7/14.3/19.5/22.3	9.3/18.6/25.0/28.4
sun	74.1/82.3/87.8/89.4	54.6/68.3/75.7/81.7	70.0/82.4/87.6/89.3	48.0/64.3/84.8/92.4
night	63.2/77.3/83.1/84.5	2.1/3.9/7.1/7.3	9.4/17.1/23.7/26.9	11.2/19.2/29.0/33.6

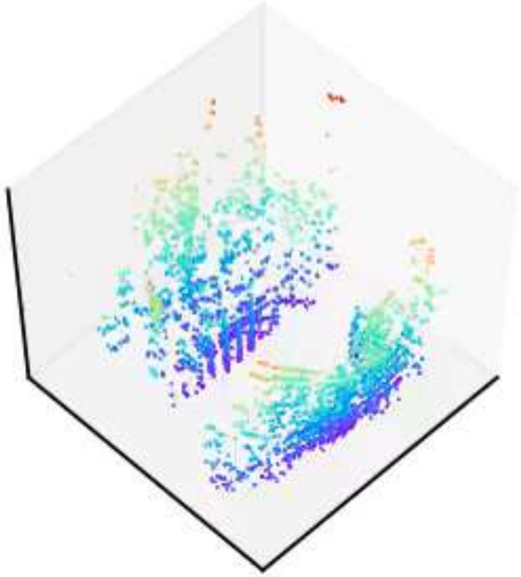
[1] In this case, the point cloud is reconstructed using a 2D LiDAR and VO. The inaccuracy of VO causes the point cloud to be distorted, hence resulting in a reduced result. But we can still observe that our method significantly outperforms other approaches.



(a) Retrieval results by LPD-Net (b) Query (c) Retrieval results by NV+SP

Experiment:
place recognition and uniqueness evaluation in KITTI dataset



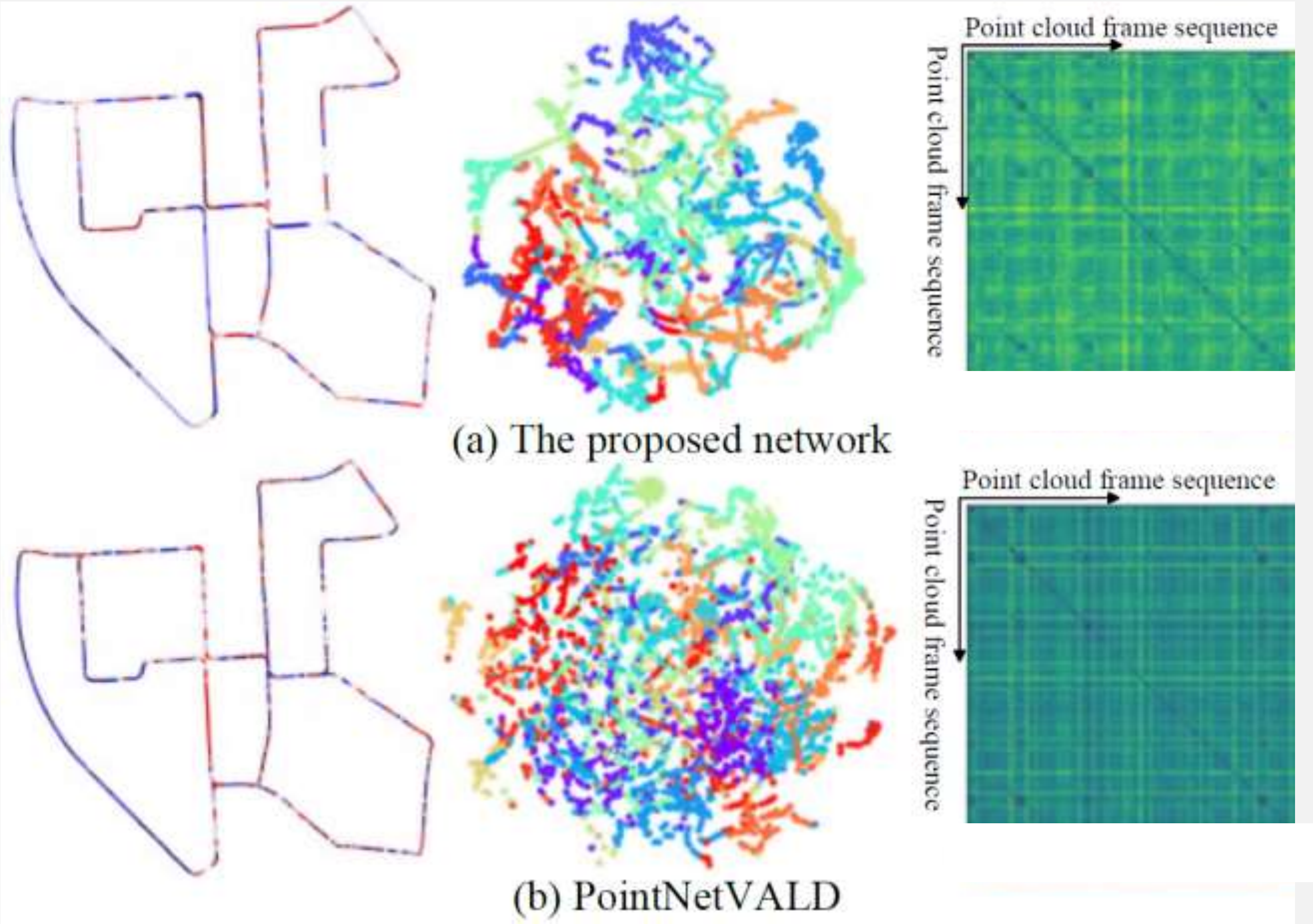


LPD-Series

- Global descriptor clustering

t-SNE visualization of the clustering results

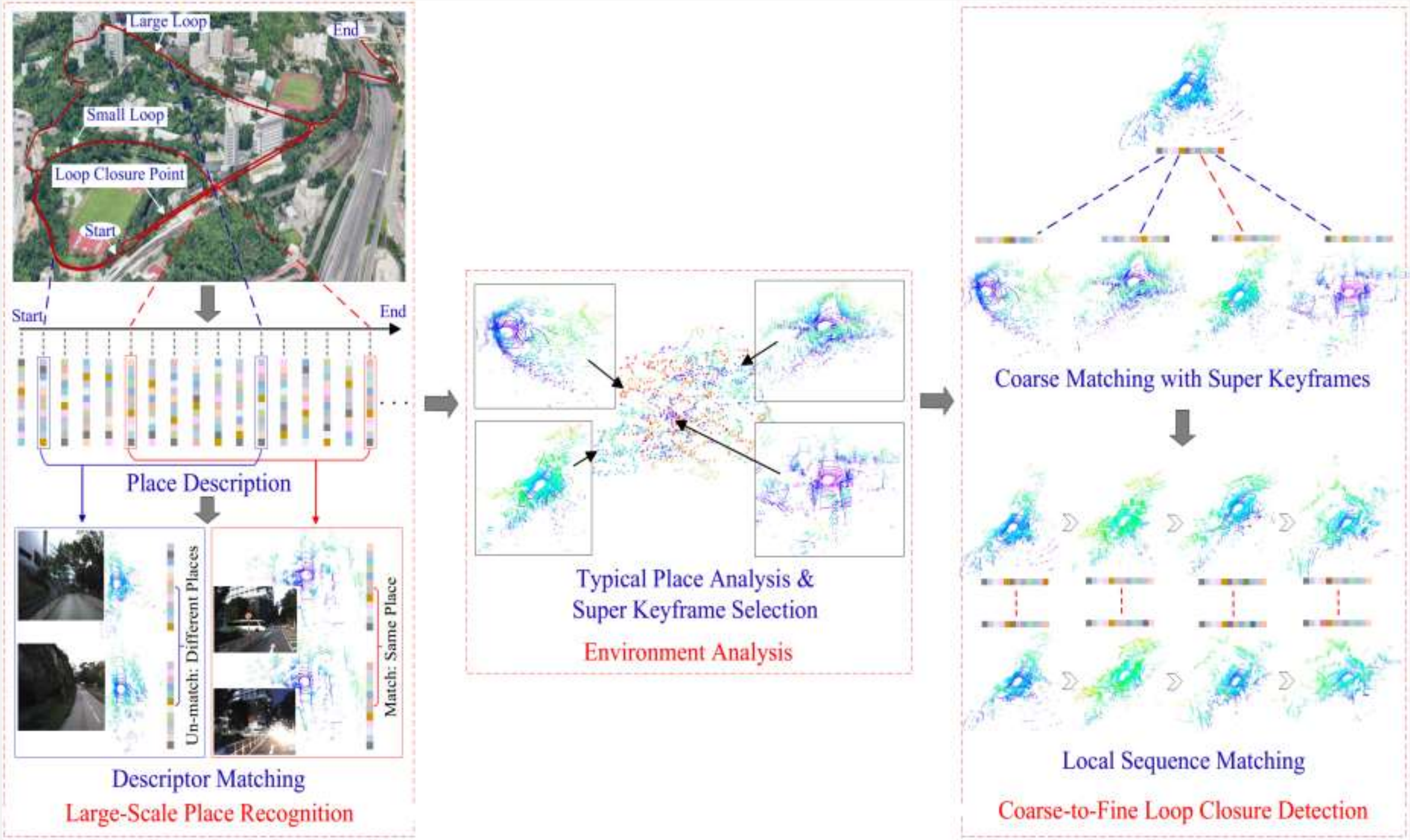
L2 distance between each place pair



Place clustering results compared with PointNetVLAD

LPD-Series

- Loop-closure detection

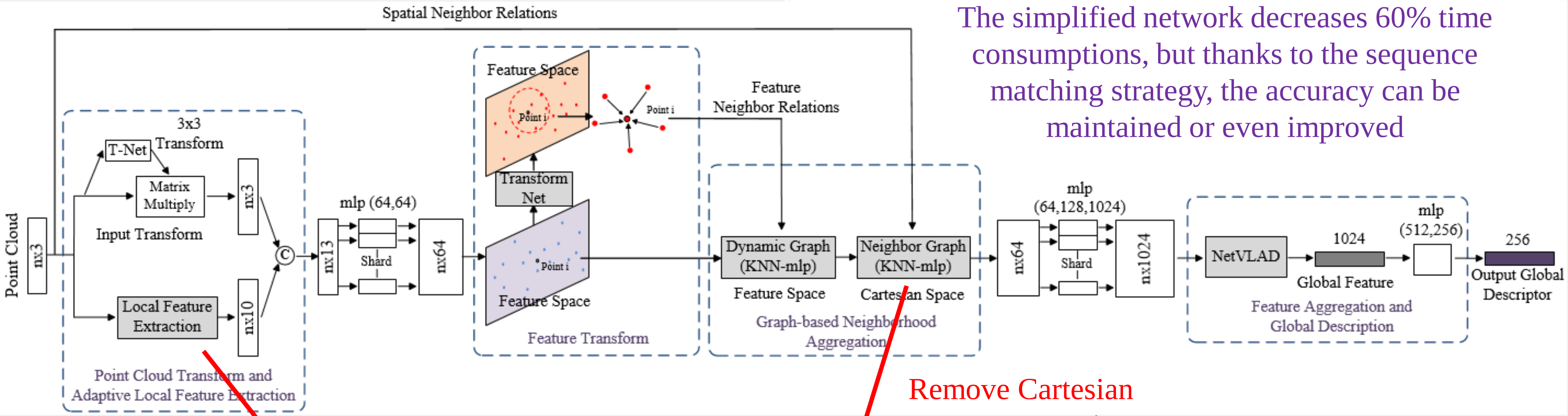


LPD-Series

☐ SeqLPD: Implementation

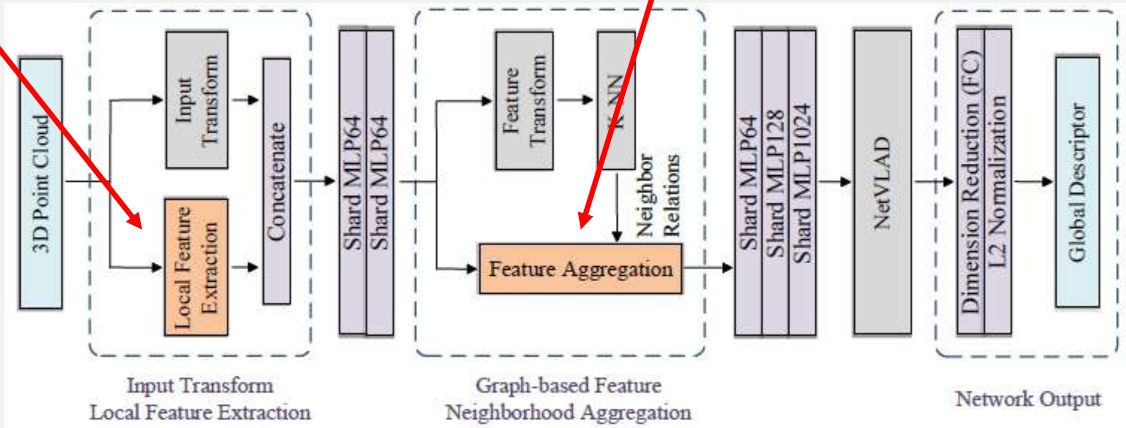
	Ave. Recall @1%	Ave. Recall @1
PN-VLAD baseline [7]	81.01	62.76
PN-VLAD refine [7]	80.71	63.33
LPD-Net [6]	94.92	86.28
LPD-Light (our)	89.55	77.92
LPD-Light + SequenceMatching (our)	95.81	87.15

The simplified network decreases 60% time consumptions, but thanks to the sequence matching strategy, the accuracy can be maintained or even improved



Remove Cartesian space aggregation

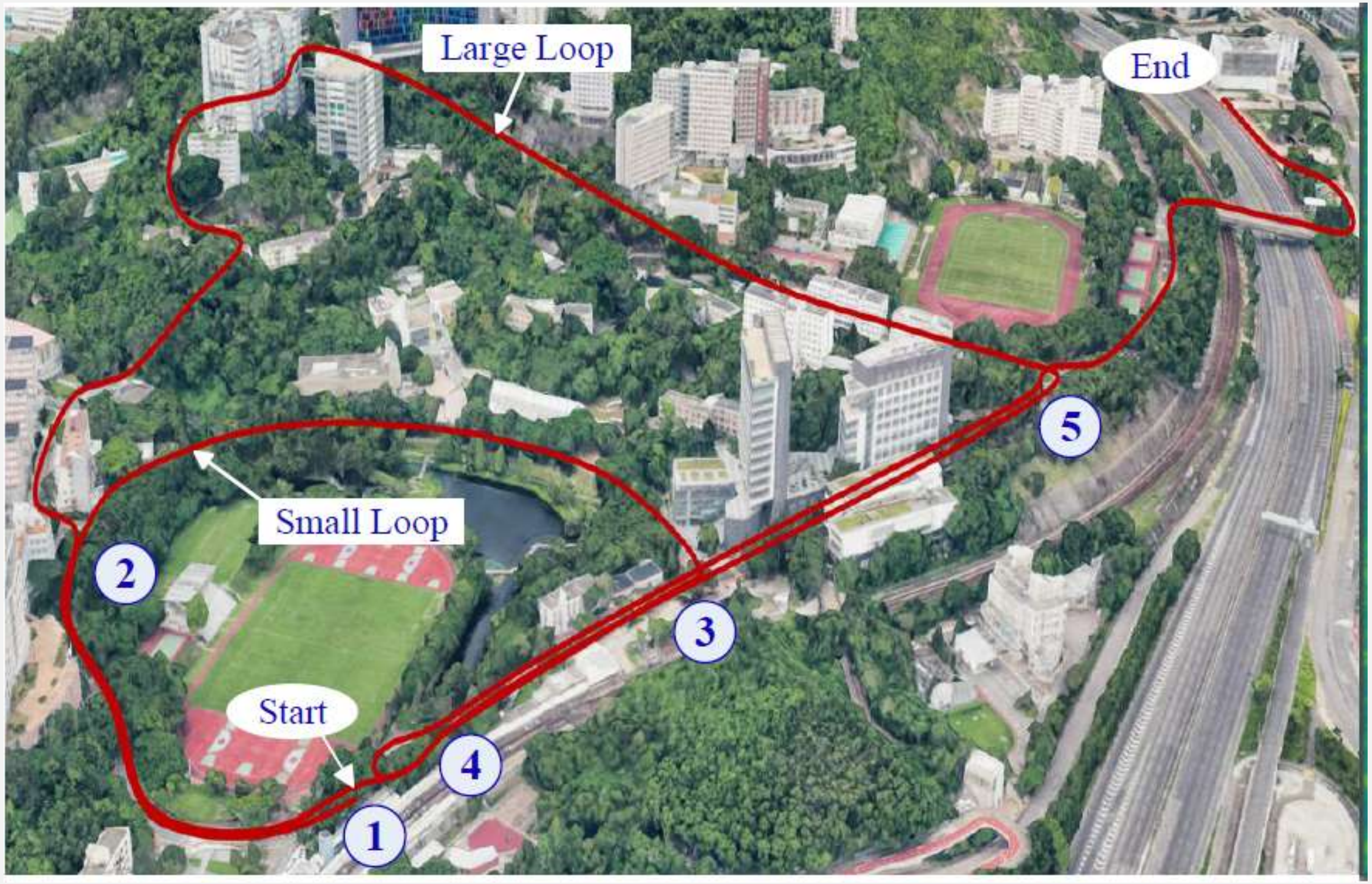
Remove 3D features



Network Simplification

LPD-Series

Loop-closure detection result

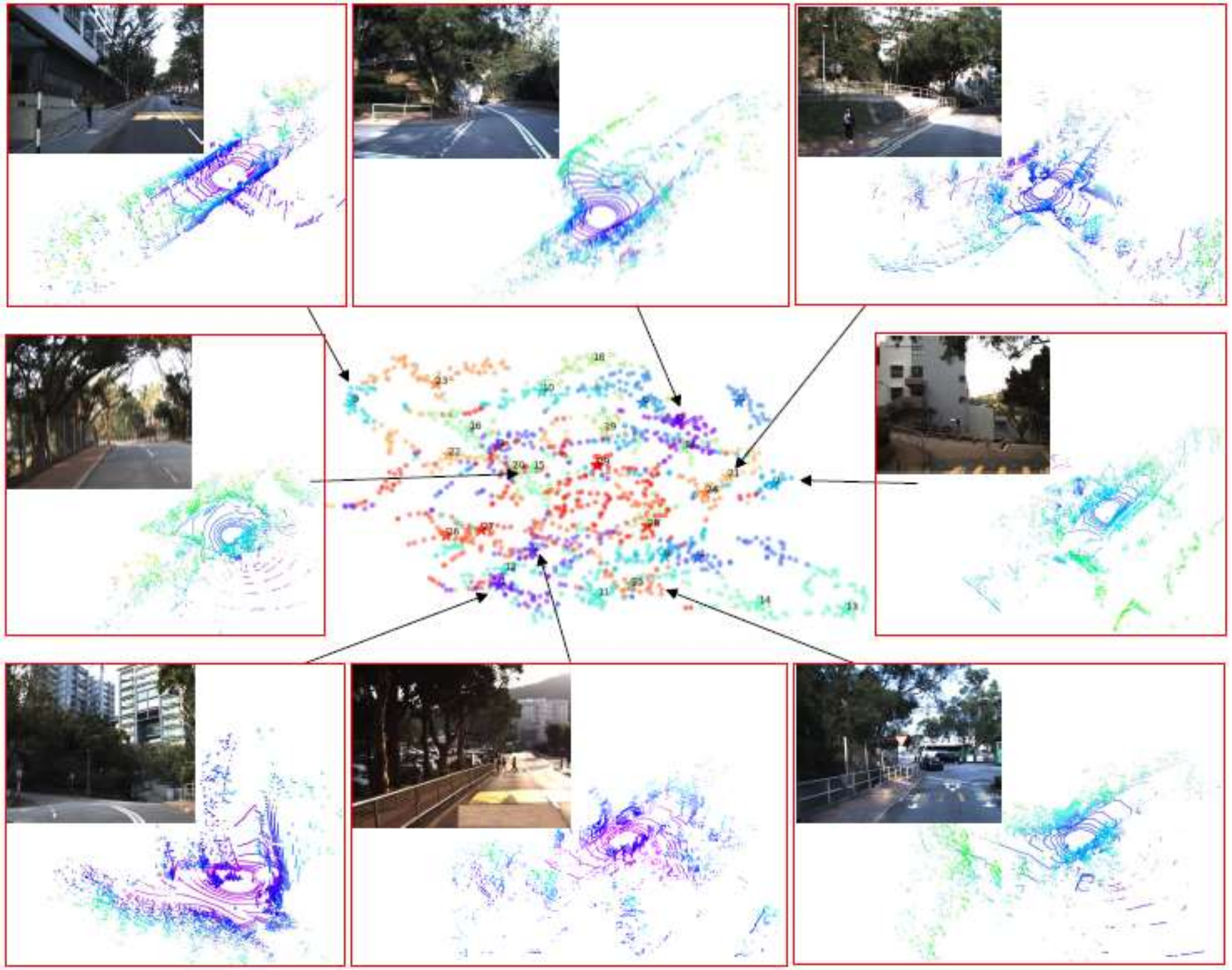


Experiments in CUHK campus:

- Outdoor route for two loops (large loop and small loop)
- Length: 4km

LPD-Series

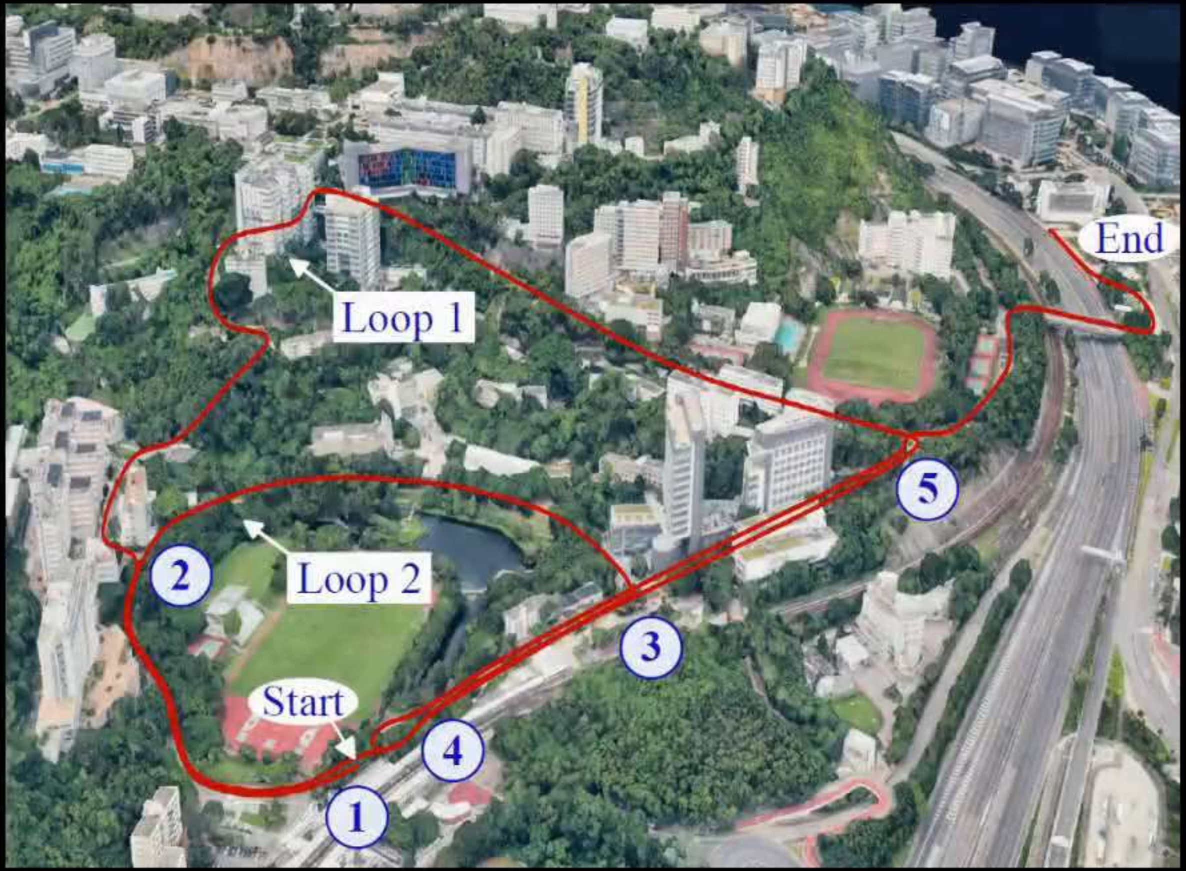
- Loop-closure detection result



Descriptor clustering and typical place selection results

Part 3:

Sequence Matching and Loop-Closure Detction

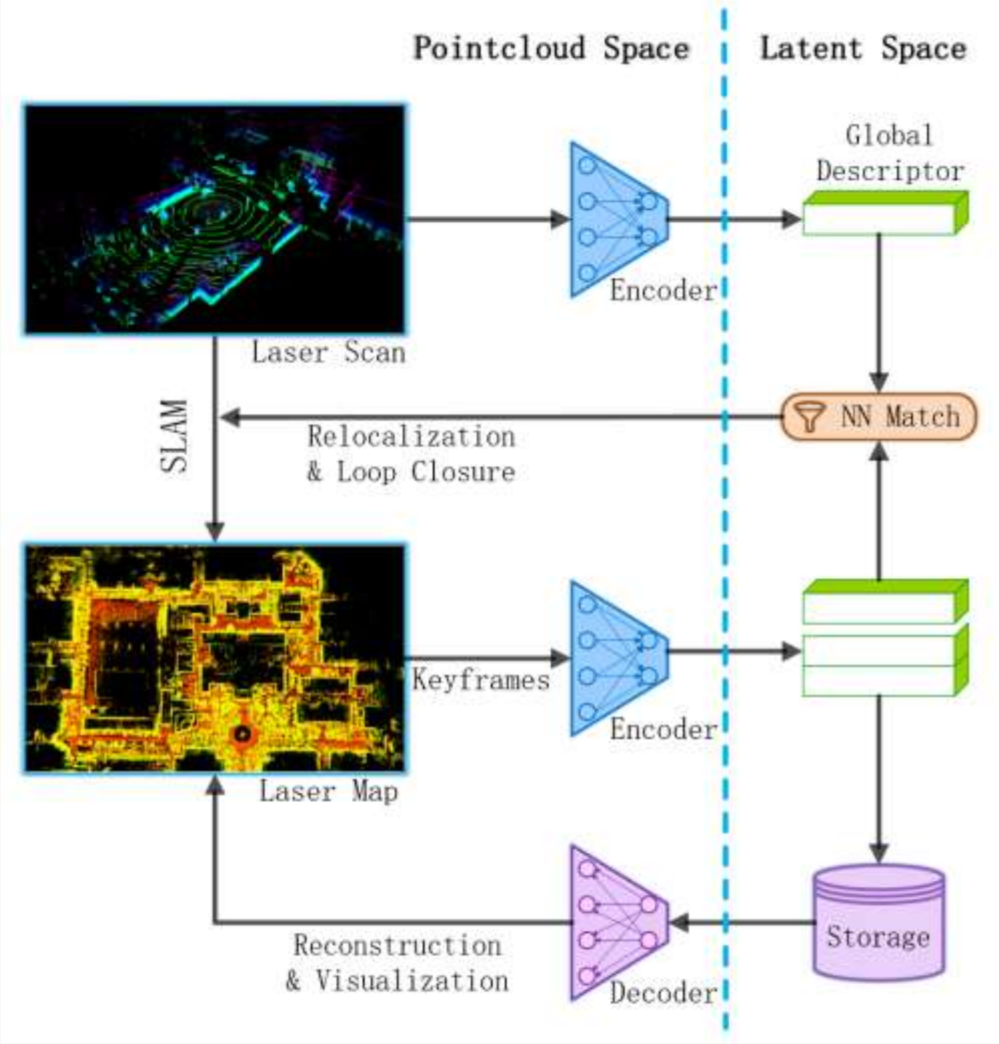


Environment:

Large-scale University Campus with Slope Terrain

LPD-Series

- One More Thing



LPD-AutoEncoder

LPD-Pose

Online course introduction

从这个课中，你将学到什么内容?

帮助学习者建立系统化的点云知识体系，了解三维点云算法在自动驾驶、机器人行业的实际应用案例

系统学习点云处理基础知识，掌握点云表征、滤波、聚类、分割、识别等几大核心问题；学习认知点云、曲面、体素等不同表征方法，点云描述特征，点云采样方法和外点去除

熟练掌握点云空间索引结构和搜索方法，包括KD-Tree，OC-Tree等，并进行工程实践训练

真实公开数据集，分析应用不同点云聚类方法及优缺点，包括K均值、Mean shift、DBSCAN、EM Clustering using GMM、HAC、Spectral clustering和Graph clustering

掌握基于点云特征的激光SLAM框架和原理，包括前端配准（ICP、PL-ICP、NDT等）、后端优化（高斯牛顿方法、LM方法）、回环检测等

学习基于特征工程的激光点云识别与跟踪算法，介绍多传感器融合原理和方法；结合理论深入探究自动驾驶点云应用实践，包括激光雷达采集、校准、跟踪、识别和建图等核心任务

全面了解深度学习在点云研究的热点问题，学习利用深度学习解决三维点云处理问题，包括点云分类、分割、注册配准、重识别、重定位、物体识别等方向

深入学习经典点云深度学习模型，利用工程实践复现经典模型深化理解，包括PointNet、PointNet++、DGCNN、PointCNN、PointPillars、PointRCNN、3D Point-Capsule Net、PointNetVLAD、PointNetLK、Deep Closest Point等

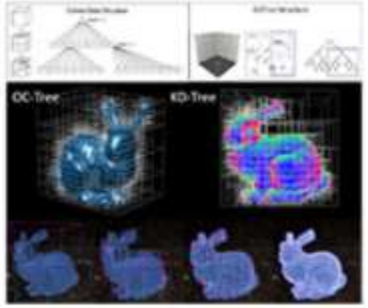
Online course introduction

你将练习的实践项目

练习项目1

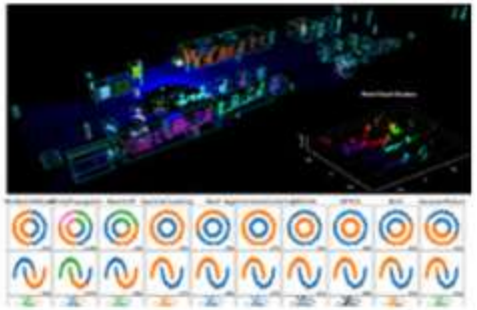
点云基础操作与PCL库编程实践

练习项目2



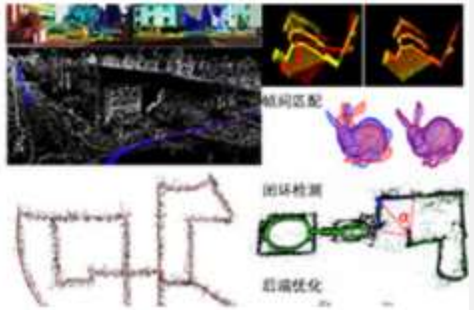
点云空间索引与搜索方法编程实践

练习项目3



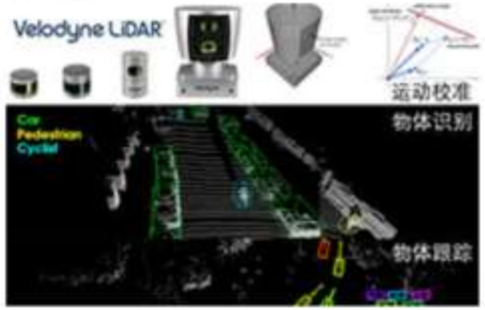
点云经典聚类方法实现及真实数据集应用实践

练习项目4



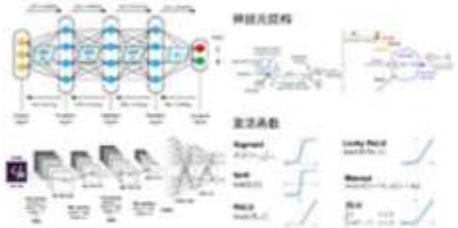
激光SLAM帧间匹配与后端优化计算实践

练习项目5



自动驾驶三维点云预处理、识别、跟踪算法实践

练习项目6

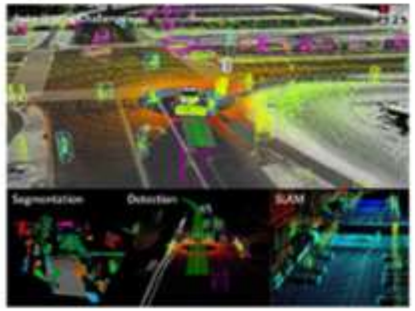


深度学习分类、回归基础知识计算与实践

练习项目7

经典点云深度学习模型复现、训练与推理实践

练习项目8



开放型真实自动驾驶深度学习应用任务实践

Thanks

Professor Liu, Yun-hui(刘云辉)
Fellow IEEE
Director, CUHK T Stone Robotics Institute,
Choh-Ming Li Professor,
Mechanical and Automation

東南大學 SOUTHEAST UNIVERSITY FutureX Lab

CUHK T Stone Robotics Institute
香港中文大學天石機器人研究所

CURI