Outdoor Visual Place Recognition Based on 3D Point Cloud

Supervisor: Prof. TEO Chee Leong **Examiner:** Prof. TAY Eng Hock

YUAN Chengran

Contents

I. Introduction

Visual Place Recognition (VPR)

What is VPR?

- For an agent (a robot or a vehicle), the ability to recognize the same place despite significant changes in appearance and viewpoints.

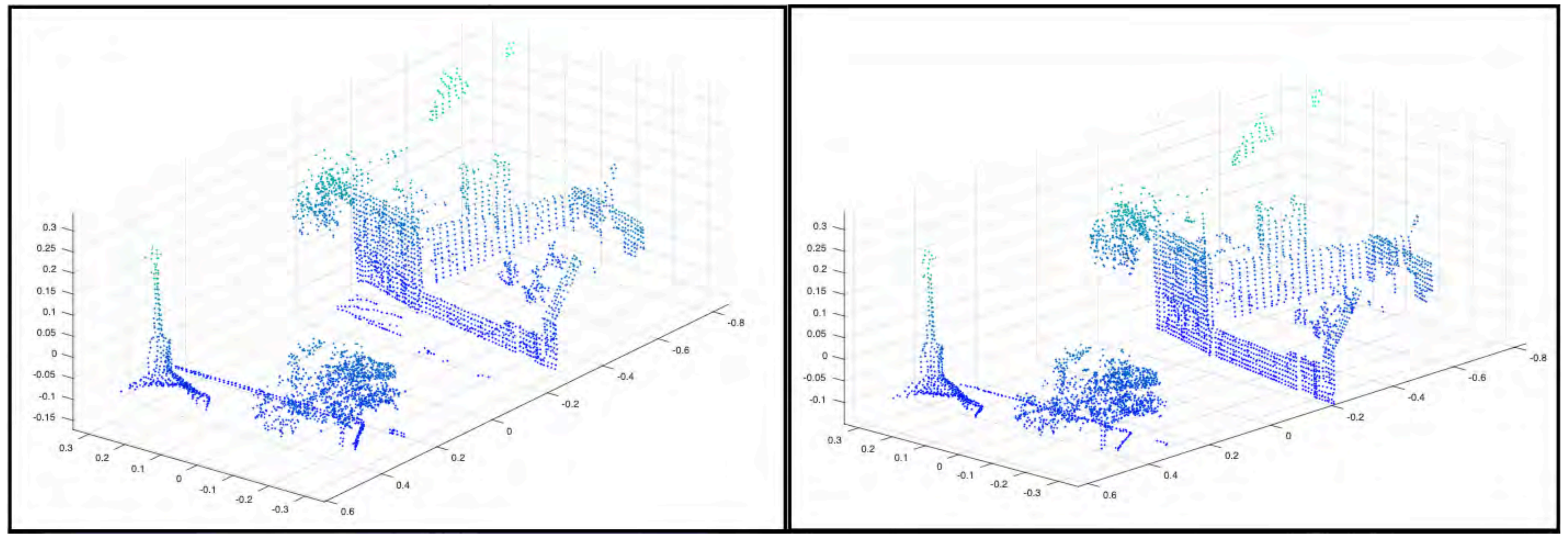
Two pictures taken at the same place

(different seasons, weather and illumination conditions)

2D & 3D Methods for VPR

- Based on the data input, there are mainly two approaches to solving VPR,
  - 2D method using images as input
  - 3D method using 3D data (usually point cloud) as input.

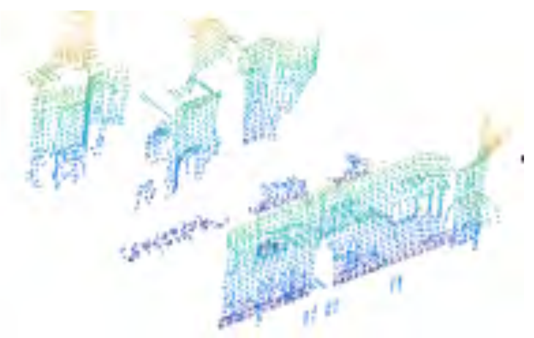
Image input



Point Cloud input

II. Clarification

General framework for 3D VPR



Point cloud

Data Preprocessing

Algorithom for VPR

Output: the place most structurally similar to the present sence

Clarification

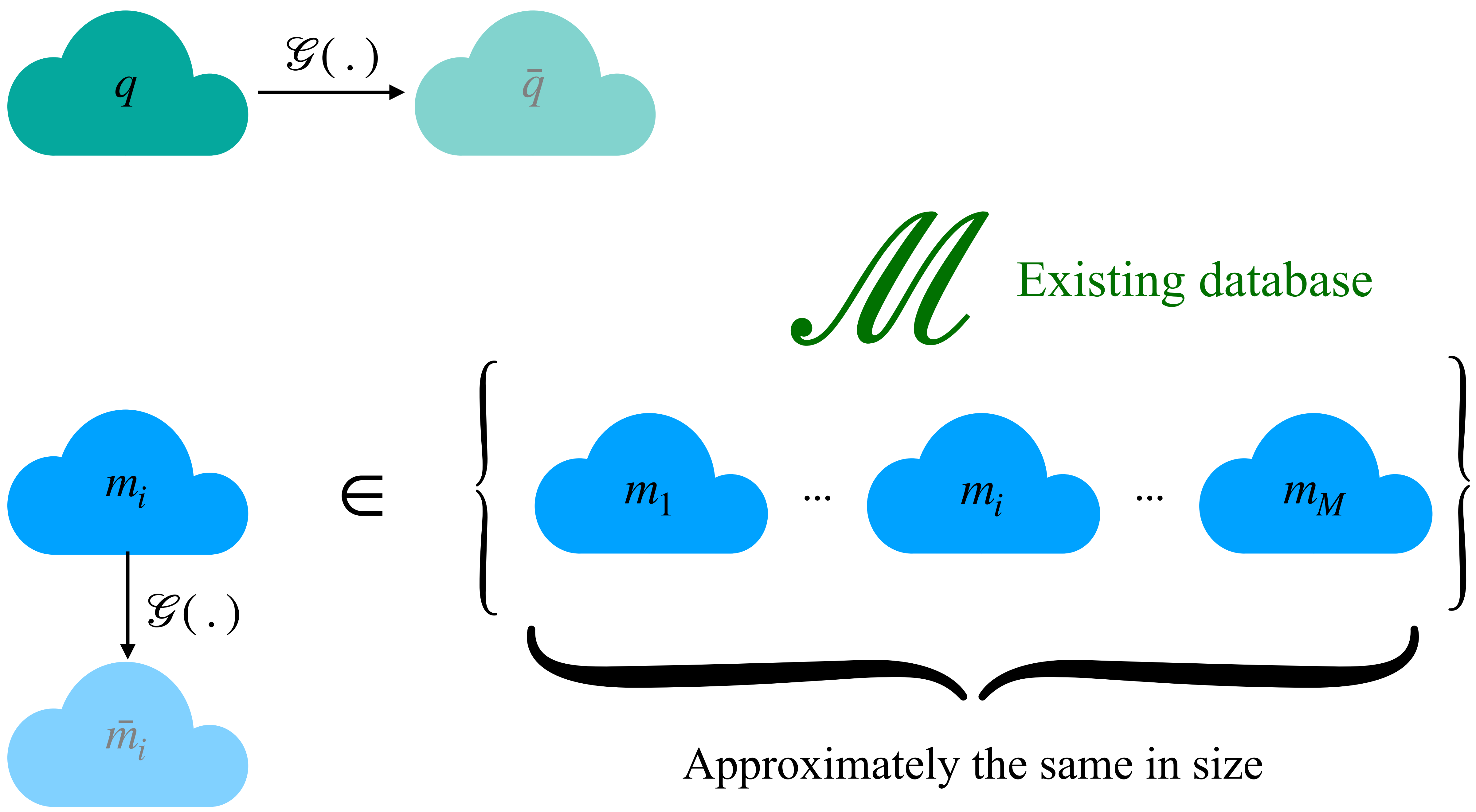
The problem defined as follows:

Given a query 3D point cloud denoted as q, where

$$AOC(q) \approx AOC(m_i)$$
and
$$|\mathcal{G}(q)| = |\mathcal{G}(m_i)|,$$

AOC: area of coverage $\mathcal{G}(\,.\,)$: downsampling filter

The goal is to retrieve the submap m_* from the database $\mathcal{M}$ that is structurally most similar to q.



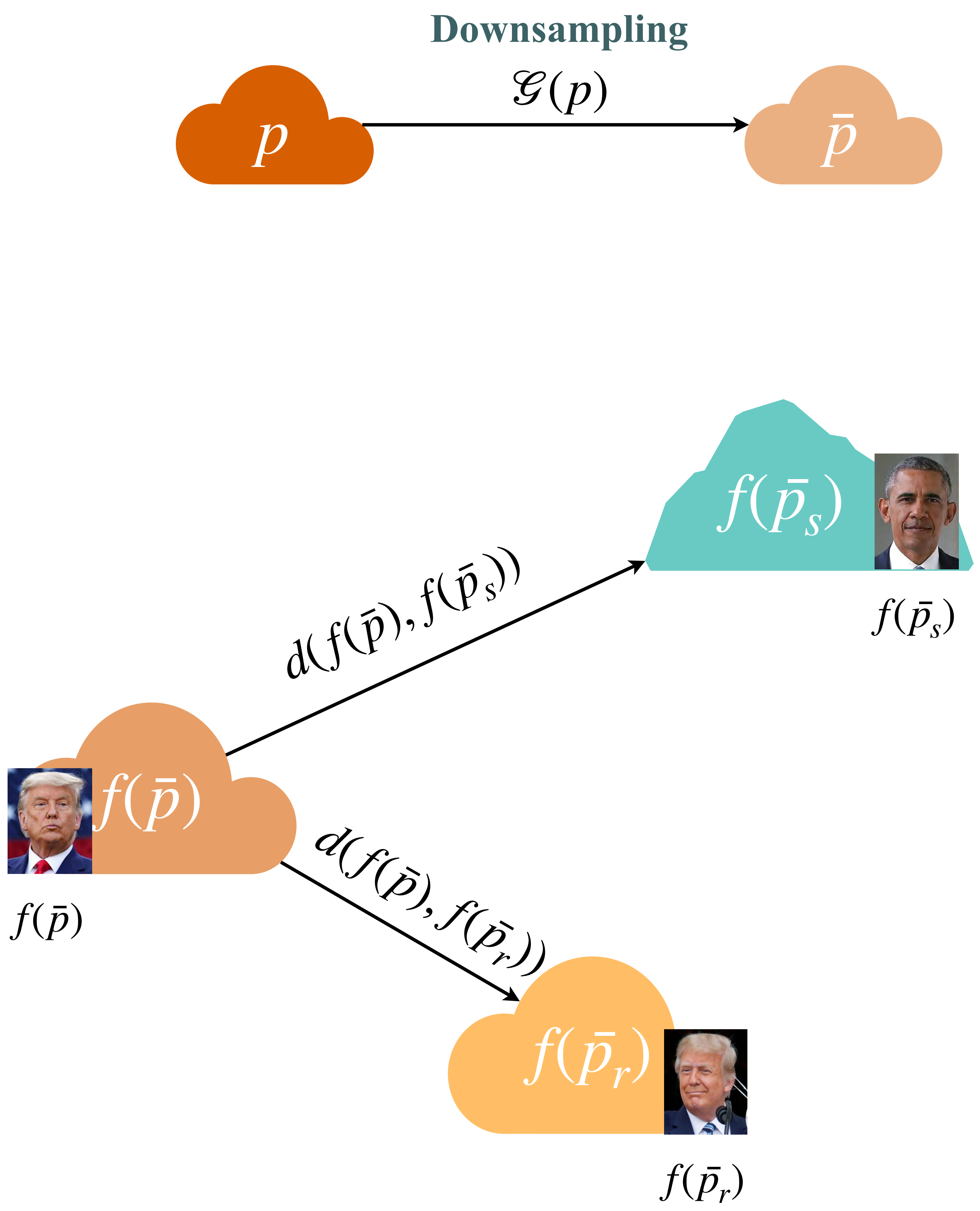
Function Definition

Towards this goal, a deep network is devised to learn a function $f(\,.\,)$ that maps a given downsampled 3D point cloud $\bar{p} = \mathcal{G}(p)$ to a fixed size **global descriptor vector** $f(\bar{p})$ such that

$$d(f(\bar{p}), f(\bar{p}_r)) < d(f(\bar{p}), f(\bar{p}_s)),$$

if p is structurally similar to p_r but dissimilar to p_s.

$d(\,.\,)$ is some distance function, e.g. Euclidean distance function.



Simplification

Our problem then simplifies to finding the submap

$$m_* \in \mathcal{M}$$

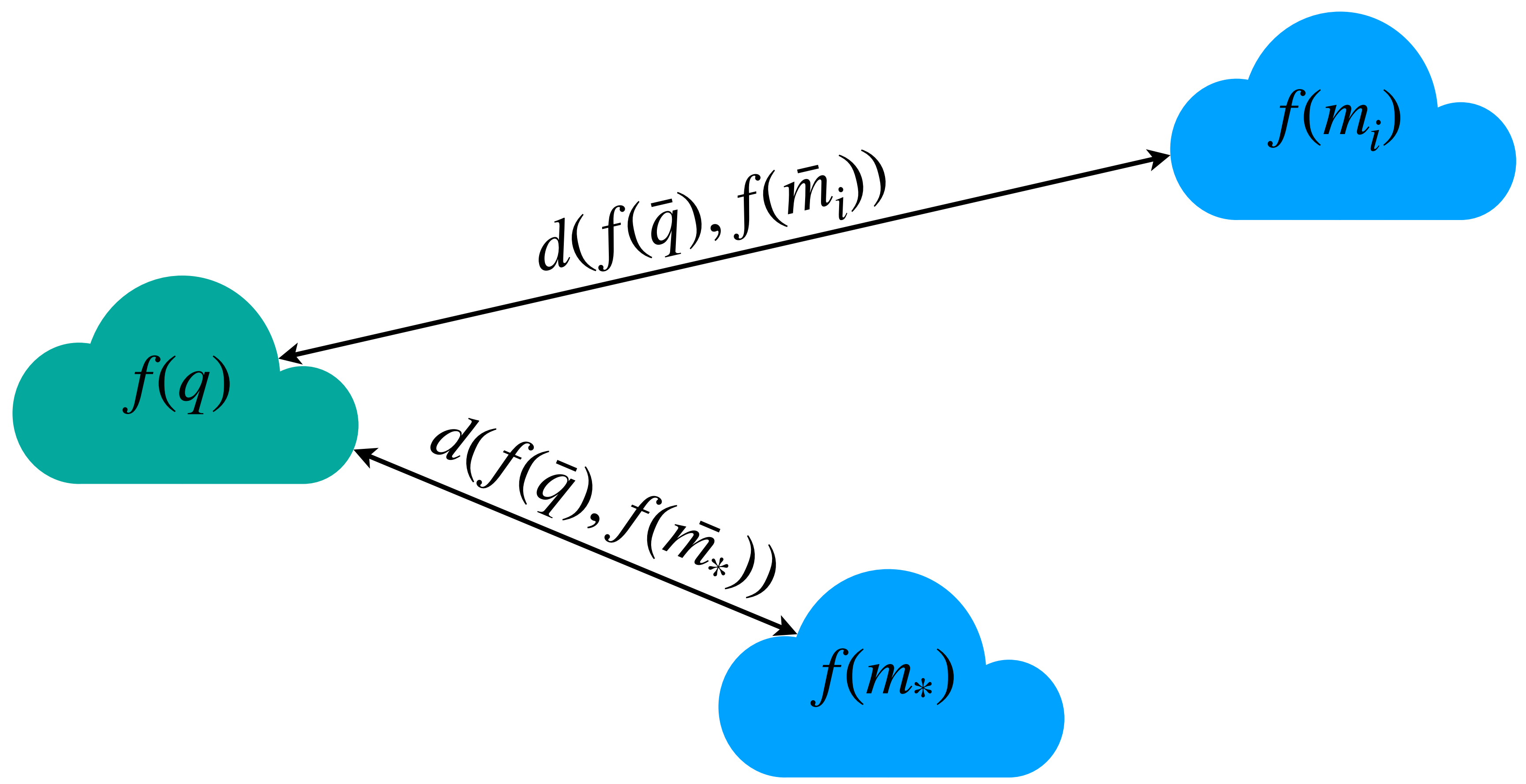
such that its global descriptor vector $f(\bar{m}_*)$ gives the minimum distance with the global descriptor vector $f(\bar{q})$ from the query q, i.e.

$$d(f(\bar{q}), f(\bar{m}_*)) < d(f(\bar{q}), f(\bar{m}_i)), \forall i \neq * .$$

In practice, this can be done by the nearest neighbor search through a list of global descriptors

$$\{ f(\bar{m}_i) \,|\, i \in 1,2,..,M \}$$

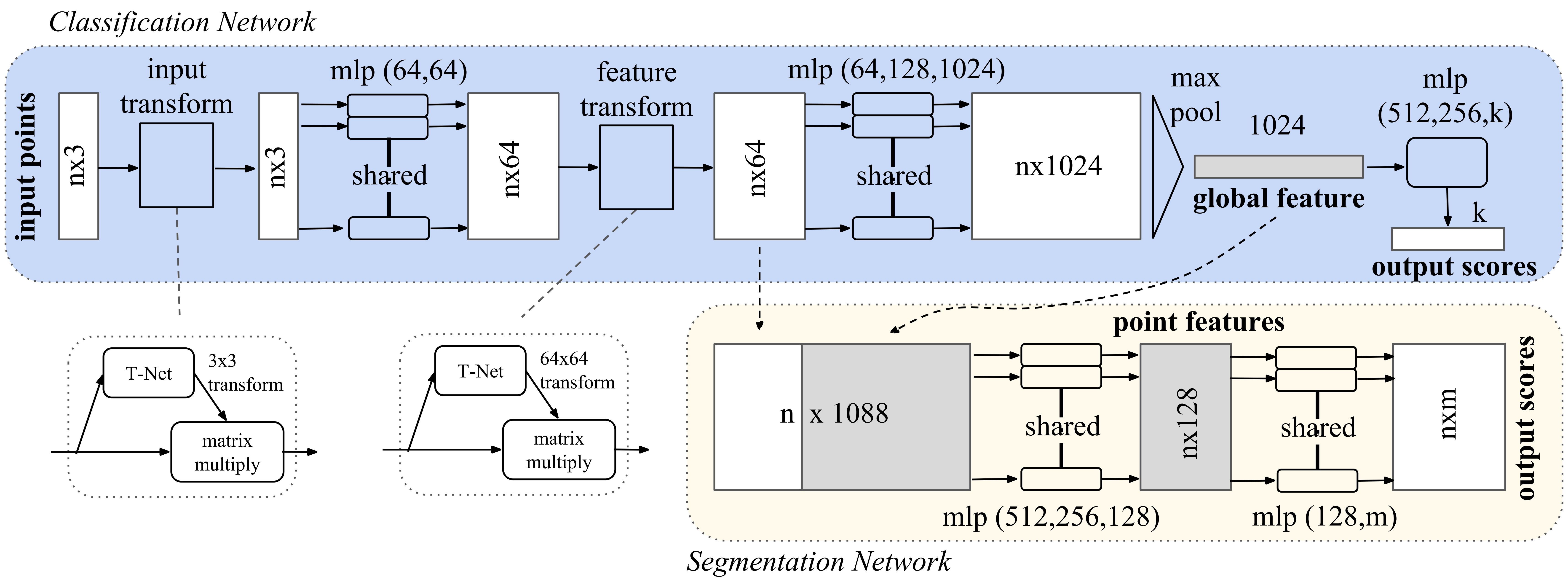
that can be computed once **offline** and stored in memory, while $f(\bar{q})$ is computed **online**.



III. Literature review

PointNet

Pioneer in Point Cloud Processing

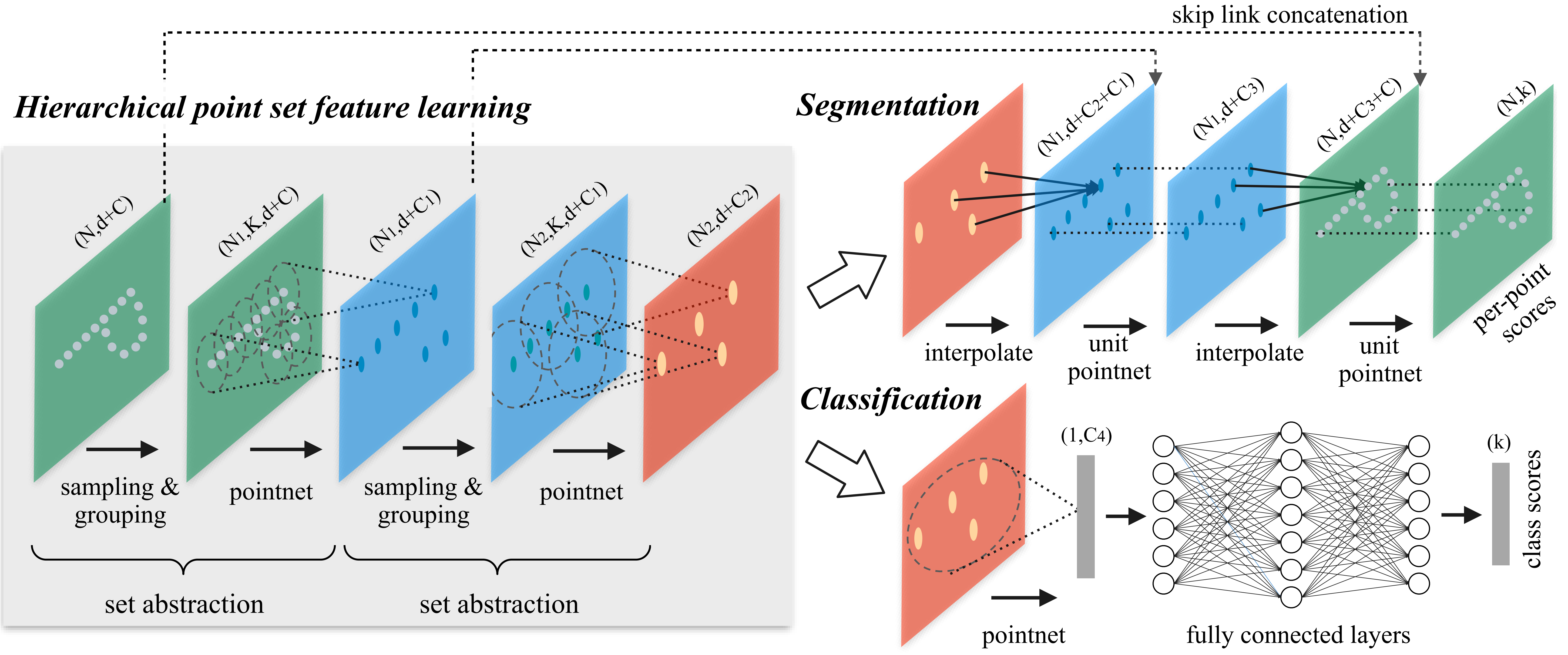


PointNet Architecture.

Qi, C. et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017): 77-85.

PointNet++

Hierarchical architecture

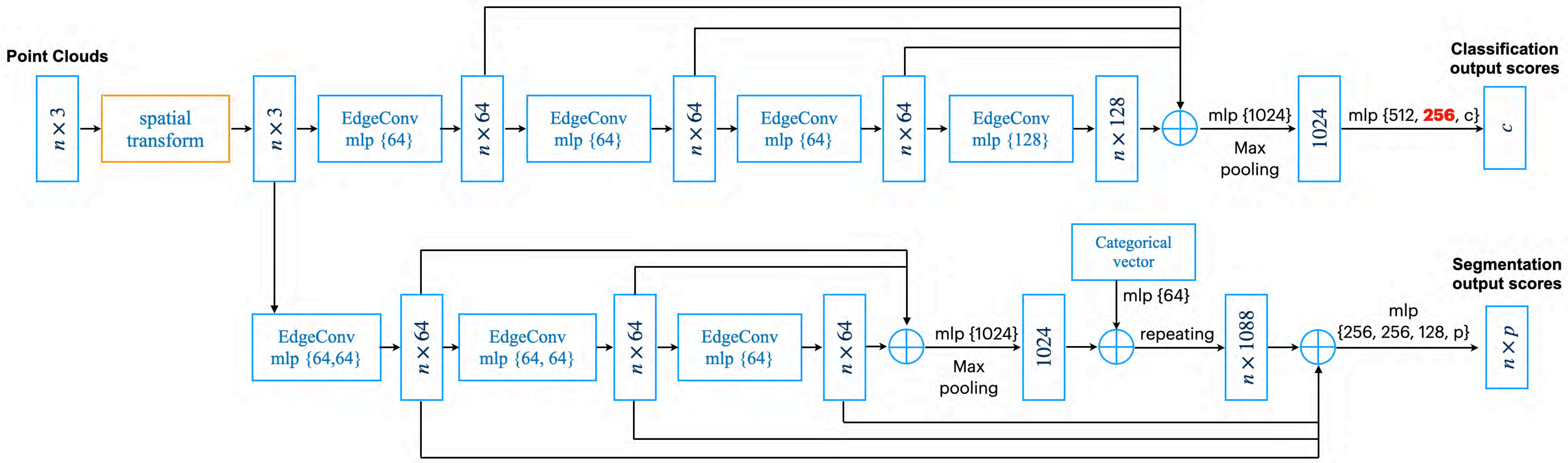


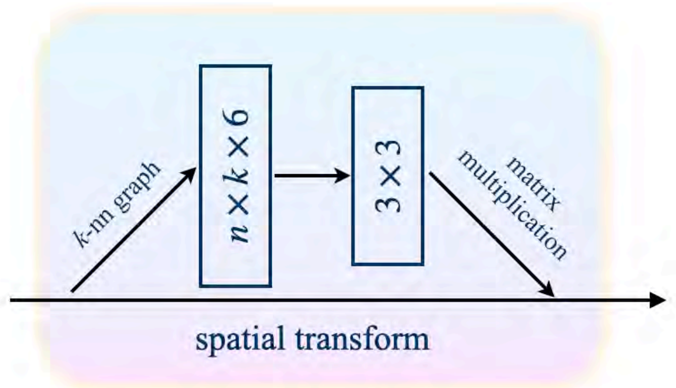
PointNet++: Hierarchical feature learning architecture and its application.

Qi, C. et al. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space." *NIPS* (2017).

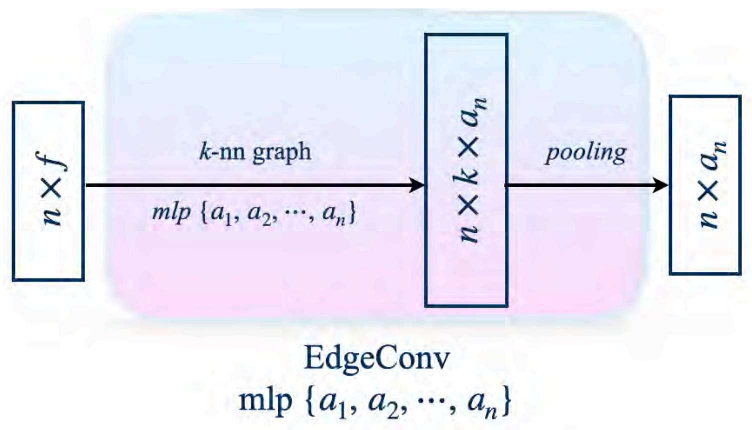
DGCNN

Better extract local geometric features

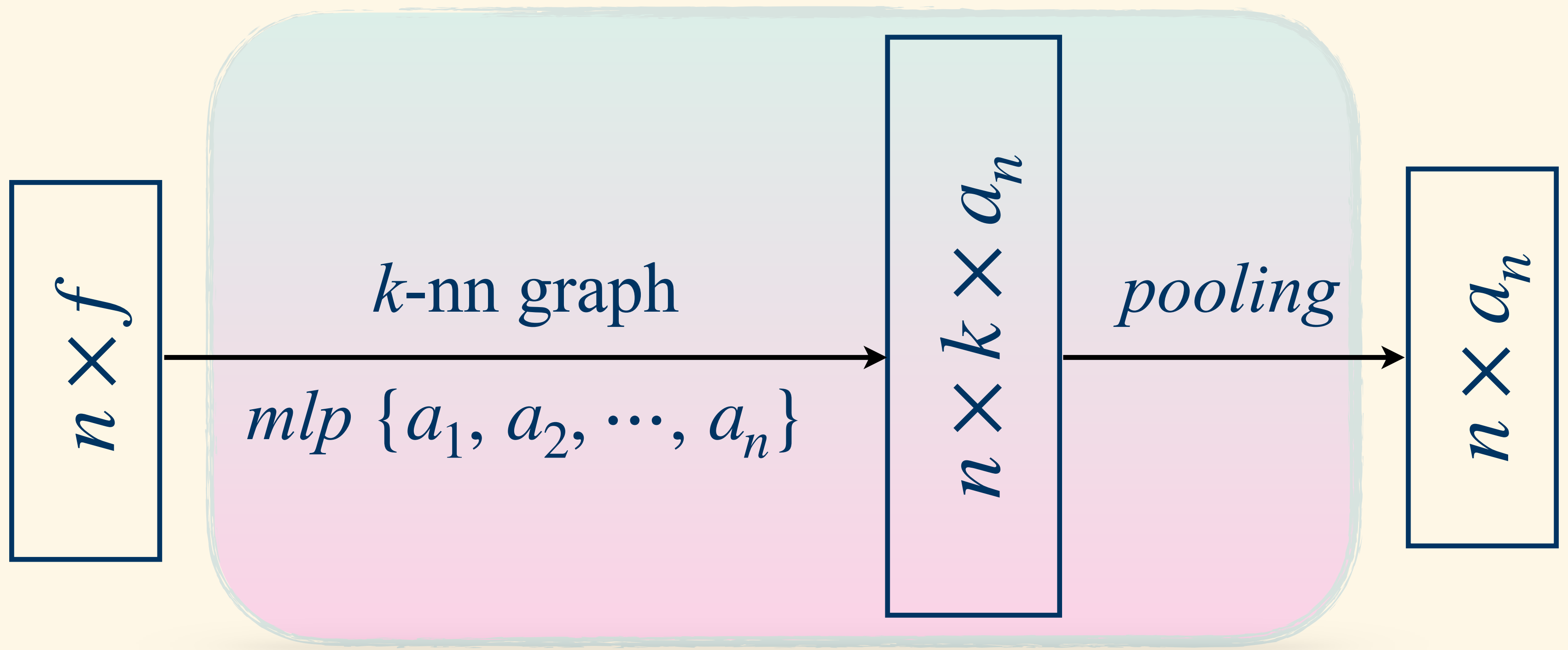




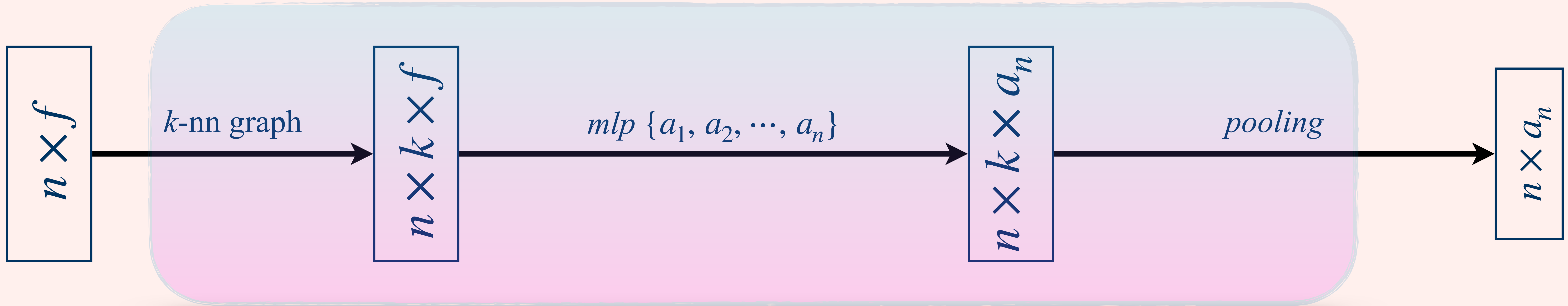
Model architectures of Dynamic Graph CNN.

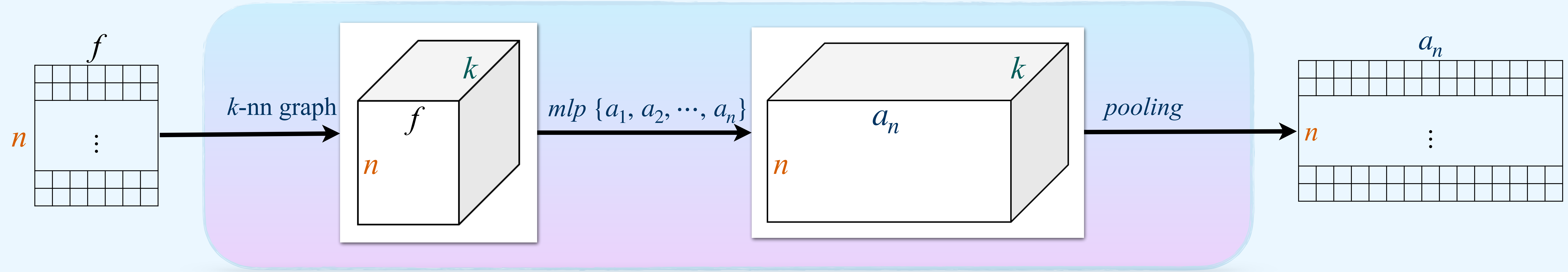


Wang, Yue et al. "Dynamic Graph CNN for Learning on Point Clouds." *ACM Transactions on Graphics (TOG)* 38 (2019): 1 - 12.



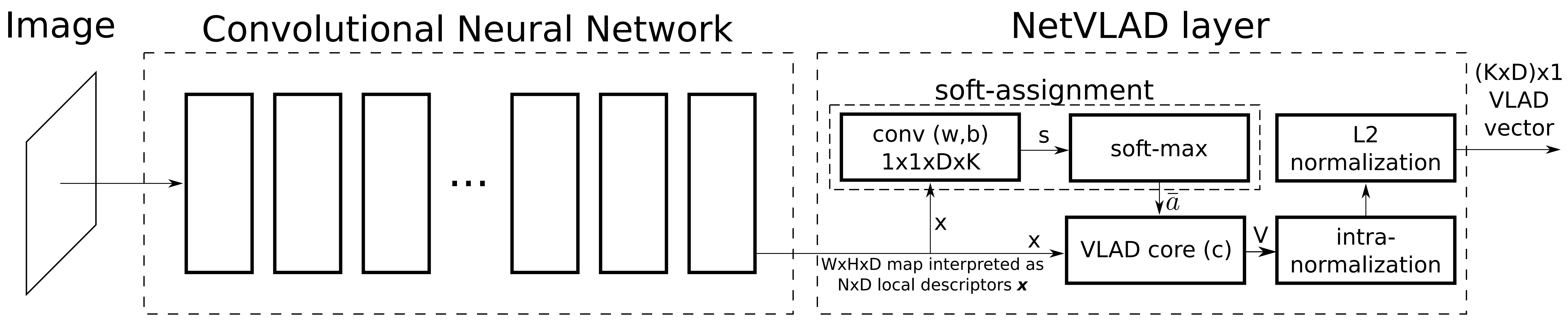
EdgeConv
mlp $\{a_1, a_2, \cdots, a_n\}$





NetVLAD

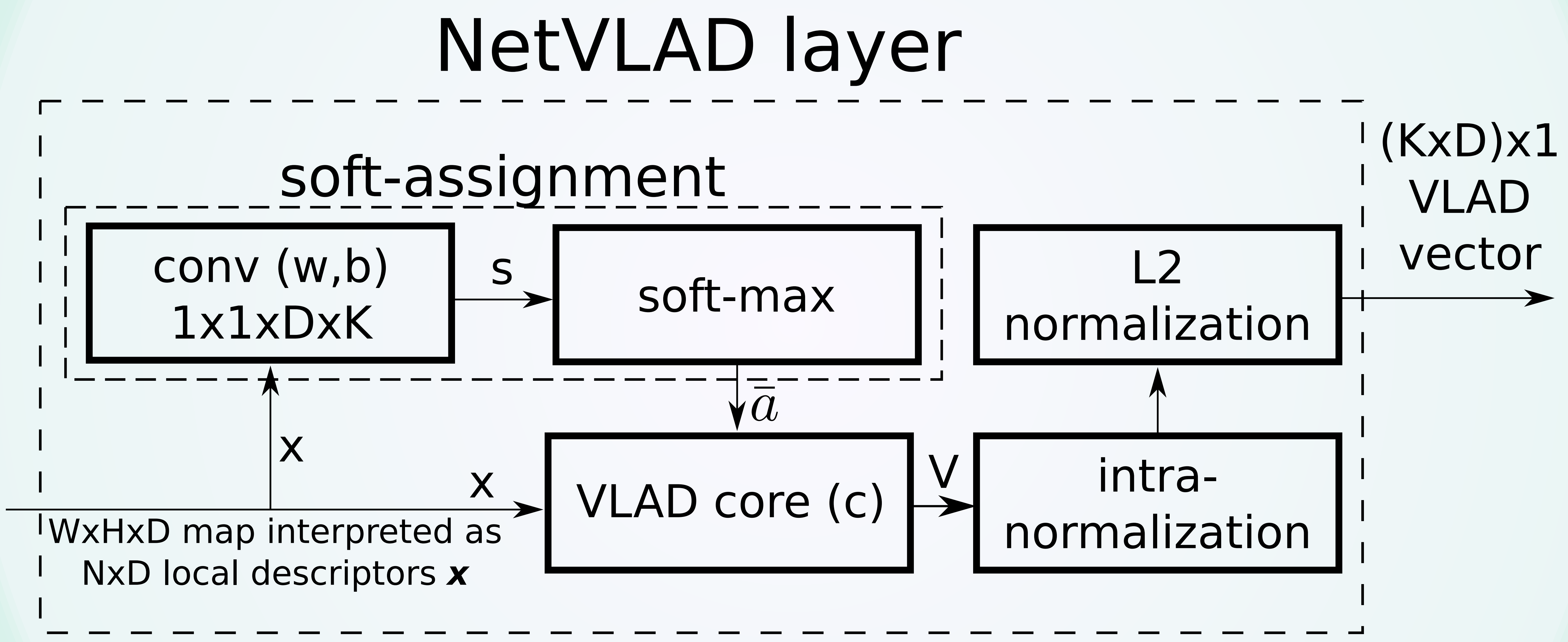
A Backbone for 2D VPR



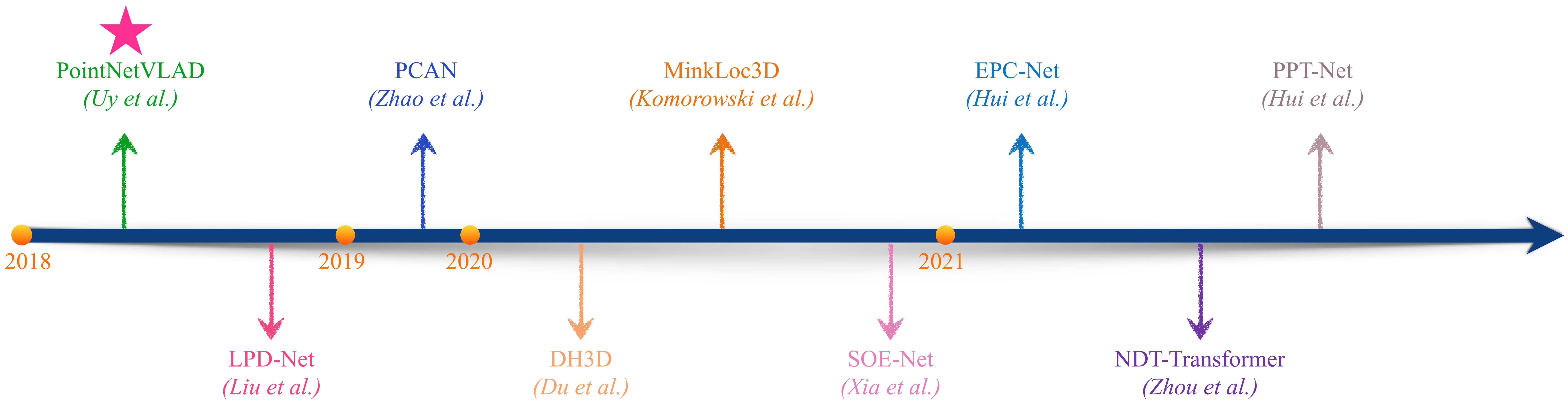
CNN architecture with the NetVLAD layer.

Arandjelović, Relja et al. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2018): 1437-1451.

Aggregate local descriptors

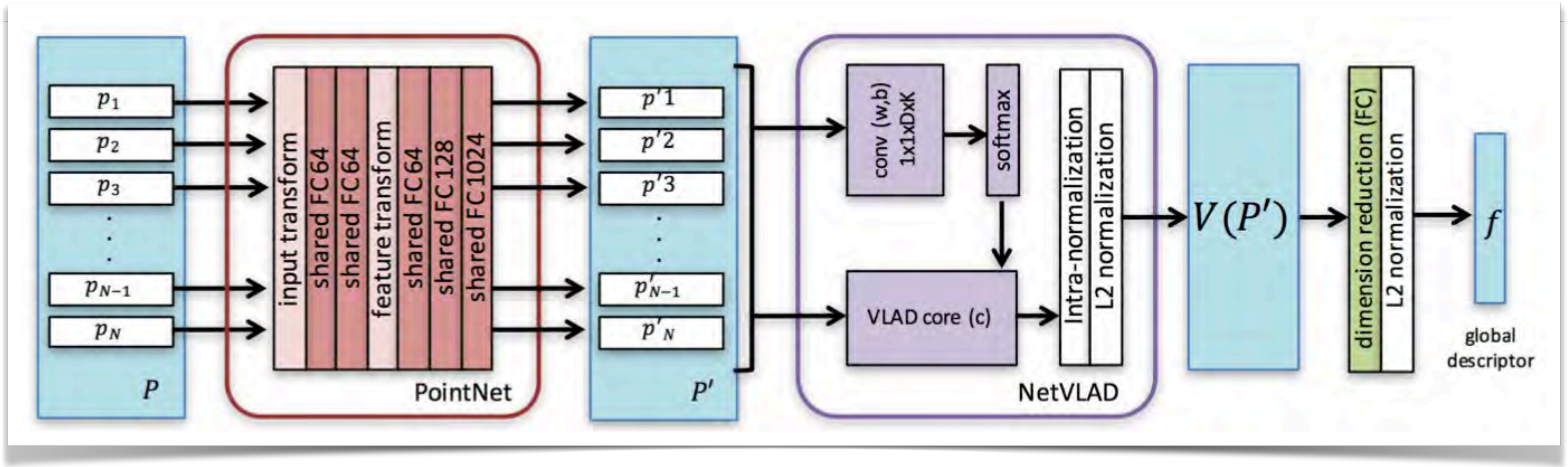


Chronological overview of 3D VPR



PointNetVLAD

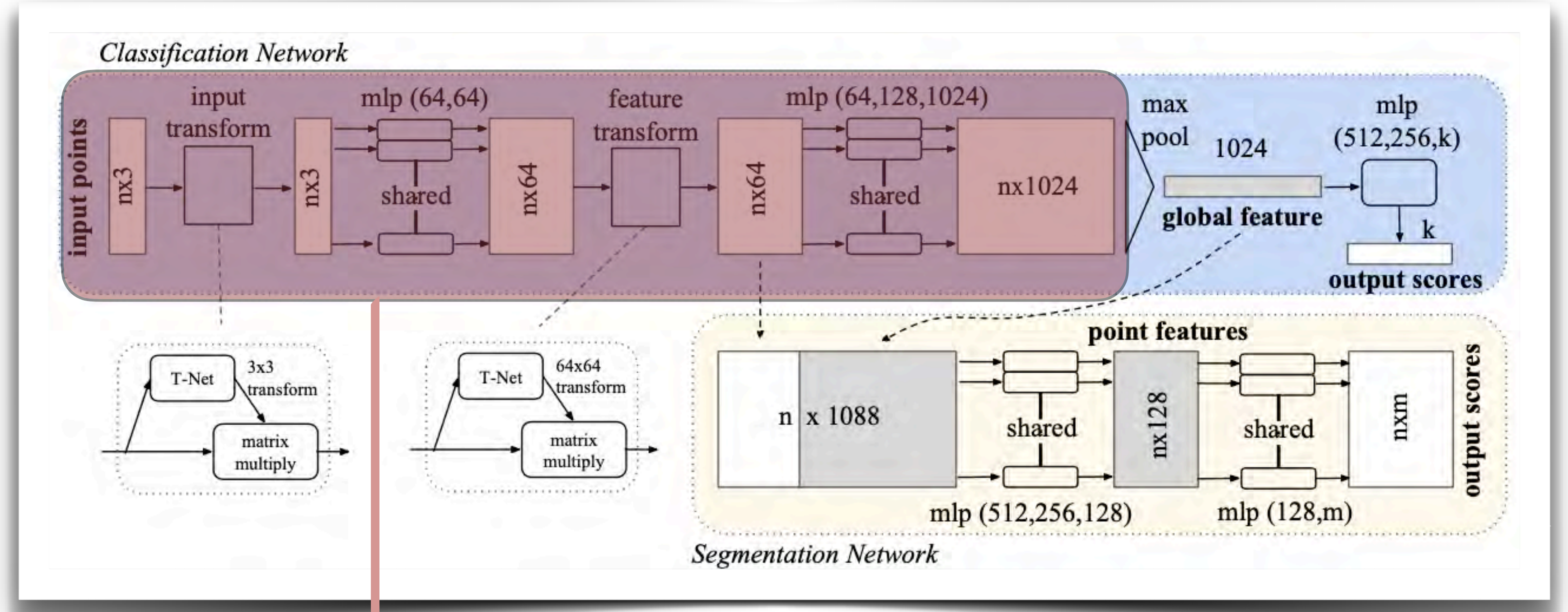
Pioneer of 3D VPR



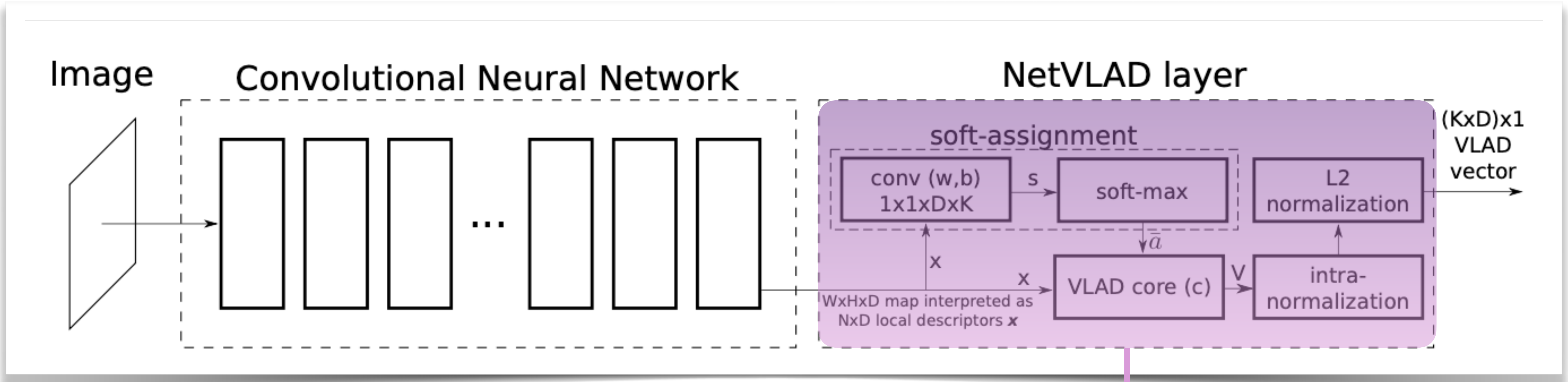
Network Architecture of PointNetVLAD.

Uy, Mikaela Angelina and Gim Hee Lee. “PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018): 4470-4479.

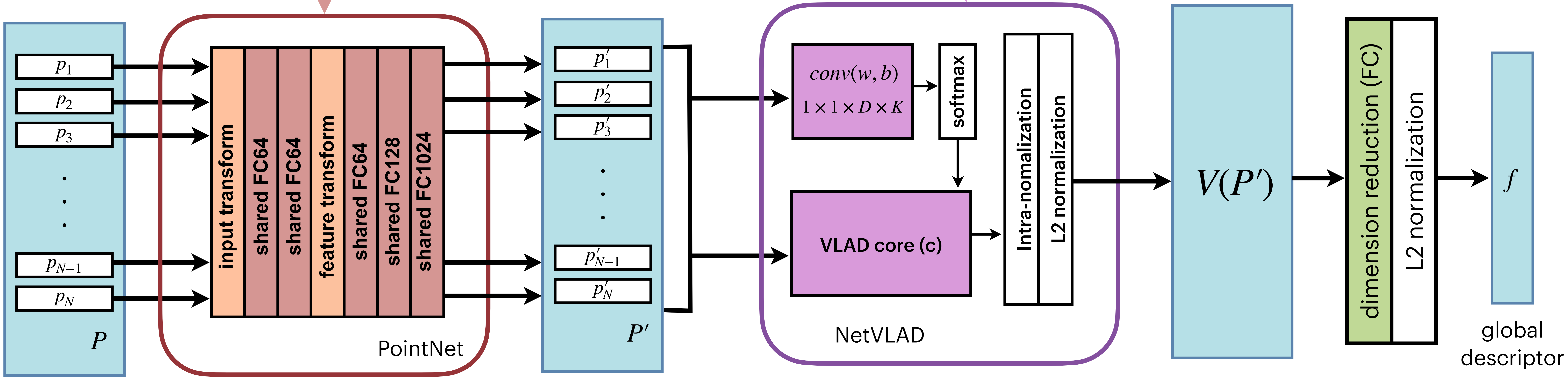
PointNetVLAD Backbone



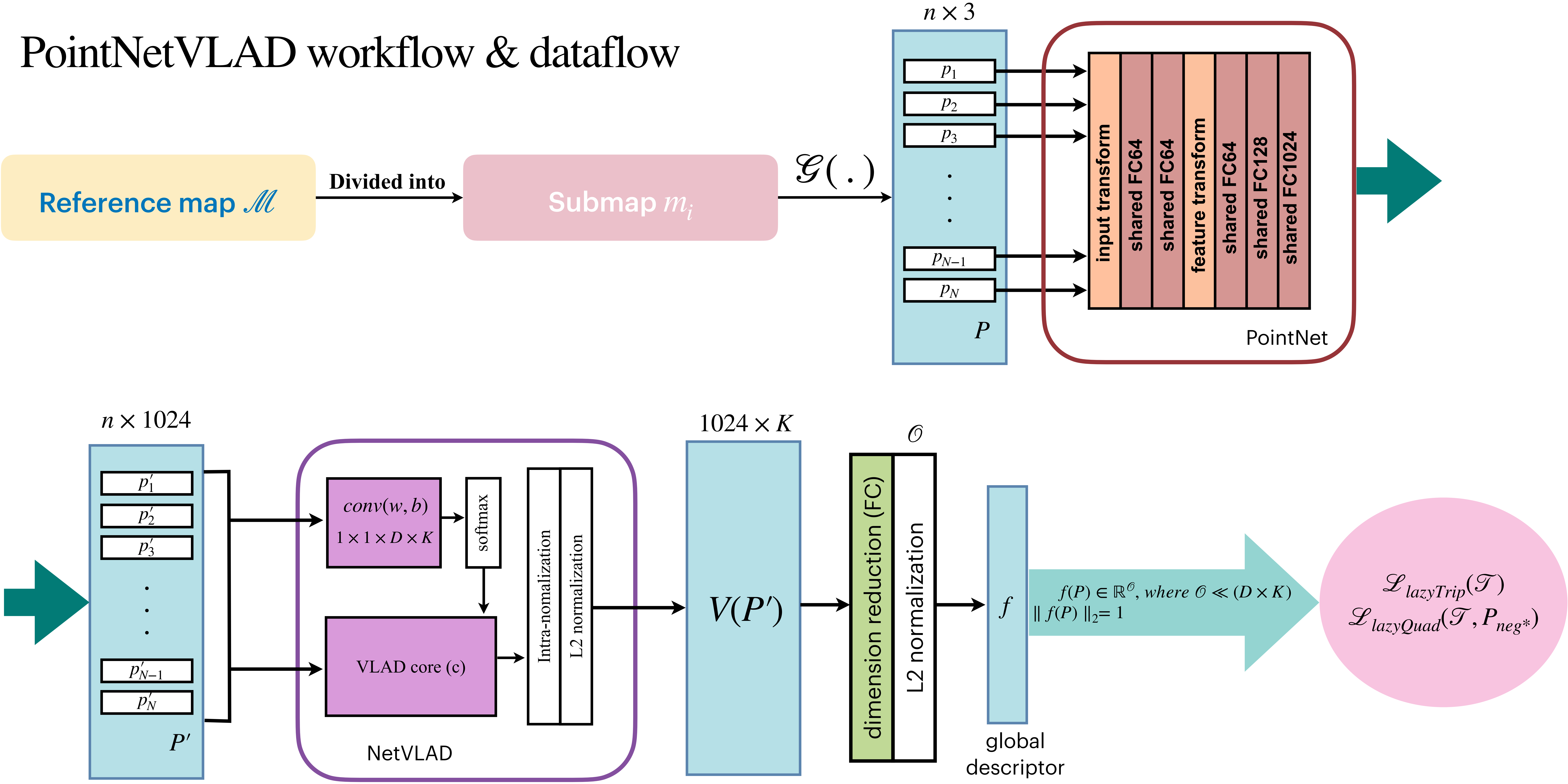
Network architecture of PointNet



CNN architecture with the NetVLAD layer

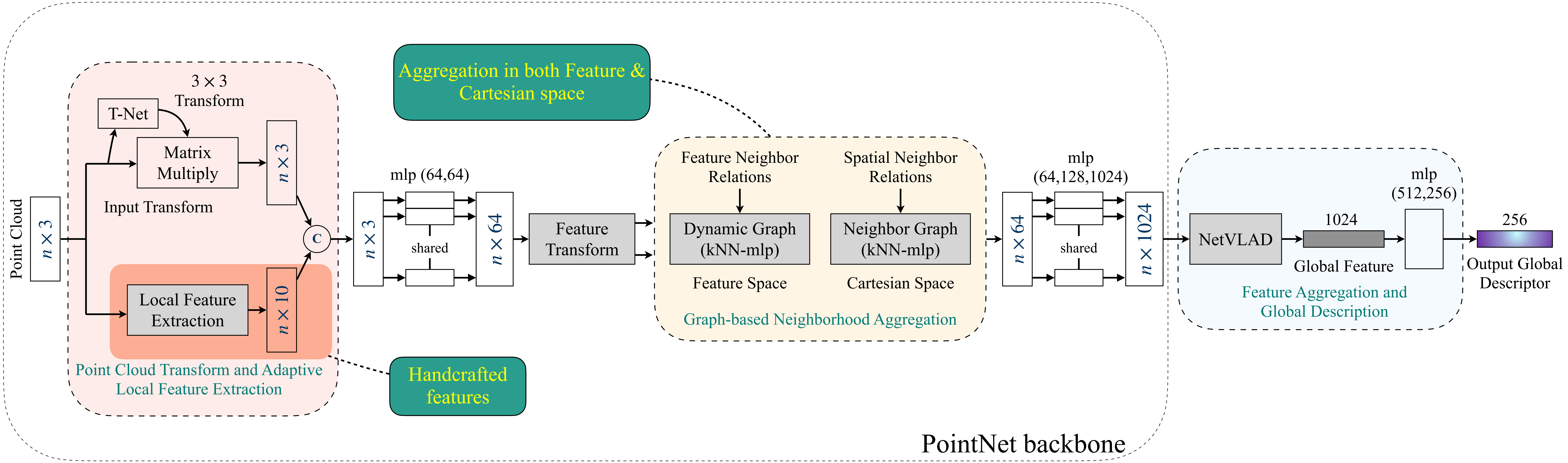


PointNetVLAD workflow & dataflow



LPD-Net

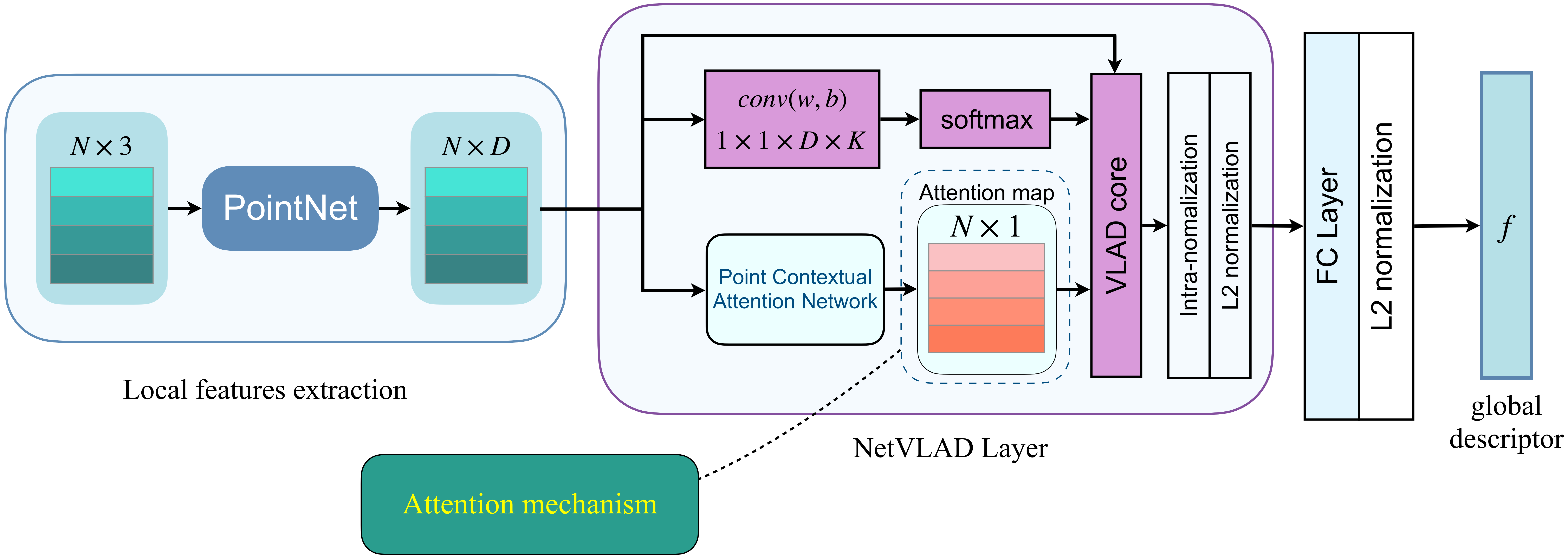
Handcrafted features, coordinate & feature space



Liu, Zhe et al. "LPD-Net: 3D Point Cloud Learning for Large-Scale Place Recognition and Environment Analysis." *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019): 2831-2840.

PCAN

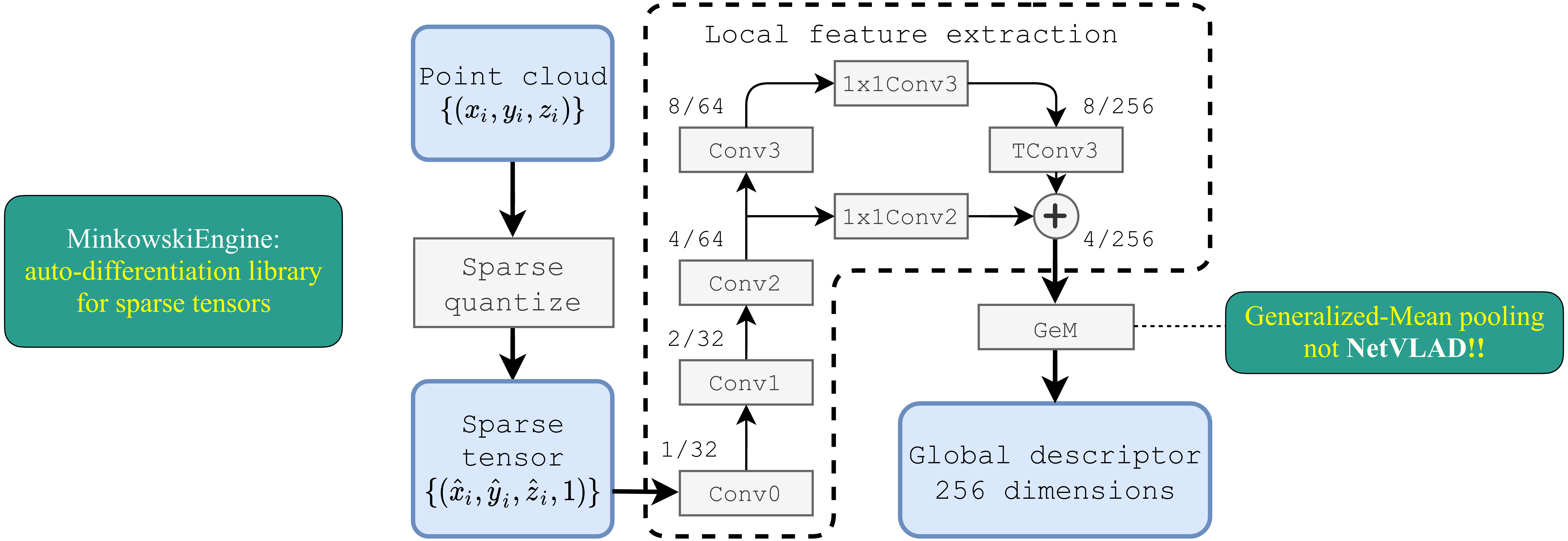
The first one to introduce the attention mechanism



Zhang, Wenxiao and Chunxia Xiao. "PCAN: 3D Attention Map Learning Using Contextual Information for Point Cloud Based Retrieval." *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019): 12428-12437.

MinkLoc3D

A **novel** backbone for 3D VPR

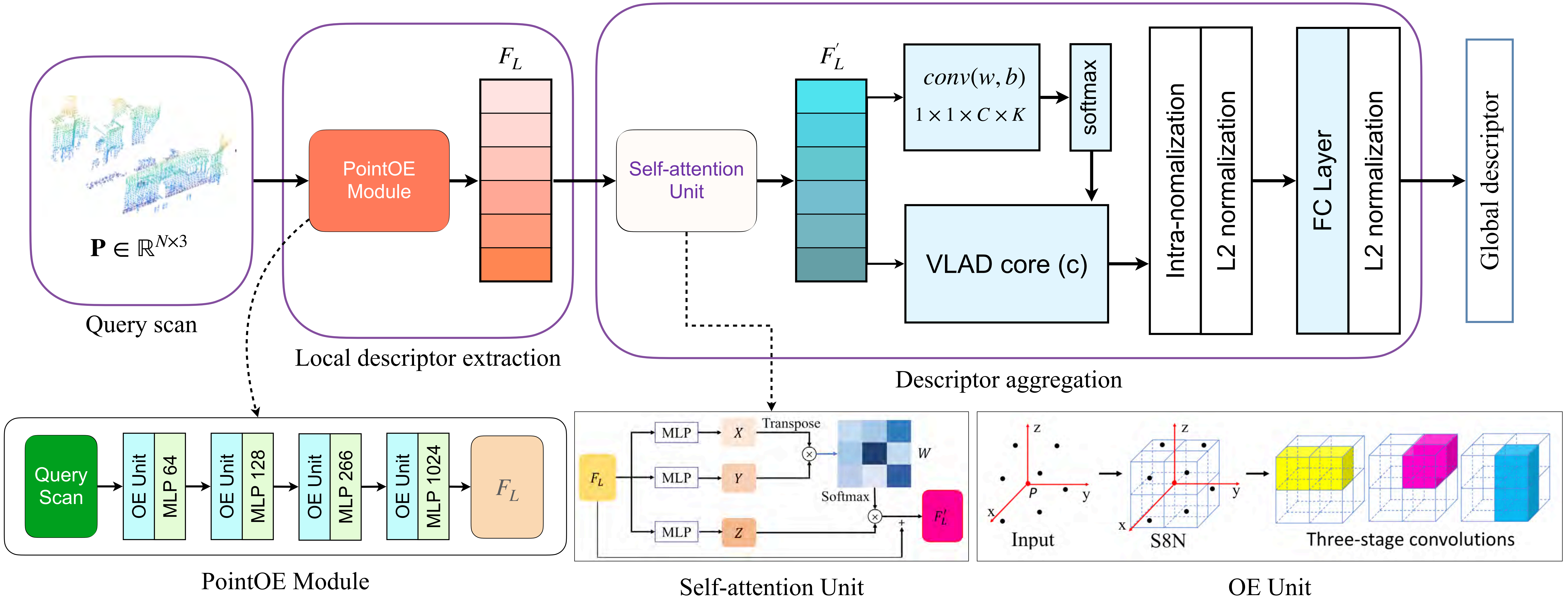


MinkLoc3D Architecture

Komorowski, Jacek. "MinkLoc3D: Point Cloud Based Large-Scale Place Recognition." *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2021): 1789-1798.

SOE-Net

Introduce self-attention & a new loss function



Xia, Yan et al. "SOE-Net: A Self-Attention and Orientation Encoding Network for Point Cloud based Place Recognition." *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021): 11343-11352.

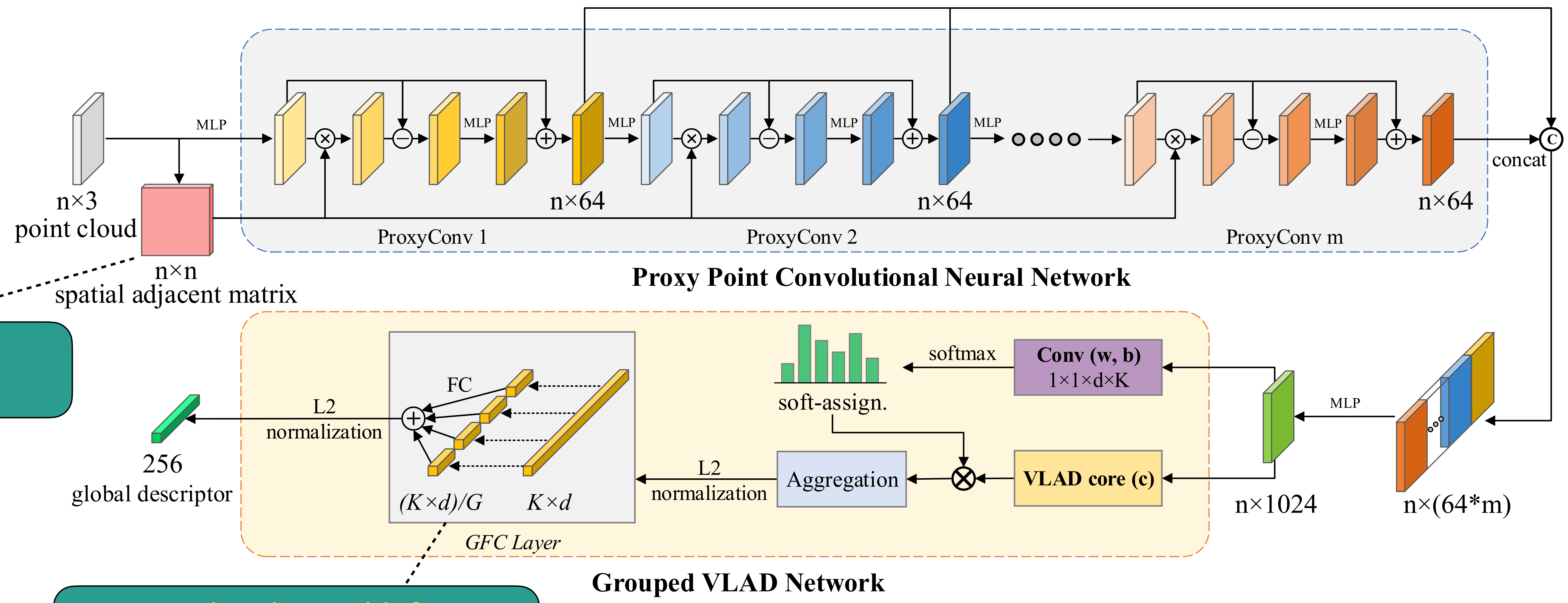
A New Loss Function for VPR:

Hardest **P**ositive **H**ardest **N**egative quadruplet loss (**HPHN loss**)

$$L_{HPHN} = \left[\left\| f\left(\delta_a\right) - f\left(\delta_{hp}\right) \right\|_2^2 - d_{hn} + \gamma \right]_+$$

EPC-Net

ProxyConv & grouped VLAD



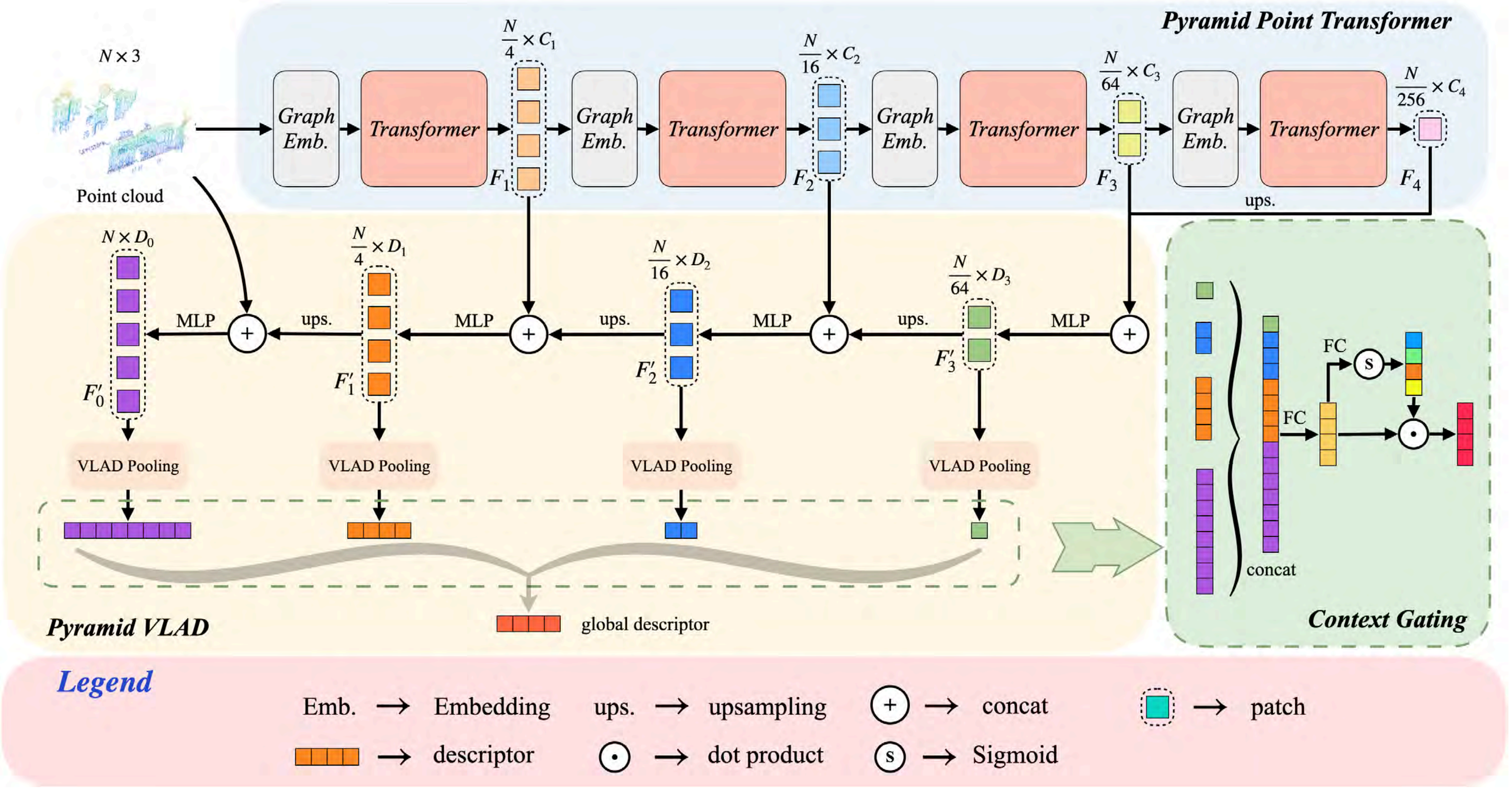
To achieve average operation

Better descriptor with fewer parameters

Hui, Le et al. "Efficient 3D Point Cloud Feature Learning for Large-Scale Place Recognition." *IEEE Transactions on Image Processing* 31 (2022): 1258-1270.

PPT-Net

A master of various algorithms



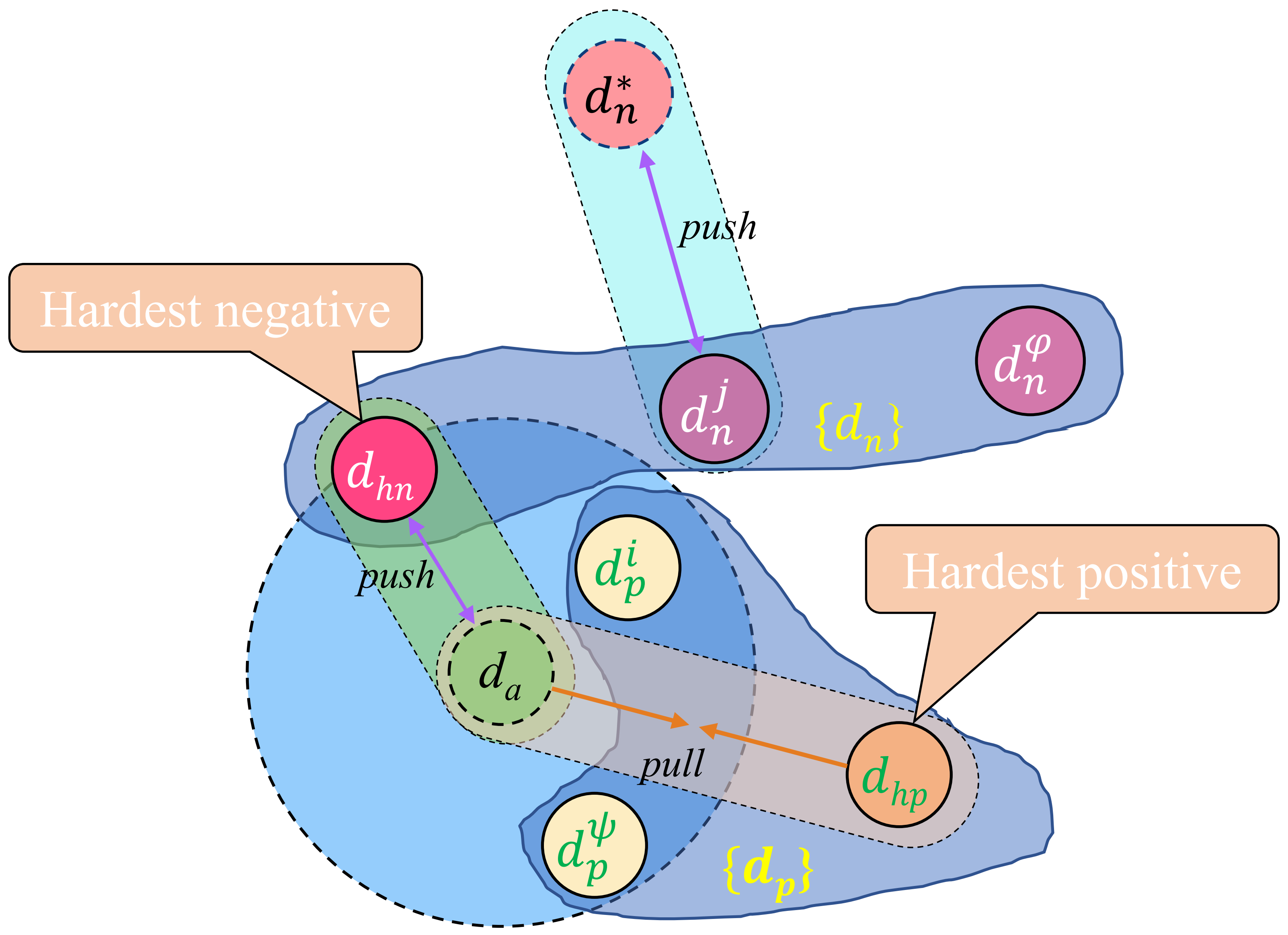
The pipeline of the pyramid point cloud transformer network (PPT-Net)

Hui, Le et al. "Pyramid Point Cloud Transformer for Large-Scale Place Recognition." *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021): 6078-6087.

IV. Methodology

HPHN quadruplet loss

Hardest **P**ositive **H**ardest **N**egative



- For a lazy quadruplet $Q_l = (d_a, \{d_p\}, \{d_n\}, d_n^*)$,
  where d_a is the anchor point cloud,
  $\{d_p\}$ is a collection of ψ positive point clouds,
  $\{d_n\}$ is a collection of φ negative point clouds,
  d_n^* is a randomly sampled point cloud, structurally dissimilar to d_a, d_p and d_n.

HPHN quadruplet loss

- The hardest positive point cloud d_{hp} is the least structurally similar to the anchor point cloud.
- The hardest negative point cloud is the most structurally similar to the anchor point cloud or the randomly sampled point cloud d_n^*.

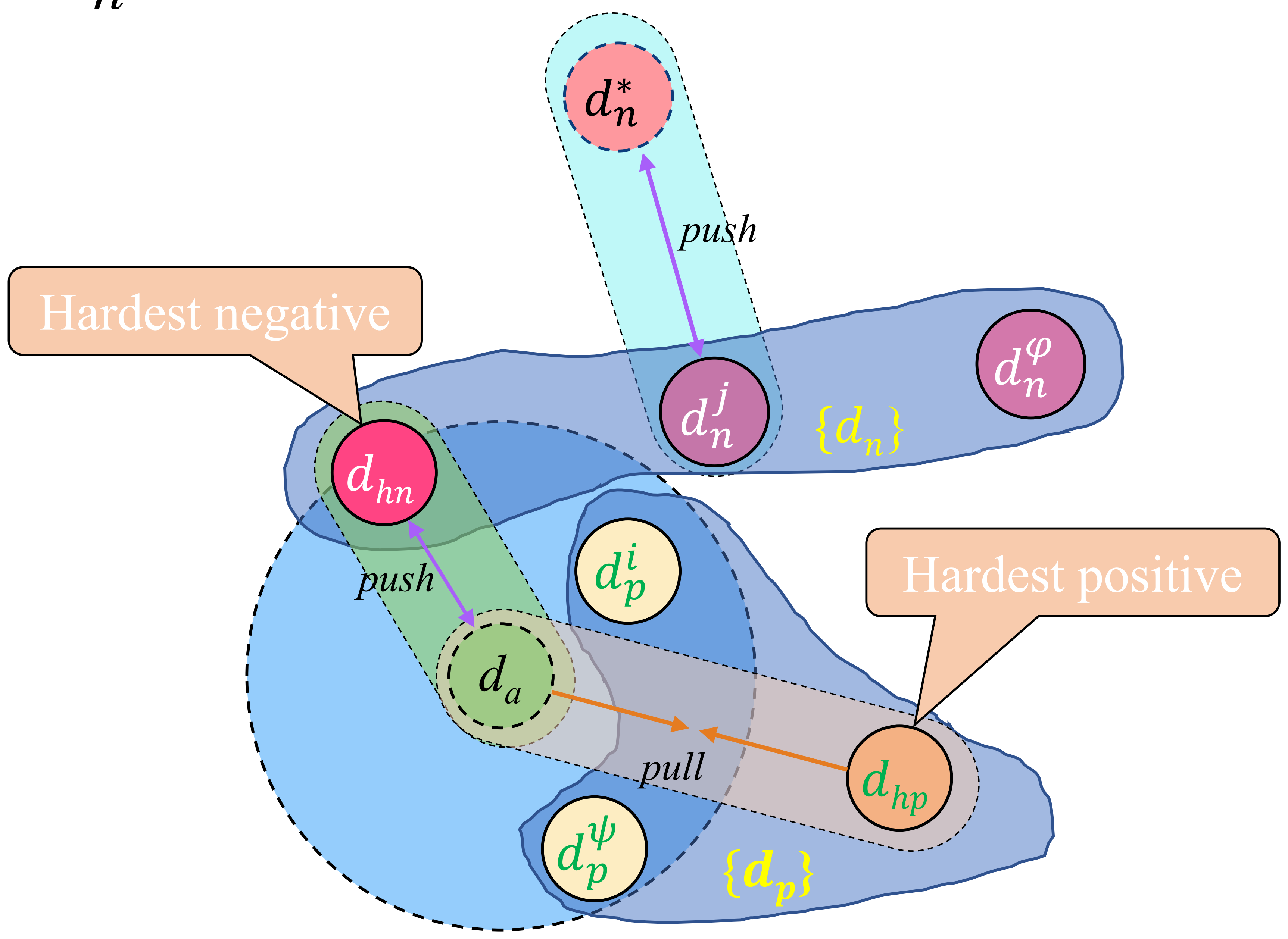
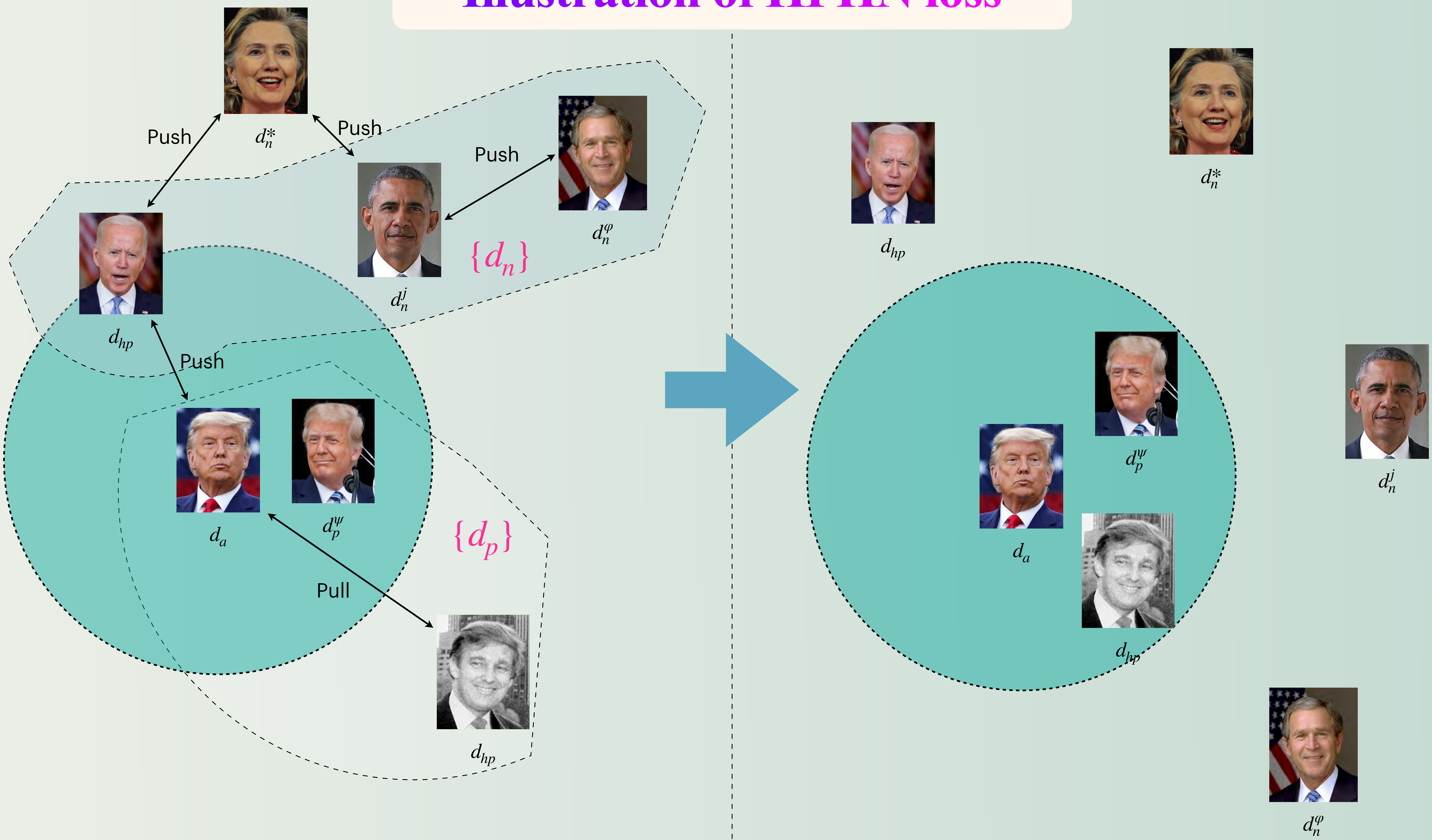
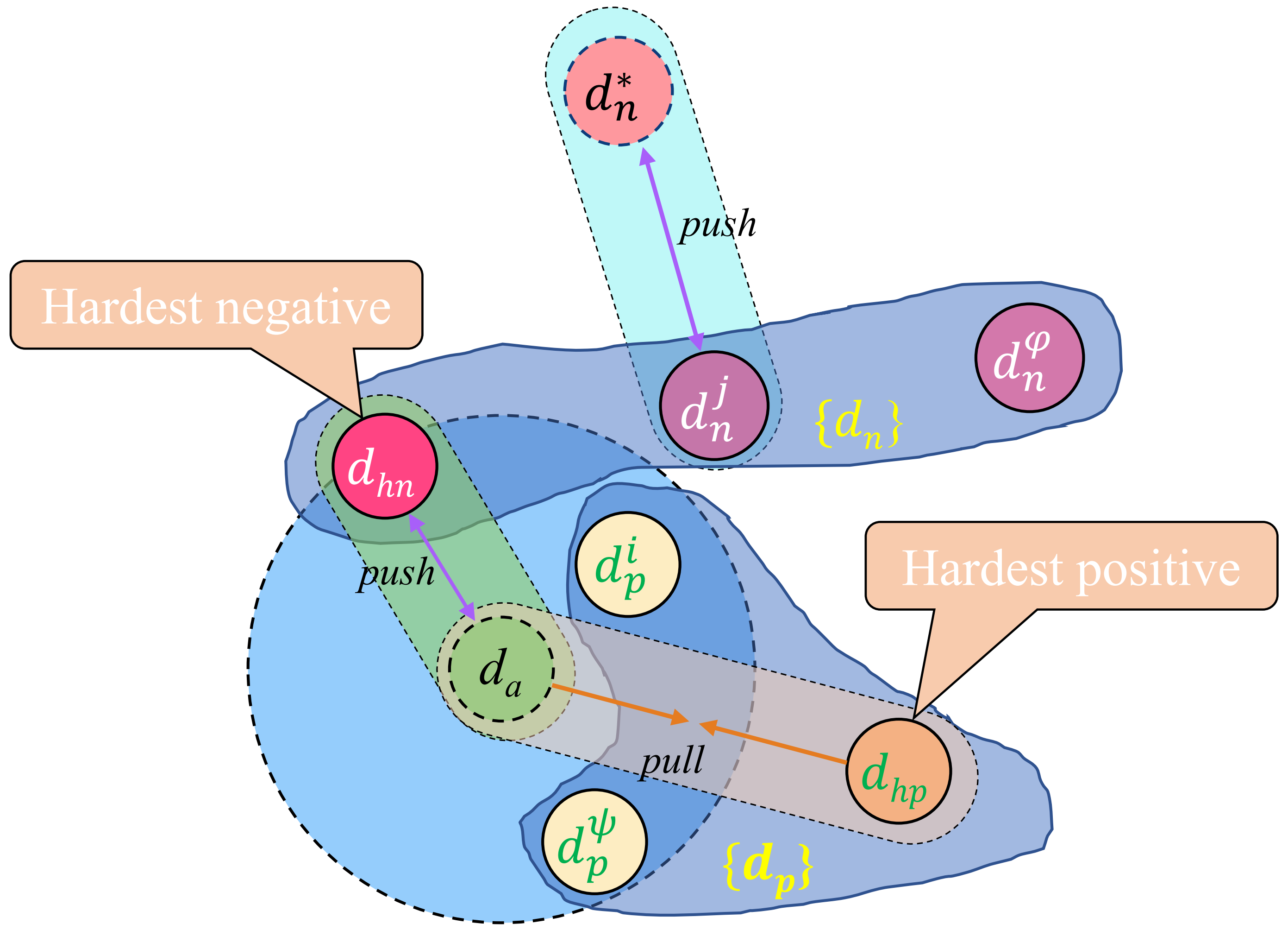


Illustration of HPHN loss

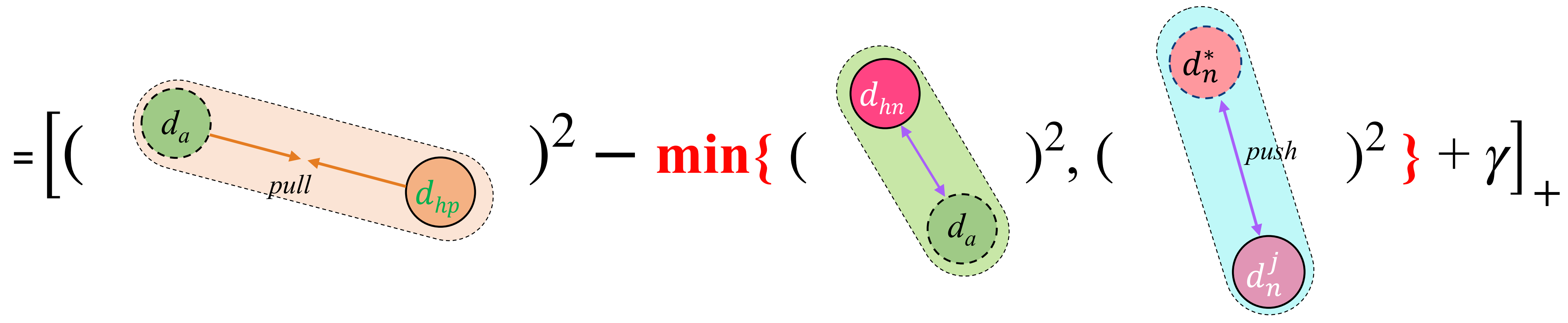


HPHN quadruplet loss

- In conclusion, the final HPHN quadruplet loss can be formulated as:

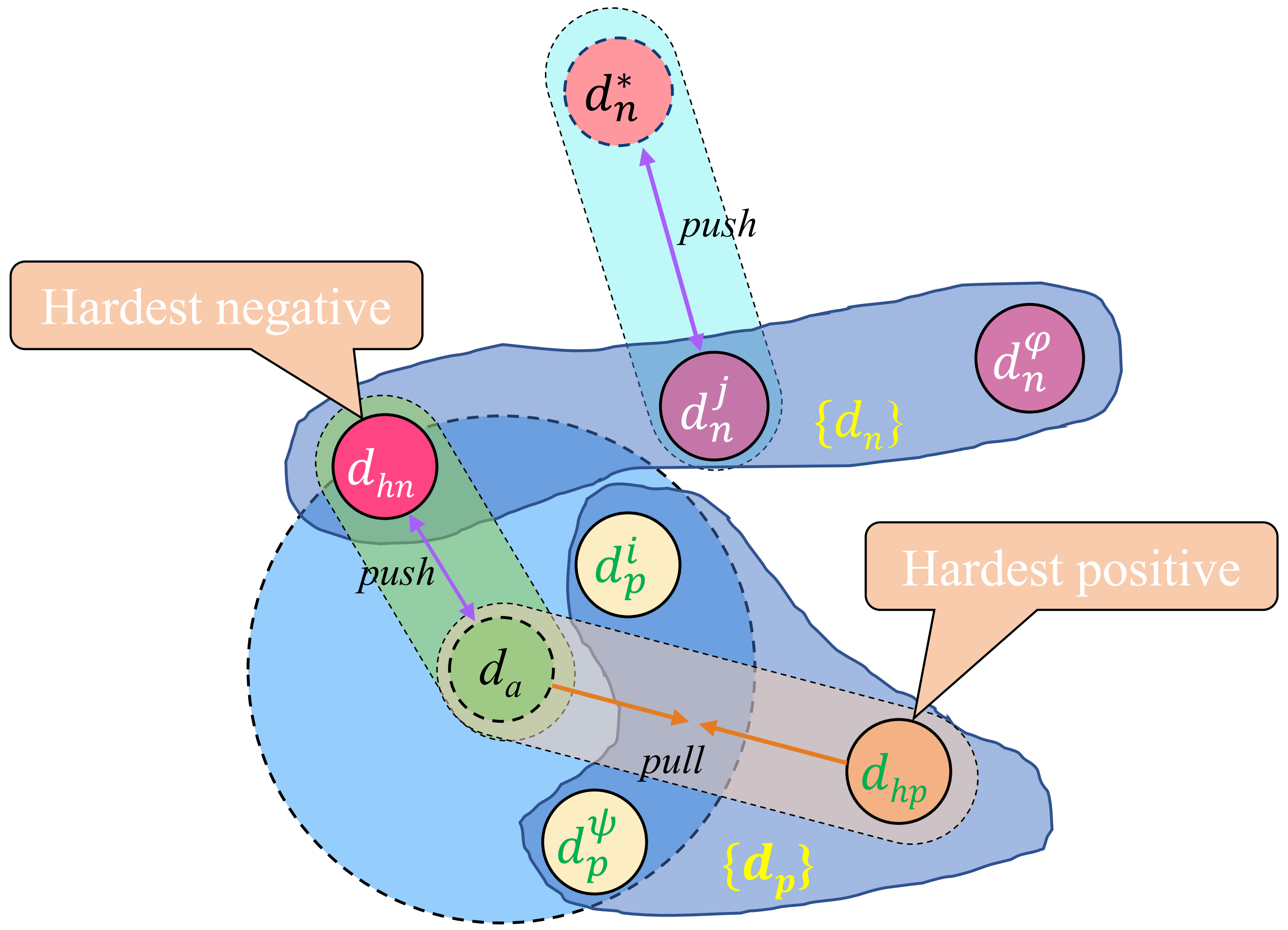


$$L_{HPHN} = \left[\left\| f(d_a) - f\left(d_{hp}\right) \right\|_2^2 - D_{hn} + \gamma \right]_+$$

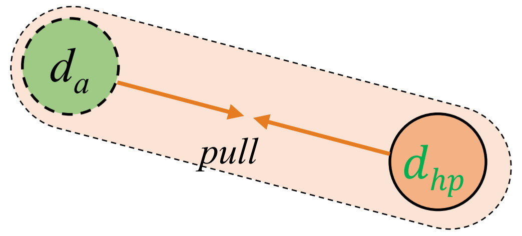
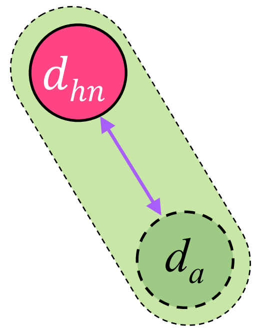
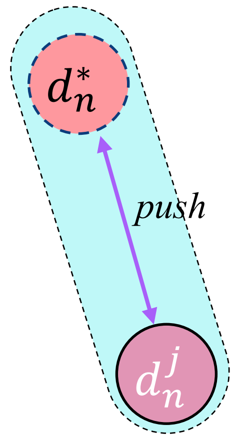


Scaled-HPHN quadruplet loss

Scaled-HPHN loss



$$L_{HPHN} = \left[\left\| f(d_a) - f(d_{hp}) \right\|_2^2 - D_{hn} + \gamma \right]_+$$

$=\Big[\Big($  $\Big)^2 -$ **min{** $\Big($  $\Big)^2, \Big($  $\Big)^2$ **}** $+ \gamma\Big]_+,$

introduce a **scale factor** κ, then we have

$$L_{S-HPHN} = \left[\left\| f(d_a) - f(d_{hp}) \right\|_2^2 - \kappa D_{hn} + \gamma \right]_+$$

V. Experiments

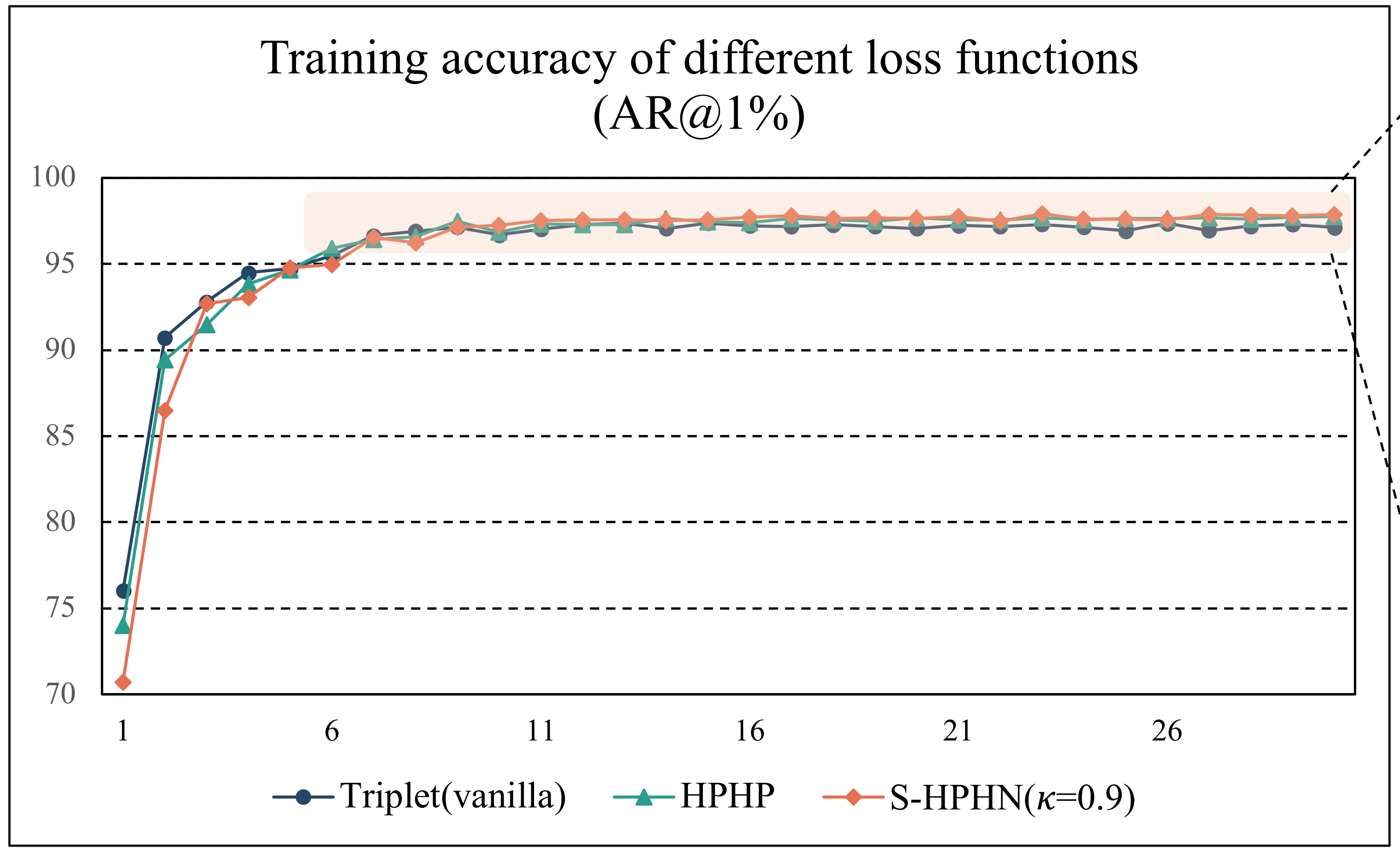
HPHN in PPT-Net

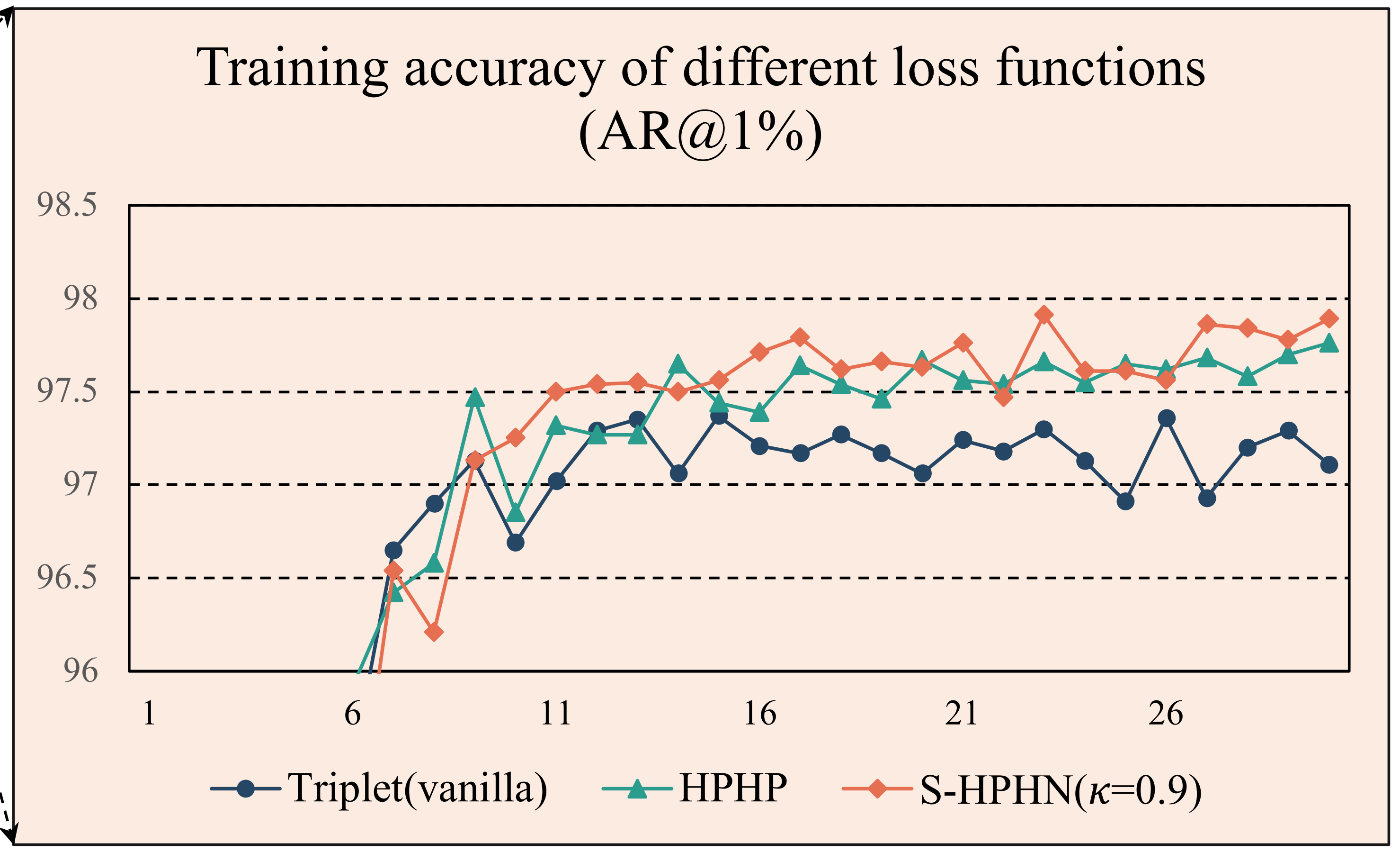
Introduce HPHN to PPT-Net

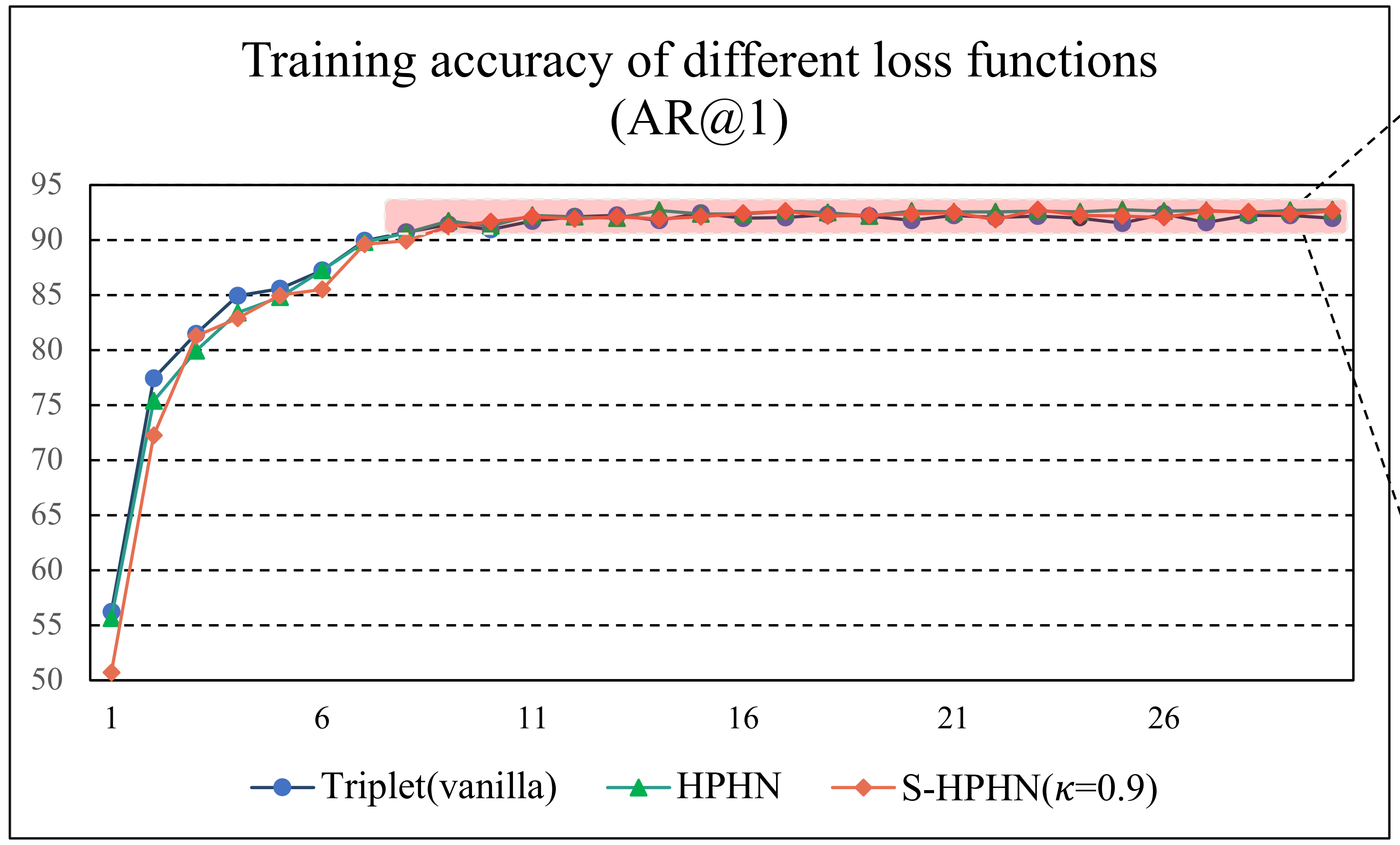
Loss	AR@1%	AR@1
Triplet (vanilla)	97.11	92.03
HPHN	**97.76** **(+0.65)**	**92.78** **(+0.75)**

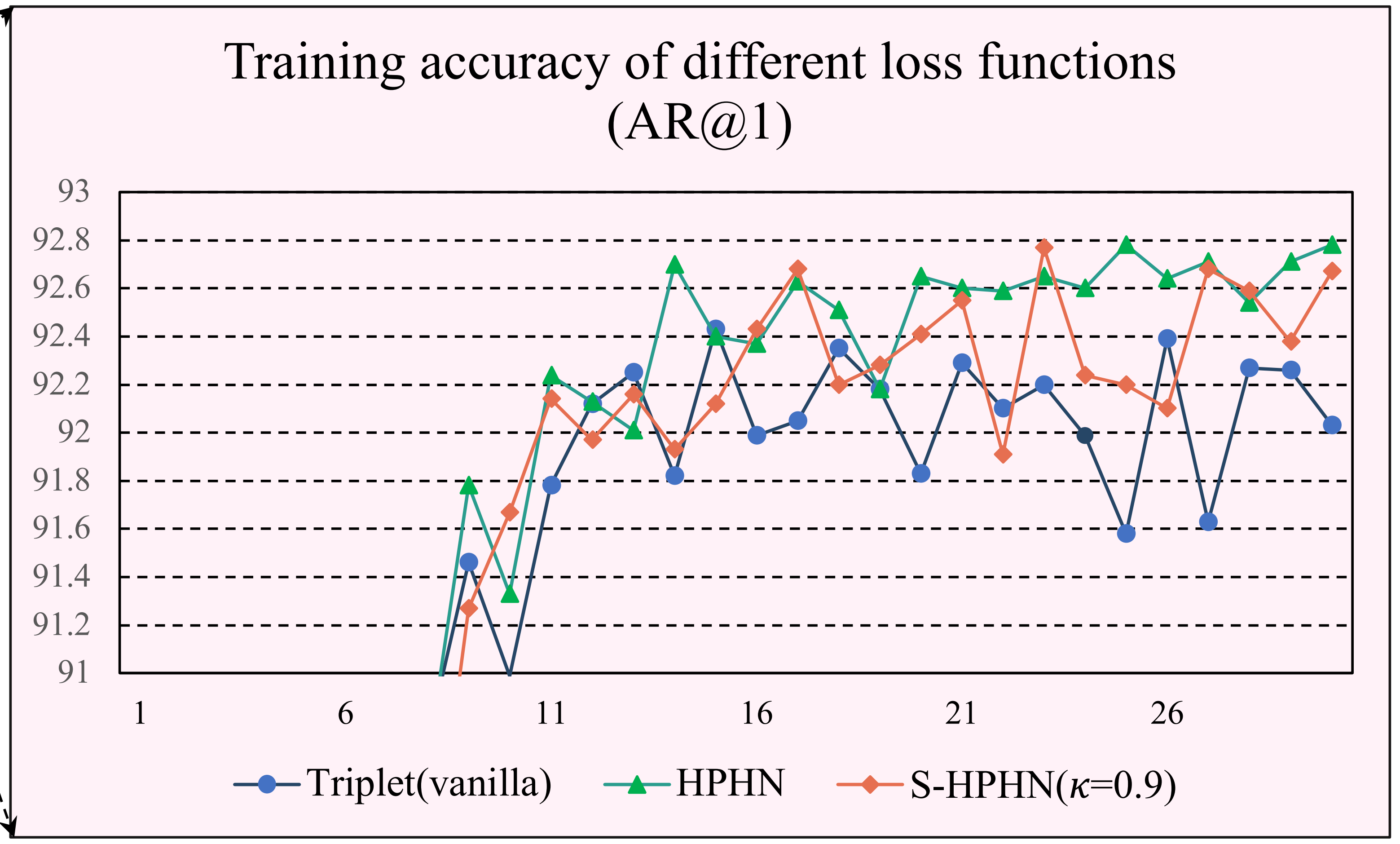
Ablation study

κ	AR@1%	AR@1
0.9	**97.89**	92.67
1	97.76	**92.78**
1.001	97.60	92.65
1.01	96.98	91.59
1.1	97.17	91.99


Training accuracy of different loss functions
(AR@1%)
100
95
90
85
80
75
70
1
6
11
16
21
26
Triplet(vanilla)
HPHP
S-HPHN(κ=0.9)


Training accuracy of different loss functions
(AR@1%)
98.5
98
97.5
97
96.5
96
1
6
11
16
21
26
Triplet(vanilla)
HPHP
S-HPHN(κ=0.9)


Training accuracy of different loss functions
(AR@1)
95
90
85
80
75
70
65
60
55
50
1
6
11
16
21
26
Triplet(vanilla)
HPHN
S-HPHN(κ=0.9)


Training accuracy of different loss functions
(AR@1)
93
92.8
92.6
92.4
92.2
92
91.8
91.6
91.4
91.2
91
1
6
11
16
21
26
Triplet(vanilla)
HPHN
S-HPHN(κ=0.9)

Comparison between three losses

Loss	AR@1%	AR@1
Triplet (vanilla)	97.11	92.03
HPHN	97.76 (+0.65)	**92.78 (+0.75)**
S-HPHN (κ=0.9)	**97.89 (+0.78)**	92.67 (+0.64)

VI. Conclusion

- The HPHN loss is implemented in PPT-Net.

- Based on HPHN loss, a scale factor is introduced to propose the scaled-HPHN loss function.

- Experiments show that both HPHN and scaled-HPHN are better than the original triplet loss.

VII. Future work

Trainable scale factor

$$L_{S-HPHN} = \left[\left\| f(d_a) - f\left(d_{hp}\right) \right\|_2^2 - \kappa D_{hn} + \gamma \right]_+$$

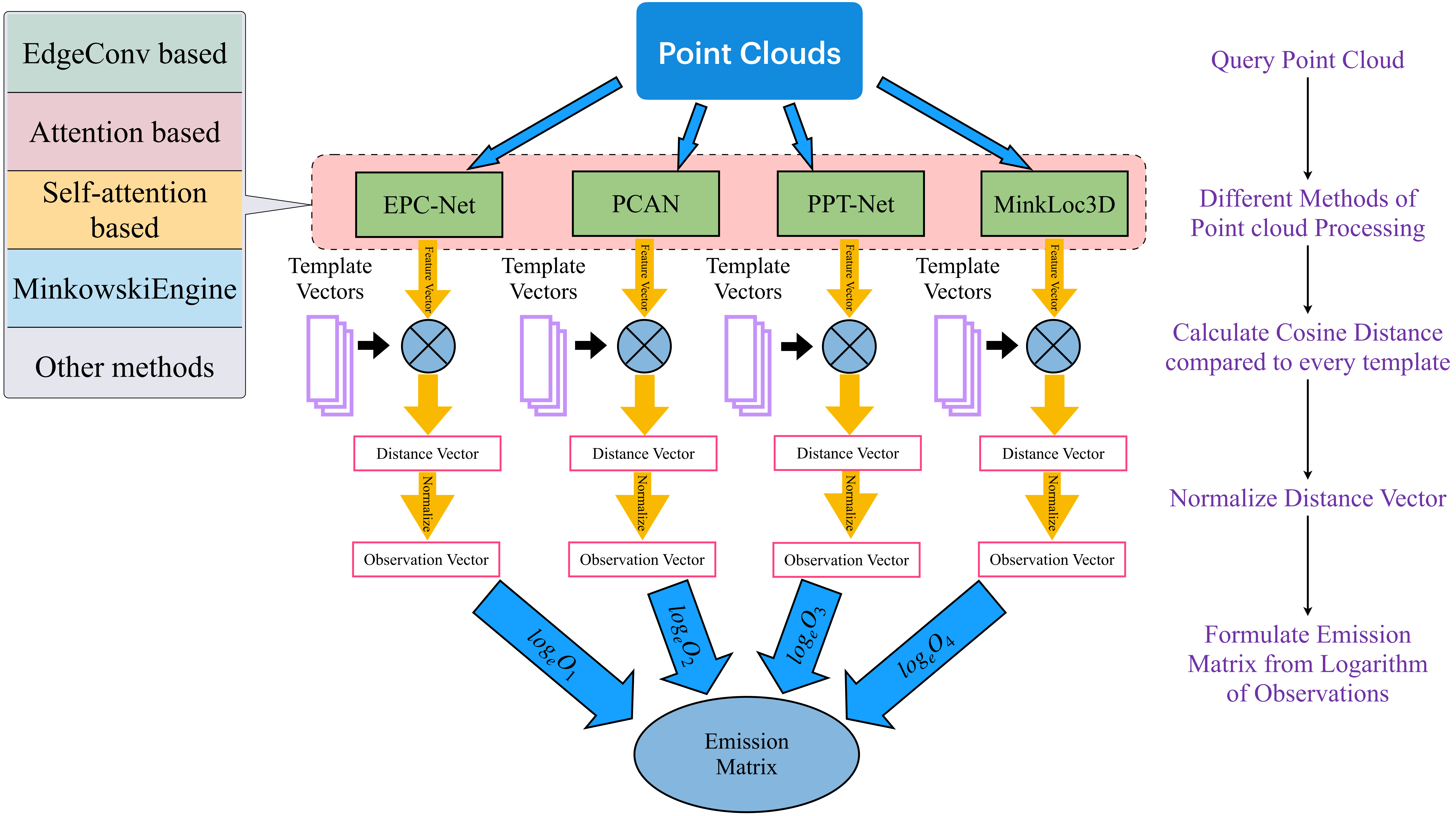
Make it trainable

Classification of 3D VPR Algorithms

EdgeConv based	Attention based	Self-attention based	MinkowskiEngine
LPD-Net EPC-Net PPT-Net	PCAN DH3D	SOE-Net PPT-Net	MinkLoc3D

Method	Parameters	FLOPs	Runtime per frame
PN_VLAD	19.78M	4.21G	20ms
PCAN	20.42M	7.73G	58ms
LPD-Net	19.81M	7.80G	28ms
MinkLoc3D	**1.10M**	**1.81G**	**17ms**
EPC-Net	4.70M	3.25G	20ms
PPT-Net	13.12M	3.23G	18ms

Multi-process Fusion



Q & A