

Investigate Indistinguishable Points in Semantic Segmentation of 3D Point Cloud

Mingye Xu^{1, 2*}, Zhipeng Zhou^{1, 4*}, Junhao Zhang¹, Yu Qiao^{1, 3†}

¹ShenZhen Key Lab of Computer Vision and Pattern Recognition,
SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

²University of Chinese Academy of Sciences, China

³Shanghai AI Lab, Shanghai, China

⁴SIAT Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society
{my.xu, zp.zhou, zhangjh, yu.qiao}@siat.ac.cn

Abstract

This paper investigates the indistinguishable points (difficult to predict label) in semantic segmentation for large-scale 3D point clouds. The indistinguishable points consist of those located in complex boundary, points with similar local textures but different categories, and points in isolate small hard areas, which largely harm the performance of 3D semantic segmentation. To address this challenge, we propose a novel Indistinguishable Area Focalization Network (IAF-Net), which selects indistinguishable points adaptively by utilizing the hierarchical semantic features and enhances fine-grained features for points especially those indistinguishable points. We also introduce multi-stage loss to improve the feature representation in a progressive way. Moreover, in order to analyze the segmentation performances of indistinguishable areas, we propose a new evaluation metric called Indistinguishable Points Based Metric (IPBM). Our IAF-Net achieves the comparable results with state-of-the-art performance on several popular 3D point cloud datasets e.g. S3DIS and ScanNet, and clearly outperforms other methods on IPBM. Our code will be available at <https://github.com/MingyeXu/IAF-Net>

1 Introduction

Deep learning on point cloud analysis has been attracting more and more attention recently. Among the tasks of point cloud analysis, efficient semantic segmentation of large-scale 3D point cloud is a challenging task with huge applications (Rusu et al. 2008; Chen et al. 2017; Chen et al. 2020). A key challenge is that 3D point cloud semantic segmentation relies on unstructured data which is typically irregularly sampled and unordered. Due to the complexity of large-scale 3D point cloud, this task also requires the understanding of the fine-grained details for each point.

For point cloud semantic segmentation, there exist some areas which are hard to be segmented, and we name these areas as “indistinguishable” areas. In order to analyze the image semantic segmentation results in detail, (Li et al. 2017) divide pixels into different difficulty levels. Inspired by (Li et al. 2017), we can also categorize these “indistinguishable” areas into three types (Figure 1): The first type

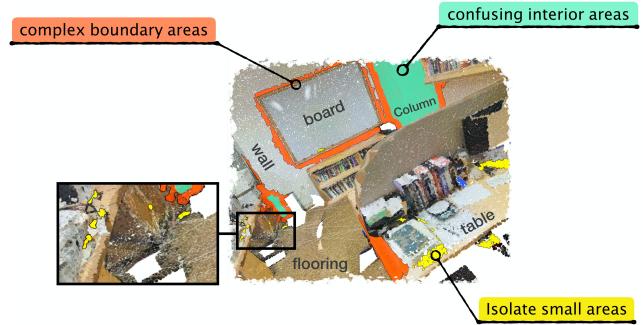


Figure 1: Three types of the indistinguishable areas.

is called **complex boundary areas** (orange areas in Figure 1) which belong to the boundary points (object boundaries and prediction boundaries). In most cases, it is difficult to identify the boundaries between different objects accurately. Because the features of each point are characterized by the information of local regions, the predictions of the boundary points will be over smooth between objects of different categories which are close in Euclidean space. The second type is named **confusing interior areas** (cyan areas in Figure 1) which contain interior points from objects of different categories with similar textures and geometric structures. For example, door and wall have similar appearance which are almost flat and share similar colors. Even for human being, it is hard to identify part of door and wall accurately in these cases. The last type is called **isolate small areas** (yellow areas in Figure 1), which are scattered and hard to be predicted. In addition, objects in the scenes would not be fully captured by the devices because of the occlusion. All of the challenges mentioned above hinder the accuracy of semantic segmentation of 3D point cloud. As far as we know, these “indistinguishable” points are not deeply explored in most of the previous methods (Jiang et al. 2018; Yang et al. 2019) on point cloud semantic segmentation task.

To improve the segmentation performance on indistinguishable points mentioned above, we design an efficient neural network which is able to enhance the features of points especially indistinguishable points. However, this task has two challenges to be addressed: 1) How to discover indistinguishable points adaptively in the training pro-

*M.Xu and Z.Zhou contributed equally.

†Corresponding author.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

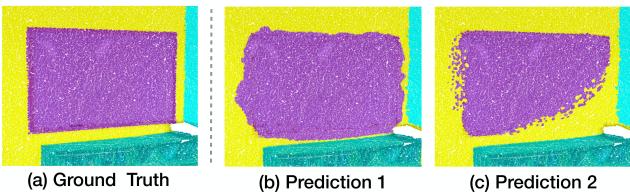


Figure 2: (a) is the ground truth of a room. (b) and (c) are two different predictions which result in similar m-IoU.

cess; 2) How to enhance the features of these points. To this end, we propose a new module called Indistinguishable Areas Focalization (IAF) which can adaptively select indistinguishable points considering hierarchical semantic features. To enhance the features of indistinguishable points, IAF module firstly acquires the fine-grained features and high-level semantic features of indistinguishable points, then enhances the features through a nonlocal operation between these points and the corresponding whole point set. Furthermore, we introduce a multi-stage loss function L_{ms} to strengthen the feature descriptions of corresponding points in each layer. In this way, we can capture the features in a more progressive manner, which guarantees the accuracy of features in each layer.

Mean IoU (m-IoU) and Overall Accuracy (OA) are two widely-used evaluation metrics of 3D semantic segmentation. OA describes the average degree of accuracy which ignores the various distribution of different categories of objects. m-IoU can reflect the accuracy of model on the identification of each category independently. Under certain circumstances (Figure 2 shows), the visualizations of two predictions with similar m-IoU can be totally different in details. In order to cooperate with the indistinguishable points' partitions and to provide a supplementary metric for OA and m-IoU, we propose a novel evaluation metric named Indistinguishable Points Based Metric (IPBM). This evaluation metric focuses on different types of indistinguishable areas. With this evaluation metric, we can evaluate the effectiveness of segmentation methods more objectively and more granularly. It has a certain contribution to the segmentation task evaluation in the future.

The main contributions are summarized as follows,

- We propose the Indistinguishable Areas Focalization (IAF) module which can select indistinguishable points adaptively and enhance the features of each point.
- We utilize the multi-stage loss to strengthen feature descriptions in each layer, which guarantees the features can represent points more accurately in a progressive way.
- Our method achieves the comparable performances with state-of-the-art methods on several popular datasets for 3D point cloud semantic segmentation.
- We introduce Indistinguishable Points Based Metric (IPBM) which focuses on the performances of segmentation methods on different types of indistinguishable areas.

2 Related Work

Point-Based Networks. Point-based networks work on irregular point clouds directly. Inspired by PointNet (Charles et al. 2017) and PointNet++ (Qi et al. 2017b), many recent works (Hu et al. 2020; Han et al. 2020; Zhang, Hua, and Yeung 2019; Xu, Zhou, and Qiao 2020; Wu, Qi, and Fuxin 2019) propose different kinds of modules based on pointwise MLP. ShellNet (Zhang, Hua, and Yeung 2019) introduce ShellConv which can allow efficient neighbor point query simultaneously and resolve point order ambiguity by defining a convolution order from inner to the outer shells on a concentric spherical domain. RandLANet (Hu et al. 2020) utilize a local feature aggregation module to automatically preserve complex local structures by progressively increasing the receptive field for each point. Some works construct novel and efficient point convolutions. A-CNN (Komarichev, Zhong, and Hua 2019) propose a multi-level hierarchical annular convolution which can set arbitrary kernel sizes on each local ring-shaped domain to better capture shape details. KPConv (Thomas et al. 2019) apply a new convolution based operator which uses a set of kernel points to define the area where each kernel weight is applied. However, most methods do not consider the indistinguishable points on point cloud semantic segmentation task specially. By contrast, we propose a novel IAF module which enhances the features of points, especially points in the indistinguishable areas. IAF module in each layer uses a specially designed points selection operation to mine the indistinguishable points adaptively and applies the nonlocal operation to fuse the features between indistinguishable points and corresponding whole point set in each layer. Multi-stage loss is conducive to abstracting representative features in a progressive way.

Local-Nonlocal Mechanism. Nonlocal mechanism has been applied to various tasks of computer vision (Yan et al. 2020; Cao et al. 2019). The pioneer work Nonlocal (Wang et al. 2018b) in video classification present non-local operations as an efficient, simple and generic component for capturing long range dependencies with deep neural networks. It computes the response at a position as a weighted sum of the features at all positions in the input feature maps. Point2Node (Han et al. 2020) utilize both local and non-local operations to dynamically explore the correlation among all graph nodes from different levels and adaptively aggregate the learned features. Local correlation and non-local correlation are used in a serial way which largely enhances nodes characteristic from different scale correlation learning. To apply the local and non-local mechanism in a targeted way, we use the non-local operation to fuse the features of different layers, which help to enhance the features of indistinguishable points. Moreover, the local features in our network are enhanced by using the multi-stage loss progressively.

3 Method

We denote the point cloud as $P = \{p_i \in R^{3+d}, i = 1, 2, \dots, N\}$, where N is the number of points and $3 + d$ denotes the xyz-dimension and additional properties, such

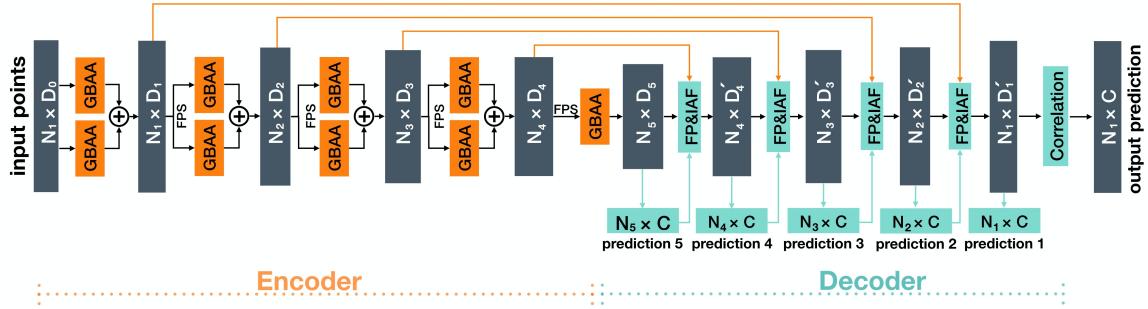


Figure 3: The detailed architecture of our IAF-Net. (N, D) represents the number of points and feature dimension respectively. FPS: Farthest Point Sampling.

as colors, normal vectors. Like previous works, our IAF-Net mainly consists of encoder and decoder modules which are shown in Figure 3. We will show the details of each part in the following subsections.

3.1 Encoder Module: Geometry Based Attentive Aggregation

This subsection describes the encoder module GBAA (Geometry Based Attentive Aggregation) of our network architecture. As Figure 4 shows, the GBAA module is constituted of local feature aggregation and attentive pooling operations.

Local Feature Aggregation

In order to get a better description of each point, we enhance the local features with eigenvalues at each point. We utilize KNN algorithm to get the K -nearest neighbors of each point in Euclidean space. As introduced in (Xu, Zhou, and Qiao 2020), we use the coordinates of neighbors of point p_i to get eigenvalue-tuples denoted by $(\lambda_i^1, \lambda_i^2, \lambda_i^3)$. The original input features of each point are denoted as $x_i^0 = (\lambda_i^1, \lambda_i^2, \lambda_i^3)$. For GBAA module in the l -th layer, we take original points P and output of last layer \mathbf{X}^{l-1} as input. We choose the K -nearest neighbors in Euclidean space and eigenvalue space respectively for each point p_i . Let $\{p_{i_1}, \dots, p_{i_K}\}$ be the K -nearest neighbors of p_i in Euclidean space, and their corresponding features are $\{x_{i_1}^{l-1}, \dots, x_{i_K}^{l-1}\}$. The features of K -nearest neighbors in eigenvalue space of point p_i are $\{x_{i_1}^{l-1}, \dots, x_{i_K}^{l-1}\}$. We define the local feature aggregation operation as $g_{\Theta_l}^1 : \mathbb{R}^{2 \times (3+d)} \times \mathbb{R}^{2 \times D_{l-1}} \rightarrow \mathbb{R}^{D_l}$, where $g_{\Theta_l}^1$ is a nonlinear function with a set of learnable parameters Θ_l , and D_{l-1}, D_l are dimension of output features of l -th and $(l-1)$ -th layer respectively. In our module, $g_{\Theta_l}^1$ is a two-layer 2D-convolution. The local feature aggregation for each point is

$$x_i^{local,l} = g_{\Theta_l}^1(\|_{k=1}^K (p_{i_k} - p_i) \oplus p_i \oplus x_{i_k}^{l-1} \oplus x_{i_k}^{l-1}). \quad (1)$$

where $1 \leq i \leq N$, \oplus is concatenation and $x_i^{local,l} \in \mathbb{R}^{K \times D_l}$. $\|$ is the concatenation among K dimension.

Attentive Pooling

For each point p_i , its local feature aggregation is $x_i^{local,l} \in \mathbb{R}^{K \times D_l}$. Instead of max pooling or average pooling, we ap-

ply an attentive pooling to $x_i^{local,l}$.

$$x_i^l = \sum_{k=1}^K g_{\Theta_l}^2(x_i^{local,l}[k]) \cdot x_i^{local,l}[k]. \quad (2)$$

where $g_{\Theta_l}^2$ is a 1-layer 2D convolution.

Moreover, as Figure 3 shows, we utilize the local feature aggregation and attentive pooling to obtain features of each point from two different receptive fields in order to enhance the representation of each point. The receptive field depends on K . We choose K_1 and K_2 nearest neighbors for feature aggregation and attentive pooling. Finally, we get the output of l -th layer which denoted as $\mathbf{X}^l = \mathbf{X}_{K_1}^l + \mathbf{X}_{K_2}^l$.

3.2 Decoder Module: FP & IAF

This subsection elucidates the decoder module FP & IAF which is shown in Figure 5. For convenience, we use Y^l to represent the features of decoder module in the l -th layer. The decoder module contains two parts: feature propagation and indistinguishable areas focalization (IAF).

Feature Propagation

In encoder part, the original point set is sub-sampled. We adopt a hierarchical propagation strategy with distance based interpolation and across level skip links as (Qi et al. 2017b). In a feature propagation process, we propagate point features $Y^l \in \mathbb{R}^{N_l \times D_l}$ and label predictions $Z^l \in \mathbb{R}^{N_l \times C}$ to $Y_{up}^{l-1} \in \mathbb{R}^{N_{l-1} \times D_{l-1}}$ and $Z_{up}^{l-1} \in \mathbb{R}^{N_{l-1} \times C}$, where N_l and N_{l-1} ($N_l \leq N_{l-1}$) are point set size of input and output of the l -th layer, C is the number of categories for semantic segmentation.

$$y_{i,fp}^{l-1} = g_{\Psi_{l-1}}(y_{i,up}^l \oplus x_i^{l-1}). \quad (3)$$

where $g_{\Psi_{l-1}}$ is convolution operations with batch normalization and activate-function, x_i^{l-1} is the features from the encoder module in the $(l-1)$ -th layer.

Indistinguishable Areas Focalization

Indistinguishable points mining. In order to discover indistinguishable points adaptively in the training process, both low level geometric and high level semantic information can be used to mine these points. Local difference is the difference between each point and its neighbors. To a certain extent, local difference reflects the distinctiveness of each point which depend on low-level geometry, latent space and

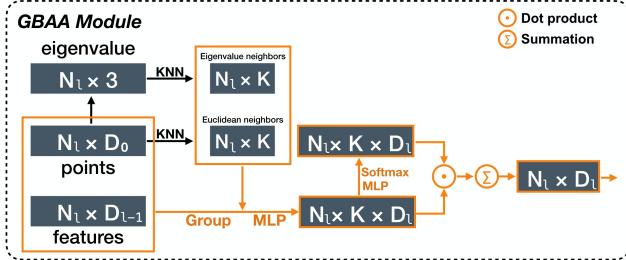


Figure 4: Encoder Module: Geometry Based Attentive Aggregation. The input are point-wise coordinates, colors and features. In GBAA, we aggregate features in both eigenvalue space and Euclidean space, then use attentive pooling to generate the output feature of each point.

high-level semantic features. So we use local difference as a criterion of mining indistinguishable points. For each point p_i , we get the K -nearest neighbors in Euclidean space, then we have the following local difference of each point in each layer:

$$LD_1^l(p_i) = \sum_{k=1}^K \|p_i - p_{i_k}\|_2. \quad (4)$$

$$LD_2^l(p_i) = \sum_{k=1}^K \|z_{i,up}^{l-1} - z_{i_k,up}^{l-1}\|_2. \quad (5)$$

$$LD_3^l(p_i) = \sum_{k=1}^K \|y_{i,fp}^{l-1} - y_{i_k,fp}^{l-1}\|_2. \quad (6)$$

Then we accumulate these local differences together:

$$LD^l(p_i) = \sum_{j=1}^3 \mu_j \times \frac{LD_j^l(p_i) - \min(LD_j^l(p))}{\max(LD_j^l(p)) - \min(LD_j^l(p))}. \quad (7)$$

where $0 \leq \mu_j \leq 1$.

LD^l indicates the accumulation of fine-grained features' difference LD_3^l among each point's local region, high-level semantic predictions' local difference LD_2^l and low-level properties' local difference LD_1^l , where $\{\mu_j\}$ is used to adjust the weight of these three local differences. We align the points in a descending order according to LD^l , then choose top $M_{l-1} = \frac{N_{l-1}}{\tau}$ points as the indistinguishable points. There are three types of points mentioned in Introduction as Figure 1, 5 shows. These indistinguishable points change dynamically as the network updates iteratively (Figure 6). It is noted that at the beginning of training, the indistinguishable points are distributed over the areas where the original properties (coordinates and colors) change rapidly. As the training process goes on, the indistinguishable points locate at the indistinguishable areas mentioned in the introduction.

Indistinguishable points set focalization. We aggregate intermediate features and label predictions of the indistinguishable points, then use the MLP (Hornik 1991) to extract the features for indistinguishable points separately.

$$x_{j \in M_{l-1}}^{l-1} = g_{\Omega_l}^1(y_{j,fp}^{l-1} \oplus z_{j,up}^{l-1}) \in \mathbb{R}^{D_{l-1}}. \quad (8)$$

where $j \in M_{l-1}$ means that the points belong to the indistinguishable points set and $g_{\Omega_l}^1$ is MLP operations.

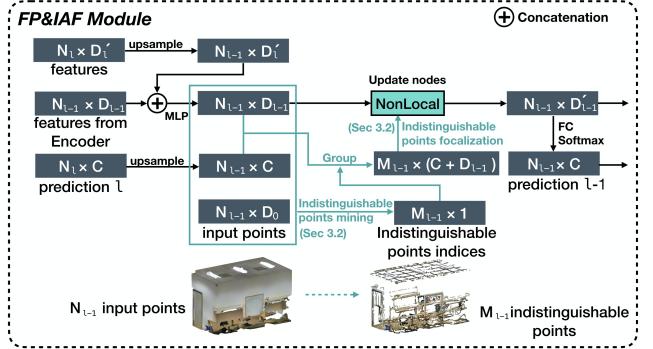


Figure 5: Decoder Module: It contains two stages: feature propagation and indistinguishable areas focalization.

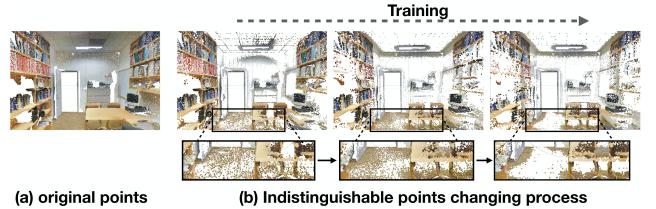


Figure 6: The adaptive change process of indistinguishable points during the training process. The background is colored white. **Best viewed in color with 300% zoom.**

Update nodes. To enhance the features of points, especially the indistinguishable points, here we utilize the Non-Local mechanism to update the features of all points by the following equations and it can enhance the features of indistinguishable points implicitly.

$$y_i^{l-1} = g_{\Omega_l}^2 \left(\sum_{j \in M_{l-1}} (g_{\Omega_l}^3(x_j^{l-1}) \odot g_{\Omega_l}^4(y_{i,fp}^{l-1})) \cdot g_{\Omega_l}^5(y_{i,fp}^{l-1}) \right). \quad (9)$$

where $g_{\Omega_l}^2, g_{\Omega_l}^3, g_{\Omega_l}^4, g_{\Omega_l}^5$ are MLPs. Also, we have the label prediction probability z_i^{l-1} of point p_i in $(l-1)$ -th layer.

$$z_i^{l-1} = \text{Softmax}(g_{\Omega_{l-1}}^6(y_i^{l-1}) \in \mathbb{R}^C). \quad (10)$$

3.3 Loss for Segmentation

In order to progressively refine the features of indistinguishable areas, we apply the multi-stage loss as follows,

$$L_{ms}^l = \text{CrossEntropy}(Z_{gt}^l, Z^l). \quad (11)$$

where $Z_{gt}^l \in \mathbb{R}^{N^l \times 1}$ is the ground truth points' labels in l -th layer.

As the output of the last layer is y_i^1 , inspired by (Han et al. 2020), we use the self correlation, local correlation and non-local correlation operation to augment features of each point p_i . Finally, we get features of each point p_i as the accumulation of three correlations' outputs. The final loss for training is as follows:

$$L_f = \sum_{l=1}^5 L_{ms}^l + L_p. \quad (12)$$

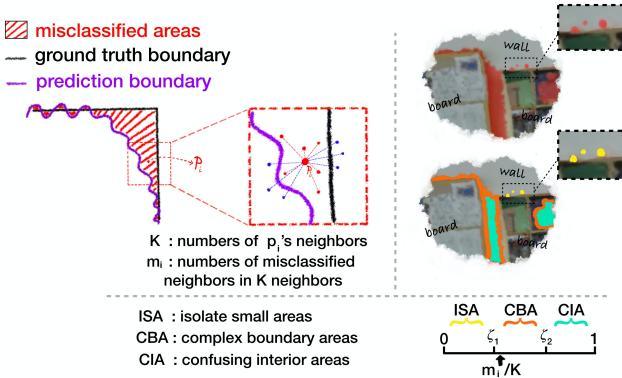


Figure 7: The evaluation process of the indistinguishable points based metric (IPBM). The black line is the boundary of the ground truth, the purple is the prediction boundary, and the red areas are misclassified areas. ζ_1, ζ_2 are the parameters for partitioning the three types of indistinguishable points.

where the L_p is the loss between final label predictions Z and ground truth labels Z_{gt} .

3.4 Indistinguishable Points Based Metric

To better distinguish the effect of different methods in 3D semantic segmentation, we propose a novel evaluation metric named “Indistinguishable Points Based Metric” (IPBM). This evaluation metric focuses on the effectiveness of segmentation methods on the indistinguishable areas.

For the whole points $P = \{p_1, p_2, \dots, p_N\}$, we have the predictions $Pred = \{z_i, 1 \leq i \leq N\}$ and ground truth labels $Label = \{z_{i,gt}, 1 \leq i \leq N\}$. Figure 7 shows the processing details of the IPBM. Firstly, for point p_i satisfying the factor $z_i \neq z_{i,gt}$, its neighbors in Euclidean space are $\{z_{ij}, 1 \leq j \leq K\}$. Then we denote the number of neighbor points that satisfy $z_{ik} \neq z_{i_k,gt}$ as m_i for each point p_i . Next, we divide interval $[0, 1]$ (domain of $\frac{m_i}{K}$) into three partitions with endpoints as $0, \zeta_1, \zeta_2, 1$.

We determine $\zeta_1 = 0.33, \zeta_2 = 0.66$ empirically by considering the curve in Figure 8. To be more specific, Figure 8 shows that the growth trend of number of points with value $\frac{m_i}{K}$ on S3DIS dataset. The curve can be divided into three partitions. From a large number of visualizations, we find that these three partitions roughly reflect the value distribution of the three types of indistinguishable areas. The examples of visualization under different choices of ζ_1, ζ_2 are shown in Figure 9.

Finally, we use $\frac{S_1}{N}, \frac{S_2}{N}, \frac{S_3}{N}$ as our new evaluation metric where S_1, S_2, S_3 are number of points in three types of indistinguishable areas. As Figure 9 shows, $\frac{S_1}{N}$ is used to evaluate the method’s performance on isolate small areas (colored yellow), $\frac{S_2}{N}$ is for complex boundary areas (colored orange), and $\frac{S_3}{N}$ is for confusing interior areas (colored cyan).

For a more comprehensive evaluation, three subsets of the point cloud are sampled for the above evaluation. As Figure 10 shows, they are original point cloud, category boundary point cloud and geometry boundary point cloud. The specific

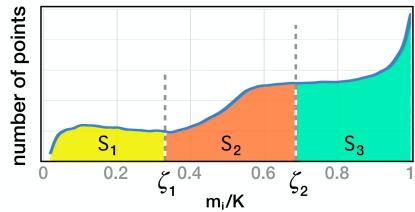


Figure 8: The curve of the points number as $\frac{m_i}{K}$ changes on S3DIS dataset. $\zeta_1 = 0.33, \zeta_2 = 0.66$. S_1, S_2, S_3 are number of points in three types of indistinguishable areas.

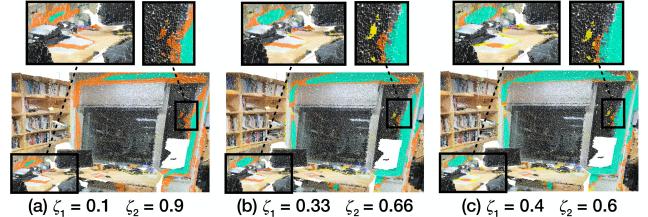


Figure 9: The visual experiments of ζ_1, ζ_2 on S3DIS dataset. Yellow areas are isolate small areas; Orange areas are complex boundary areas; Cyan areas are confusing interior areas. **Best viewed in color with 300% zoom.**

methods of subset point cloud acquisition is explained in the supplementary materials.

4 Experiments

4.1 Experimental Evaluations on Benchmarks

S3DIS Semantic Segmentation

Dateset. The S3DIS (Armeni et al. 2016) dataset contains six sets of point cloud data from three different buildings (including 271 rooms). We follow (Boulch 2020) to prepare the dataset.

For training, we randomly select points in the considered point cloud, and extract all points in an infinite column centered on this point, where the column section is 2 meters. For each column, we randomly select 8192 points as the input points. During the testing, for a more systematic sampling of the space, we compute a 2D occupancy pixel map with pixel size 0.1 meters. Then, we consider each occupied cell as a center for a column (same for training). Finally, the output scores are aggregated (sum) at point level and points not seen by the network receive the label of its nearest neighbors. Following (Boulch 2020), we report the results under two settings: testing on Area 5 and 6-fold cross validation.

Performance Comparison. Table 1 and Table 3 show the quantitative results of different methods under two settings mentioned above, respectively. Our method achieves on-par performance with the SOTA methods. It is noted that some methods (Qi et al. 2017a; Zhao et al. 2019) use small column section(1 meter), and it may not contain enough holistic information. It will be more common to the situation which the column section cannot contain the whole object. By contrast, our method use IAF module to deal with the in-

7 Acknowledgments

This work was supported in part by the Shanghai Committee of Science and Technology, China (Grant No. 20DZ1100800), in part by the National Natural Science Foundation of China under Grant (61876176, U1713208), and in part by the Shenzhen Basic Research Program (CXB201104220032A), Guangzhou Research Program (201803010066).

References

- Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *CVPR*.
- Boulch, A. 2020. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics* .
- Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* .
- Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCV Workshops*.
- Charles, R. Q.; Su, H.; Kaichun, M.; and Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *CVPR* .
- Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *CVPR*.
- Chen, Z.; Zeng, W.; Yang, Z.; Yu, L.; Fu, C. W.; and Qu, H. 2020. LassoNet: Deep Lasso-Selection of 3D Point Clouds. *IEEE Transactions on Visualization and Computer Graphics* 26(1): 195–204. doi:10.1109/TVCG.2019.2934332.
- Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*.
- Han, W.; Wen, C.; Wang, C.; Li, X.; and Li, Q. 2020. Point2Node: Correlation Learning of Dynamic-Node for Point Cloud Feature Modeling. *AAAI* .
- Hornik, K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks* .
- Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; and Markham, A. 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *CVPR* .
- Huang, Q.; Wang, W.; and Neumann, U. 2018. Recurrent slice networks for 3d segmentation of point clouds. In *CVPR*.
- Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; and Lu, C. 2018. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652* .
- Komarichev, A.; Zhong, Z.; and Hua, J. 2019. A-CNN: Annularly Convolutional Neural Networks on Point Clouds. *CVPR* .
- Landrieu, L.; and Simonovsky, M. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*.
- Li, X.; Liu, Z.; Luo, P.; Change Loy, C.; and Tang, X. 2017. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*.
- Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. In *NIPS*.
- Lin, Y.; Yan, Z.; Huang, H.; Du, D.; Liu, L.; Cui, S.; and Han, X. 2020. FPConv: Learning Local Flattening for Point Convolution. In *CVPR*.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*.
- Rusu, R. B.; Marton, Z. C.; Blodow, N.; Dolha, M.; and Beetz, M. 2008. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems* .
- Tatarchenko, M.; Park, J.; Koltun, V.; and Zhou, Q.-Y. 2018. Tangent convolutions for dense prediction in 3d. In *CVPR*.
- Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; and Savarese, S. 2017. Segcloud: Semantic segmentation of 3d point clouds. In *3DV*.
- Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. *arXiv preprint arXiv:1904.08889* .
- Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; and Shan, J. 2019. Graph Attention Convolution for Point Cloud Semantic Segmentation. In *CVPR*.
- Wang, S.; Suo, S.; Ma, W.-C.; Pokrovsky, A.; and Urtasun, R. 2018a. Deep parametric continuous convolutional neural networks. In *CVPR*.
- Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local Neural Networks. *CVPR* .
- Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2018c. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829* .
- Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*.
- Xu, M.; Zhou, Z.; and Qiao, Y. 2020. Geometry Sharing Network for 3D Point Cloud Classification and Segmentation. *AAAI* .
- Yan, X.; Zheng, C.; Li, Z.; Wang, S.; and Cui, S. 2020. PointASNL: Robust Point Clouds Processing using Nonlocal Neural Networks with Adaptive Sampling. In *CVPR*.
- Yang, J.; Zhang, Q.; Ni, B.; Li, L.; Liu, J.; Zhou, M.; and Tian, Q. 2019. Modeling point clouds with self-attention and gumbel subset sampling. In *CVPR*.

- Ye, X.; Li, J.; Huang, H.; Du, L.; and Zhang, X. 2018. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *ECCV*.
- Zhang, C.; Luo, W.; and Urtasun, R. 2018. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *3DV*.
- Zhang, Z.; Hua, B.-S.; and Yeung, S.-K. 2019. ShellNet: Efficient Point Cloud Convolutional Neural Networks using Concentric Shells Statistics. In *ICCV*.
- Zhao, H.; Jiang, L.; Fu, C.-W.; and Jia, J. 2019. PointWeb: Enhancing local neighborhood features for point cloud processing. In *CVPR*.

References

- Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *CVPR*.
- Boulch, A. 2020. ConvPoint: Continuous convolutions for point cloud processing. *Computers & Graphics* .
- Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203* .
- Cao, Y.; Xu, J.; Lin, S.; Wei, F.; and Hu, H. 2019. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *ICCV Workshops*.
- Charles, R. Q.; Su, H.; Kaichun, M.; and Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *CVPR* .
- Chen, X.; Ma, H.; Wan, J.; Li, B.; and Xia, T. 2017. Multi-view 3d object detection network for autonomous driving. In *CVPR*.
- Chen, Z.; Zeng, W.; Yang, Z.; Yu, L.; Fu, C. W.; and Qu, H. 2020. LassoNet: Deep Lasso-Selection of 3D Point Clouds. *IEEE Transactions on Visualization and Computer Graphics* 26(1): 195–204. doi:10.1109/TVCG.2019.2934332.
- Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*.
- Han, W.; Wen, C.; Wang, C.; Li, X.; and Li, Q. 2020. Point2Node: Correlation Learning of Dynamic-Node for Point Cloud Feature Modeling. *AAAI* .
- Hornik, K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural networks* .
- Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigoni, N.; and Markham, A. 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *CVPR* .
- Huang, Q.; Wang, W.; and Neumann, U. 2018. Recurrent slice networks for 3d segmentation of point clouds. In *CVPR*.
- Jiang, M.; Wu, Y.; Zhao, T.; Zhao, Z.; and Lu, C. 2018. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652* .
- Komarichev, A.; Zhong, Z.; and Hua, J. 2019. A-CNN: Annularly Convolutional Neural Networks on Point Clouds. *CVPR* .
- Landrieu, L.; and Simonovsky, M. 2018. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*.
- Li, X.; Liu, Z.; Luo, P.; Change Loy, C.; and Tang, X. 2017. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*.
- Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. In *NIPS*.
- Lin, Y.; Yan, Z.; Huang, H.; Du, D.; Liu, L.; Cui, S.; and Han, X. 2020. FPCConv: Learning Local Flattening for Point Convolution. In *CVPR*.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*.
- Rusu, R. B.; Marton, Z. C.; Blodow, N.; Dolha, M.; and Beetz, M. 2008. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems* .
- Tatarchenko, M.; Park, J.; Koltun, V.; and Zhou, Q.-Y. 2018. Tangent convolutions for dense prediction in 3d. In *CVPR*.
- Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; and Savarese, S. 2017. Segcloud: Semantic segmentation of 3d point clouds. In *3DV*.
- Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. *arXiv preprint arXiv:1904.08889* .
- Wang, L.; Huang, Y.; Hou, Y.; Zhang, S.; and Shan, J. 2019. Graph Attention Convolution for Point Cloud Semantic Segmentation. In *CVPR*.
- Wang, S.; Suo, S.; Ma, W.-C.; Pokrovsky, A.; and Urtasun, R. 2018a. Deep parametric continuous convolutional neural networks. In *CVPR*.
- Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018b. Non-local Neural Networks. *CVPR* .
- Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2018c. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829* .
- Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*.
- Xu, M.; Zhou, Z.; and Qiao, Y. 2020. Geometry Sharing Network for 3D Point Cloud Classification and Segmentation. *AAAI* .
- Yan, X.; Zheng, C.; Li, Z.; Wang, S.; and Cui, S. 2020. PointASNL: Robust Point Clouds Processing using Nonlocal Neural Networks with Adaptive Sampling. In *CVPR*.

Yang, J.; Zhang, Q.; Ni, B.; Li, L.; Liu, J.; Zhou, M.; and Tian, Q. 2019. Modeling point clouds with self-attention and gumbel subset sampling. In *CVPR*.

Ye, X.; Li, J.; Huang, H.; Du, L.; and Zhang, X. 2018. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. In *ECCV*.

Zhang, C.; Luo, W.; and Urtasun, R. 2018. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *3DV*.

Zhang, Z.; Hua, B.-S.; and Yeung, S.-K. 2019. ShellNet: Efficient Point Cloud Convolutional Neural Networks using Concentric Shells Statistics. In *ICCV*.

Zhao, H.; Jiang, L.; Fu, C.-W.; and Jia, J. 2019. PointWeb: Enhancing local neighborhood features for point cloud processing. In *CVPR*.