DTU

DTU Compute Visual Computing

diffConv: Analyzing Irregular Point Clouds with an Irregular View

Manxi Lin, Aasa Feragen

Visual Computing, Technical University of Denmark

{manli,afhar}@dtu.dk

Background and Motivation

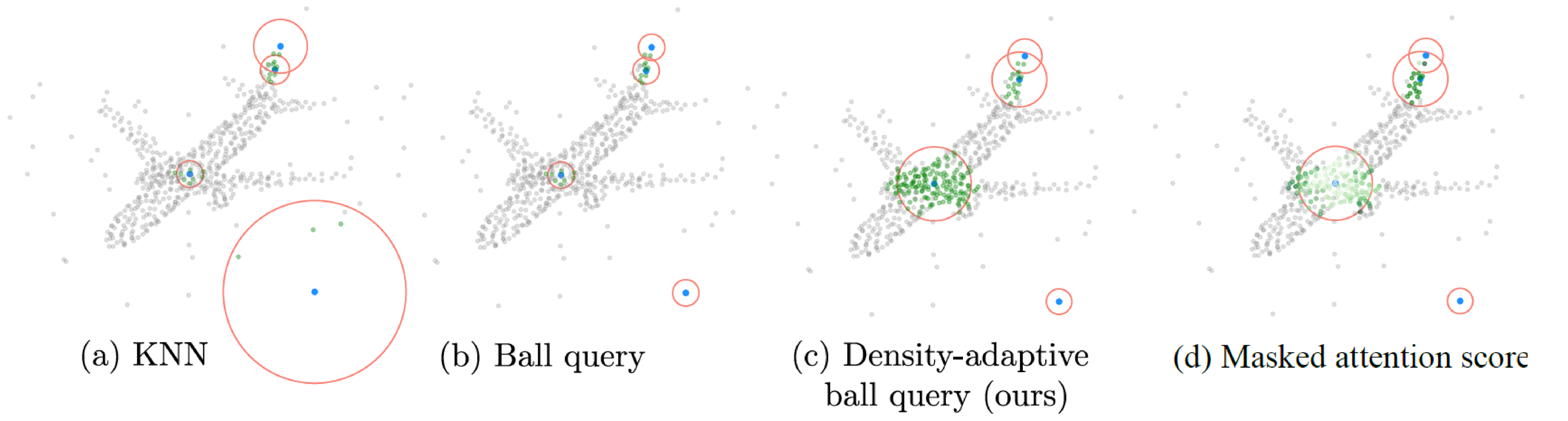

Fig 1. The blue dots refer to key points, green dots denote neighbors, orange circles indicate neighborhoods of key points. The color scale in (d) denotes the neighbor importance of the key point on the airplane body, given by masked attention score.

Point cloud is becoming a more and more popular representation of 3D objects. The industrial demand has prompted a strong call for exploiting the underlying geometric information in points, fuelling deep learning models that rely on learning effective local point features.

Standard spatial convolutions assume input data with a regular neighborhood structure. However, unlike grid data, points are unordered and unstructured. Existing methods typically generalize convolution to the irregular point cloud domain by fixing neighborhood size, where the convolution kernel size remains the same for each point.

Figure 1. illustrates the two common point groupings, namely the k nearest neighbor (KNN) grouping and ball query. KNN fixes its view to the k nearest neighbors of each key point, and the resulting groupings are sensitive to noise and outliers. Sometimes object points are automatically grouped with far away points, strongly affecting their downstream features.

Ball query, conversely, constrains its view to a pre-defined constant radius ball. We see from figure (b) that using the same radius, or view, at every location also causes problems: A small radius is needed to avoid grouping noise, but this means that the less dense parts of the airplane, such as its tail, easily become disconnected and mistaken for noise.

In addition, points from flat areas, which usually have similar and unrecognizable local feature differences, need larger receptive field to learn discrimitive features. Neither KNN nor ball query is giving them large enough neighborhoods.

Methods

Differing from the previous generalizations of CNNs, which all rely on taking a regular view of the point clouds in order to define convolutional operators, we suggest an irregular view of point cloud analysis in this paper, namely diffConv.

Density-dilated ball query. Given a point cloud consisting of N points $P = \{p_i | i = 1,2,\ldots,N\}$, and a data matrix containing their feature vectors $\mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N]^T$, we group the point p_i's neighbors by a Gaussian Kernel:

$$d_i = \frac{1}{Nh}\sum_{j=1}^{N}\frac{1}{\sqrt{2\pi}}e^{-\frac{\|p_i - p_j\|_2^2}{2h^2}},$$

where h is the bandwidth. Then, with the normalized estimated density, the neighborhood is adaptively dilated by:

$$r_i = \sqrt{r^2(1+\hat{d}_i)},$$

where r is the minimum searching radius.

Masked attention. The dilated ball query still treats all the neighbors inside the ball equally. We further introduce masked attention to learn and enhance the irregularity. We use an adjacency matrix A to store the connections between points. The attentive scores are computed by:

$$A_{ij} = \begin{cases} l_\phi(x_i||p_i)l_\psi(x_j||p_j)^T & \text{if } \|p_i - p_j\|_2 < r_i \\ -\infty & \text{otherwise,} \end{cases}$$

$$\tilde{A}_{i,j} = \frac{e^{A_{i,j}}}{\sum_k^N e^{A_{i,k}}}.$$

$$\overline{A}_{ij} = \frac{\sqrt{\tilde{A}_{ij}}}{\sum_k^N \sqrt{\tilde{A}_{kj}}} \quad \text{and} \quad \hat{A}_{ij} = \frac{\overline{A}_{ij}}{\sum_k^N \overline{A}_{ik}}.$$

Laplacian smoothing. We leverage the Laplacian smoothing from graph convolutional neural networks to aggregate the point feature difference with its neighbors:

$$S = X - \hat{A}X$$

The aggregated feature difference are further fused with absolute point features and coordinates:.

$$G = \sigma(l_\theta(S||X) + l_\pi(P))$$

This is a generalization of edgeConv [35] from regular neighborhoods to irregular ones

Experiment Results

Methods	OA(%)	MA(%)
PointNet [18]	90.19±0.58	84.55±0.81
DGCNN [35]	92.89±0.43	89.62±0.83
CurveNet [38]	92.86±0.49	89.51±0.81
diffConvNet (ours)	**93.15±0.34**	**89.86±0.56**

Tab 1. Classification result on ModelNet40 [37] (resplits). OA means overall accuracy over instances; MA means mean accuracy over categories.

Methods	CER(%)
PointNet	28.3
PointNet++	23.6
DGCNN	25.9
CurveNet	22.7
RS-CNN	26.2
PCT	25.5
GDANet	25.6
diffConvNet (ours)	**21.4**

Tab 2. Classification result on ModelNet40-C [22]. CER means corruption error rate.

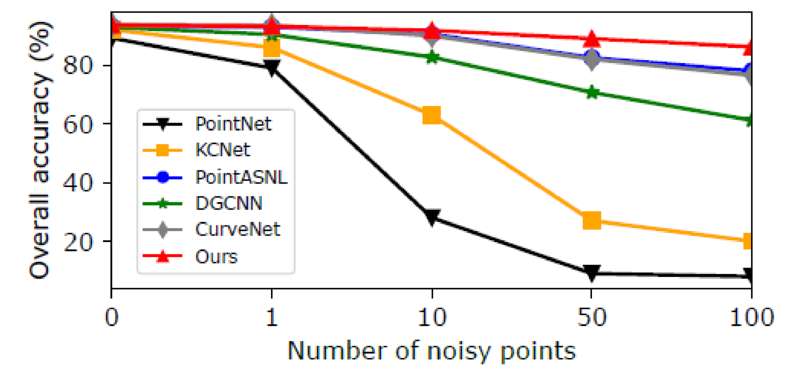

Fig 2. Robustness Study on ModelNet40.

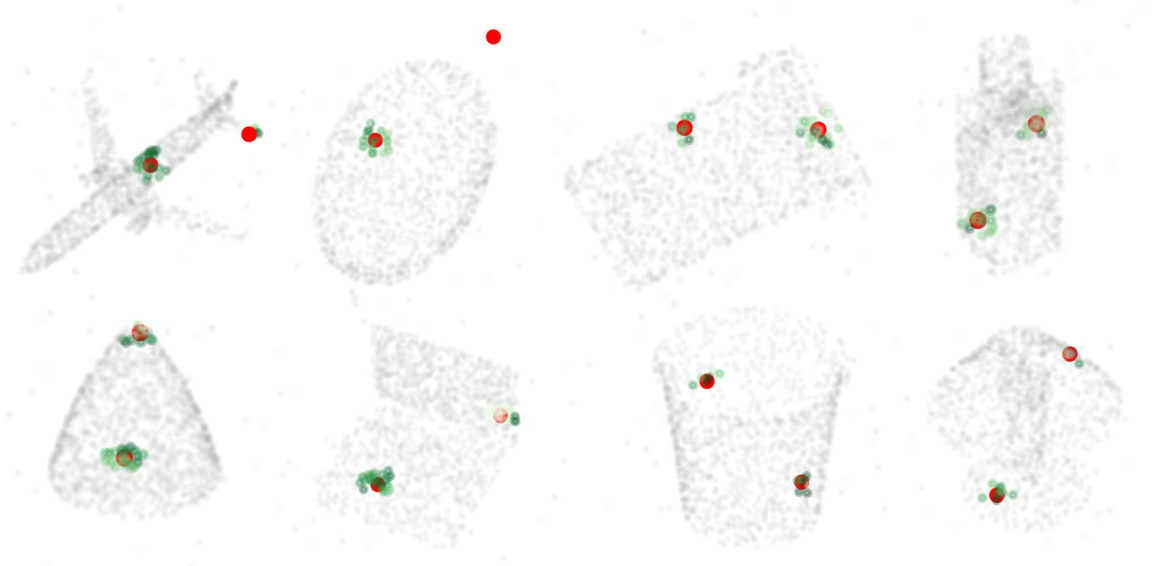

Fig 3. Visualization of attentive scores on ModelNet40-C.

**Citations in this poster are corresponding to the references in the paper.*