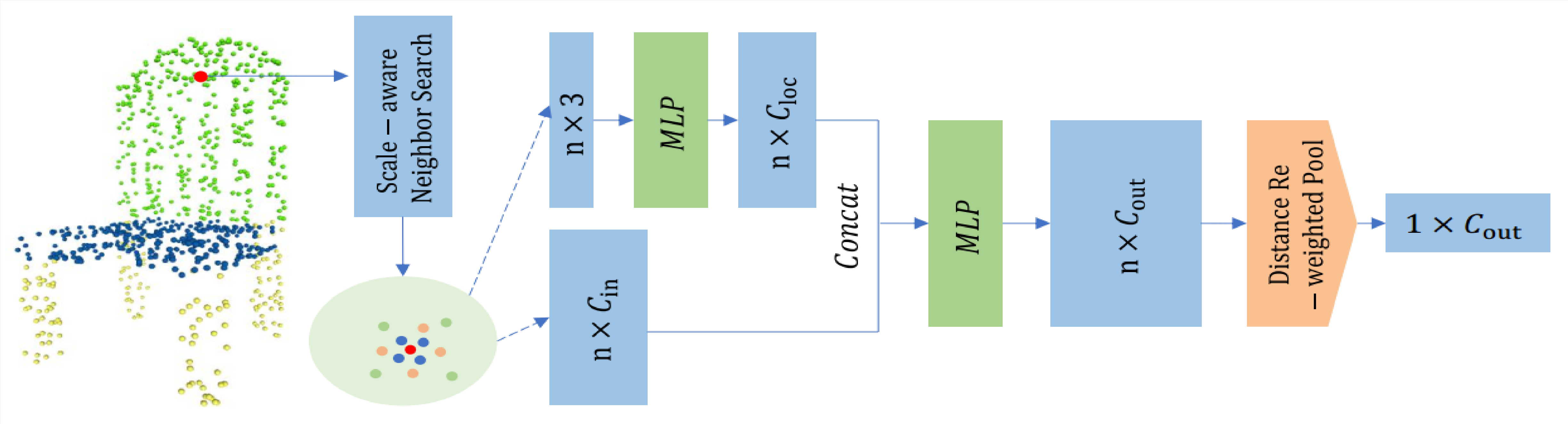
PC-Net: A Deep Network for 3D Point Clouds Analysis

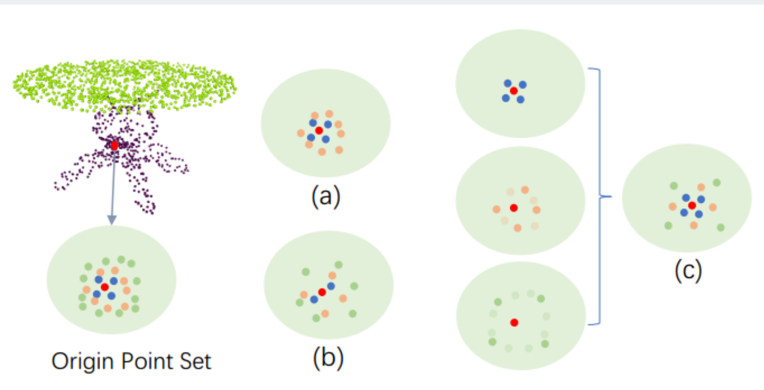
Due to the irregularity and sparsity of 3D point clouds, applying convolutional neural networks directly on them can be nontrivial. In this work, we propose a simple but effective approach for 3D Point Clouds analysis, named PC-Net. PC-Net directly learns on point sets and is equipped with three new operations: first, we apply a novel scale-aware neighbor search for adaptive neighborhood extracting; second, for each neighboring point, we learn a local spatial feature as a complement to their associated features; finally, at the end we use a distance reweighted pooling to aggregate all the features from local structure. With this module, we design hierarchical neural network for point cloud understanding. For both classification and segmentation tasks, our architecture proves effective in the experiments and our models demonstrate state-of-the-art performance over existing deep learning methods on popular point cloud benchmarks.

PC-Net



Overview of PC-Net. For a representative point, we first apply a scale-aware neighbor search on the point cloud for adaptive neighborhood extracting. Then for each neighboring point, we lift its coordinate into Cloc dimensional space and concatenate its associated feature and this local spatial feature. At the end, PC-Net uses a distance re-weighted pooling operation to aggregate information from all the neighboring points

Scale-aware neighbor search



An illustration of different neighbor search strategies to get *K* neighbors (*K* = 12 in this example). (a) Directly taking in *K* nearest neighbors. (b) Uniformly sampling K input points from $K \times D$ neighboring points (*D* denotes the dilation rate and equals to 3 in this example). (c) Scale-aware neighbor search, which uses different sampling rates to extract points at different scales of the neighboring points.

Local feature learning

The spatially-local correlation in the neighborhood of representative points is important information to help generate output features. First, learning local geometric features by applying MLP(·) function on the local coordinates of Pn as:

$$\mathbf{F_l} = MLP(\mathbf{P_n} - p),$$

which lifts local coordinates into high dimensional feature space.
Then we concatenate Fl and Fn together as the new input features of Prep and forward them to another MLP(·) function:

$$\mathbf{F_p} = MLP([\mathbf{F_l}\ \mathbf{F_n}]).$$

Distance re-weighted pooling

To exploit the distance correlation between neighboring points and *p*, we add a weight term Wp, which is defined as:

$$\mathbf{W_p}(i) = exp(-\parallel p_i - p \parallel),$$

where each feature associated with pi in Fp is re-weighted by the function of distance. The selection of Wp is based on the intuition that neighboring points close to the representative point should have more impact on each other. Consequently, the learned feature of p is:

$$\mathbf{f_p} = MaxPool_c(\mathbf{W_p} \times \mathbf{F_p}).$$

Experimental Result

Classification

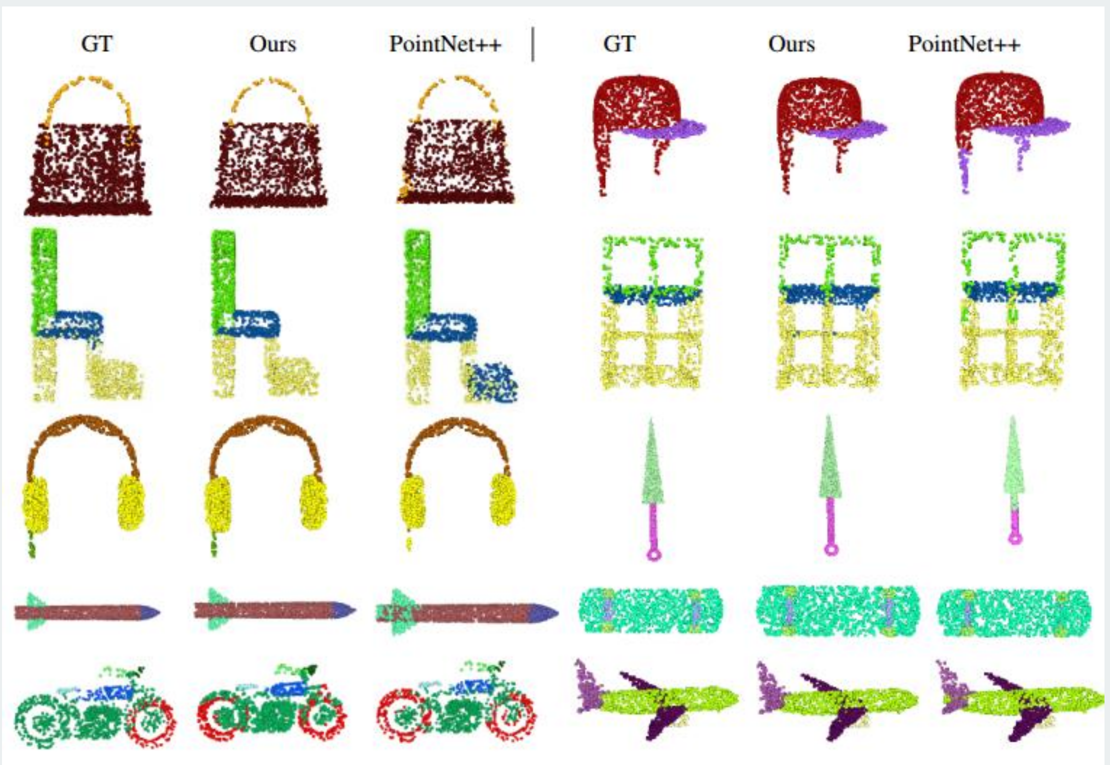
MODELNET10

Method	mA	OA
MPNN	-	92.07
PointNet++	-	92.51
SplineCNN	-	92.65
PointCNN	-	93.28
Kd-Net	93.5	94.0
KCNet	-	94.4
Ours	**95.09**	**95.37**

MODELNET40

Method	mA	OA
VoxNet	83.0	85.9
Kd-Net	88.5	90.6
PointNet	86.2	89.2
PointNet++	-	90.7 (91.9, PN5000)
SO-Net	-	90.7 (93.4, PN5000)
3DmFV-Net	-	91.4 (91.6, P2048)
PCNN	-	92.3
KCNet	-	91.0
SpecGCN	-	91.5 (92.1, PN2048)
PointCNN	88.8	92.5
PointConv	-	92.5
Ours	**89.97**	**93.33**

Segmentation



Future Work

We will explore scale-aware techniques for the classical ConvNets or graph-based ConvNets to better understand the 3D point cloud in the future.