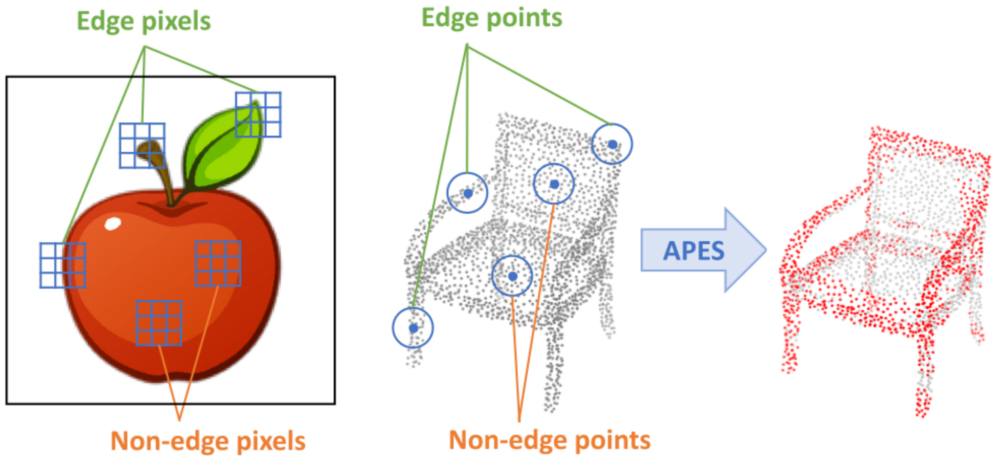
Attention-Based Point Cloud Edge Sampling

Chengzhi Wu[1], Junwei Zheng[1], Julius Pfrommer[2,3], and Jürgen Beyerer[1,2,3]

[1]Karlsruhe Institute of Technology, Germany [2]Fraunhofer IOSB, Germany

[3]Fraunhofer Center for Machine Learning, Germany

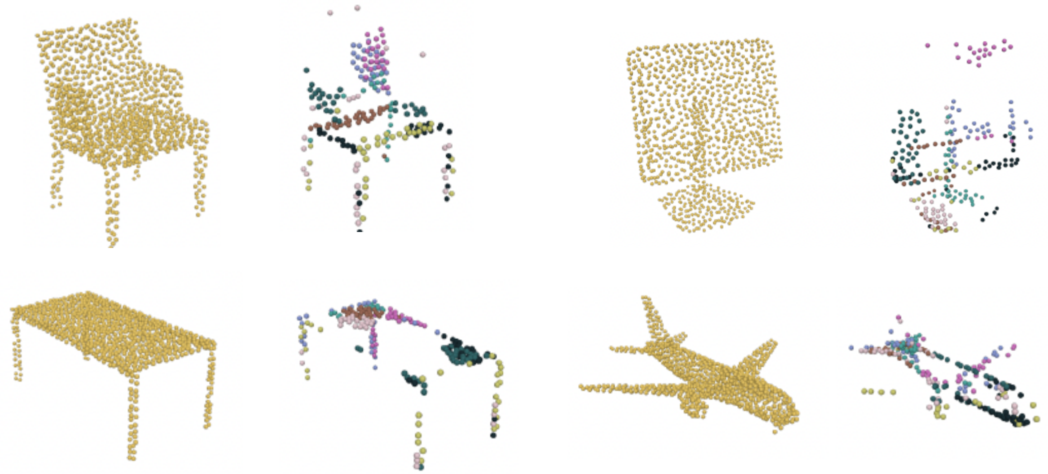


CVPR2023 Highlight
Paper tag: TUE-PM-114

Motivation

What features do CNNs learn from images?

- Texture
- Color
- Shape silhouette/edges



For 3D point clouds:

- Usually no texture information due to sparsity
- Color information sometimes is also not available
- Shape silhouette/edges are of crucial importance

For CV tasks, when sampling point clouds, would sampling edge points be a better choice?

[1] Geirhos, Robert et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." *ICLR* (2019).
[2] Engel, Nico et al. "Point Transformer." IEEE Access 9 (2020): 134826-134840.

Motivation

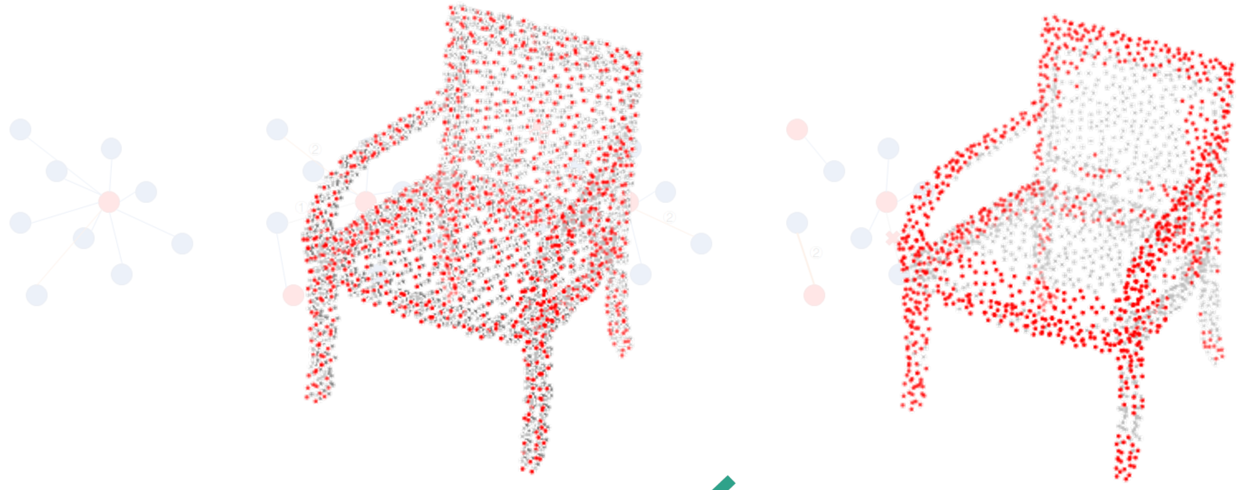
Current point cloud sampling methods

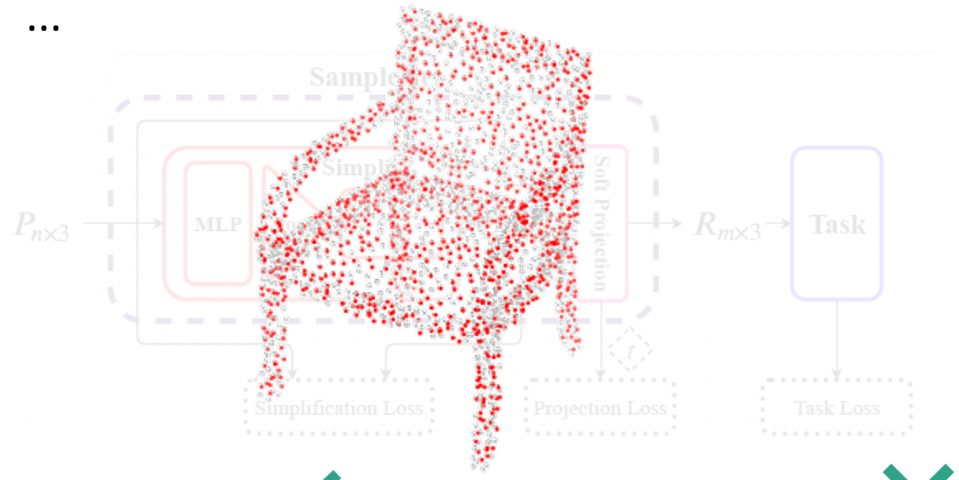
Mathematical statistics-based methods:

- Random Sampling
- Voxel-based grid sampling
- Farthest Point Sampling (FPS)
- Inverse Density Importance Sampling (IDIS)
- ...

Neural network learning-based methods:

- S-Net
- SampleNet
- DA-Net
- MOPS-Net
- LighTN
- ...





Direct point selecting ✓ + Task-oriented ✓

Generating new points ✗

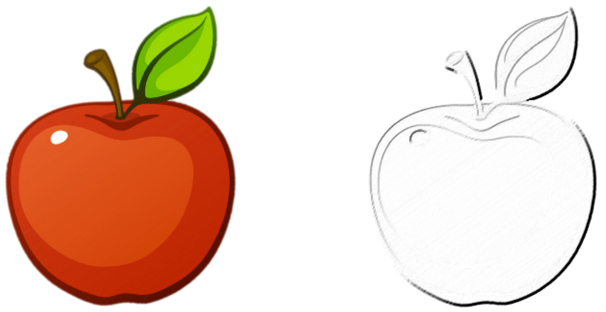
[3] Lang, Itai et al. "SampleNet: Differentiable Point Cloud Sampling." *CVPR* (2019): 7575-7585.

Revisiting Canny Edge Detection on Images

Pixels with larger intensity gradients are defined as edge pixels

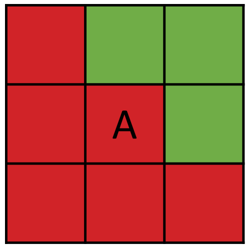
Larger differences between the pixels from a local patch set S_i

The standard deviation σ_i of the intensities in the patch is larger

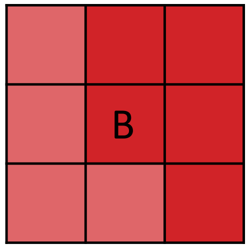


$\boldsymbol{p}_i$: feature of center pixel
$\boldsymbol{p}_{ij}$: feature of one neighbor pixel
$h(\boldsymbol{p}_i, \boldsymbol{p}_{ij})$: measure of feature correlation

Edge Pixel A



Non-Edge Pixel B



$\sigma_A > \sigma_B$

Correlation Map $(\sigma_A = 0.057)$

0.15	0.15	0.15	0.15	0.15	0.15	0.03	0.03	0.03

Correlation Map $(\sigma_B = 0.010)$

0.12	0.12	0.12	0.12	0.12	0.10	0.10	0.10	0.10

Compute normalized correlation map:

$$\boldsymbol{m}_i = \mathrm{softmax}\left(h(\boldsymbol{p}_i, \boldsymbol{p}_{ij})_{j \in S_i}\right)$$

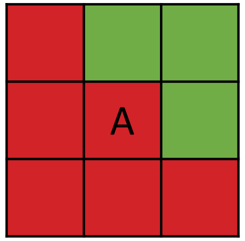
Compute σ_i over the elements of $\boldsymbol{m}_i$

Pixels with larger σ_i are selected as edge pixels

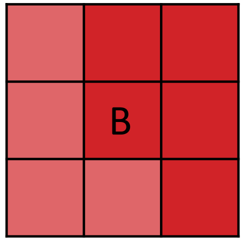
[4] Canny, John F.. "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (1986): 679-698.

Local-based **A**ttention-Based **P**oint Cloud **E**dge **S**ampling (APES)

Edge Pixel A



Non-Edge Pixel B



$\sigma_A > \sigma_B$

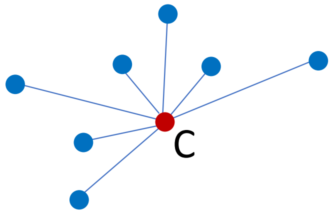
Correlation Map ($\sigma_A = 0.057$)

0.15	0.15	0.15	0.15	0.15	0.15	0.03	0.03	0.03

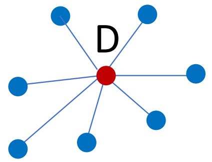
Correlation Map ($\sigma_B = 0.010$)

0.12	0.12	0.12	0.12	0.12	0.10	0.10	0.10	0.10

Edge Point C



Non-Edge Point D



$\sigma_C > \sigma_D$

Correlation Map ($\sigma_C = 0.051$)

0.18	0.16	0.16	0.16	0.16	0.06	0.06	0.06

Correlation Map ($\sigma_D = 0.022$)

0.18	0.12	0.12	0.12	0.12	0.12	0.12	0.10

Key idea: Use the local patch attention map as the normalized correlation map

Correlation measure: $h^l(\boldsymbol{p}_i, \boldsymbol{p}_{ij}) = Q(\boldsymbol{p}_i)^\top K(\boldsymbol{p}_{ij} - \boldsymbol{p}_i)$

⇩

Correlation map: $\boldsymbol{m}_i{}^l = \mathrm{softmax}\left(h^l(\boldsymbol{p}_i, \boldsymbol{p}_{ij})_{j \in S_i} / \sqrt{d}\right)$

⇩

Compute σ_i over the elements of $\boldsymbol{m}_i{}^l$

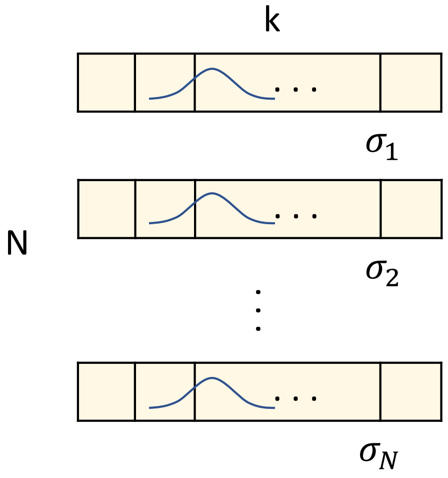
⇩

Points with larger σ_i are selected as edge points

Global-based APES

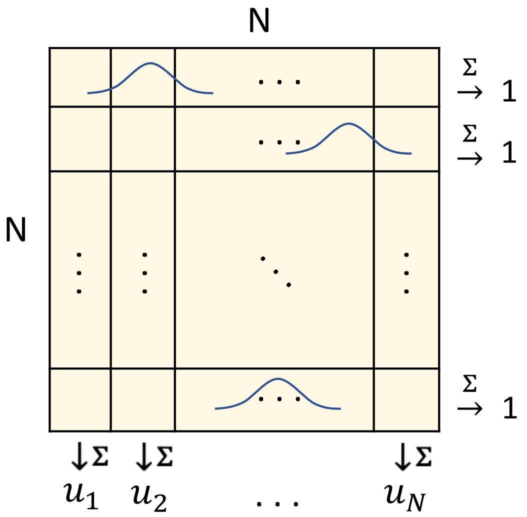
Key idea: (i) Use the global attention map as the normalized correlation map
(ii) Instead of computing row-wise standard deviations, compute column-wise sums

Local-based Correlation Map



select points with larger σ

Global-based Correlation Map



select points with larger u

Correlation measure: $h^g(\boldsymbol{p}_i, \boldsymbol{p}_j) = Q(\boldsymbol{p}_i)^\top K(\boldsymbol{p}_j)$

$\Downarrow$

Local correlation map: $\boldsymbol{m}_i{}^g = \mathrm{softmax}\left(h^g(\boldsymbol{p}_i, \boldsymbol{p}_j)_{j \in S} / \sqrt{d}\right)$

$\Downarrow$

Global correlation map: $M^g = \begin{pmatrix} - & \boldsymbol{m}_1{}^{g\top} & - \\ - & \boldsymbol{m}_2{}^{g\top} & - \\ & \vdots & \\ - & \boldsymbol{m}_N{}^{g\top} & - \end{pmatrix}$

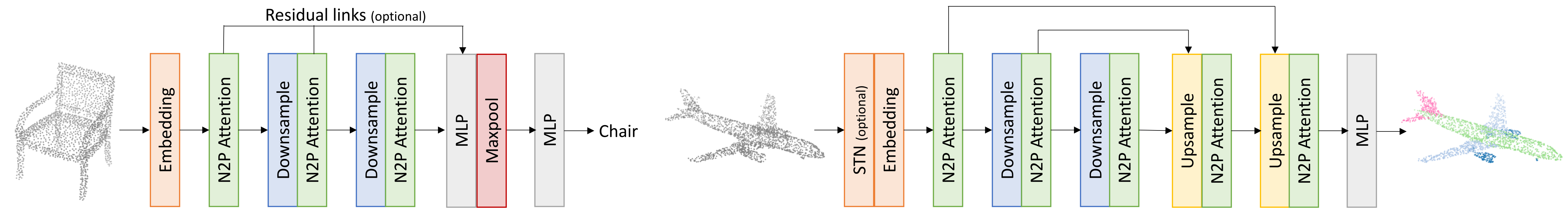
$\Downarrow$

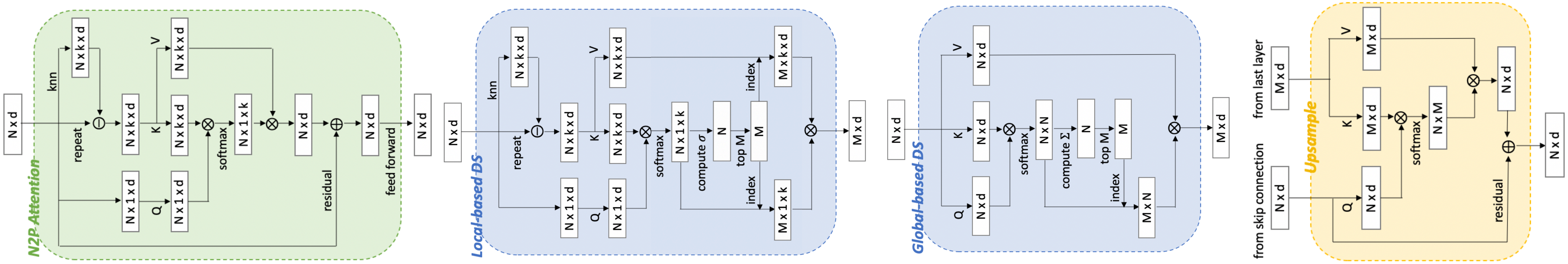
Compute column-wise sums u_i of M^g

$\Downarrow$

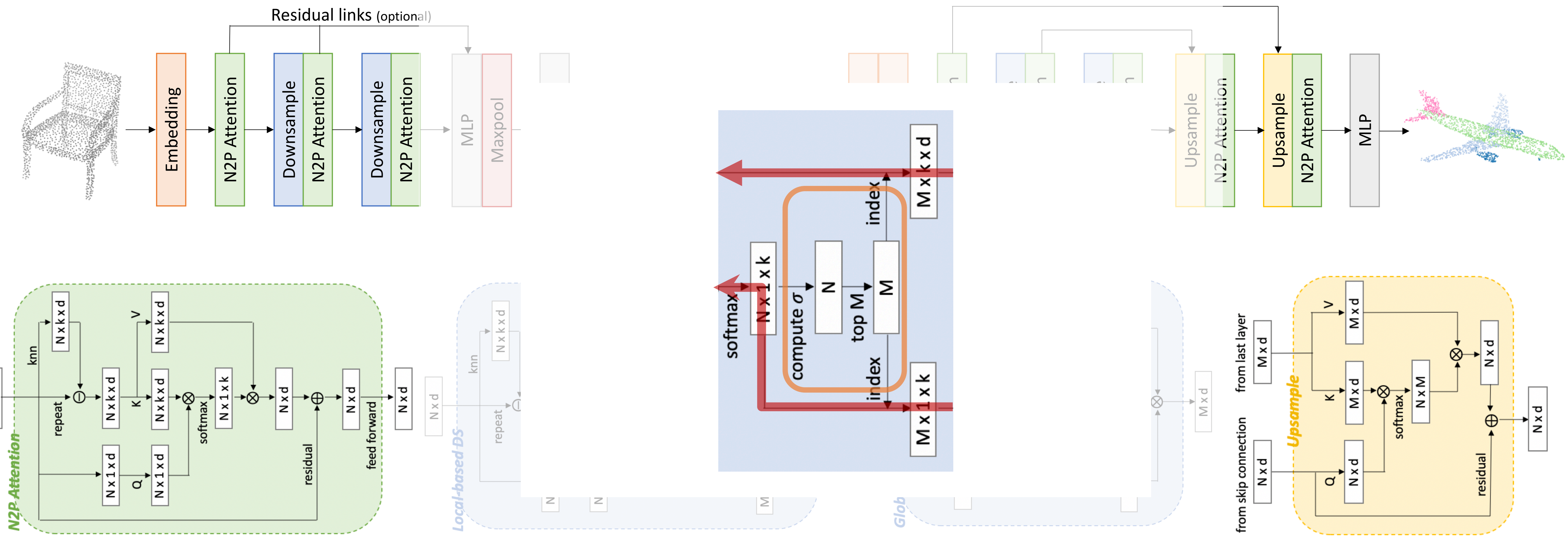
Points with larger u_i are selected as edge points

Network Architecture



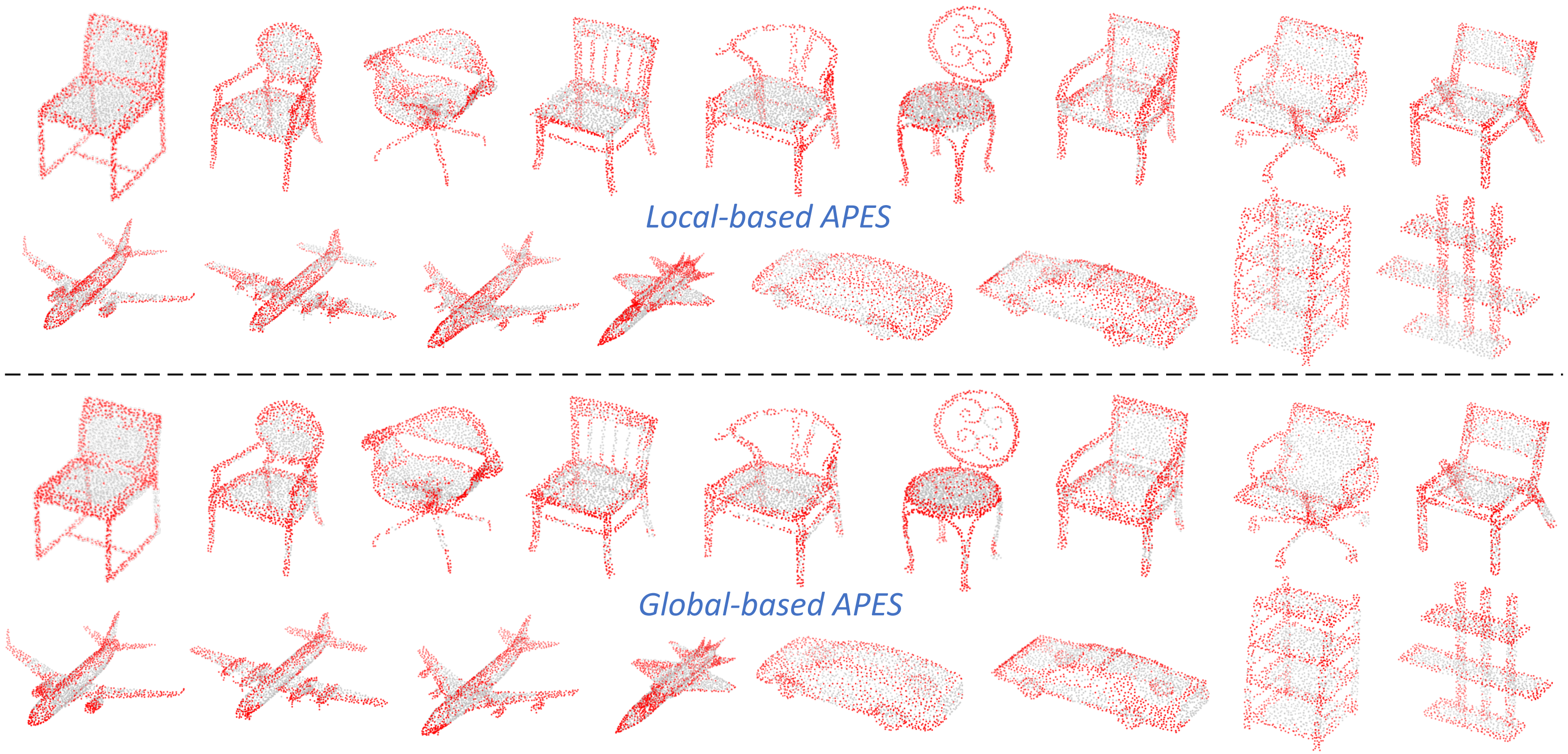


Network Architecture



Experiments - Classification

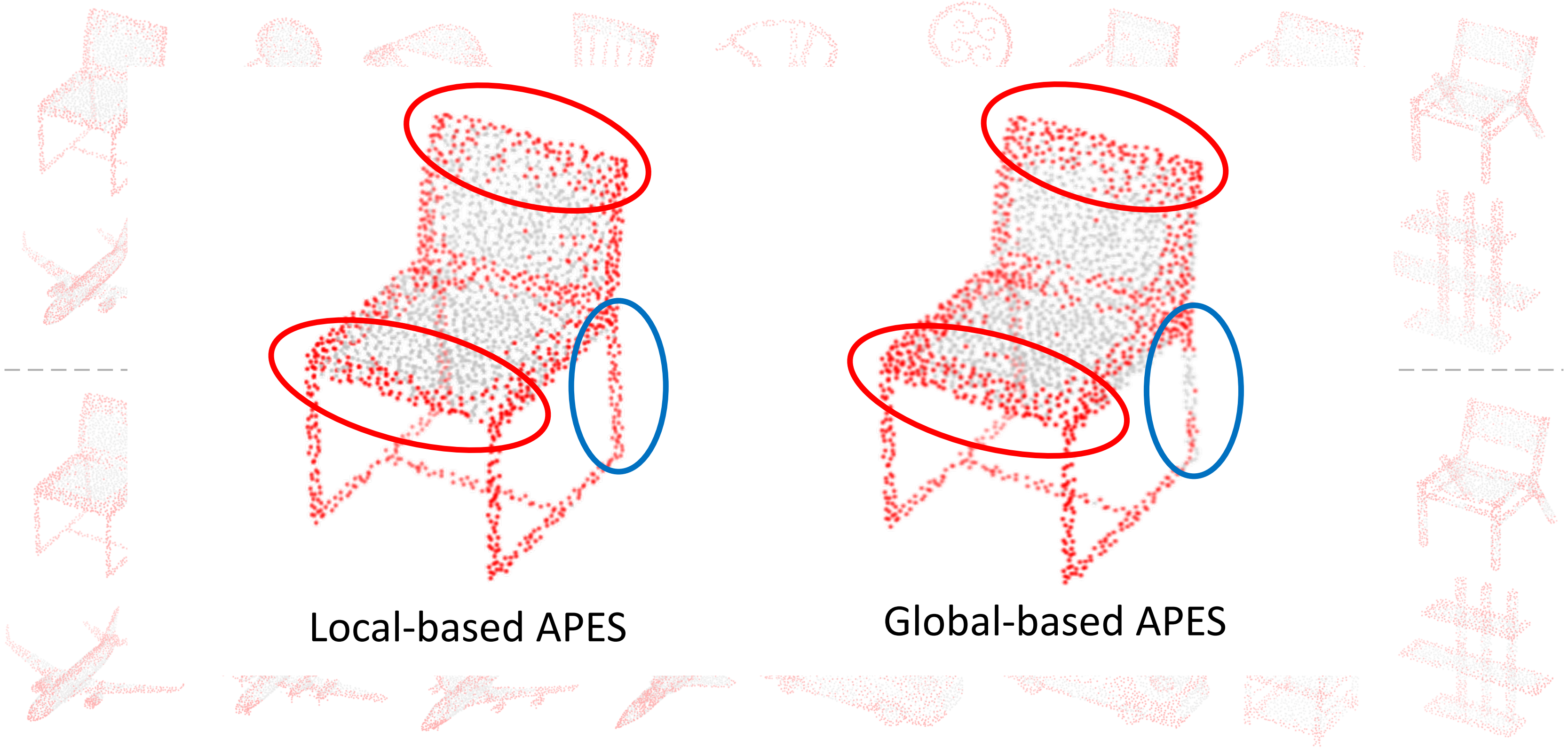
Benchmark: ModelNet40



Method	Overall Accuracy
PointNet	89.2%
PointNet++	91.9%
SpiderCNN	92.4%
DGCNN	92.9%
PointCNN	92.2%
PointConv	92.5%
PVCNN	92.4%
KPConv	92.9%
PointASNL	93.2%
PT[1]	92.8%
PT[2]	93.7%
PCT	93.2%
PRA-Net	93.7%
PAConv	93.6%
CurveNet	**93.8%**
DeltaConv	**93.8%**
APES (local-based)	93.5%
APES (global-based)	**93.8%**

Experiments - Classification

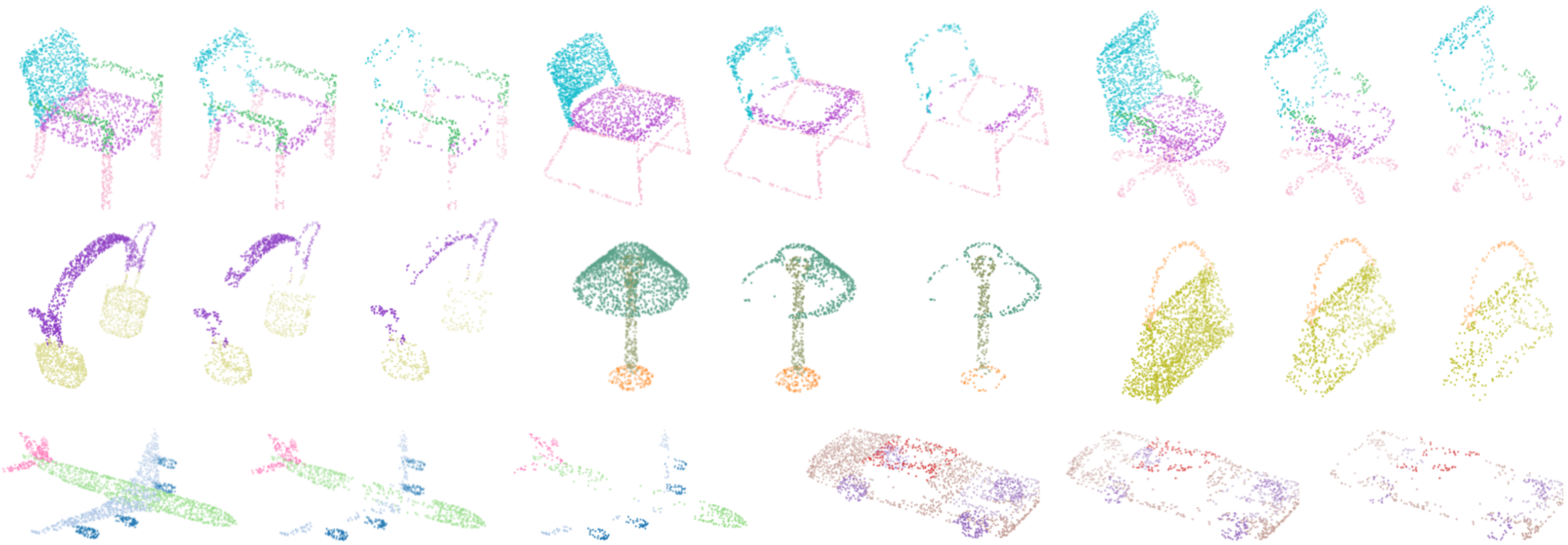
Benchmark: ModelNet40



Method	Overall Accuracy
PointNet	89.2%
PointNet++	91.9%
SpiderCNN	92.4%
DGCNN	92.9%
PointCNN	92.2%
PointConv	92.5%
PVCNN	92.4%
KPConv	92.9%
PointASNL	93.2%
PT[1]	92.8%
PT[2]	93.7%
PCT	93.2%
PRA-Net	93.7%
PAConv	93.6%
CurveNet	**93.8%**
DeltaConv	**93.8%**
APES (local-based)	93.5%
APES (global-based)	**93.8%**

Experiments - Segmentation

Benchmark: ShapeNet Part



Method	Cat. mIoU	Ins. mIoU
PointNet	80.4%	83.7%
PointNet++	81.9%	85.1%
SpiderCNN	82.4%	85.3%
DGCNN	82.3%	85.2%
SPLATNet	83.7%	85.4%
PointCNN	84.6%	86.1%
PointConv	82.8%	85.7%
KPConv	85.0%	86.2%
PT[1]	-	85.9%
PT[2]	83.7%	**86.6%**
PCT	-	86.4%
PRA-Net	83.7%	86.3%
PAConv	84.6%	86.1%
CurveNet	-	**86.6%**
StratifiedTransformer	**85.1%**	**86.6%**
APES (local-based)	83.1%	85.6%
APES (global-based)	83.7%	85.8%

The performances on intermediate downsampled point clouds are better!

Method \ Points	Cat. mIoU (%)			Ins. mIoU (%)		
	2048	1024	512	2048	1024	512
APES (local)	83.11	85.56	**86.17**	85.58	87.27	87.41
APES (global)	83.67	84.86	85.44	85.81	87.78	**88.06**

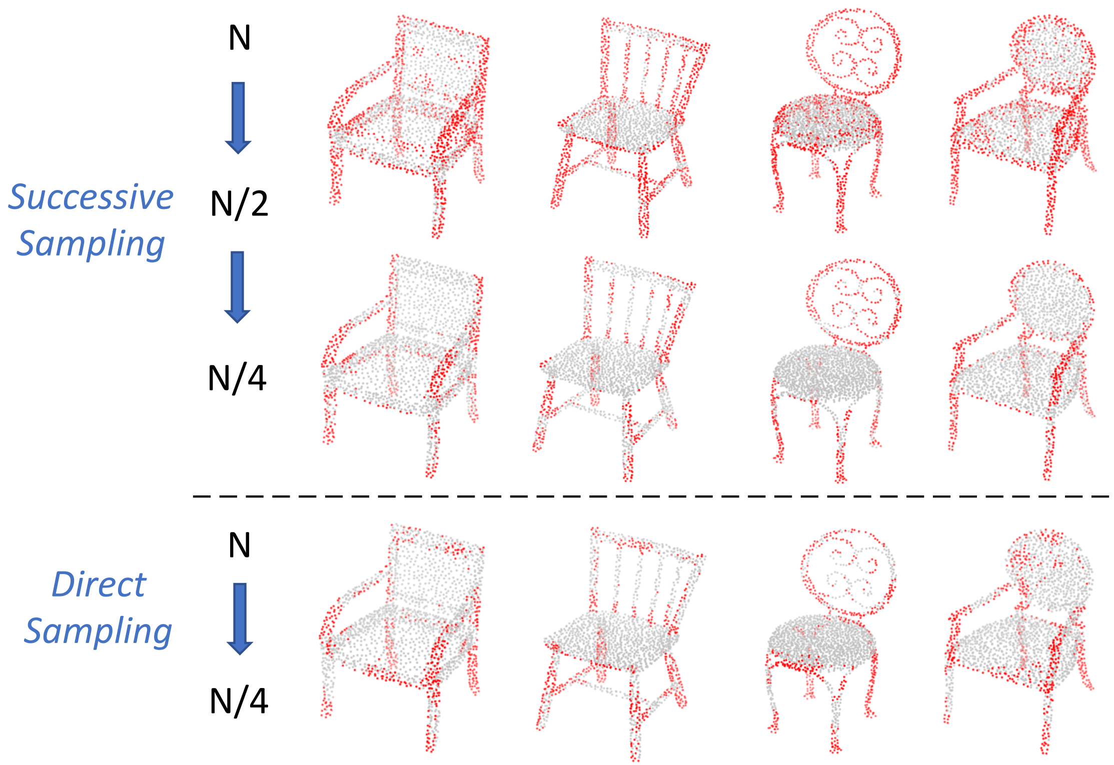
Ablation study

Method	Feature Learning Layer	OA (%)
DGCNN	EdgeConv	92.90
APES (local-based)	EdgeConv	93.02
	P2P Attention	93.30
	N2P Attention	**93.47**
APES (global-based)	EdgeConv	93.18
	P2P Attention	93.46
	N2P Attention	**93.81**

Method	Embedding Dimension	OA (%)
APES (local-based)	64	93.10
	128	93.47
	192	**93.54**
APES (global-based)	64	93.34
	128	93.81
	192	**93.83**

k	8	16	32	64	128	256	512
OA (%)	93.14	93.26	93.47	93.52	93.54	93.59	**93.63**

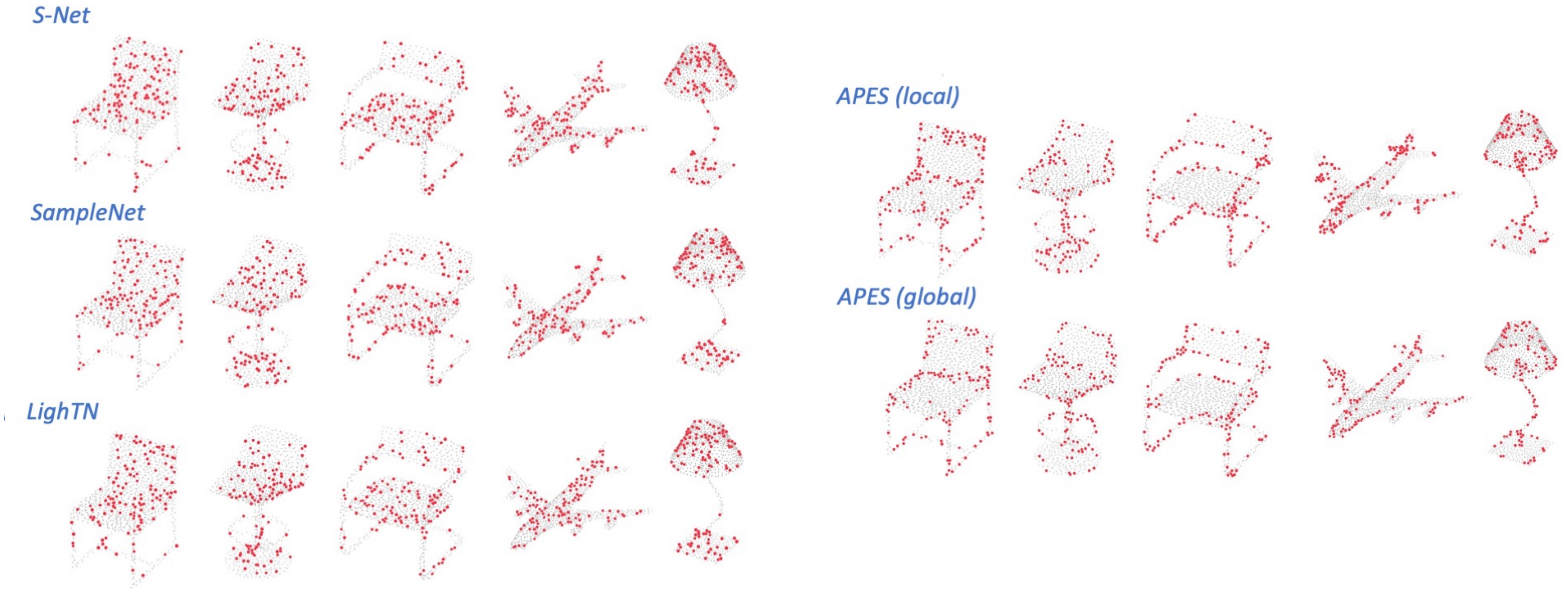
k: number of neighbors used in local-based APES



Sampling Methods Comparison

Benchmark:
ModelNet40 classification,
with simple PointNet
on sampled sub-point cloud

M	Voxel	RS	FPS	S-NET	PST-NET	SampleNet	MOPS-Net	DA-Net	LighTN	APES (local)	APES (global)
512	73.82	87.52	88.34	87.80	87.94	88.16	86.67	89.01	89.91	90.79	**90.81**
256	73.50	77.09	83.64	82.38	83.15	84.27	86.63	86.24	88.21	90.38	**90.40**
128	68.15	56.44	70.34	77.53	80.11	80.75	86.06	85.67	86.26	89.73	**89.77**
64	58.31	31.69	46.42	70.45	76.06	79.86	85.25	85.55	86.51	88.68	**89.57**
32	20.02	16.35	26.58	60.70	63.92	77.31	84.28	85.11	86.18	86.49	**88.56**



Conclusion

- We propose an attention-based point cloud edge sampling (APES) method, which uses the attention mechanism to compute correlation maps and sample edge points accordingly.
- Two variations of local-based APES and global-based APES are proposed based on two different attention modes.
- Qualitative and quantitative results show that our method successfully extracts edge points and achieves excellent performance on common point cloud benchmark tasks.

Future Work:

- Design other supplementary losses for the training.
- Propose a better upsampling method that can better cope with edge point sampling.
- ...

Thanks for watching!