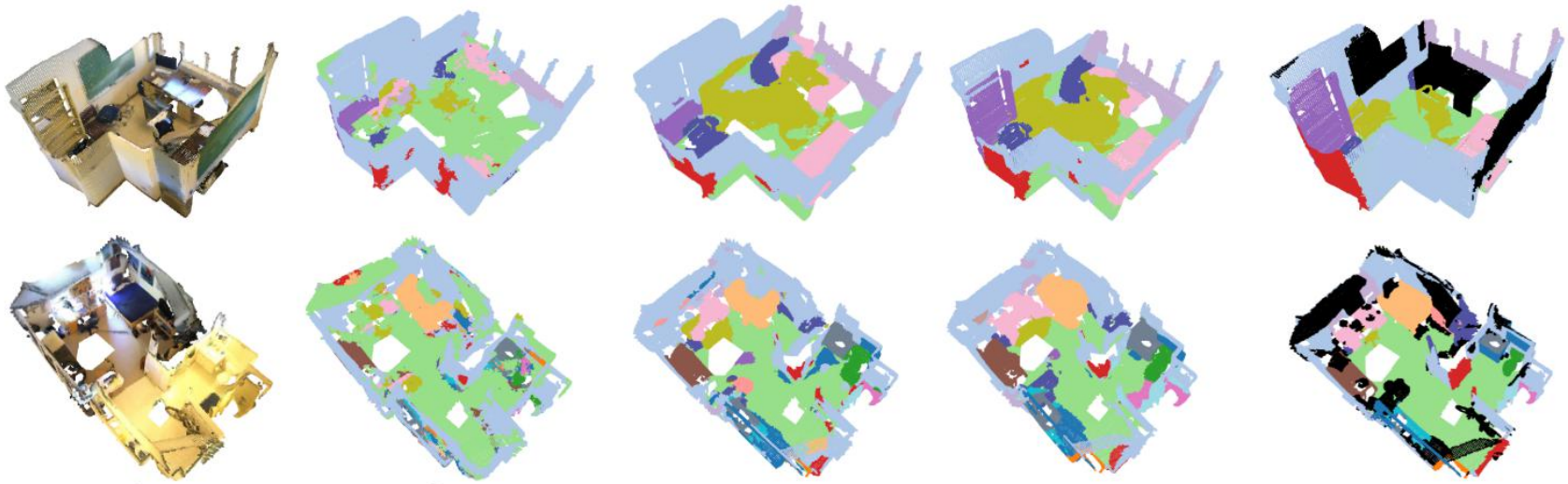
Deep Learning for 3D Point Cloud Analysis



Xu Yan

2020.8.20

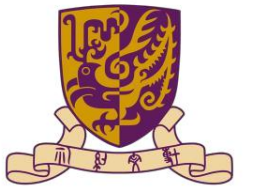
香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Paper List

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

- PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding (ECCV2020 spotlight)
- A Closer Look at Local Aggregation Operators in Point Cloud Analysis (ECCV2020)
- Towards Content-Independent Multi-Reference Super-Resolution: Adaptive Pattern Matching and Feature Aggregation (ECCV2020)

PointContrast



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding

Saining Xie[1], Jiatao Gu[1], Demi Guo[⋆], Charles R. Qi[⋆],
Leonidas Guibas[2⋆], and Or Litany[2⋆]

[1] Facebook AI Research
[2] Stanford University

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

All existing representation learning schemes are tested either on single objects or low-level tasks (e.g. registration). This status quo can be attributed to multiple reasons:

1) Lack of large-scale and high-quality **data**: compared to 2D images, 3D data is harder to collect, more expensive to label, and the variety of sensing devices may introduce drastic domain gaps;

2) Lack of unified **backbone** architectures: in contrast to 2D vision where architectures such as ResNets have proven successful as backbone networks for pre-training and fine-tuning, point cloud network architecture designs are still evolving;

3) Lack of a comprehensive set of datasets and high-level tasks for **evaluation**.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

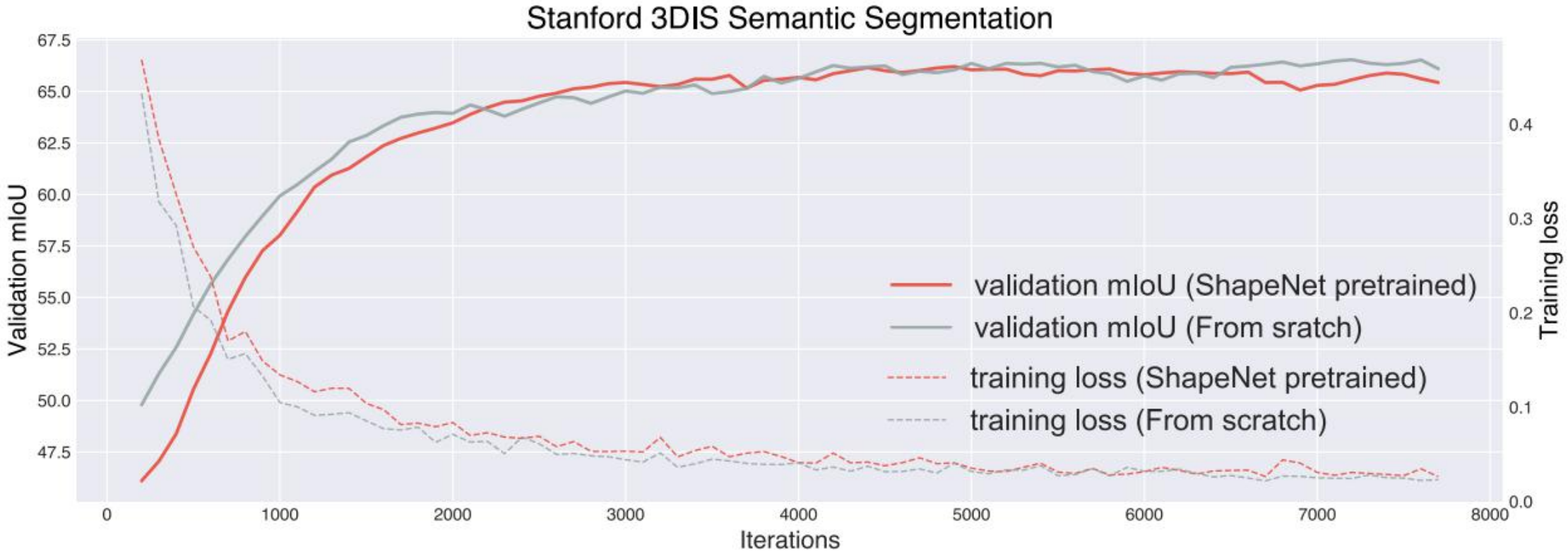


Fig. 1: Training from scratch *vs.* fine-tuning with ShapeNet pre-trained weights.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

PointContrast: Downstream Tasks for Fine-tuning					
Datasets	Real / Synth.	Complexity	Env.	Task	Rel. gain
S3DIS	Real	Entire floor, office	Indoor	Segmentation	(+2.7%) mIoU
SUN RGB-D	Real	Medium-sized cluttered rooms	Indoor	Detection	(+3.1%) mAP0.5
ScanNetV2	Real	Large rooms	Indoor	Segmentation Detection	(+1.9%) mIoU (+2.6%) mAP0.5
ShapeNet	Synth.	Single objects	Indoor & outdoor	Classification	(+4.0%) Acc.*
ShapeNetPart	Synth.	Object parts	Indoor & outdoor	Segmentation	(+2.2%) mIoU*
Synthia 4D	Synth.	Street scenes, driving envs.	Outdoor	Segmentation	(+3.3%) mIoU

Table 1: **Summary of downstream fine-tuning tasks.** Compared to the baseline learning paradigm of training from scratch, which is dominant in 3D deep learning, our unsupervised pre-training method PointContrast boosts the performance across the board when finetuning on a diverse set of high-level 3D understanding tasks.
$*$ indicates results trained using only 1% of the training data.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

– **Domain gap between source and target data:** Objects in ShapeNet are synthetic, normalized in scale, aligned in pose, and lack scene context. This makes pre-training and fine-tuning data distributions drastically different.
– **Point-level representation matters:** In 3D deep learning, the local geometric features, *e.g.* those encoded by a point and its neighbors, have proven to be discriminative and critical for 3D tasks [47, 48]. Directly training on *object instances* to obtain a global representation might be insufficient.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Algorithm 1 General Framework of PointContrast

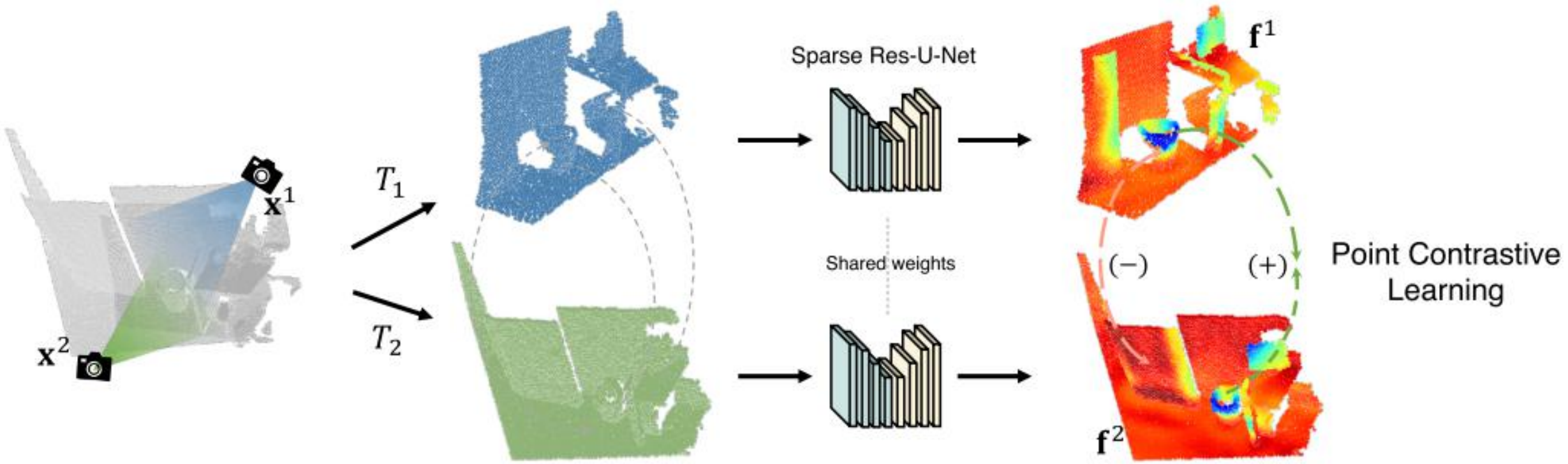
Input: Backbone architecture NN; Dataset $X = \{\mathbf{x}_i \in \mathbb{R}^{N\times 3}\}$; Point feature dimension D;
Output: Pre-trained weights for NN.
for *each point cloud* $\mathbf{x}$ *in* X **do**
- From $\mathbf{x}$, generate two views $\mathbf{x}^1$ and $\mathbf{x}^2$.
- Compute correspondence mapping (matches) M between points in $\mathbf{x}^1$ and $\mathbf{x}^2$.
- Sample two transformations $\mathbf{T}_1$ and $\mathbf{T}_2$.
- Compute point features $\mathbf{f}^1, \mathbf{f}^2 \in \mathbb{R}^{N\times D}$ by $\mathbf{f}^1 = \mathrm{NN}(\mathbf{T}_1(\mathbf{x}^1))$ and $\mathbf{f}^2 = \mathrm{NN}(\mathbf{T}_2(\mathbf{x}^2))$.
- Backprop. to update NN with contrastive loss $\mathcal{L}_c(\mathbf{f}^1, \mathbf{f}^2)$ on the matched points.

end

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Fig. 2: **PointContrast: Pretext task for 3D pre-training.**



PointContrast

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

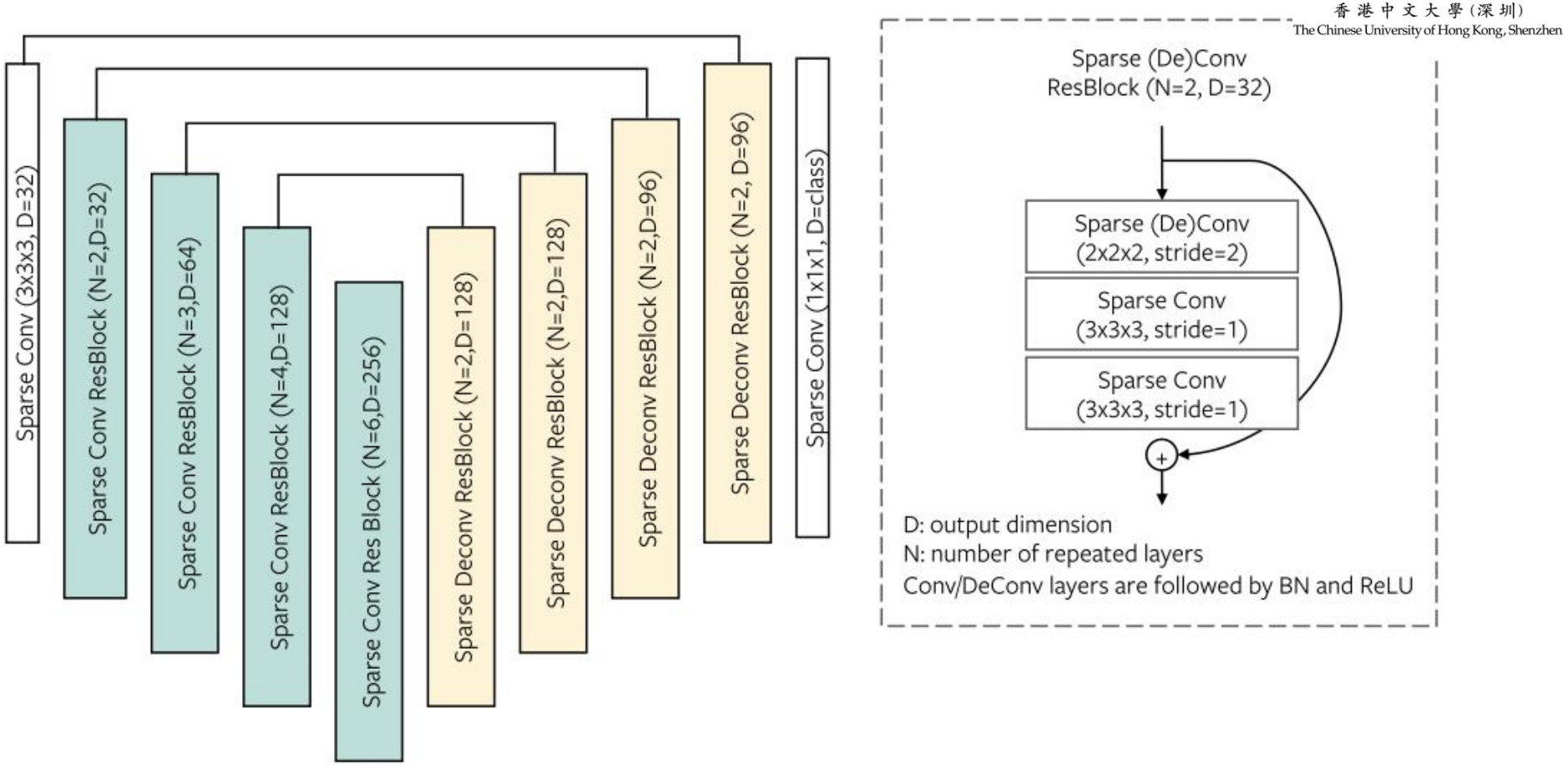


Fig. 3: SR-UNet architecture we used as a shared backbone network for pre-training and fine-tuning tasks. For segmentation and detection tasks, both the encoder and decoder weights are fine-tuned; for classification downstream tasks, only the encoder network is kept and fine-tuned.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

Hardest-Contrastive Loss

$$\mathcal{L}_c = \sum_{(i,j)\in\mathcal{P}} \left\{ [d(\mathbf{f}_i, \mathbf{f}_j) - m_p]_+^2 / |\mathcal{P}| + 0.5[m_n - \min_{k\in\mathcal{N}} d(\mathbf{f}_i, \mathbf{f}_k)]_+^2 / |\mathcal{N}_i| + 0.5[m_n - \min_{k\in\mathcal{N}} d(\mathbf{f}_j, \mathbf{f}_k)]_+^2 / |\mathcal{N}_j| \right\}$$

Here P is a set of matched (**positive**) pairs of points and N is a randomly sampled set of non-matched (**negative**) points which is used for the hardest negative mining, where the hardest sample is defined as the closest point in the L2 normalized feature space to a positive pair. [x]+ denotes function max(0, x). $m_p = 0.1$ and $m_n = 1.4$ are margins for positive and negative pairs.

PointInfoNCE Loss

$$\mathcal{L}_c = -\sum_{(i,j)\in\mathcal{P}} \log \frac{\exp(\mathbf{f}_i \cdot \mathbf{f}_j/\tau)}{\sum_{(\cdot,k)\in\mathcal{P}} \exp(\mathbf{f}_i \cdot \mathbf{f}_k/\tau)}$$

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

evaluating on all 55 classes	1% data	10% data	100% data
Trained from scratch	62.2	77.9	85.1
PointContrast (Hardest-Contrastive)	**66.2** (+4.0)	**79.0** (+1.1)	**85.7** (+0.6)
PointContrast (PointInfoNCE)	**65.8** (+3.6)	**78.8** (+0.9)	**85.7** (+0.6)
using 100% training data	10 tail classes	30 tail classes	all 55 classes
Train from scratch	65.0	70.9	85.1
PointContrast (Hardest-Contrastive)	**70.9** (+5.9)	**72.9** (+2.0)	**85.7** (+0.6)
PointContrast (PointInfoNCE)	**67.8** (+2.8)	**72.0** (+1.1)	**85.7** (+0.6)

Table 2: **ShapeNet classification.** Top: classification accuracy with limited labeled training data for finetuning. Bottom: classification accuracy on the least represented classes in the data (tail-classes). In all cases, PointContrast boosts performance. Relative improvement increases with scarcer training data and on less frequent classes.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

methods	IoU (1% data)	IoU (5% data)	IoU (100% data)
SO-Net[36]	64.0	69.0	-
PointCapsNet[85]	67.0	70.0	-
Multitask Unsupervised[22]	68.2	77.7	-
Train from scratch	71.8	79.3	84.7
PointConstrast (Hardest-Contrastive)	**74.0** (+2.2)	**79.9** (+0.6)	**85.1** (+0.4)
PointConstrast (PointInfoNCE)	**73.1** (+1.3)	**79.9** (+0.6)	**85.1** (+0.4)

Table 3: **ShapeNet part segmentation.** Replacing the backbone architecture with SR-UNet already boosts performance. PointContrast pre-training further adds a significant gain, and outshines where labels are most limited.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

methods	mIoU	mAcc
PointNet [47]	41.1	49.0
PointCNN [37]	57.3	63.9
MinkowskiNet32 [9]	65.4	71.7
Train from scratch	68.2	75.5
PointConstrast (Hardest-Contrastive)	**70.9** (+2.7)	**77.0** (+1.5)
PointConstrast (PointInfoNCE)	70.3 (+2.1)	76.9 (+1.4)

Table 4: **Stanford Area 5 Test (Fold 1) (S3DIS).** Replacing the backbone network with SR-UNet improves upon prior art. Using PointContrast adds further significant boost with a mild preference for Hardest-contrastive over the PointInfoNCE objective. See Appendix for more methods in comparison.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

methods	input	mAP@0.5	mAP@0.25
VoteNet [45]	Geo	-	57.0
VoteNet [45]	Geo+Height	32.9	57.7
Train from scratch	Geo	31.7	55.6
PointContrast(Hardest-Contrastive)	Geo	34.5 (+2.8)	**57.5** (+1.9)
PointContrast(PointInfoNCE)	Geo	**34.8** (+3.1)	**57.5** (+1.9)

Table 5: **SUN RGB-D detection results.** PointContrast demonstrates a substantial boost compared to training from scratch. We observe a larger improvement in localization as manifested by the ΔmAP being larger for @0.5 than @0.25.

methods	mIoU	mAcc
MinkowskiNet32 [9]	78.7	91.5
Train from scratch	79.8	91.5
PointContrast (Hardest-Contrastive)	82.6 (+2.8)	**93.7** (+2.2)
PointContrast (PointInfoNCE)	**83.1** (+3.3)	**93.7** (+2.2)

Table 6: **Segmentation results on the 4D Synthia test set.** All networks here are SR-UNet with 3D kernels, trained on individual 3D frames without temporal modeling.

PointContrast

methods	mIoU	mAcc
Train from scratch	72.2	80.7
PointContrast(Hardest-Contrastive)	73.3 (+1.1)	81.0 (+0.3)
PointContrast(PointInfoNCE)	**74.1** (+1.9)	**81.6** (+0.9)

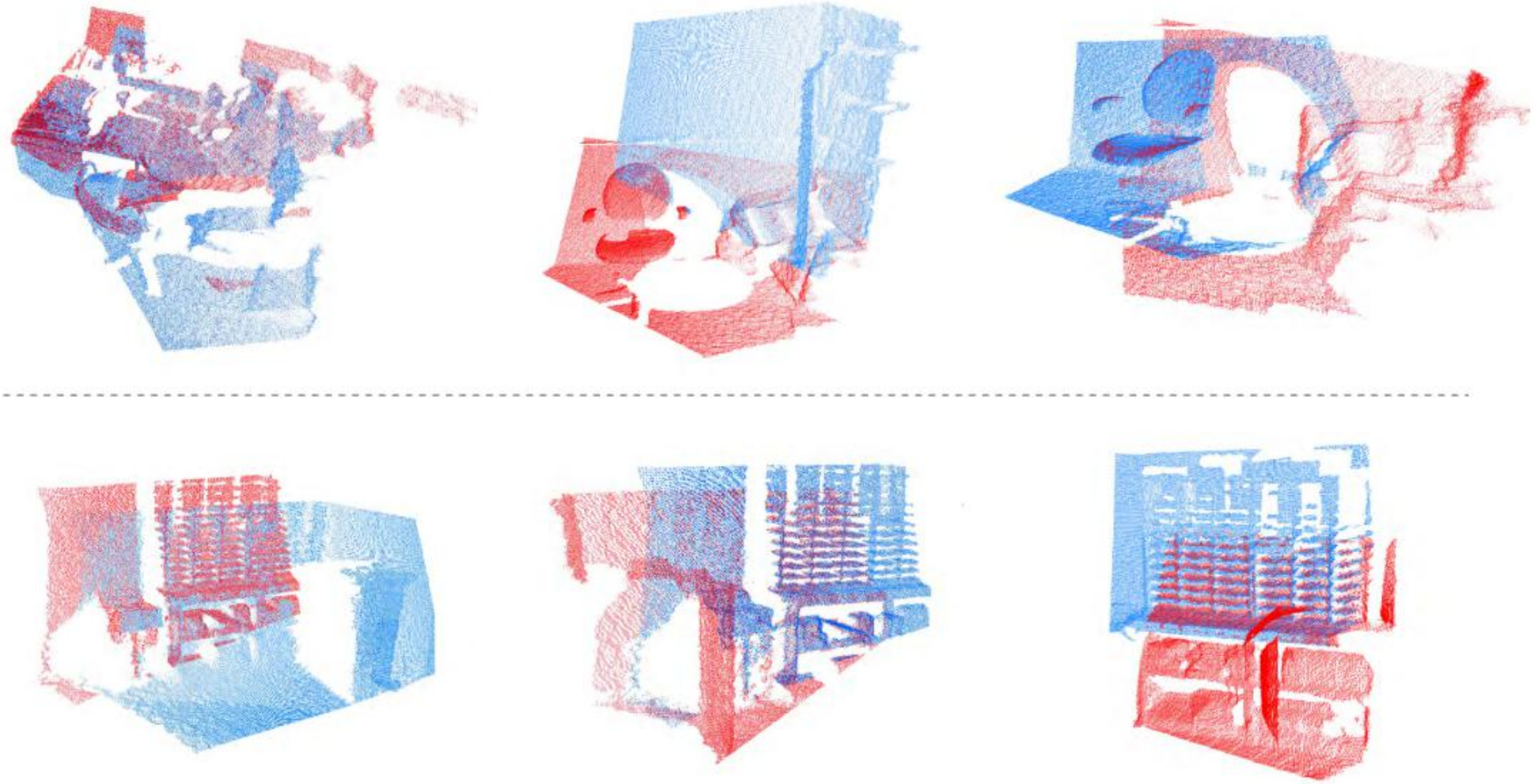
Table 7: **Segmentation results on ScanNet validation set.** PointContrast boosts performance on the "in-domain" transfer task where the pre-training and fine-tuning datasets come from a common source, showing the usefulness of pre-training even when labels are available.

methods	input	mAP@0.5	mAP@0.25
DSS [58, 29]	Geo+RGB	6.8	15.2
3D-SIS [29]	Geo+RGB (5 Views)	22.5	40.2
VoteNet [45]	Geo+Height	33.5	58.6
Train from scratch	Geo	35.4	56.7
PointContrast(Hardest-Contrastive)	Geo	37.3 (+1.9)	**59.2** (+2.5)
PointContrast(PointInfoNCE)	Geo	**38.0** (+2.6)	58.5 (+1.8)

Table 8: **3D object detection results on ScanNet validation set.** Similarly to in-domain *segmentation* task, here as well PointContrast boost performance on *detection*, setting a new best result over prior art. See Appendix for more methods in comparison.

PointContrast

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen



Closer Look

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

A Closer Look at Local Aggregation Operators in Point Cloud Analysis

Ze Liu[1,2⋆†], Han Hu[2⋆], Yue Cao[2], Zheng Zhang[2], and Xin Tong[2]

[1] University of Science and Technology of China
liuze@mail.ustc.edu.cn
[2] Microsoft Research Asia
{hanhu,yuecao,zhez,xtong}@microsoft.com

https://github.com/zeliu98/CloserLook3D

Closer Look

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

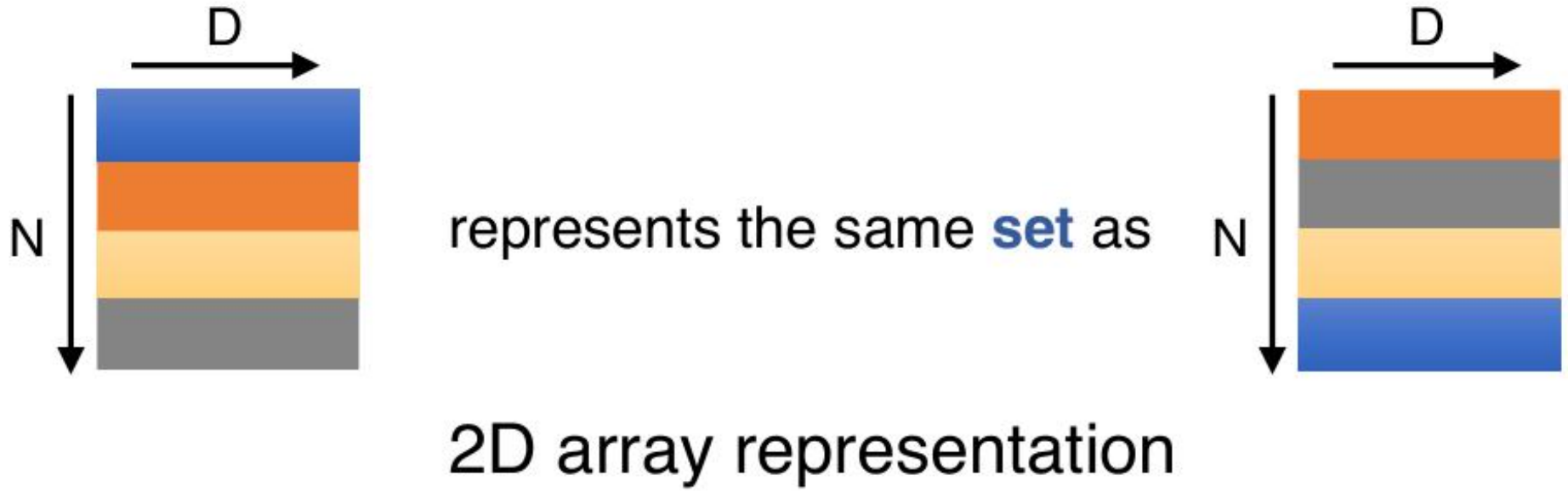
- The investigation reveals that despite the different designs of these operators, all of these operators **make surprisingly similar contributions** to the network performance under the same network input and feature numbers and result in the state-of-the-art accuracy on standard benchmarks.
- Propose a simple local aggregation operator without learnable weights, named **Position Pooling (PosPool)**, which performs similarly or slightly better than existing sophisticated operators.

Closer Look

The model has respect properties of point clouds

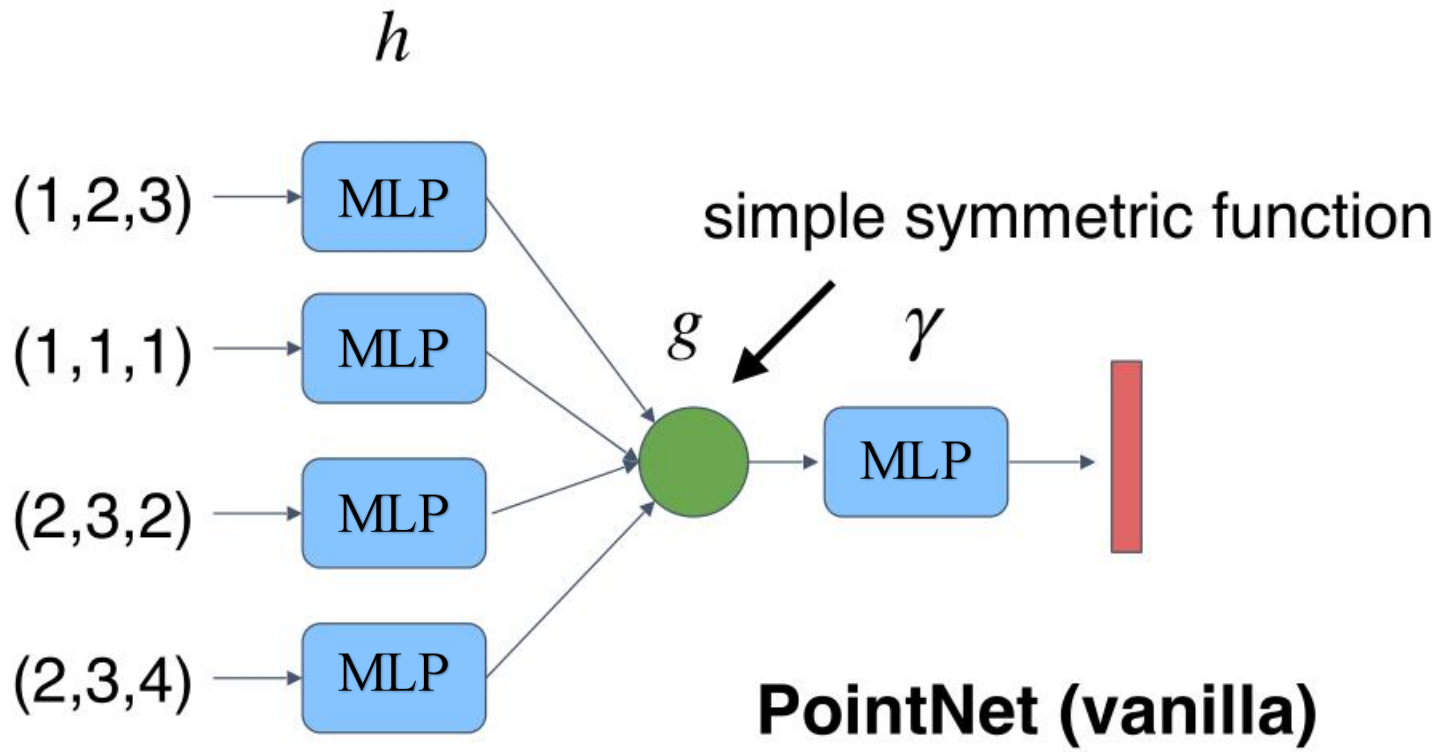
Permutation Invariance

- Point cloud is a set of unordered points



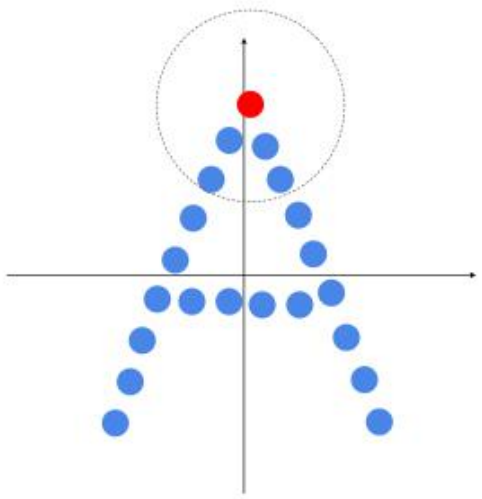
Closer Look

Symmetric function

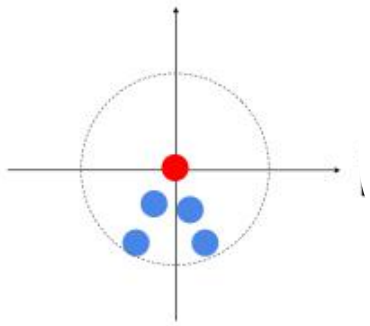


Observe: $f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric

Closer Look

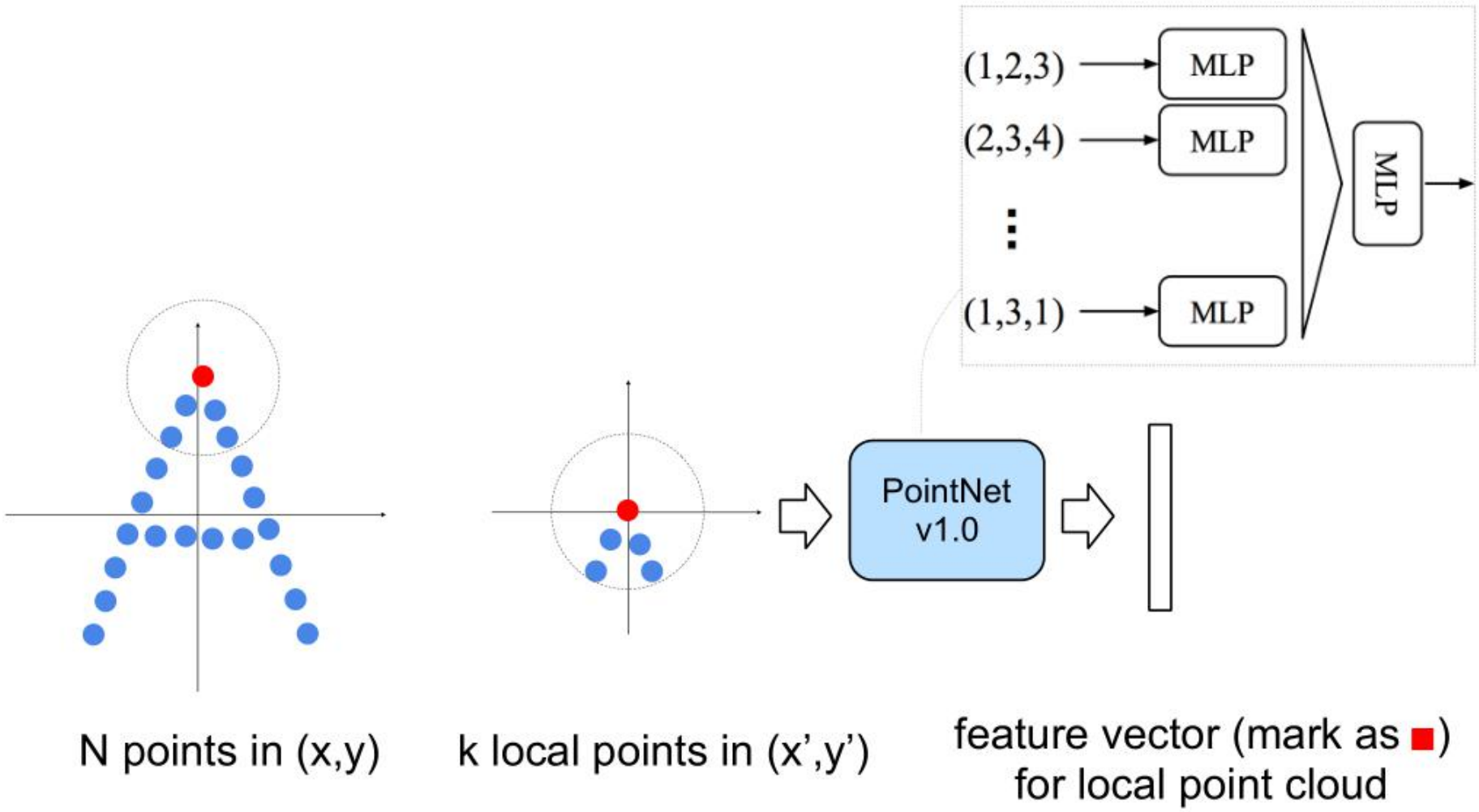


N points in (x,y)

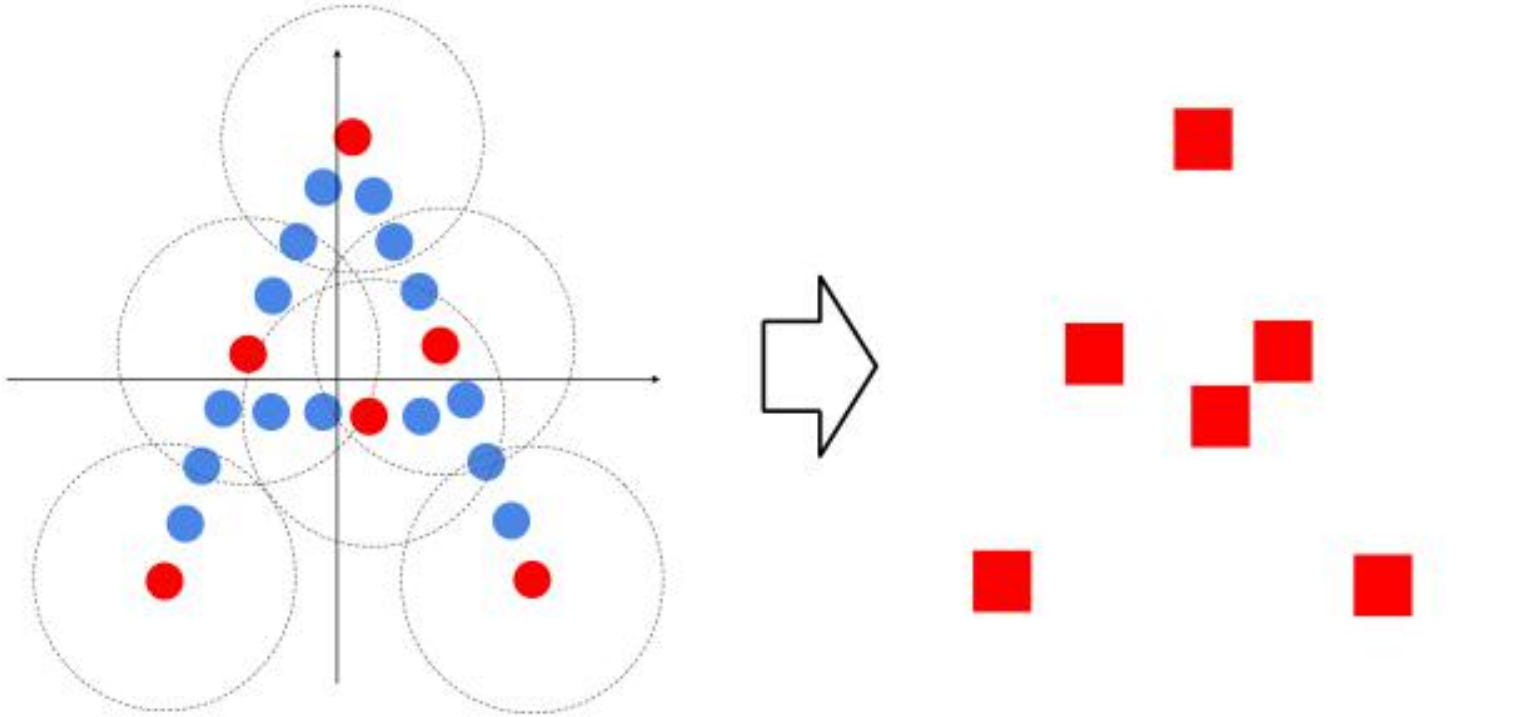


k local points in (x',y')

Closer Look


(1,2,3)
MLP
(2,3,4)
MLP
(1,3,1)
MLP
MLP
PointNet
v1.0
N points in (x,y)
k local points in (x',y')
feature vector (mark as ■)
for local point cloud

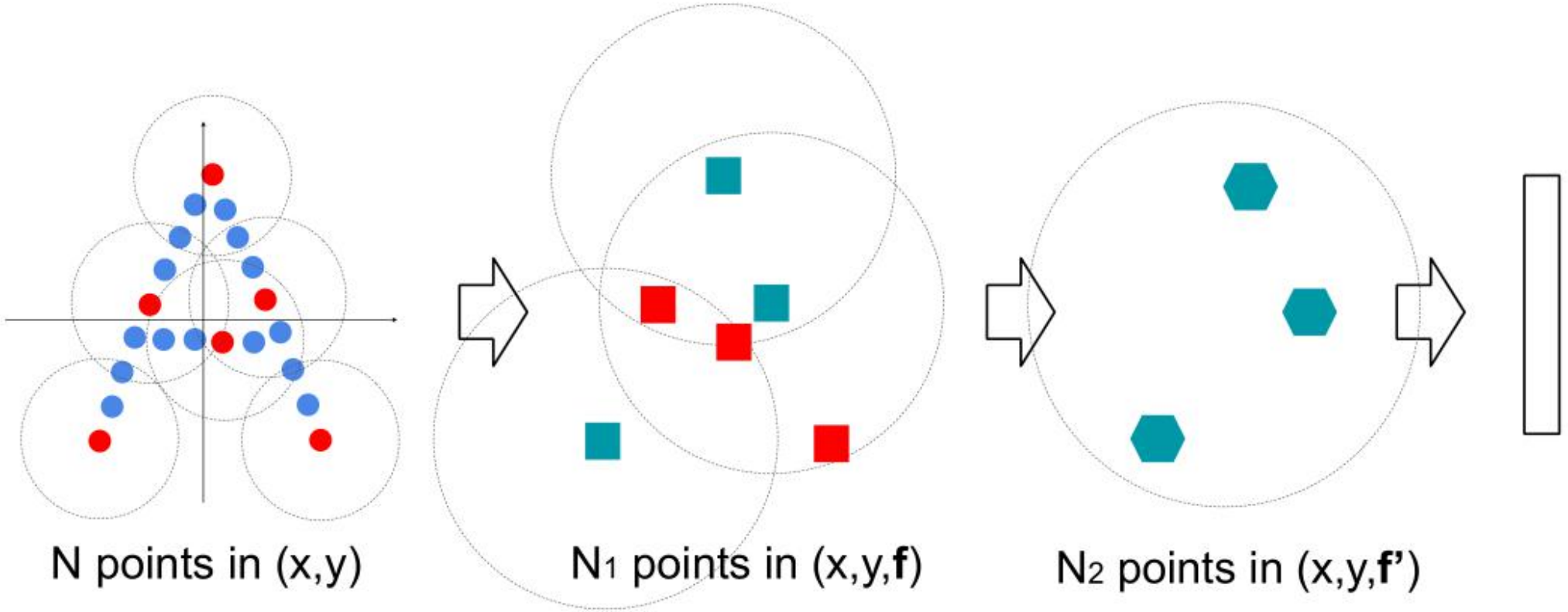
Closer Look



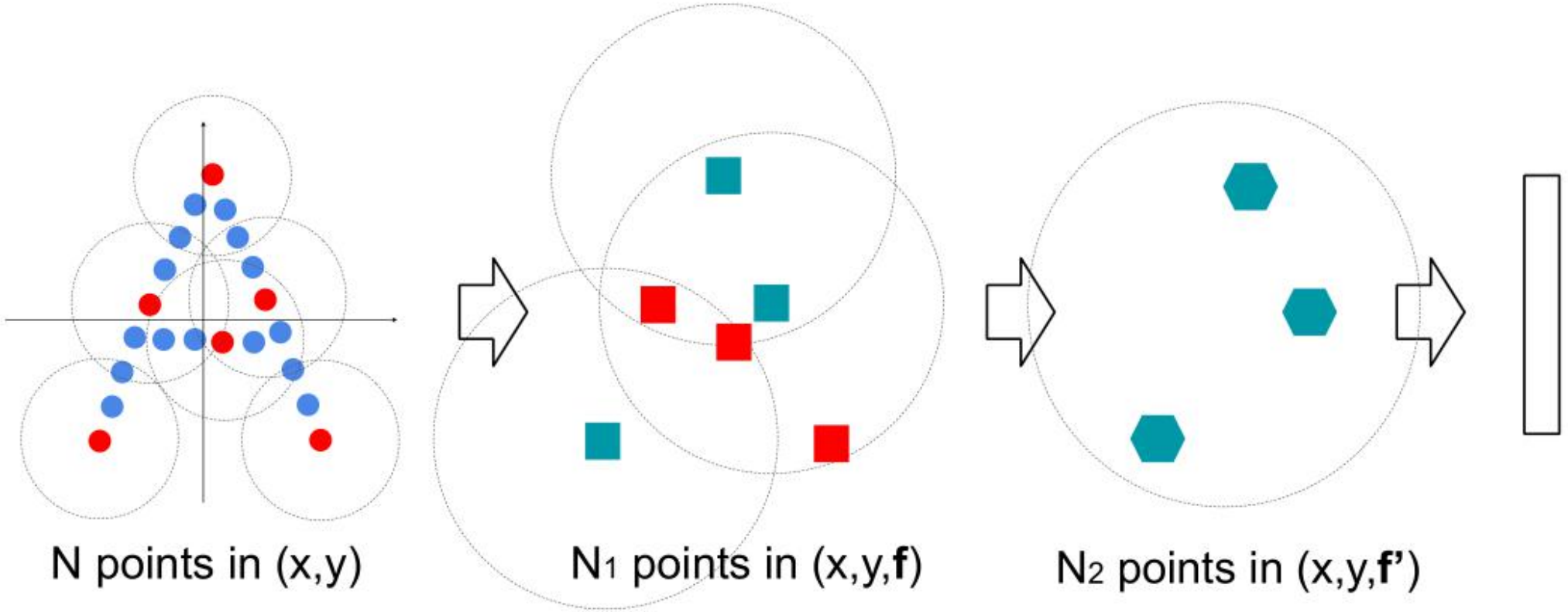
N points in (x,y)

N_1 points in (x,y,**f**)

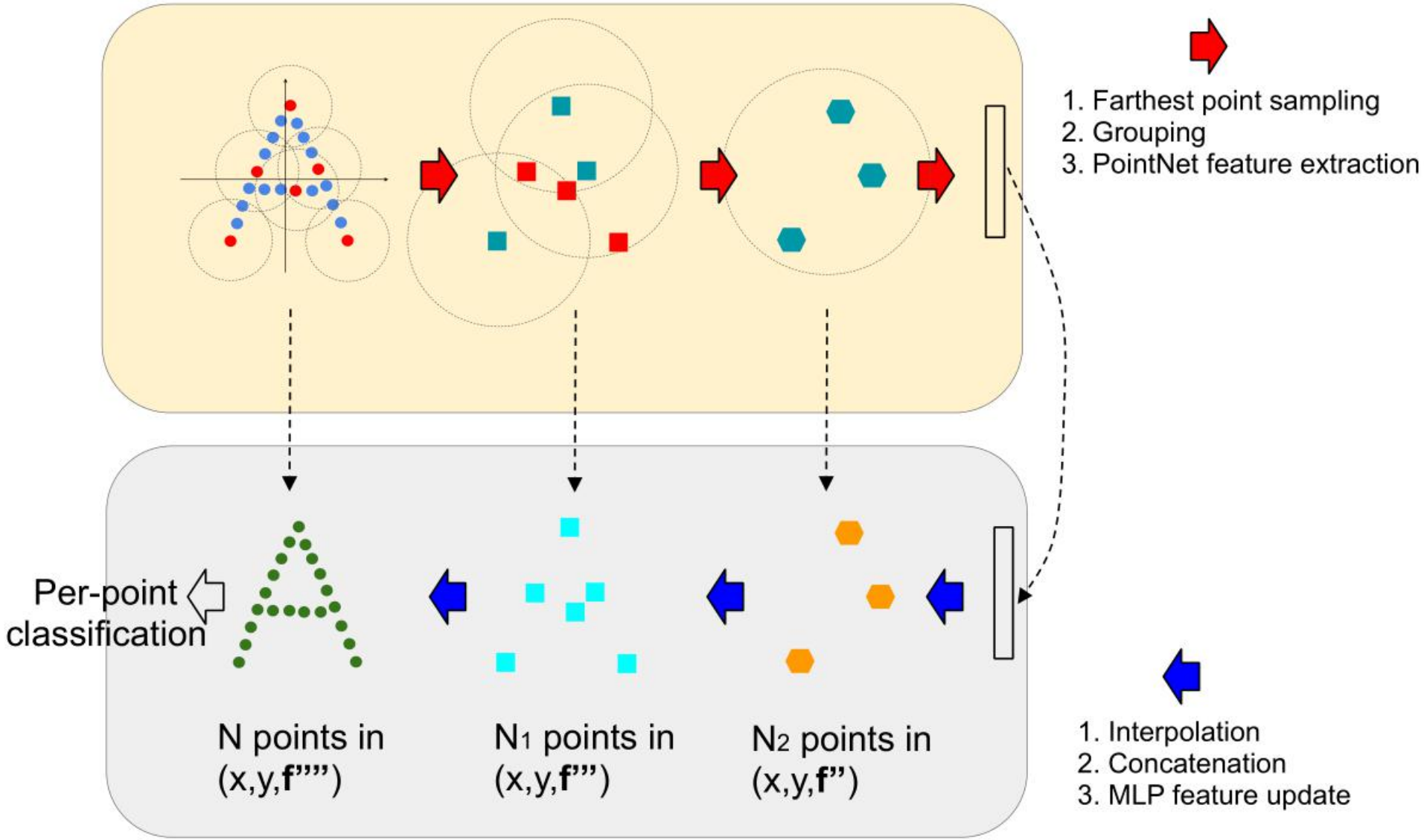
Closer Look



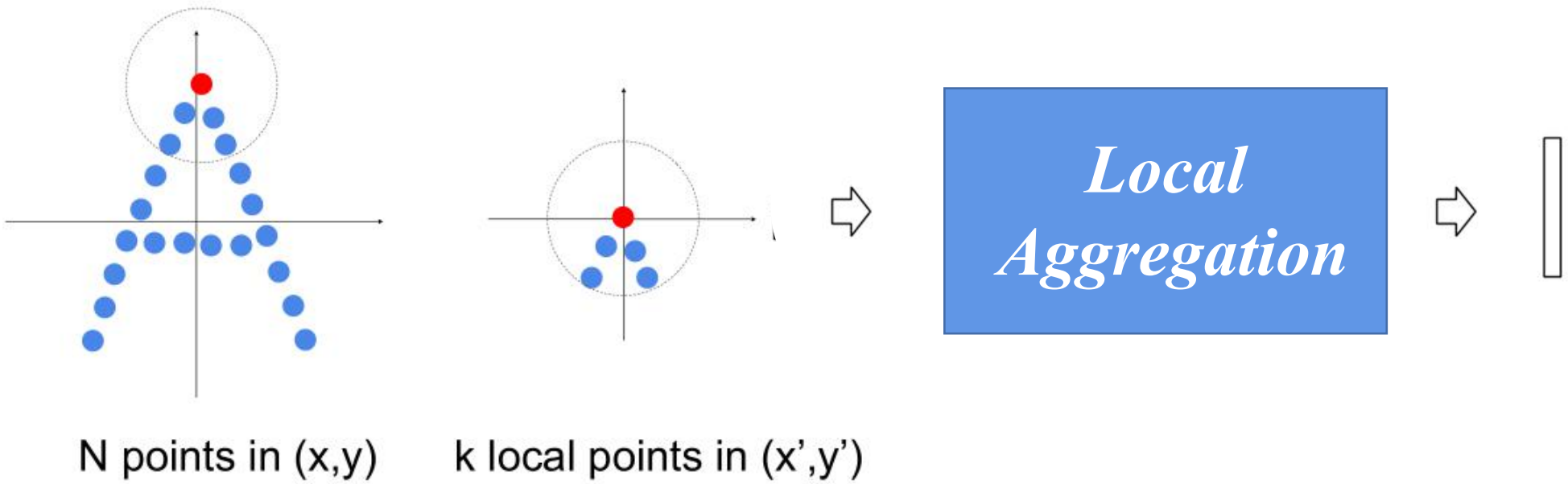
Closer Look



Closer Look


1. Farthest point sampling
2. Grouping
3. PointNet feature extraction
Per-point classification
N points in (x,y,f'''')
N1 points in (x,y,f''')
N2 points in (x,y,f'')
1. Interpolation
2. Concatenation
3. MLP feature update

Closer Look



Closer Look

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

- Point-wise multi-layer perceptions (MLP) based
- Adaptive weight based
- Pseudo grid feature based

Closer Look

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

- Point-wise multi-layer perceptions (MLP) based

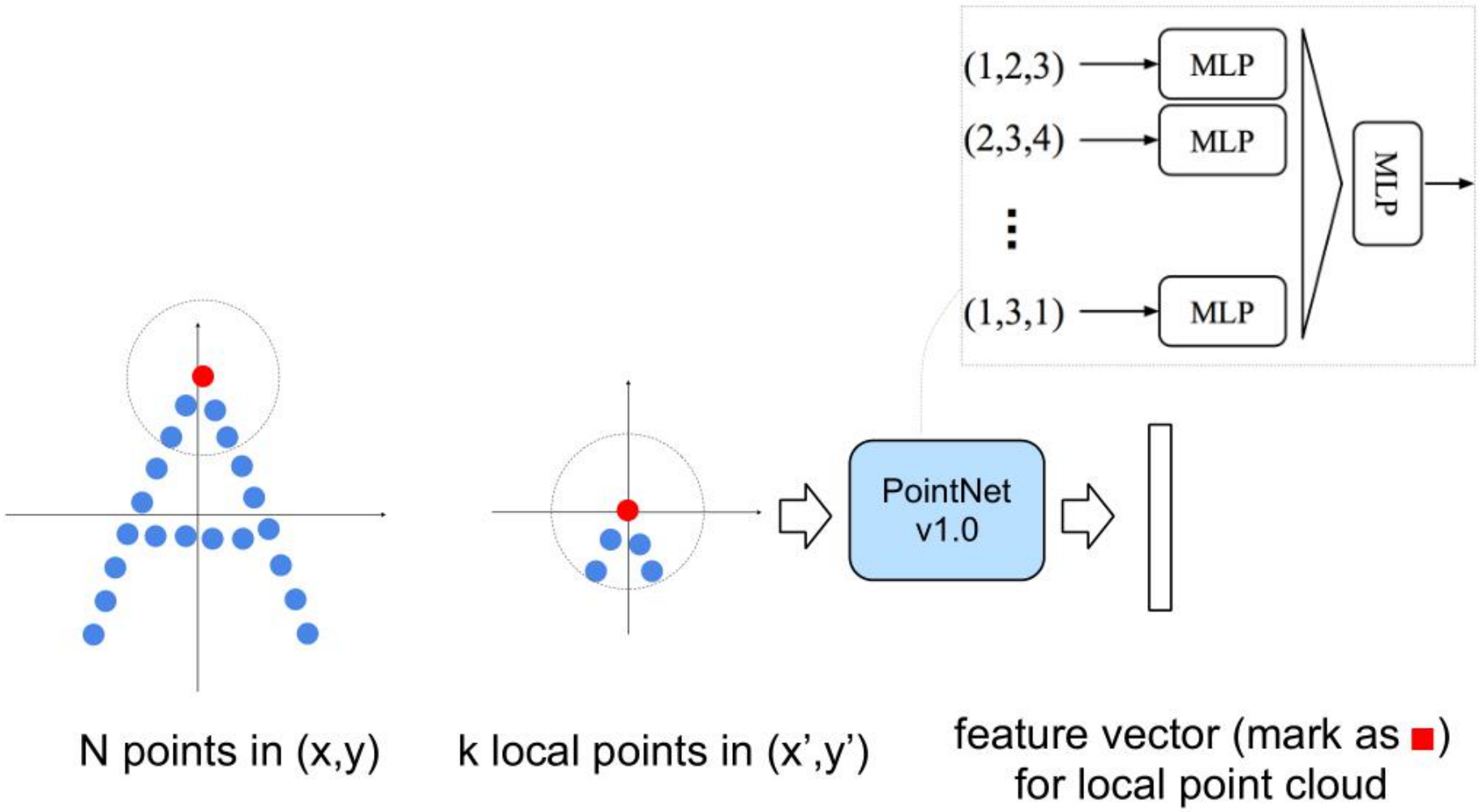
General Formulation In general, for each point i, a local aggregation layer first transforms a neighbor point j's feature $\mathbf{f}_j \in \mathbb{R}^{d\times 1}$ and its relative location $\Delta\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i \in \mathbb{R}^{3\times 1}$ into a new feature by a function $G(\cdot,\cdot)$, and then aggregate all transformed neighborhood features to form point i's output feature by a reduction function R (typically using MAX, AVG or SUM), as

$$\mathbf{g}_i = R\left(\{G(\Delta\mathbf{p}_{ij}, \mathbf{f}_j) | j \in \mathcal{N}(i)\}\right), \tag{1}$$

where $\mathcal{N}(i)$ represents the neighborhood of point i. Alternatively, edge features $\{\mathbf{f}_i, \Delta\mathbf{f}_{ij}\}$ ($\Delta\mathbf{f}_{ij} = \mathbf{f}_j - \mathbf{f}_i$) can be used as input instead of $\Delta\mathbf{p}_{ij}$ [35].

$$G(\Delta\mathbf{p}_{ij}, \mathbf{f}_j) = \mathrm{MLP}\left(\mathrm{concat}(\Delta\mathbf{p}_{ij}, \mathbf{f}_j)\right). \tag{2}$$

PointNet2


(1,2,3)
MLP
(2,3,4)
MLP
MLP
(1,3,1)
MLP
PointNet
v1.0
N points in (x,y)
k local points in (x',y')
feature vector (mark as ■)
for local point cloud

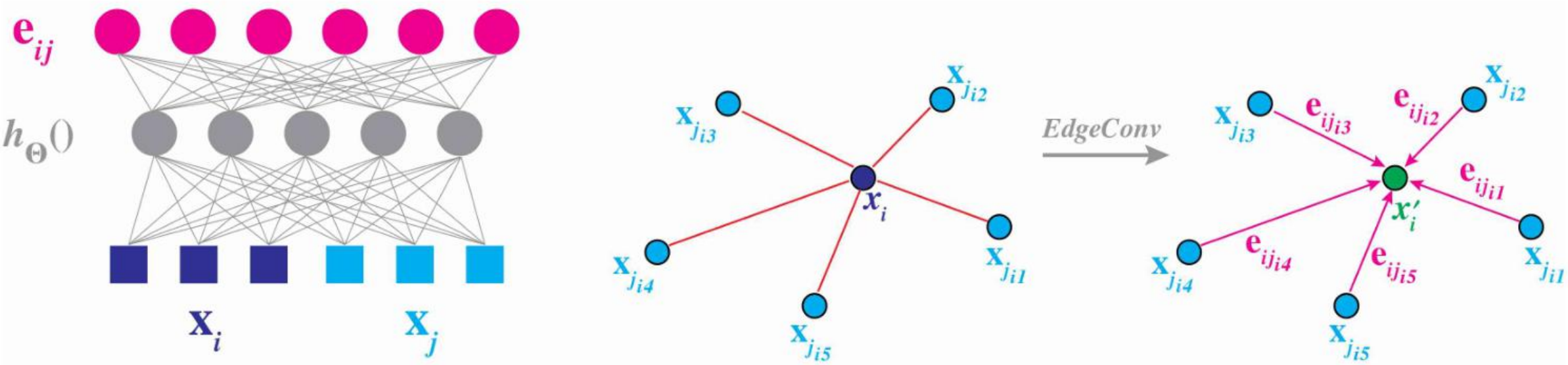
DGCNN

Graph Convolution for local neighborhoods

Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs

Martin Simonovsky
Université Paris Est, École des Ponts ParisTech
martin.simonovsky@enpc.fr

Nikos Komodakis
Université Paris Est, École des Ponts ParisTech
nikos.komodakis@enpc.fr



Closer Look

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

- Adaptive weight based

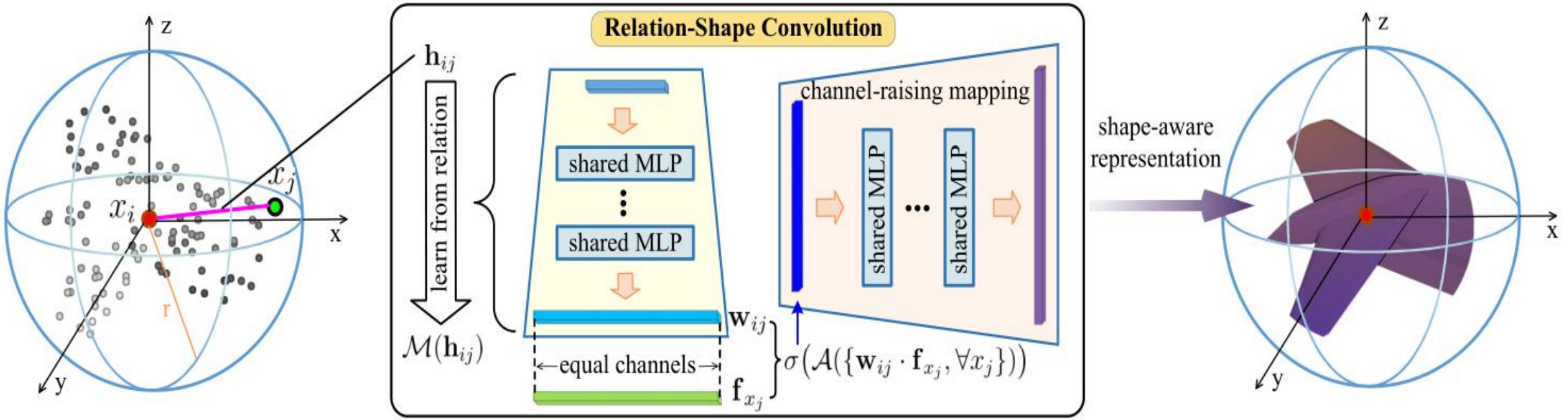
Adaptive Weight based Methods The adaptive weight based methods define convolution filters over arbitrary relative positions, and hence can compute aggregation weights on all neighbor points:

$$G(\Delta \mathbf{p}_{ij}, \mathbf{f}_j) = H\left(\Delta \mathbf{p}_{ij}\right) \odot \mathbf{f}_j, \tag{5}$$

RS-CNN

Relation-Shape Convolutional Neural Network for Point Cloud Analysis

Yongcheng Liu[†‡] Bin Fan[*†] Shiming Xiang[†‡] Chunhong Pan[†]
[†] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[‡] School of Artificial Intelligence, University of Chinese Academy of Sciences
Email: {yongcheng.liu, bfan, smxiang, chpan}@nlpr.ia.ac.cn



Results with different relations

model	low-level relation $\mathbf{h}$	channels	acc.
A	(3D-Ed)	1	92.5
B	(3D-Ed, $x_i - x_j$)	4	93.0
C	(3D-Ed, $x_i - x_j, x_i, x_j$)	10	**93.6**
D	(3D-cosd, $x_i^{\text{nor}}, x_j^{\text{nor}}$)	7	92.8
E	(2D-Ed, $x'_i - x'_j, x'_i, x'_j$)	10	≈ 92.2

Closer Look

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

- Pseudo grid feature based

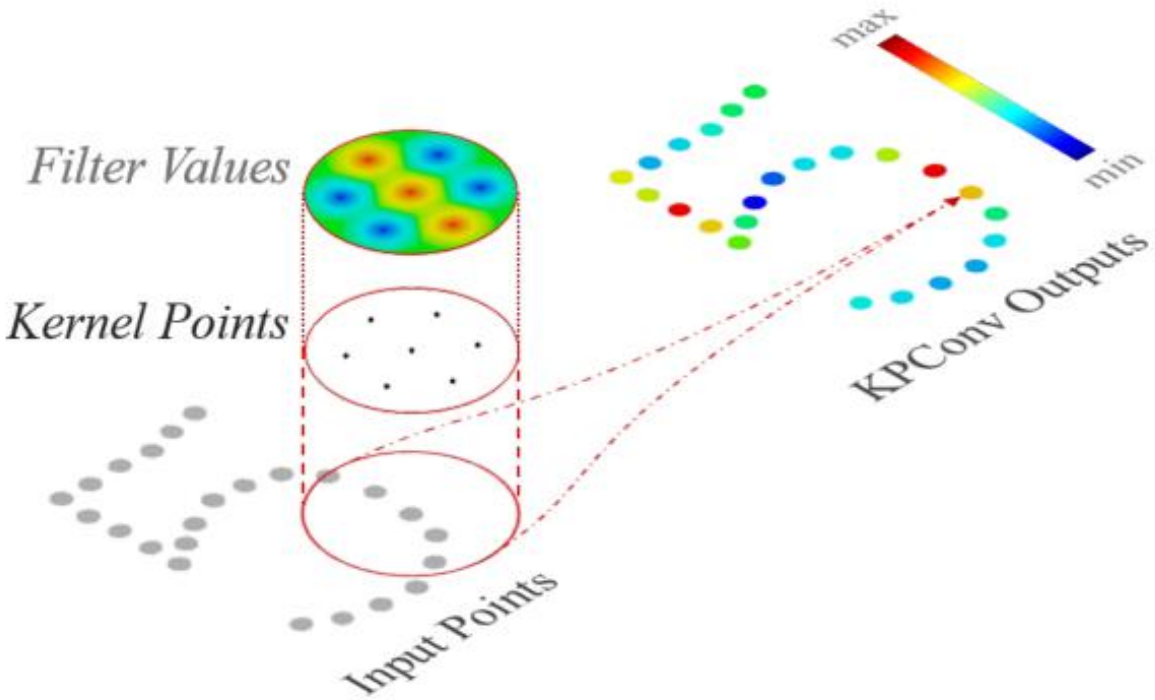
Pseudo Grid Feature based Methods The pseudo grid feature based methods generate pseudo features on several sampled regular grid points, such that regular convolution methods can be applied. A representative method is KPConv [30], where equally distributed spherical grid points are sampled and the pseudo features on the k^{th} grid point is computed as

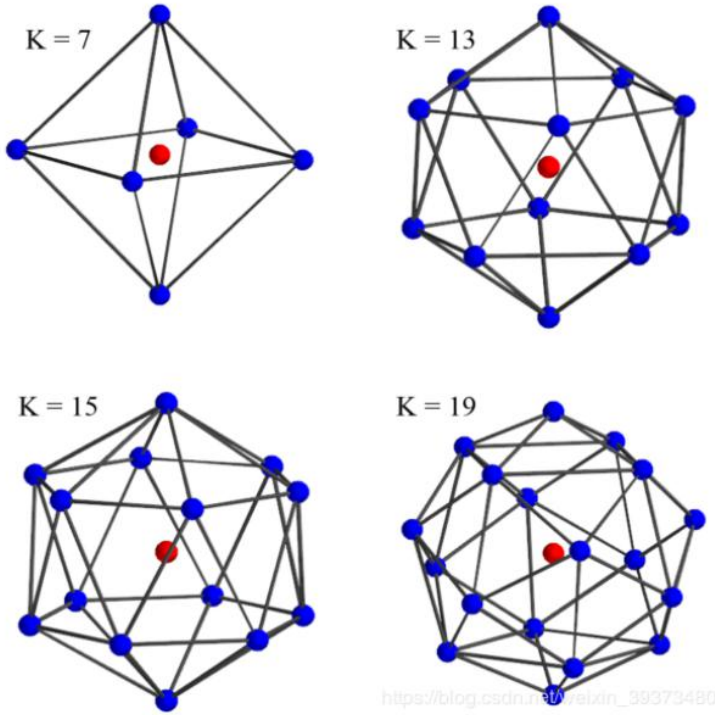
$$\mathbf{f}_{i,k} = \sum_{j \in \mathcal{N}(i)} \max(0, 1 - \frac{\|\Delta \mathbf{p}_{jk}\|_2}{\sigma}) \cdot \mathbf{f}_j. \tag{3}$$

The index of each grid point k will have strict mapping with the relative position to center point $\Delta \mathbf{p}_{ik}$. Hence, a (depth-wise) convolution operator with parametrized weights $\mathbf{w}_k \in \mathbb{R}^{d \times 1}$ defined on each grid point can be used to achieve feature transformation:

$$G(\Delta \mathbf{p}_{ik}, \mathbf{f}_{i,k}) = \mathbf{w}_k \odot \mathbf{f}_{i,k}. \tag{4}$$

KPConv


max
min
Filter Values
Kernel Points
KPConv Outputs
Input Points


K = 7
K = 13
K = 15
K = 19

KPConv

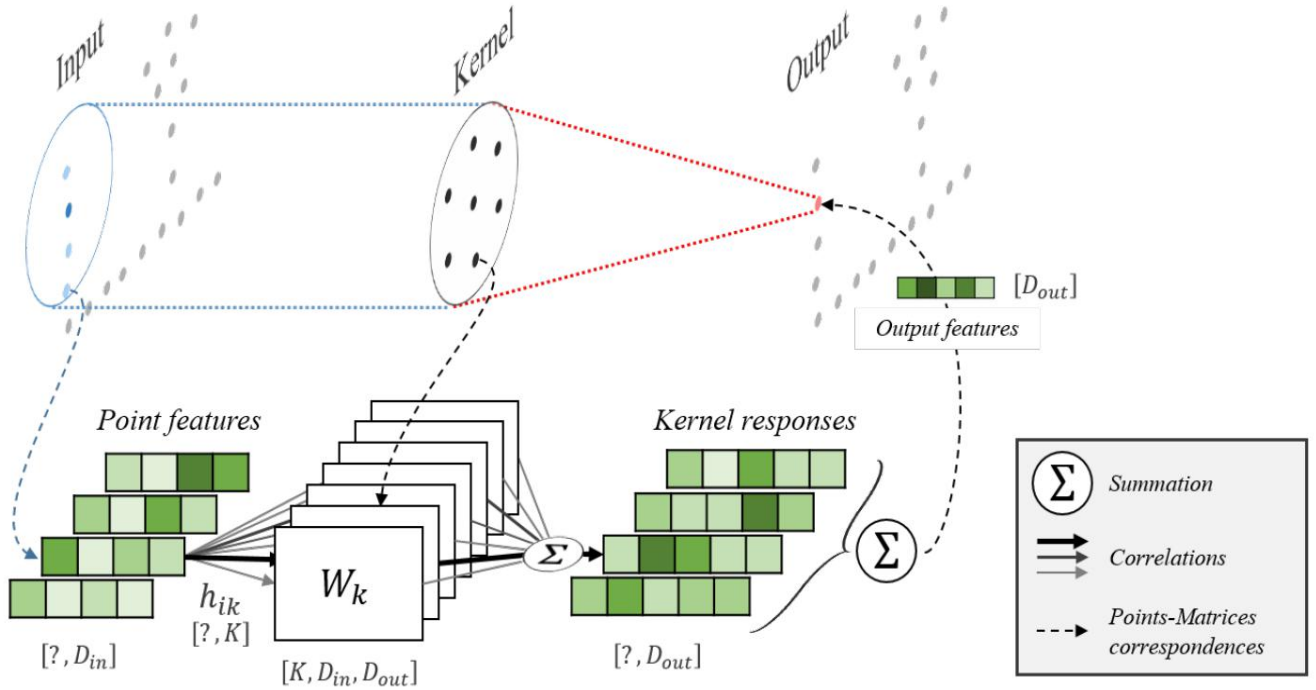
The general point convolution of F by a kernel g

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i$$

Function g for KPCONV

$$g(y_i) = \sum_{k<K} h\left(y_i, \widetilde{x}_k\right) W_k$$

$$h\left(y_i, \widetilde{x}_k\right) = \max\left(0, 1 - \frac{\|y_i - \widetilde{x}_k\|}{\sigma}\right)$$



KPConv

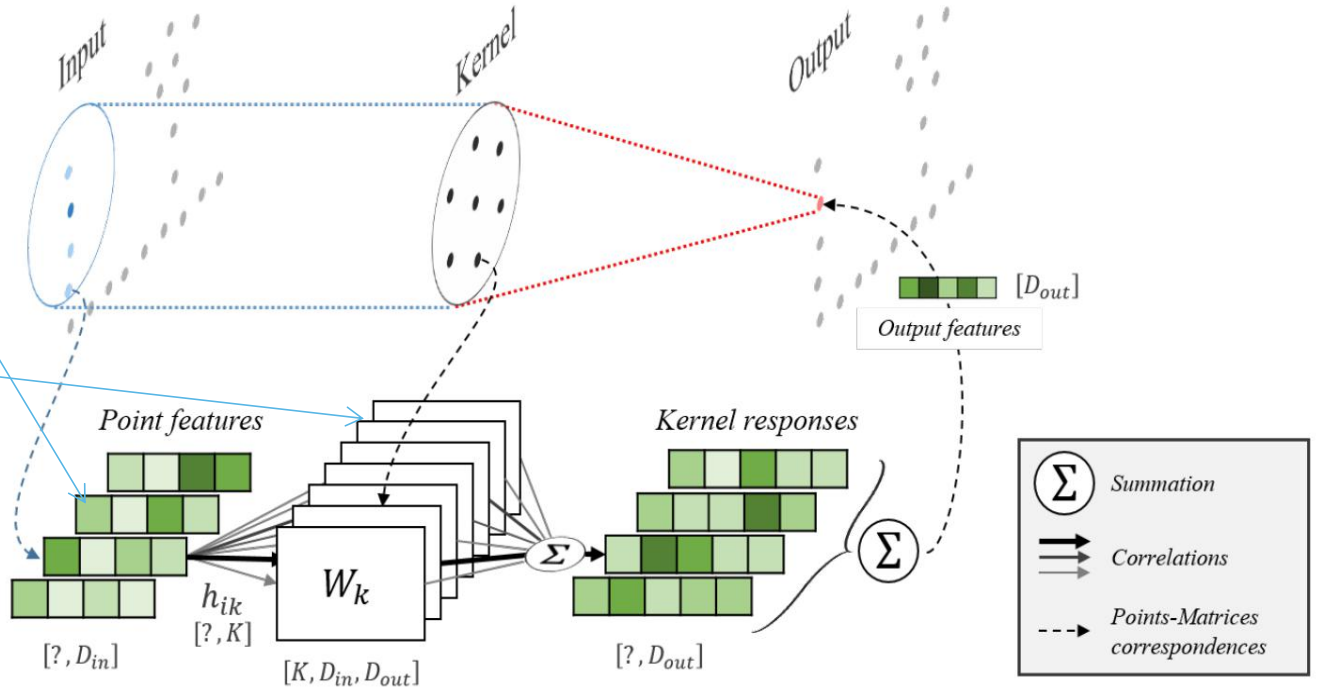
The general point convolution of F by a kernel g

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i$$

Function g for KPCONV

$$g(y_i) = \sum_{k<K} h\left(y_i, \widetilde{x}_k\right) W_k$$

$$h\left(y_i, \widetilde{x}_k\right) = \max\left(0, 1 - \frac{\|y_i - \widetilde{x}_k\|}{\sigma}\right)$$



PosPool

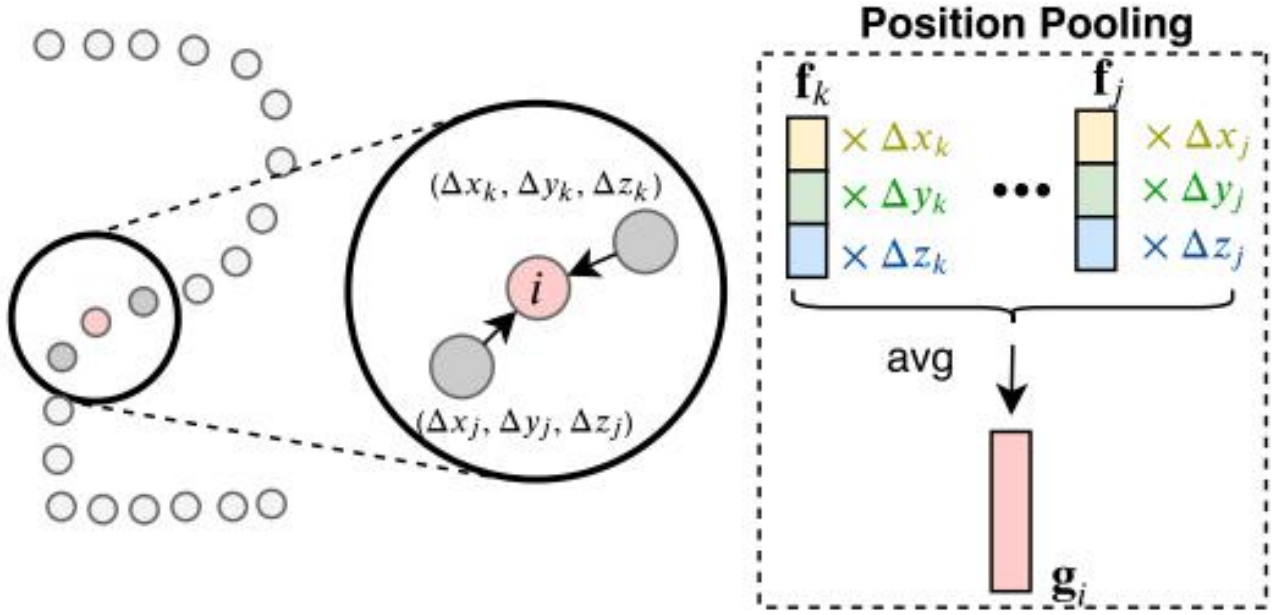


Fig. 2. Illustration of the proposed position pooling (PosPool) operator.

$$G(\Delta \mathbf{p}_{ij}, \mathbf{f}_j) = \text{Concat}\left[\Delta x_{ij}\mathbf{f}_j^0; \Delta y_{ij}\mathbf{f}_j^1; \Delta z_{ij}\mathbf{f}_j^2\right],$$

Closer Look

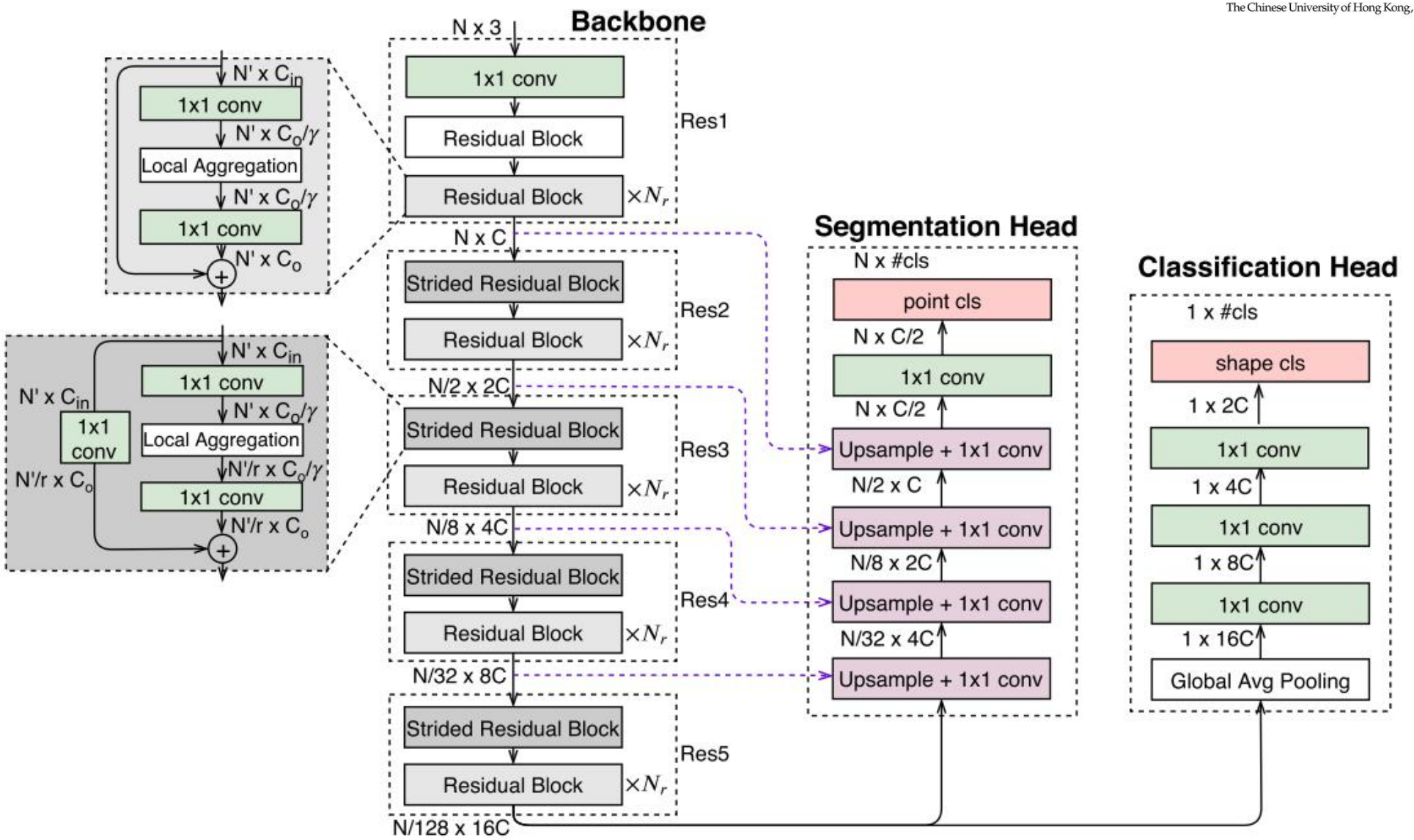

Backbone
N x 3
1x1 conv
Residual Block
Residual Block
×N_r
Res1
N x C
Strided Residual Block
Residual Block
×N_r
Res2
N/2 x 2C
Strided Residual Block
Residual Block
×N_r
Res3
N/8 x 4C
Strided Residual Block
Residual Block
×N_r
Res4
N/32 x 8C
Strided Residual Block
Residual Block
×N_r
Res5
N/128 x 16C
N' x C_{in}
1x1 conv
N' x C_o/γ
Local Aggregation
N' x C_o/γ
1x1 conv
N' x C_o
N' x C_{in}
1x1 conv
N' x C_{in}
1x1 conv
N' x C_o/γ
Local Aggregation
N'/r x C_o/γ
1x1 conv
N'/r x C_o
N'/r x C_o
Segmentation Head
N x #cls
point cls
N x C/2
1x1 conv
N x C/2
Upsample + 1x1 conv
N/2 x C
Upsample + 1x1 conv
N/8 x 2C
Upsample + 1x1 conv
N/32 x 4C
Upsample + 1x1 conv
Classification Head
1 x #cls
shape cls
1 x 2C
1x1 conv
1 x 4C
1x1 conv
1 x 8C
1x1 conv
1 x 16C
Global Avg Pooling

Table 1. The performance of baseline operators, sweet spots of point-wise MLP based, pseudo grid feature based and adaptive weight based operators, and the proposed PosPool operators on three benchmark datasets. Baseline* denotes Eq. (6) and baseline† (AVG/MAX) denotes Eq. (7) AVG/MAX, respectively. PosPool and PosPool* denote the operators in Eq. (8) and (10), respectively. (S) after each method denotes a smaller configuration of this method ($N_r = 1$, $\gamma = 2$ and $C = 36$), which is about 16× more efficient than the regular configuration (the other row) of $N_r = 1$, $\gamma = 2$ and $C = 144$. Previous best performing methods on three benchmarks in literature are shown in the first block of this table

method	ModelNet40			S3DIS			PartNet			
	acc	param	FLOP	mIoU	param	FLOP	val	test	param	FLOP
DensePoint [15]	93.2	0.7M	0.7G	-	-	-	-	-	-	-
KPConv [30]	92.9	15.2M	1.7G	65.7	15.0M	6.5G	-	-	-	-
PointCNN [14]	92.5	0.6M	25.3G	65.4	4.4M	36.7G	-	46.4	4.4M	23.1G
baseline*	91.4	19.4M	1.8G	51.5	18.4M	7.2G	42.5	44.6	18.5M	6.7G
baseline† (AVG, S)	90.7	1.2M	0.1G	50.3	1.1M	0.5G	39.5	40.6	1.1M	0.4G
baseline† (AVG)	91.4	19.4M	1.8G	51.0	18.4M	7.2G	44.2	45.8	18.5M	6.7G
baseline† (MAX, S)	91.5	1.2M	0.1G	57.4	1.1M	0.5G	39.8	41.2	1.1M	0.4G
baseline† (MAX)	91.8	19.4M	1.8G	58.4	18.4M	7.2G	45.4	47.4	18.5M	6.7G
point-wise MLP (S)	92.6	1.7M	0.2G	56.7	1.6M	0.8G	45.3	47.0	1.6M	0.7G
point-wise MLP	92.8	26.5M	2.7G	66.2	25.5M	9.8G	48.1	51.5	25.6M	9.1G
pseudo grid (S)	92.3	1.2M	0.3G	64.3	1.2M	1.0G	44.2	45.2	1.2M	0.9G
pseudo grid	93.0	19.5M	2.0G	65.9	18.5M	9.3G	50.8	53.0	18.5M	8.5G
adapt weights (S)	92.1	1.2M	0.2G	61.9	1.2M	0.6G	44.1	46.1	1.2M	0.5G
adapt weights	93.0	19.4M	2.3G	66.5	18.4M	7.8G	50.1	53.5	18.5M	7.2G
PosPool (S)	92.5	1.2M	0.1G	64.2	1.1M	0.5G	44.6	47.2	1.1M	0.5G
PosPool	92.9	19.4M	1.8G	66.5	18.4M	7.3G	50.0	53.4	18.5M	6.8G
PosPool* (S)	92.6	1.2M	0.1G	61.3	1.1M	0.5G	46.1	47.2	1.1M	0.5G
PosPool*	93.2	19.4M	1.8G	66.7	18.4M	7.3G	50.6	53.8	18.5M	6.8G

Table 2. Evaluating different settings of the point-wise MLP method. The option ∇, $\triangle$, $\square$ and $\diamond$ denote input features using $\{\Delta\mathbf{p}_{ij}, \mathbf{f}_j\}$, $\{\mathbf{f}_i, \Delta\mathbf{f}_{ij}\}$, $\{\Delta\mathbf{p}_{ij}, \mathbf{f}_i, \Delta\mathbf{f}_{ij}\}$, and $\{\Delta\mathbf{p}_{ij}, \mathbf{f}_i, \mathbf{f}_j, \Delta\mathbf{f}_{ij}\}$, respectively. "Sweet spot" denotes balanced settings regarding both efficacy and efficiency. The accuracy on PartNet test set is not tested in ablations to avoid the tuning of test set

method	γ	input ∇	input $\triangle$	input $\square$	input $\diamond$	#FC	$R(\cdot)$	ModelNet40	S3DIS	PartNet (val/test)
PointNet++ [22]	-	✓				3	MAX	90.7	-	-/42.5
PointNet++*	-	✓				3	MAX	91.6	55.3	43.1/45.3
sweet spot	8		✓			1	MAX	92.8	62.9	48.2/50.8
	2		✓			1	MAX	92.8	66.2	48.1/51.2
FC num	8		✓			2	MAX	92.5	59.5	47.9/-
	8		✓			3	MAX	92.0	59.9	48.7/-
input	8	✓				1	MAX	92.6	59.8	47.1/-
	8		✓			1	MAX	92.5	61.4	47.6/-
	8				✓	1	MAX	92.7	51.0	47.9/-
reduction $R(\cdot)$	8		✓			1	AVG	92.3	55.1	46.8/-
	8		✓			1	SUM	92.2	44.7	46.7/-

Sweet spots for point-wise MLP methods. Regarding both the efficacy and efficiency, the sweet spot settings are applying 1 FC layer to an input combination of relative position and edge features. Table 2 also shows that using $\gamma = 2$ for this method can approach or surpass the state-of-the-art on all three datasets.

Table 3. Evaluating different settings of the adaptive weight based methods. $d\mathbf{p}*$ denotes the 9-dimensional position vector as in [16]. "Sweet spot" denotes balanced settings regarding both efficacy and efficiency. The accuracy on PartNet test set is not tested for ablations to avoid tuning the test set.

method	γ	input				#FC	$R(\cdot)$	S.M.	ModelNet	S3DIS	PartNet (val)
		$\{d\mathbf{p}\}$	$\{d\mathbf{f}\}$	$\{d\mathbf{p}, d\mathbf{f}\}$	$\{d\mathbf{p}^*\}$						
PConv[34]	-	✓				2	SUM		-	58.3	-
FlexConv[5]	-	✓				1	SUM		90.2	56.6	-
sweet spot	8	✓				1	AVG		92.7	62.6	50.0
sweet spot*	2	✓				1	AVG		93.0	66.5	50.1
FC num	8	✓				2	AVG		92.6	61.3	49.9
	8	✓				3	AVG		92.5	58.5	49.6
input	8		✓			1	AVG		85.3	46.6	46.9
	8			✓		1	AVG		82.2	55.7	46.4
	8				✓	1	AVG		92.1	57.0	49.1
reduction	8	✓				1	SUM		92.6	61.7	49.1
	8	✓				1	MAX		92.4	62.3	49.7
SoftMax	8	✓				1	AVG	✓	91.7	55.9	45.8

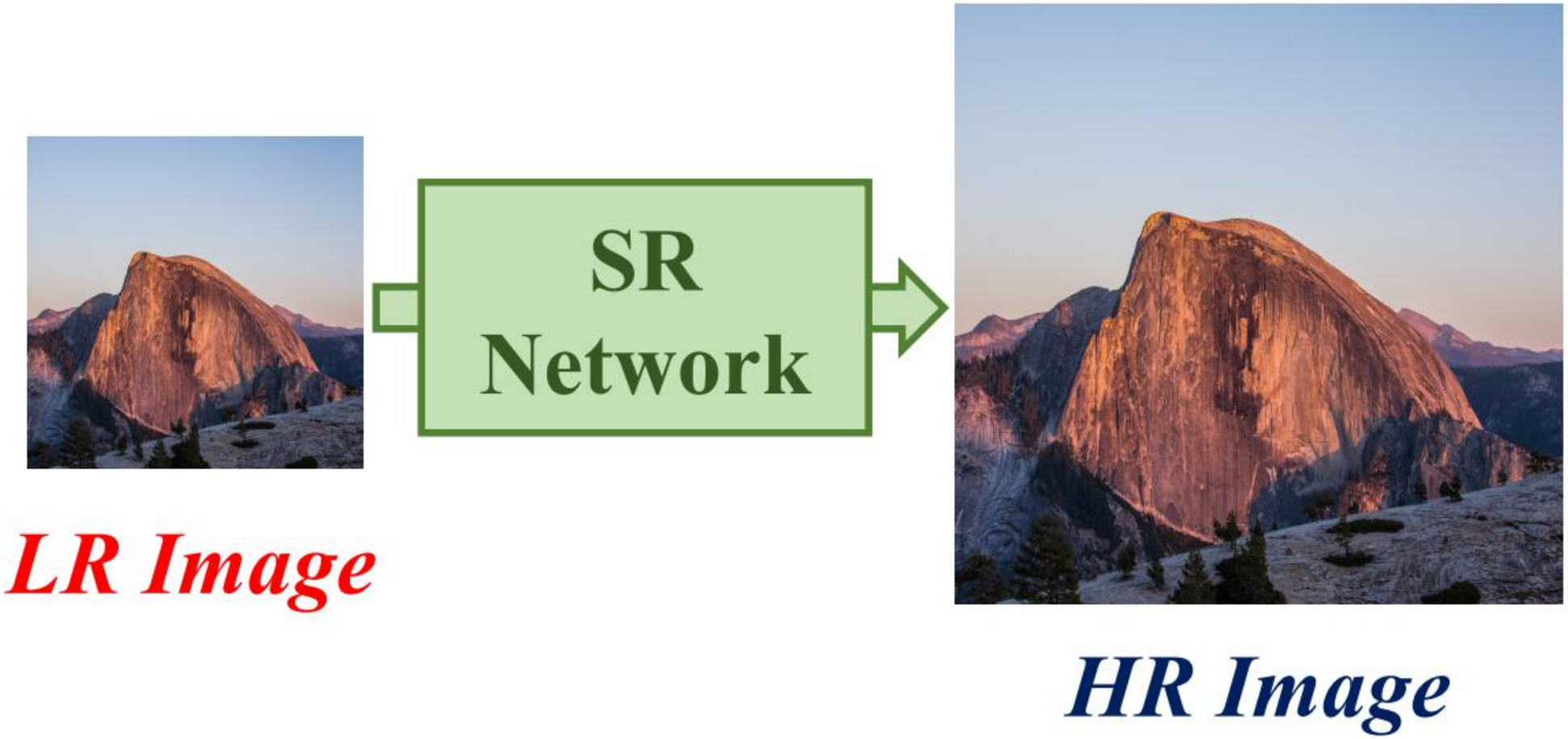
Sweet spots for adaptive weight based methods. The best performance is achieved by applying 1 FC layer without SoftMax normalization on relative positions alone to compute the adaptive weights. This method also approaches or surpasses the state-of-the-art on all three datasets using a deep residual network.

Towards Content-Independent Multi-Reference Super-Resolution: Adaptive Pattern Matching and Feature Aggregation

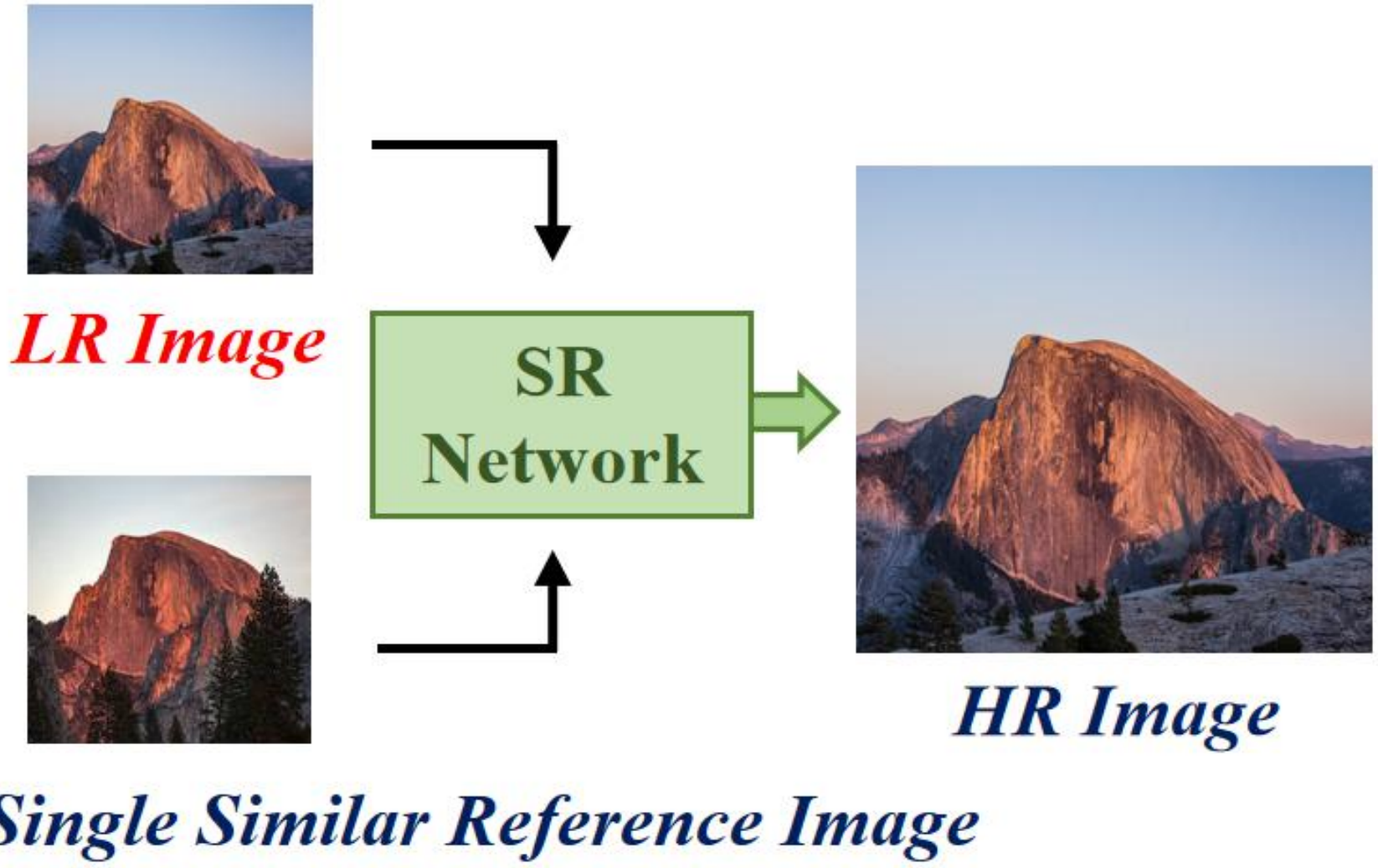
Xu Yan [1], Weibing Zhao* [1], Kun Yuan [1,2], Ruimao Zhang [3], Zhen Li† [1] and Shuguang Cui [1]*

[1] The Chinese University of Hong Kong (Shenzhen),
Shenzhen Research Institute of Big Data,
[2] University of Ottawa, [3] SenseTime Research

Single image super-resolution

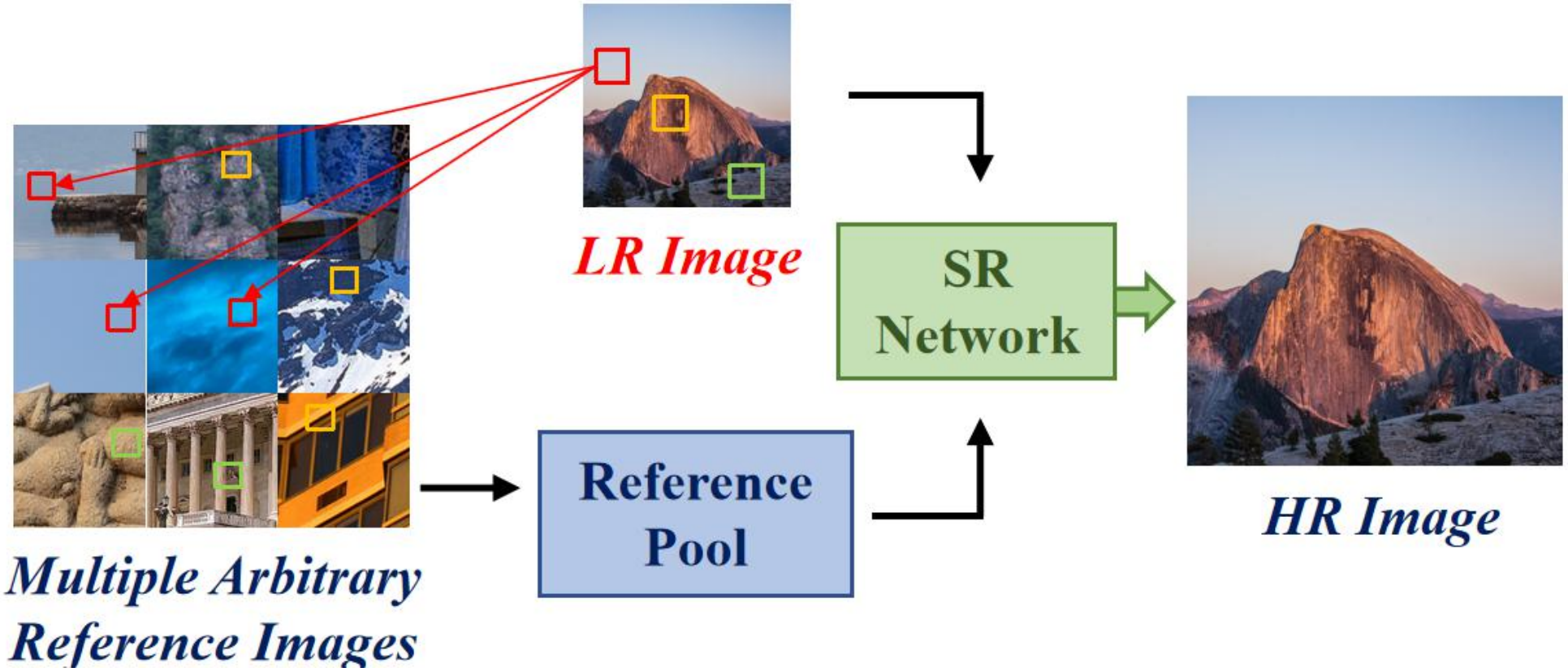


Reference-based super-resolution

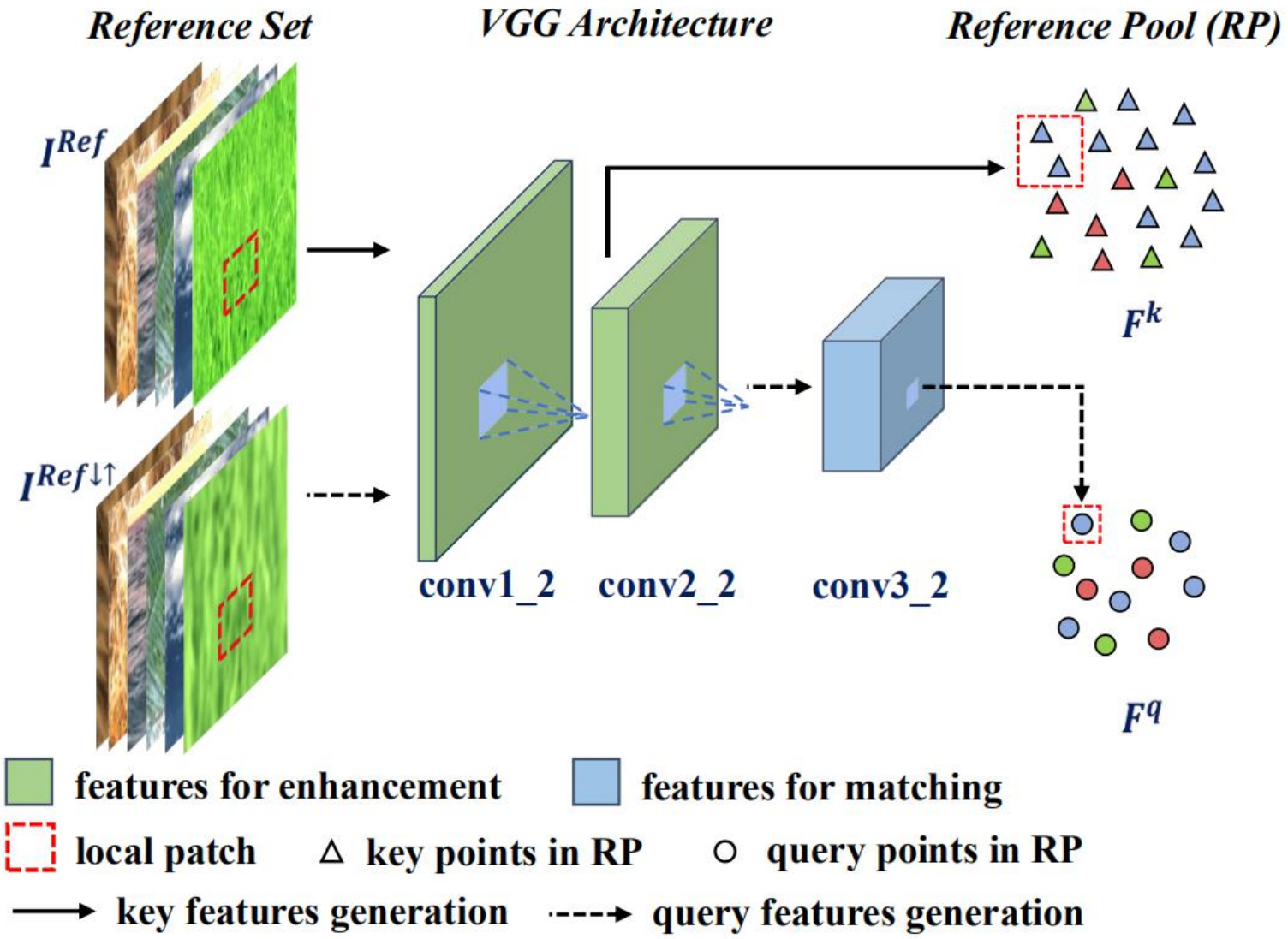


CIMR-SR

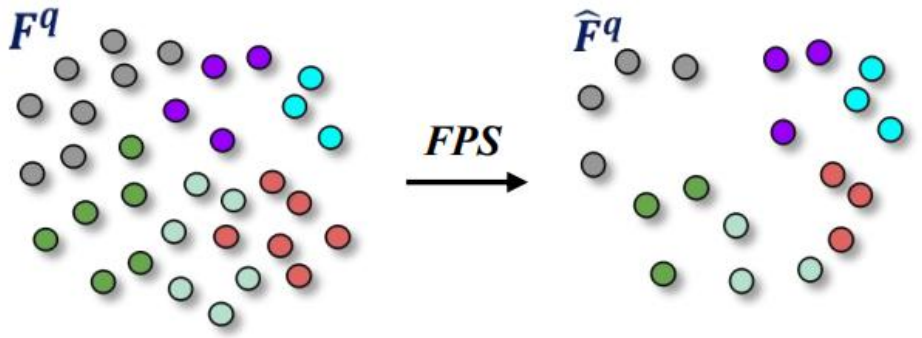
Content-independent multi-reference super-resolution

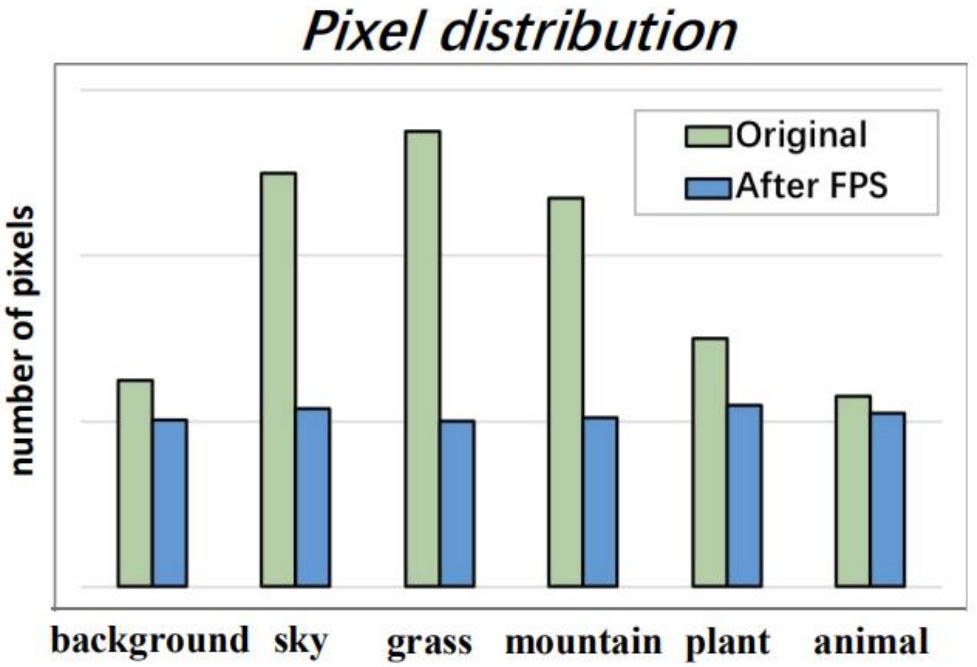


CIMR to feature space



Farthest Point Sampling (FPS)



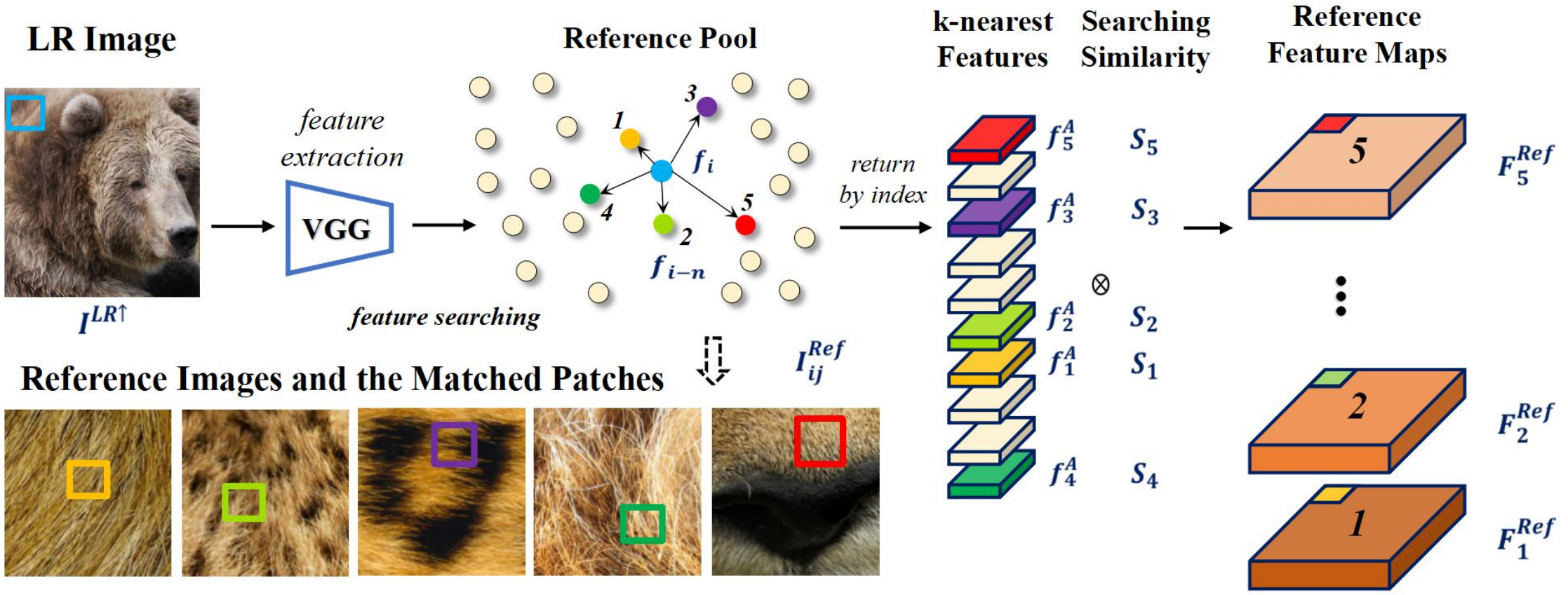


sky mountain background plant grass animal building

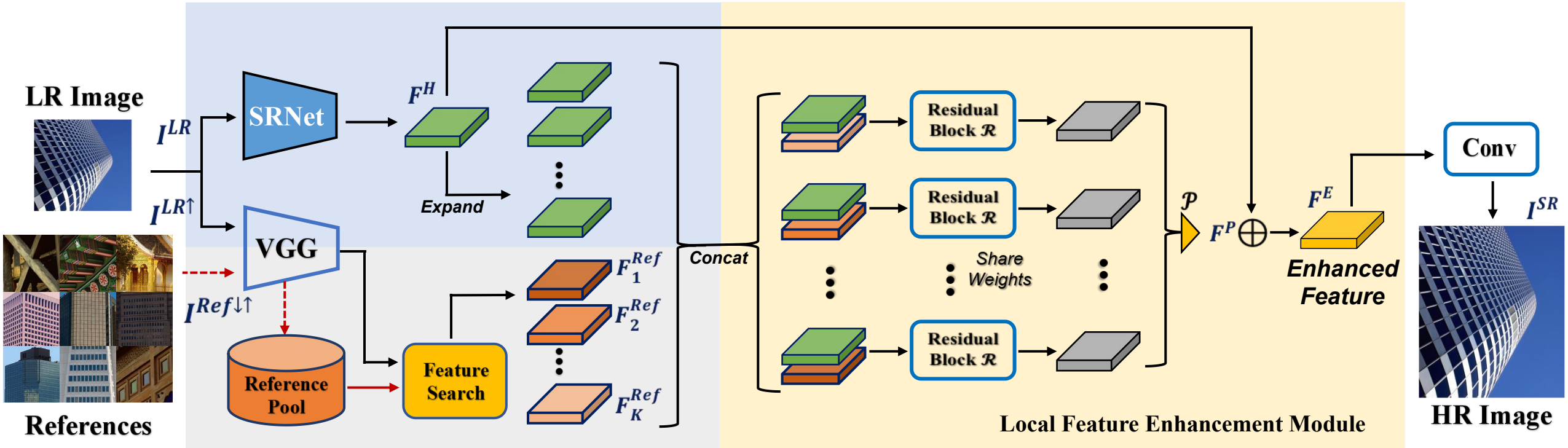
CIMR

- **Feature Searching** retrieves K most similar feature points from query features for each local patch of input LR
- **Feature Aggregation** aligns and fuses the searched key features and generates enhanced feature maps for HR reconstruction

Feature searching



Feature aggregation



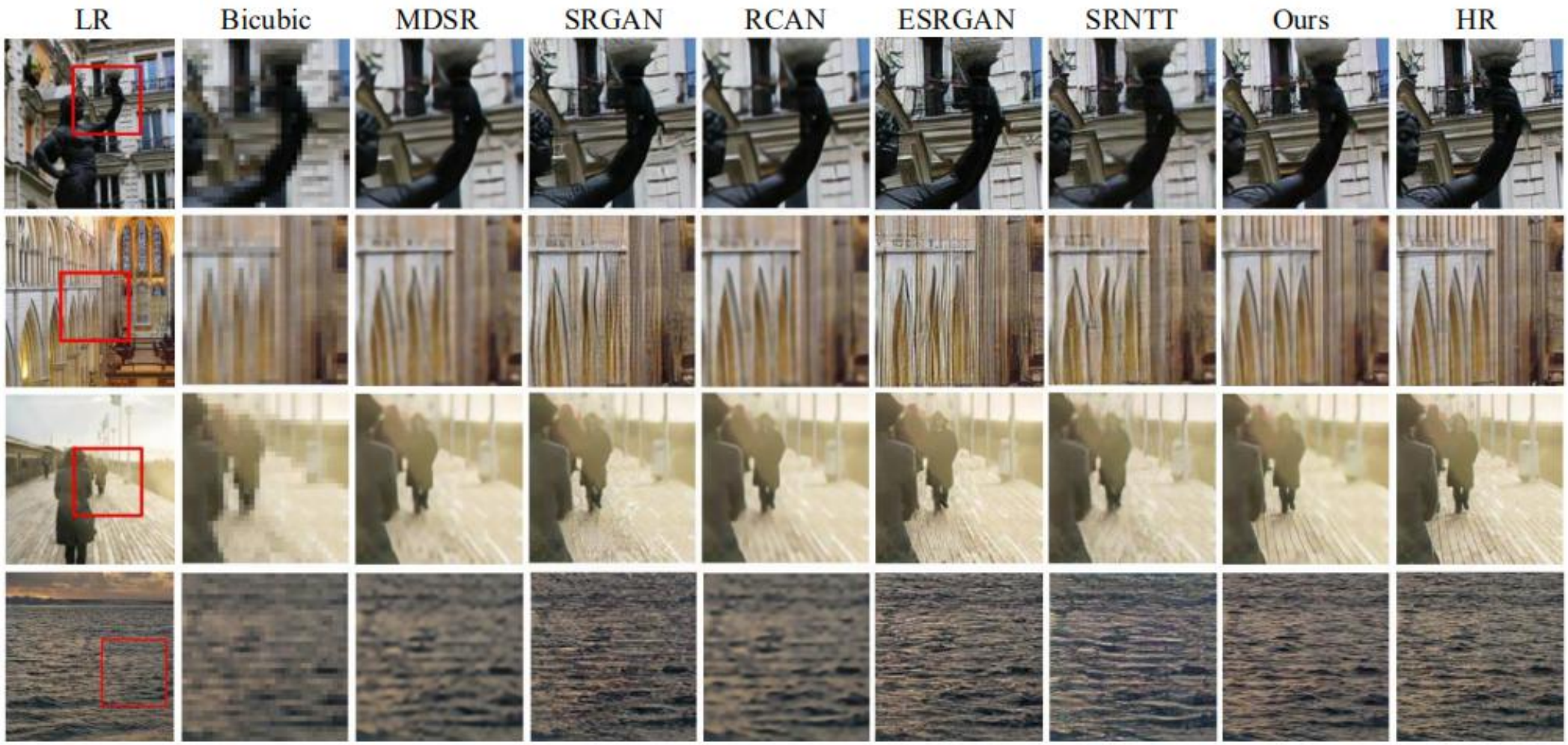
Results

Content-independent references

Table 1. PSNR/SSIM comparison among different SR methods on four datasets: methods are grouped by SISR (top) and RefSR (bottom), and the best result is in bold. All the SR results are evaluated by PSNR and SSIM metrics on the Y channel of transformed YCbCr space.

Algorithm	CUFED5	Urban100	Sun80
Bicubic	24.18 / 0.684	23.14 / 0.674	27.24 / 0.739
SRCNN [4]	25.33 / 0.745	24.41 / 0.738	28.26 / 0.781
SCN [26]	25.45 / 0.743	24.52 / 0.741	27.93 / 0.786
DRCN [12]	25.26 / 0.734	25.14 / 0.760	27.84 / 0.785
LapSRN [13]	24.92 / 0.730	24.26 / 0.735	27.70 / 0.783
MDSR [15]	25.93 / 0.777	25.51 / 0.783	28.52 / 0.792
EDSR [15]	25.90 / 0.776	25.50 / 0.783	28.49 / 0.789
SRGAN [14]	24.40 / 0.702	24.07 / 0.729	26.76 / 0.725
RCAN [33]	**26.32 / 0.789**	**25.65 / 0.785**	**28.67 / 0.795**
SAN [3]	26.29 / 0.789	25.63 / 0.783	28.66 / 0.795
LandMark [30]	24.91 / 0.718	-	27.68 / 0.776
CrossNet [36]	25.48 / 0.764	25.11 / 0.764	28.52 / 0.793
SRNTT [35]	25.61 / 0.764	25.09 / 0.774	27.59 / 0.756
CIMR	26.16 / 0.781	25.24 / 0.778	29.67 / 0.806
SRNTT*-ℓ_2 [35]	25.98 / 0.776	25.54 / 0.784	28.49 / 0.791
SRNTT-ℓ_2 [35]	26.24 / 0.784	25.50 / 0.783	28.54 / 0.793
CIMR-ℓ_2	**26.35 / 0.789**	**25.77 / 0.792**	**30.07 / 0.813**

Results



Results

Content-similar references

Table 2. PSNR/SSIM at different reference levels on CUFED5 dataset. The "warp" denotes the data augmentation with random translation (quarter to half width/height), rotation (10∼30 degree), and scaling (1.2∼2.0× upscaling) from the original HR image.

Algorithm	HR (warp)	L1	L2	L3	L4	All
CrossNet	25.49/.764	25.48/.764	25.48/.764	25.47/.763	25.46/.763	-
SRNTT-ℓ_2	29.29/.889	26.15/.781	26.04/.776	25.98/.775	25.95/.774	26.24/.784
CIMR-ℓ_2	29.82/.903	**27.32/.805**	**27.05/.799**	**26.92/.796**	**26.86/.794**	**27.44/.810**
SRNTT	**33.87/.959**	25.42/.758	25.32/.752	25.24/.751	25.23/.750	25.61/.764
CIMR	30.73/.918	26.50/.786	26.47/.784	26.45/.784	26.44/.784	26.63/.790

ECCV'20 ONLINE 23-28 AUGUST 2020

Results

Table 3. Comparison of adopting different `conv` layers for CIMR. Left part shows PSNR at different reference levels on CUFED5 test set. Right part describes running time, including offline feature searching and forward inference time.

Method-ℓ_2	HR(warp)	L1	L2	L3	L4	FS	Forward
SRNTT	29.29	26.15	26.04	25.98	25.95	2.726	**0.351**
`conv1`	28.24	26.97	26.89	26.84	26.79	2.045	0.783
`conv2`	28.77	27.06	26.93	26.92	26.85	1.551	0.535
`conv3`	27.31	26.26	26.06	26.02	26.03	**1.020**	0.361
`conv2/3`	27.87	27.27	27.01	26.89	26.82	2.571	1.077
`conv1/2/3`	**29.82**	**27.32**	**27.05**	**26.92**	**26.86**	4.616	1.551

香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

SHENZHEN RESEARCH INSTITUTE OF BIG DATA
SRIBD

Thanks for watching!

Xu Yan

2020.8.20