IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 31, 2021, accepted June 21, 2021, date of publication July 5, 2021, date of current version September 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3094624

PointMTL: Multi-Transform Learning for Effective 3D Point Cloud Representations

YIFAN JIAN[1], YUWEI YANG[2], ZHI CHEN[1], XIANGUO QING[1], YANG ZHAO[1], LIANG HE[1], XUEKUN CHEN[1], AND WEI LUO[1]

[1]Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, Chengdu 610064, China
[2]College of Electronics and Information Engineering, Sichuan University, Chengdu 610064, China

Corresponding author: Zhi Chen (chenzhinpic@126.com)

ABSTRACT Effectively learning and extracting the feature representations of 3D point clouds is an important yet challenging task. Most of existing works achieve reasonable performance in 3D vision tasks by modeling the relationships among points appropriately. However, the feature representations are only learned with a specific transform through these methods, which are easy to overlap and thus limit the representation ability of the model. To address these issues, we propose a novel Multi-Transform Learning framework for point clouds (PointMTL), which can extract diverse features from multiple mapping transform to obtain richer representations. Specifically, we build a module named Multi-Transform Encoder (MTE), which encodes and aggregates local features from multiple non-linear transforms. To further explore global context representations, a module named Global Spatial Fusion (GSF) is proposed to capture global information and selectively fuse with local representations. Moreover, to guarantee the richness and diversity of learned representations, we further propose a Spatial Independence Criterion (SIC) strategy to enlarge the differences between the transforms and reduce information redundancies. In contrast to previous works, our approach fully exploits representations from multiple transforms, thus having strong expressiveness and good robustness for point clouds related tasks. The experiments on three typical tasks (i.e., semantic segmentation on S3DIS and ScanNet, part segmentation on ShapeNet and shape classification on ModelNet40) demonstrates the effectiveness of our method.

INDEX TERMS 3D point clouds, feature representations, multi-transform learning, semantic segmentation.

I. INTRODUCTION

The rapid development of sensor technology enables a more convenient way to obtain 3D point clouds data, which boosts the research for many intelligent systems, such as autonomous driving [1]–[3], robotics [4], [5] and virtual/augmented reality [6]–[8]. However, a major challenge is that the raw point clouds are typically unstructured (i.e., irregular, disordered and uneven density as shown in Fig. 1). Therefore, the recent success of deep Convolutional Neural Networks (CNNs) developed for the structured 2D data cannot be applied directly for the analysis of unstructured 3D point clouds. This makes the typical tasks of point clouds analysis, e.g., semantic segmentation, part segmentation and shape classification, still remain challenging.

To tackle with such type of unstructured data, some early works [9]–[15] have transformed the point clouds to regular multi-view images or voxels for a direct application of deep CNNs. Nevertheless, both multi-view images and voxels usually cause the loss of the inherent geometric information of point clouds. Moreover, such transformations require a high computational complexity, which is infeasible for

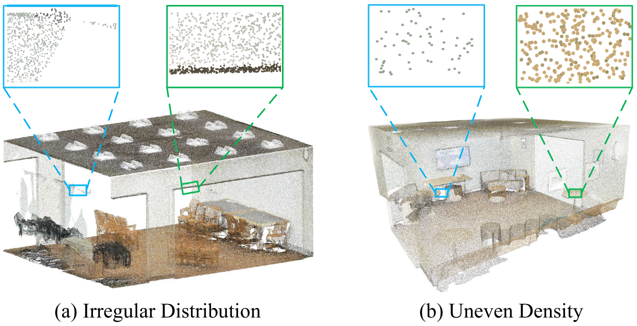

(a) Irregular Distribution (b) Uneven Density

FIGURE 1. Typical properties of 3D point clouds.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Zhang.

real-world applications. In contrast, some works have emerged to directly process point clouds. The pioneering PointNet [16] first learns the point-wise features using shared Multi-Layer Perceptrons (MLPs) with the consideration of permutation invariance. And then gathers such features to obtain a compact representation via a max-pooling operation. However, PointNet extracts a global representation without considering any fine-grained local information, which is proven to be effective for capturing the details of point clouds. To overcome such drawback, some other works [17]–[22] further explore the local relations by partitioning the raw point clouds into a set of local subsets and then hierarchically aggregating them into a high-level contextual representation. Another category of works [23]–[30] represent point clouds by constructing graph structures, and perform a convolution-like operation on the spatially close nodes for aggregating the information along the neighbor nodes in a sub-graph. Based on the nature of point clouds, the receptive field of such convolution-like operation in a graph structure can dynamically accommodate to the shape of the objects for encoding the local contextual information.

Most of aforementioned methods appropriately capture the contextual representations of the point clouds with a specific transform. However, due to the close distance between different objects, they may result in feature overlap (as shown in the dotted circle from Fig. 2). Representative works [31]–[35] in other fields have demonstrated that more powerful models can be built by learning diversified representations of multiple transforms.

Inspired by the multi-transform approaches, we propose a novel Multi-Transform Learning framework for 3D Point Cloud (PointMTL), which can effectively learn and fuse different feature representations from the embedding transforms. Specifically, the point clouds first go through the Multi-Transform Encoder (MTE), which transforms the original point clouds separately to obtain the representation of different transforms. Then, local receptive fields are dynamically constructed by K-Nearest Neighbors (KNN) algorithm [36] and aggregated by attention mechanism. In addition, the Global Spatial Fusion (GSF) module is introduced to extract the global contextual representations. Finally, to guarantee more diverse feature representations, Spatial Independence Criterion (SIC) module is proposed as a constraint for feature learning. All the modules constitute the MTL units, which are added between the encoding layers, and the decoding layers are optional for different tasks. Our proposed method has been evaluated on three different tasks (four benchmarks), i.e., semantic segmentation on S3DIS [37] and ScanNet [38], part segmentation on ShapeNet [39], shape classification on ModelNet40 [11]. Experimental results have demonstrated the effectiveness of our model.

Our contributions can be summarized as follows:

- We propose a PointMTL framework that projects the original point clouds data with multiple feature transforms. This framework partly solves the feature overlap problem in a specific transform, and obtains richer information from multiple transforms.
- We introduce the MTE and GSF module, that can encode local and global feature representations from multiple transforms respectively.
- We propose a novel SIC strategy to enlarge the differences between multiple features. By minimizing the similarity of features, SIC can effectively guarantee information diversity and help the network to capture more meaningful representations.
- Our model performs better than other methods on various tasks and benchmarks, which demonstrate its effectiveness.

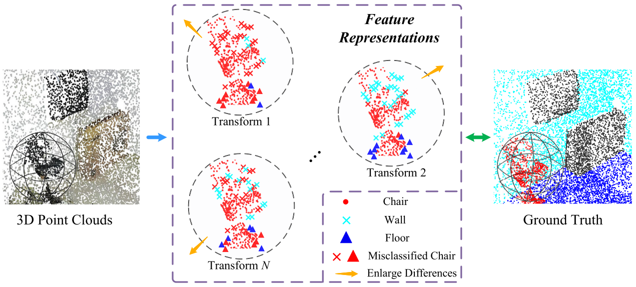


FIGURE 2. Illustration of the feature representations from multiple transforms. The crosses and triangles represent different classes, and the selected classes are highlighted in red. Learning feature representations from a specific transform can result in different degrees of feature overlap.

II. RELATED WORKS

A. INDIRECT METHODS FOR POINT CLOUDS

With the success of deep CNNs in 2D images, one kind of indirect methods usually transform point clouds into a set of images rendered from different views and utilize deep CNNs to process the rendered view images [9], [12], [14], [40]. However, the view-based methods usually lead to the loss of the inherent geometric relationship during the view rendering. Therefore, the view-based methods tend to fail when dealing with the dense labeling tasks which requires rich structural and contextual information associated with the point clouds.

Another kind of indirect methods transform point clouds into the regular volumetric occupancy grids [10], [11], [35]. Then, 3D deep CNNs are used to extract features from the corresponding 3D voxel structures. However, the volumetric-based methods usually need high computational complexity, which leads the 3D voxel structures to a very low spatial resolution (e.g., typically 64 × 64 × 64). To address this issue, some improved methods (e.g., KD-Net [41] and OctNet [42]) only consider the occupied voxels. Nevertheless, the low-resolution voxel operation inevitably involves lots of geometric information loss, decreasing the effectiveness of the extracted features.

B. DIRECT METHODS FOR POINT CLOUDS

To overcome the problems encountered by the view-based and volumetric-based methods, recent methods usually process 3D point clouds directly. The pioneering work is PointNet [16], which extracts point-wise features by using shared Multiple Layer Perceptions (MLPs). However, PointNet results in global representations which loses the fine-grained local information. Some extended works, such as PointNet++ [17] and PointWeb [21], further exploit the local information by partitioning the point clouds into local sub-regions. Meanwhile, some other works modify the traditional convolution to adapt the unstructured 3D point clouds. For example, Atzmon *et al.* [43] propose parametric continuous convolutions which are performed on the raw point clouds. Li *et al.* [19] first use a χ-transformation to capture a potentially canonical order of the point clouds, and then apply typical convolutions on them. Feng *et al.* [44] propose a local attention-edge convolution operation to learn rich contextual correlations on the point clouds. To achieve the translation and permutation invariance, Wu *et al.* [22] extend the 3D continuous convolution operation for point clouds. Komarichev *et al.* [45] design a ring convolution operator to capture contextual signatures for each point.

Recently, graph structures are considered [46]–[48] which can naturally preserve the geometrical cues of point clouds. For example, Landrieu and Simonovsky [23] first construct super-graphs on large-scale point clouds, then graph convolutions are applied to learn the super-point signatures. Wang *et al.* [24] construct a local neighbor graph on point clouds, and then edge convolutions are performed on the neighbor graph. Wang *et al.* [26] propose a graph attention convolution to weight the neighbor points, helping to capture the local structured features. With Graph Convolution Networks (GCNs), Liu *et al.* [27] propose a dynamic agglomeration operation for point clouds. Wang *et al.* [49] utilize graph attention blocks to exploit the local and global structural information of point clouds. Technically, Han *et al.* [29] and Yan *et al.* [50] both propose to dynamically capture the self, local and non-local correlations among points, and effectively integrate the learned features. In addition, Hu *et al.* [30] propose a lightweight network and a random sampling mechanism for efficient point cloud processing. Our method also directly processes the point clouds by constructing the graph structure, fully considering local and global contextual information. However, unlike the aforementioned methods, we adopt a multi-transform strategy to extract diverse features.

C. MULTI-TRANSFORM FEATURE LEARNING

Due to the close distance between different categories of points, feature learning with a specific transform (like the aforementioned methods) often results in feature overlap and performance decrease. Instead of single-transform feature learning, multi-transform feature learning has been widely used in many research fields [32], [33], [51]–[53]. It demonstrates a powerful ability to acquire diverse features. For example, the works in [31]–[34] have demonstrated that multi-head attention, compared with single attention, can enrich different aspects of features. To explore the differences between features learned in multiple embedding transforms, Clark *et al.* [54] examine the performance of different transforms in the same layer and the effect of the transforms in the different layers. They observe that the transforms in the same layer often exhibit similar behaviors. Thus, for better feature learning, some additional supervision information [55] needs to be added to the network. Therefore, some works [32], [56], [57] introduce extra penalty items to enlarge the differences between learned features from multiple transforms.

Learning features from multiple transforms can help the model obtain richer representations, but these representations need to be fused efficiently to overcome the adverse effects of unnecessary information. The gating strategy [58]–[62] can suit this requirement. Since the gating strategy can fully fuse multi-level features, Wang *et al.* [49] introduce it to integrate the information of each point with the surrounding points. Meanwhile, with the gating strategy, Han *et al.* [29] aggregate various levels of correlation representations in a non-linear and data-adaptive way.

Different from the above methods, our network introduces multiple transforms for feature learning. In addition, we use an novel gating strategy to effectively enhance the flow of useful information and suppress useless information. We also propose the SIC strategy to ensure the feature diversity of transforms.

III. OUR METHODOLOGY

A. ARCHITECTURE OVERVIEW

Fig. 3 shows the overall structure of our framework. The framework adopts an encoder-decoder structure, which aggregates point clouds layer-by-layer. More specifically, the framework first takes 3D point clouds as input, and applies Multi-Transform Encoder (MTE) to extract diverse features. The MTE utilizes multiple MLPs to transform points for different features, and then encodes the local information with the attention mechanism. Afterwards, the Global Spatial Fusion (GSF) module is applied to the original features and other transforms for capturing the global information. Finally, the framework introduces the Spatial Independence Criterion (SIC) to enlarge the differences between transforms and reduce information redundancies. The above three modules (MTE, GSF and SIC) constitute the multi-transform learning (MTL) units, which are introduced into encoding layers to capture diverse feature representations. The decoding layers are optional for different tasks. In the following section, we will elaborate the key components.

B. MULTI-TRANSFORM ENCODER

In the point cloud feature representation, it is a common practice to build local regions within specific radius to aggregate neighbor points. However, due to the uniqueness and encoding limitations, some implicit features of points cannot be well explored. At the same time, the aggregation of neighbors' information is based on a fixed radius, which weakens the network's ability to dynamically explore a larger range of

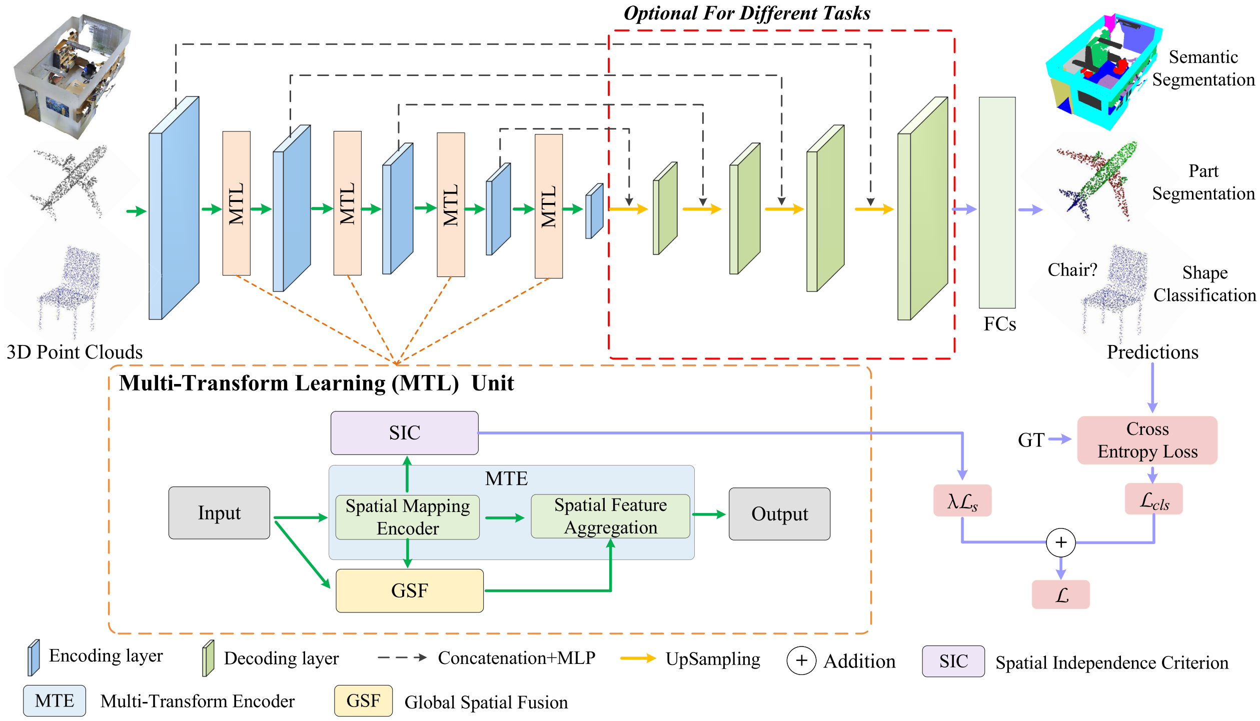


FIGURE 3. The overall architecture of the PointMTL. The MTE module is used to transform the original feature into multiple feature representations, and aggregates the local features. The GSF module aims to extract global information from various transforms. The SIC module is introduced to enlarge the differences between transforms to obtain richer representations. All the above-mentioned modules constitute multi-transform learning (MTL) units, which are added to encoding layers. The decoding layers are optional for different tasks.

point sets. Based on the above facts, we propose the Multi-Transform Encoder (MTE) to better encode and fuse point features. The structure of the MTE is shown in Fig. 4. This module mainly consists of two parts, i.e., Spatial Mapping Encoder and Spatial Feature Aggregation.

1) SPATIAL MAPPING ENCODER

The spatial mapping encoder is proposed to obtain diverse representations from multiple transforms. Specifically, given a point cloud $P^0 = \{p_1^0, p_2^0, \cdots, p_N^0\} \in \mathbb{R}^{3+F}$, where P^0 represents the point clouds without any transformation, and N is the number of points. Each point in P^0 contains 3-dimensional xyz coordinates and F-dimensional feature representations. To obtain the point cloud representations of multiple transforms, we apply multiple independent MLPs to P^0, which can be defined as follows:

$$P^i = MLP_i\left(P^0\right), \quad i = 1, \ldots, M \tag{1}$$

where P^i denotes the point cloud with the i-th transform, M is the number of transforms. In this paper, we use a MLP to instantiate the transform. Through such MLPs structure, high-dimensional implicit features can be extracted by transforms. In addition, the parameters of MLPs are not shared, so the feature representations with multiple transforms are diversified.

To achieve above goals, we construct attentive graphs with multiple transforms. Consider a graph $G^j(V, E)$ derived from P^j, $j = 0, \ldots, M$. V and E define the set of points and edges, respectively. In addition, we denote $\mathcal{N}^j(c) = \{k : (c, k) \in E\} \cup \{c\}$ as the neighbor set of center point c. c and $\mathcal{N}^j(c)$ are obtained by the Farthest Point Sampling (FPS) [17] and the K-Nearest Neighbors (KNN) algorithm [36] respectively. In this way, the local receptive fields are dynamically explored due to the diverse representations

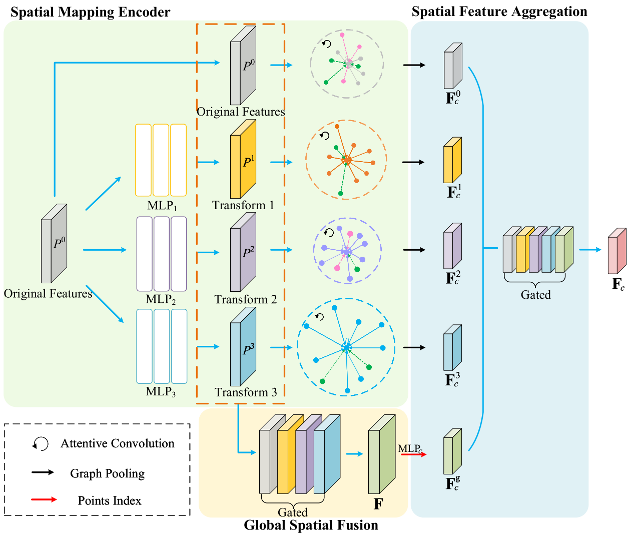


FIGURE 4. Illustration of the MTE and GSF. The MTE is divided into two parts, Spatial Mapping Encoder and Spatial Feature Aggregation. The GSF is used to obtain global information from multiple transforms.

with multiple transforms. However, the neighbor points in $\mathcal{N}^j(c)$ may contain feature of other classes. Therefore, they should be selectively aggregated in a local region. To this end, we adopt the attention mechanism [26] to effectively handle the size-varying neighbors. Specifically, defining $H^j = \{h_1^j, h_2^j, \cdots, h_N^j\} \in \mathbb{R}^3$ and $F^j = \{f_1^j, f_2^j, \cdots, f_N^j\} \in \mathbb{R}^F$ as the set of xyz coordinates and corresponding features of P^j, respectively. Then, relative position Δh_{ck}^j and relative features Δf_{ck}^j are used to measure the spatial relationships and feature differences between neighbor points and center points. The formulas are as follows:

$$\Delta h_{ck}^j = h_k^j - h_c^j, \ k \in \mathcal{N}^j(c) \tag{2}$$

$$\Delta f_{ck}^j = MLP_s^j\left(f_k^j\right) - MLP_s^j\left(f_c^j\right), \quad k \in \mathcal{N}^j(c) \tag{3}$$

where $MLP_s^j : \mathbb{R}^F \rightarrow \mathbb{R}^S$ is a MLP, S denotes the encoded feature dimension. Combined with the above relative information, attention weights $\boldsymbol{W}_{ck}^j$ are defined as:

$$\boldsymbol{W}_{ck}^j = softmax\left(MLP_w^j\left(\Delta h_{ck}^j \oplus \Delta f_{ck}^j\right)\right), \tag{4}$$

where $MLP_w^j : \mathbb{R}^{3+F} \rightarrow \mathbb{R}^S$, $\oplus$ denotes the concatenation operation. Finally, the features of neighbor points are weighted and summed as follows:

$$\mathbf{F}_c^j = \sum_{k \in \mathcal{N}(c)} \boldsymbol{W}_{ck}^j \odot MLP_s^j(f_k^j), \ \mathbf{F}_c^j \in \mathbb{R}^S \tag{5}$$

where $\mathbf{F}_c^j$ denotes the encoded features of central points with j-th transform, and $\odot$ represents point-wise product. Since multiple transforms can extract more implicit features than a single transform, it helps to solve the problem of feature overlap. In addition, local information are encoded with each transform separately, so that the obtained feature representations can be more diversified. These representations are beneficial for capturing fine structures.

2) SPATIAL FEATURE AGGREGATION

From the above transforms, the local information $\mathbf{F}_c^j$ can be obtained. These diverse local information needs to be effectively aggregated. Therefore, we use the gated mechanism to adaptively aggregate multiple transformed features. Specifically, different MLPs are employed to transform $\mathbf{F}_c^j$, aiming to further explore the high-dimensional features. The gates $\mathbf{G}_c^j \in \mathbb{R}^S$ can be obtained by applying a softmax on the transformed features:

$$\mathbf{G}_c^j = softmax\left(MLP_g^j\left(\mathbf{F}_c^j\right)\right), \tag{6}$$

where MLP_g^j is defined as the gated function. Generally, the result of feature aggregation can be obtained through $\mathbf{G}_c^j \odot \mathbf{F}_c^j$. However, the differences among transforms are not fully considered. Inspired by [63], we propose to effectively enhance useful information and suppress useless information. The final aggregation result is defined as follows:

$$\mathbf{F}_c = \left(1 - \mathbf{G}_c^0\right) \odot \sum_{j=1}^{M} \mathbf{G}_c^j \odot \mathbf{F}_c^j + \left(1 + \mathbf{G}_c^0\right) \odot \mathbf{F}_c^0, \tag{7}$$

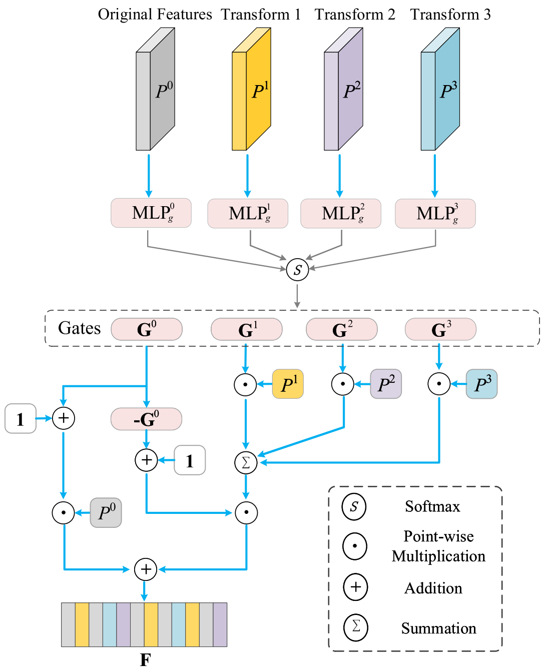


FIGURE 5. The detailed architecture of the GSF module. This module effectively aggregates global information from multiple transformed features.

In this way, only when there are useless features at the current position, the model will receive useful information derived from other transforms. Therefore, the network can not only adjust useful information to the suitable position, but also effectively suppress useless information.

C. GLOBAL SPATIAL FUSION

Local features extracted by multiple transforms have been effectively explored by the MTE module. However, global contextual information may be ignored. Therefore, we propose a Global Spatial Fusion (GSF) module to obtain global representations, as shown in Fig. 4 and 5.

More specifically, after getting the representations with multiple transforms, i.e., P^j, we use the same gated strategy in the previous subsection to directly fuse them:

$$\mathbf{F} = \left(1 - \mathbf{G}^0\right) \odot \sum_{j=1}^{M} \mathbf{G}^j \odot P^j + \left(1 + \mathbf{G}^0\right) \odot P^0, \tag{8}$$

where $\mathbf{F}$ denotes the global features of points with multiple transforms, $\mathbf{G}^j \in \mathbb{R}^{3+F}$ represents the gates obtained by the P^j (i.e., using P^j instead of $\mathbf{F}_c^j$). Since the obtained $\mathbf{F}$ is at a low feature level, we should embed it into high-dimensional features and integrate with the local representations $\mathbf{F}_c^j$. Specifically, we first extract the global representations of the center point c from $\mathbf{F}$, defined as $\mathbf{F}_c'$. Then, we use MLPs to embed $\mathbf{F}_c'$ into the same dimensions of $\mathbf{F}_c^j$, formally as:

$$\mathbf{F}_c^g = MLP_c\left(\mathbf{F}_c'\right), \tag{9}$$

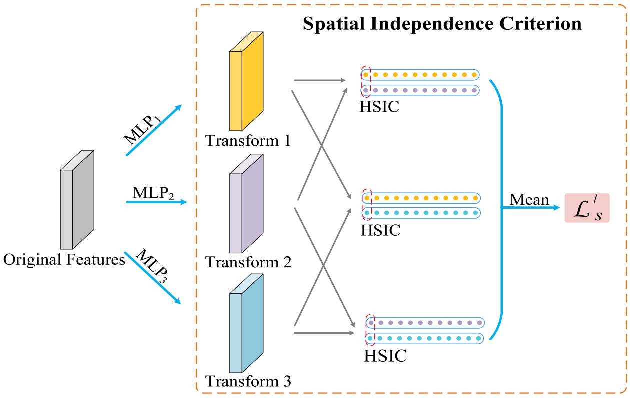


FIGURE 6. Illustration of the SIC strategy. The different colored points come from different transforms.

where $\mathbf{F}_c^g$ denotes the embedded global features of the center points, $MLP_c : \mathbb{R}^{3+F} \rightarrow \mathbb{R}^S$ is the embedding mapping for the center point c. Finally, $\mathbf{F}_c^g$ is fed into the spatial feature aggregation module as additional global information, as shown in Fig. 4. Through this way, local features $\mathbf{F}_c^j$ ($j = 0, \ldots, M$) and global features $\mathbf{F}_c^g$ can be fully integrated to obtain the final representations.

D. SPATIAL INDEPENDENCE CRITERION

With the MTE and GSF, rich feature representations can be derived from multiple transforms. However, if no constraints are imposed on the learned features, the feature representations will tend to be similar. In order to guarantee diverse representations, we propose a Spatial Independence Criterion (SIC) (as shown in Fig. 6) to measure the feature similarity and enlarge the differences of features between different transforms.

Specifically, we choose Hilbert-Schmidt Independence Criterion (HSIC) as the measurement of feature similarity. HSIC [64] is a kernel-based independent discrimination method. Since HSIC is based on the kernel function, it is able to explore more high-dimensional potential associations of the points. We note that there are several other measurements, such as Euclidean distance and cosine similarity. However, Euclidean distance only represents the spatial position relationships, which is sensitive to outliers. The cosine similarity is only inclined to measure the differences in the direction. While HSIC can naturally represent the points in a point cloud as many samples from a distribution, and thus the HSIC has the natural interpretation as computing the independence between two random variables. In addition, HSIC has been proved to have the advantages of simplicity and fast convergence in theory. Following [64], we can get the expression of HSIC as follows:

$$HSIC(X, Y) = (N-1)^{-2} tr\left(\boldsymbol{K}_X \boldsymbol{J} \boldsymbol{K}_Y \boldsymbol{J}\right), \tag{10}$$

where X and Y represent random variables, and N denotes the number of points sampled from the joint distribution of X and Y. $tr()$ is the trace of the matrix. $\boldsymbol{K}_X \in \mathbb{R}^{N \times N}$ and $\boldsymbol{K}_Y \in \mathbb{R}^{N \times N}$ have entries $\boldsymbol{K}_{X_{ij}} = k(x_i, x_j)$ and $\boldsymbol{K}_{Y_{ij}} = k(y_i, y_j)$, $k()$ is the kernel function. $\boldsymbol{J} \in \mathbb{R}^{N \times N}$ is the centering matrix, $\boldsymbol{J} = \boldsymbol{I}_N - \frac{1}{N}\mathbf{1}_N \mathbf{1}_N^T$. By selecting a suitable kernel function such as Gaussian $k(x, y) \sim \exp(-\frac{1}{2} \| x - y \|^2 / \sigma^2)$, HSIC can be zero when X and Y are independent.

For our network, we introduce HSIC to measure the similarity between corresponding points in paired transforms. We define the spatial independence loss $\mathcal{L}_s$ as follows:

$$\mathcal{L}_s^l = \frac{2}{M\,(M+1)} \sum_{X=1}^{M} \sum_{Y=1}^{M} HSIC(P^X, P^Y), \tag{11}$$

$$\mathcal{L}_s = \frac{1}{L} \sum_{l=1}^{L} \mathcal{L}_s^l, \tag{12}$$

where both $P^X$and $P^Y \in \{P^i \mid i = 1, \ldots M\}$, l represents the feature encoding layer. We use the aforementioned Gaussian distribution as the kernel function to get $\boldsymbol{K}_{P^X}$ and $\boldsymbol{K}_{P^Y}$. Through Equ. 11, we can obtain the mean unbiased estimate of HSIC in pairwise transforms. We average the $\mathcal{L}_s^l$ of all encoding layers to get the final loss $\mathcal{L}_s$. Since this value should be minimized, we add it to the loss function for network training.

E. NETWORK TRAINING AND TESTING

Given the point cloud training dataset $\{(P_b, T_b)\}_{b=1}^B$, where B denotes the total number of training blocks, $P_b = \{p_h^b, h = 1, \ldots N\}$ and $T_b = \{t_h^b, h = 1, \ldots N\}$ are the input point cloud block and the corresponding ground-truth with N points, respectively. In addition, N points are sampled from all the points Q in the block ($N \leq Q$). Without loss of generality, we subsequently drop the superscript b and consider each block independently.

We adopt the softmax cross-entropy loss function to train our model, which is proven effective in most existing methods to accomplish the point cloud recognition task. For the segmentation (i.e., point-wise classification) tasks, the loss function can be calculated by:

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{h=1}^{N} \sum_{j=0}^{C-1} \log \Pr(t_h = j | P; \theta), \tag{13}$$

where C denotes the number of total classes, $\Pr(t_h = j | P; \theta)$ is the probability that measures how likely the point belong to the j-th class. For the object classification task, the loss function can be calculated by:

$$\mathcal{L}_{cls} = -\sum_{j=0}^{C-1} \log \Pr(T = j | P; \theta), \tag{14}$$

where $\Pr(T = j | P; \theta)$ is the probability that measures how likely the whole point clouds belong to the j-th class. Therefore, the total loss is defined as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_s, \tag{15}$$

where λ is a hyper-parameter for balancing the specific loss terms. The above defined loss is continuous and differentiable, so we can use Adam algorithm [65] to quickly optimize network parameters.

In the network inference phase, we take all the test blocks as input. For the segmentation, we predict the labels of N points in each block, and obtain the labels of all Q points through interpolation ($N \leq Q$). The interpolated labels are obtained by the weighted sum of the predicted probabilities of the three nearest points, and the weight is inversely proportional to the distance. For the classification, we directly predict which class each test block belongs to.

IV. EXPERIMENTAL SETUPS

To demonstrate the effectiveness of our proposed approach, we evaluate it on three different tasks: Semantic Segmentation, Part Segmentation and Shape Classification. First, we introduce details of some publicly available datasets. Then, the evaluation metrics and implementation details are given. Afterwards, we compare results of our model with other methods. Finally, we construct ablation experiments to analyze the impact of each module.

A. DATASETS DESCRIPTION

The **S3DIS** dataset [37] includes 3D point clouds from 271 rooms in 6 indoor areas of three different buildings. Each point labeled with a semantic label in one of the 13 categories contains *xyz* coordinates and RGB information.

The **ScanNet** dataset [38] is an RGB-D video dataset that contains scanned and reconstructed indoor scenes and rich 3D semantic annotations, including 1210 training and 312 validation scans. The latest version (ScanNetv2) provides 100 new test scans and all semantic labels are publicly unavailable. Each point in the scene is labeled as one of 21 categories. We submit our predictions to the official server[1] for evaluation.

The **ShapeNet** dataset [39] is a large, richly-annotated part segmentation dataset. This dataset contains 16881 3D composite models of 16 shape categories and 50 annotated parts in total. Each shape has 2 to 5 annotated parts.

The **ModelNet40** dataset [11] is a standard dataset to evaluate the shape classification of point clouds, including 12311 meshed CAD models from 40 man-made object categories.

B. EVALUATION METRICS

To evaluate the performance, we follow other related methods [21], [29], [30]. Three widely-used metrics [70] are adopted, i.e., Overall Accuracy (OA), class-wise mean Intersection Over Union (mIoU) and mean Accuracy (mAcc). The formulas of these metrics are as follow:

$$OA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}} \tag{16}$$

$$mIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{17}$$

$$mAcc = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}, \tag{18}$$

where $k+1$ denotes the number of classes, p_{ii} represents the number of correctly classified points for which the category is i, p_{ij} and p_{ji} are the amount of points of class i or j inferred to class j or i.

C. IMPLEMENTATION DETAILS

The proposed PointMTL framework is implemented with Tensorflow 1.12 and one NVIDIA Tesla M40 GPU (with 24G memory). In the following, our data preprocessing strategy and model parameter settings are described in detail.

1) DATA PREPROCESSING

1) For the S3DIS dataset, we follow the same data preprocessing method in [26]. Specifically, we split the dataset room by room and then sample them into 1.2m × 1.2m blocks with a 0.1 buffer area each side for the training data. The points in each block are randomly sampled into a uniform number of 4096. During the testing phase, we test on all the points in the scene.
2) For the ScanNet dataset, we follow the same experimental setup in [22] for a fair comparison. We randomly sample 3m × 1.5m × 1.5m cubes (each with 8192 points) from the indoor room data to generate the training samples, and take into account the entire scans via a sliding window manner for the testing samples.
3) For the ShapeNet dataset, we split all samples into 14007 and 2874 for training and testing respectively. Each sample contains 2048 points, and a normal vector is calculated as additional features to better describe the underlying shape. Moreover, following previous works [71], [72], we remove the parts that only contain one single point since it is impossible to distinguish.
4) For the ModelNet40 dataset, we uses the official split with 9843 shapes for training and 2468 shapes for testing. We create the point clouds by uniformly sampling 1024 points with computed normal vextors on the grid surface. Meanwhile, we select a certain ratio of points in the point clouds to rotate randomly around the z-axis and jitter the position of these points by a Gaussian noise with zero mean and 0.02 standard deviation.

2) PARAMETER SETTINGS

In all the experiments, we use Adam optimizer [65] with an initial learning rate of 0.001. After every 300000 steps, the learning rate decays by 0.5. The number of transforms (i.e., M) is set to 3.

1) For semantic segmentation task, the points are downsampled and upsampled with the network (4096-1024-256-64-32-64-256-1024-4096) on the S3DIS dataset and (8192-1024-256-64-32-64-256-1024-8192) on the

[1] http://www.scan-net.org/

TABLE 1. Quantitative results on S3DIS dataset evaluated on Area 5. '–' denotes value not available. Best results are in bold.

Methods	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [16]	-	48.98	41.09	88.80	97.33	69.80	0.05	3.92	46.26	10.76	58.93	52.61	5.85	40.28	26.38	33.22
SegCloud [66]	-	57.35	48.92	90.06	96.05	69.86	0.00	18.37	38.35	23.12	70.40	75.89	40.88	58.42	12.96	41.60
PointNet++ [17]	86.43	-	54.98	91.41	97.92	69.45	0.00	16.27	66.13	14.48	72.32	81.10	35.12	59.67	59.45	51.42
SPG [23]	86.38	66.50	58.04	89.35	96.87	78.12	0.00	**42.81**	48.93	61.58	75.41	84.66	52.60	69.84	2.10	52.22
PointCNN [19]	85.91	63.86	57.26	92.31	98.24	79.41	0.00	17.60	22.77	62.09	74.39	80.59	31.67	66.67	62.05	56.74
PCCN [67]	-	67.01	58.27	92.26	96.20	75.89	**0.27**	5.98	69.49	**63.45**	66.87	65.63	47.28	68.91	59.10	46.22
PointWeb [21]	86.97	66.64	60.28	91.95	98.48	79.39	0.00	21.11	59.72	34.81	76.33	**88.27**	46.89	69.30	64.91	52.46
GACNet [26]	87.79	-	62.85	92.28	98.27	81.90	0.00	20.35	59.07	40.85	**85.80**	78.54	**70.75**	61.70	74.66	52.82
ELGS [49]	88.43	-	60.06	92.80	98.48	72.65	0.01	32.42	68.12	28.79	74.91	85.12	55.89	64.93	47.74	**58.22**
Point2Node [29]	88.81	70.02	62.96	93.88	98.26	**83.30**	0.00	35.65	55.31	58.78	79.51	84.67	44.07	**71.13**	58.72	55.17
Ours	**88.92**	**70.17**	**65.15**	**95.47**	**98.63**	79.97	0.00	22.40	**70.87**	53.94	78.91	87.70	55.29	67.66	**79.11**	56.97

TABLE 2. Quantitative results on S3DIS dataset with 6-fold cross validation. '–' denotes value not available. Best results are in bold.

Methods	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [16]	78.50	66.20	47.60	88.00	88.70	69.30	42.40	23.10	47.50	51.60	54.10	42.00	9.60	38.20	29.40	35.20
RSNet [68]	-	66.45	56.47	92.48	92.83	78.56	32.75	34.37	51.62	68.11	60.13	59.72	50.22	16.42	44.85	52.03
SPG [23]	85.50	73.00	62.10	89.90	95.10	76.40	62.80	47.10	55.30	68.40	73.50	69.20	63.20	45.90	8.70	52.90
PointCNN [19]	88.14	75.61	65.39	**94.78**	**97.30**	75.82	63.25	51.71	58.38	57.18	71.63	69.12	39.08	61.15	52.19	58.59
DGCNN [24]	84.10	-	56.10	-	-	-	-	-	-	-	-	-	-	-	-	-
PointWeb [21]	87.31	76.19	66.73	93.54	94.21	80.84	52.44	41.33	64.89	68.13	71.35	67.05	50.34	62.68	62.20	58.49
ShellNet [69]	87.10	-	66.80	90.20	93.60	79.90	60.40	44.10	64.90	52.90	71.60	84.70	53.80	64.60	48.60	59.40
ELGS [49]	87.60	-	66.30	93.70	95.60	76.90	42.60	46.70	63.90	69.00	70.10	76.00	52.80	57.20	54.80	62.50
Point2Node [29]	89.01	79.10	70.00	94.08	97.28	83.42	62.68	**52.28**	**72.31**	64.30	75.77	70.78	**65.73**	49.83	60.26	60.90
RandLA-Net [30]	88.00	**82.00**	70.00	93.10	96.10	80.60	62.40	48.00	64.40	69.40	69.40	**76.40**	60.00	64.20	65.90	60.10
Ours	**89.69**	81.57	**71.09**	93.70	93.95	**84.43**	**67.56**	45.22	70.69	**73.91**	**77.86**	73.78	34.82	**69.11**	**72.64**	**66.47**

ScanNet dataset respectively. The loss balancing hyper-parameter λ is set to 0.6 and 0.4 while the batch size is set to 8 and 4 for S3DIS and ScanNet respectively.

2) For part segmentation task, we downsample and upsample the points 3 times each, and the number of points increase in the encoder and decrease in the decoder with (2048-512-128-64-128-512-2048). The loss balancing hyper-parameter λ is set to 0.4, and the batch size is set to 8.
3) For shape classification task, the points are downsampled 3 times and the number of points drops double every time (i.e., 1024-512-256-128). We concatenate the outputs after max pooling of each layer, and use two fully connected layers to classify the results. We set the loss balancing hyper-parameter λ to 0.6 and the batch size to 8.

V. EXPERIMENTAL RESULTS

A. SEMANTIC SEGMENTATION ON S3DIS DATASET

We construct experiments on two settings, namely testing on Area 5 and 6-fold cross validation. Since the objects in Area 5 are different from other areas, experiments on Area 5 can better measure the generalization of the model.

Tab. 1 shows the quantitative results of different methods evaluated on Area 5 of S3DIS dataset. Compared with others, our PointMTL achieves the remarkable performance in all three metrics of OA, mAcc and mIoU. In particular, the mIoU has achieved 65.15%, which is 2.19% higher than the second-ranked method. Through the IoU results of various categories, it can be found that our method reaches competitive results on objects of different sizes in the scene, especially on window and board.

Tab. 2 shows the quantitative results of different methods on S3DIS dataset with 6-fold cross validation. In this setting, our method achieves the best performance in terms of OA (0.68% higher than the second-ranked method) and mIoU (1.09% higher than the second-ranked method) metrics. Our PointMTL is superior to other approaches on 7 out of 13 categories, resulting in overall improvement in mIoU. As can be seen, the mAcc result of our method is only 0.43% lower than RandLA-Net [30]. However, RandLA-Net improves the mAcc by taking the entire scene as input for larger shape representations. In addition, since there exists lots of similar appearances between sofa and chair, we observe that PointMTL confuses them easily, resulting in poor segmentation on the sofa.

The visualization results compared with other methods are shown in Fig. 7. Our PointMTL can better distinguish the

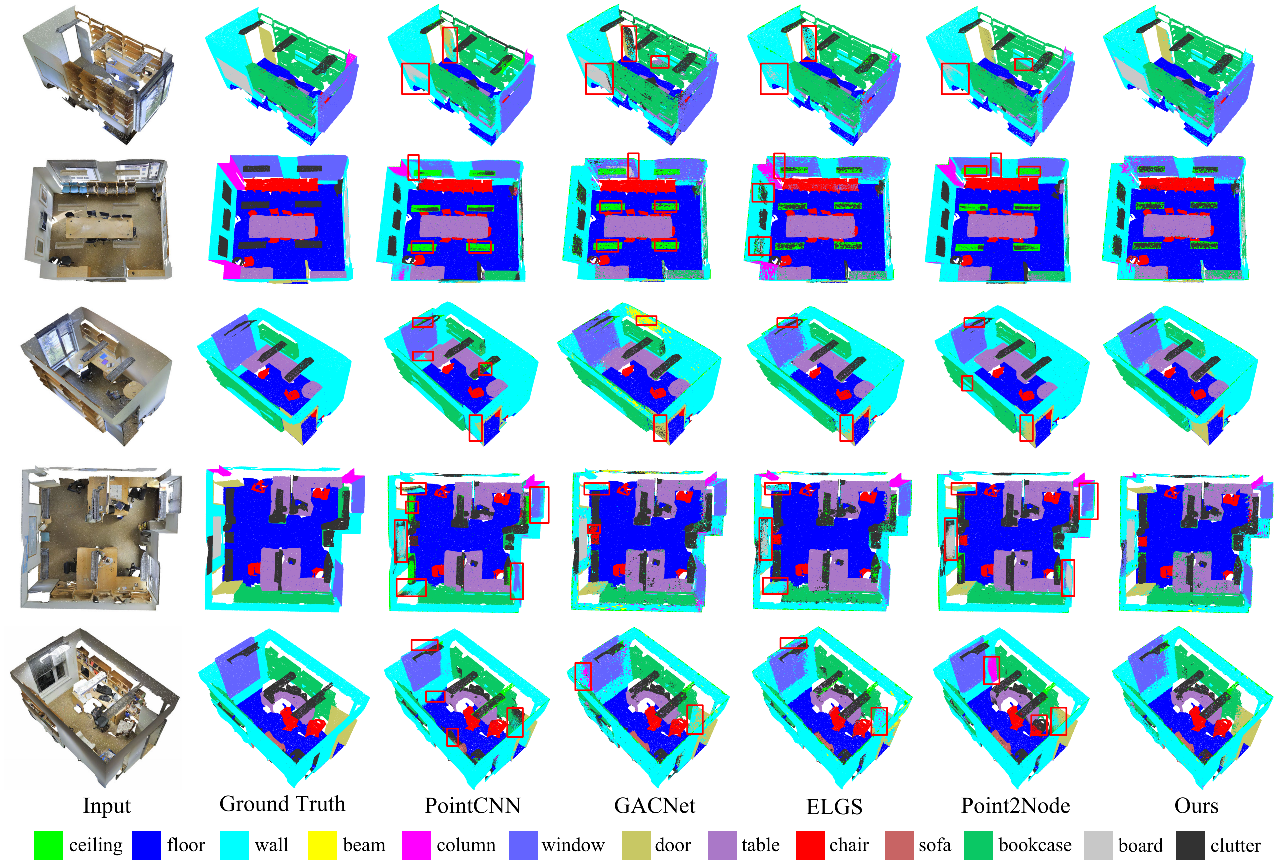


FIGURE 7. Visual comparison of different methods on S3DIS dataset. From left to right: point clouds with original colors, ground truth, PointCNN [19], GACNet [26], ELGS [49], Point2Node [29] and Ours. The regions in red boxes demonstrate the effectiveness of our method.

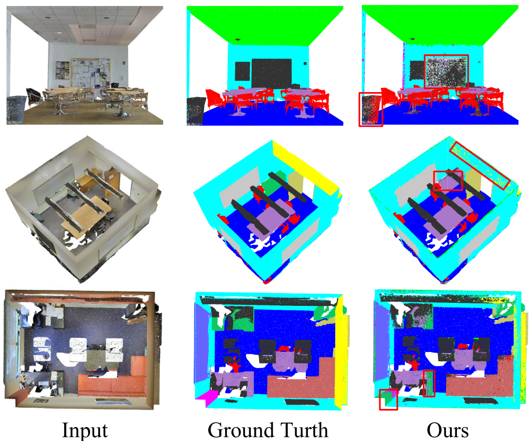


FIGURE 8. Failure examples on the S3DIS dataset.

TABLE 3. Quantitative results on the ScanNetv2 dataset. Best result is in bold.

Methods	mIoU
ScanNet [38]	30.6
PointNet++ [17]	33.9
SPLATNet [72]	39.3
FCPN [73]	44.7
PointCNN [19]	45.8
3DMV [74]	48.4
PCNN [67]	49.8
PointConv [22]	55.6
HPEIN [28]	61.8
PointASNL [50]	63.0
Ours	**63.2**

edges of some easily ambiguous objects, such as board and wall. The main reason is that the problem of feature overlap is overcome and the representations is separable with multiple transforms. Fig. 8 show three failure examples. Obviously, in the first row, part of clutter on the wall is misclassified as board. The bookcase in the second row is completely recognized as table, while the part of table in the third row is mixed with the bookcase. These phenomenon are due to the similar spatial feature between the clutter on the wall and the board as well as the bookcase and the table, which the network cannot completely separate.

B. SEMANTIC SEGMENTATION ON ScanNet DATASET

Tab. 3 shows the quantitative results on ScanNetv2 dataset. Our method achieves 63.2% mIoU, which is better than

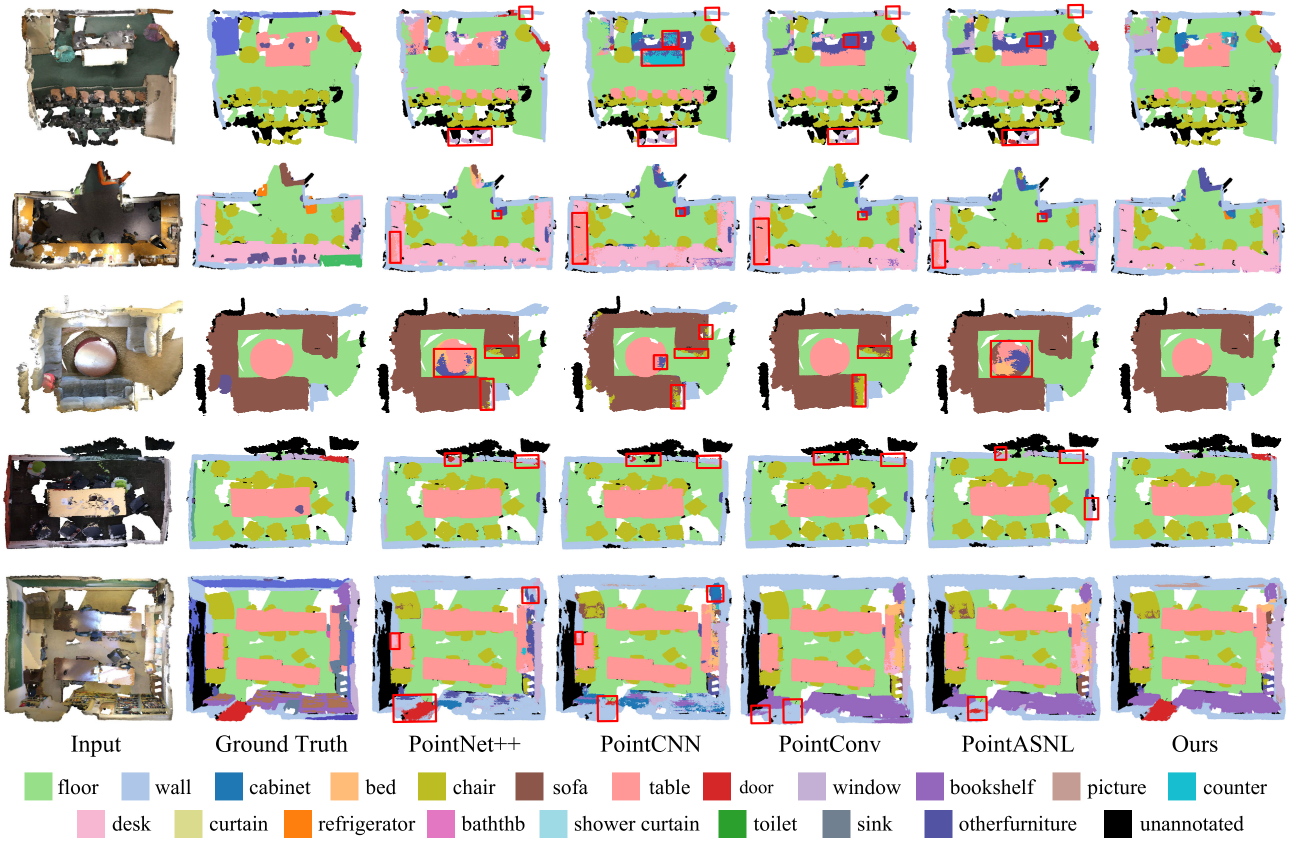


FIGURE 9. Qualitative results on ScanNetv2 dataset. From left to right: RGB point clouds, ground truth, PointNet++ [17], PointCNN [19], PointConv [22], PointASNL [50] and Ours. The regions in red boxes demonstrate the effectiveness of our method.

TABLE 4. Quantitative results on ShapeNet dataset. Best results are in bold.

Methods	mIoU	Aero	Bag	Cap	Car	Chair	Ear phone	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate board	Table
KD-Net [41]	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
PointNet [16]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
A-SCN [75]	84.6	83.8	80.8	83.5	79.3	90.5	69.8	91.7	86.5	82.9	96.0	69.2	93.8	82.5	62.9	74.4	80.8
KCNet [71]	84.7	82.8	81.5	86.4	77.6	90.3	76.8	91.0	87.2	84.5	95.5	69.2	94.4	81.6	60.1	75.2	81.3
RSNet [68]	84.9	82.7	86.4	84.1	78.2	90.4	69.3	91.4	87.0	83.5	95.4	66.0	92.6	81.8	56.1	75.8	82.2
PointNet++ [17]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN [24]	85.1	84.2	83.7	84.4	77.1	90.9	78.5	91.5	87.3	82.9	96.0	67.8	93.3	82.6	59.7	75.5	82.0
SpiderCNN [20]	85.3	83.5	81.0	87.2	77.5	90.7	76.8	91.1	87.3	83.3	95.8	70.2	93.5	82.7	59.7	75.8	82.8
PointCNN [19]	86.1	84.1	**86.5**	86.0	80.8	90.6	**79.7**	**92.3**	**88.4**	85.3	96.1	**77.2**	**95.3**	**84.2**	**64.2**	**80.0**	82.3
DPAM [27]	86.1	**84.3**	81.6	**89.1**	79.5	90.9	77.5	91.8	87.0	84.5	96.2	68.7	94.5	81.4	**64.2**	76.2	84.3
Ours	**86.4**	83.8	83.6	84.3	**81.0**	**91.3**	76.1	91.6	86.0	**85.4**	**96.4**	75.4	94.8	80.8	61.7	75.2	**84.7**

other methods. Fig. 9 shows the qualitative results. It can be seen that our PointMTL has excellent segmentation performance on various objects, especially on door and table, which demonstrates the effectiveness of our approach for capturing fine-grained geometric structures. Note that, in the comparison of the last row, our method is not as precise as PointConv [22] in distinguishing between chair and sofa. However, in the third row of comparison, our method performs better. The reason is that there exists lots of similar appearances between chair and sofa, and semantic contextual information learned by the network in various scenes is different.

C. PART SEGMENTATION ON ShapeNet DATASET

In Tab. 4, we compare our PointMTL with other methods, using the IoU of each category and the part-averaged IoU (mIoU) as the metrics. In terms of mIoU, our method achieves 86.4%, which is better than other approaches. In addition, 5 of 16 categories show the highest results,

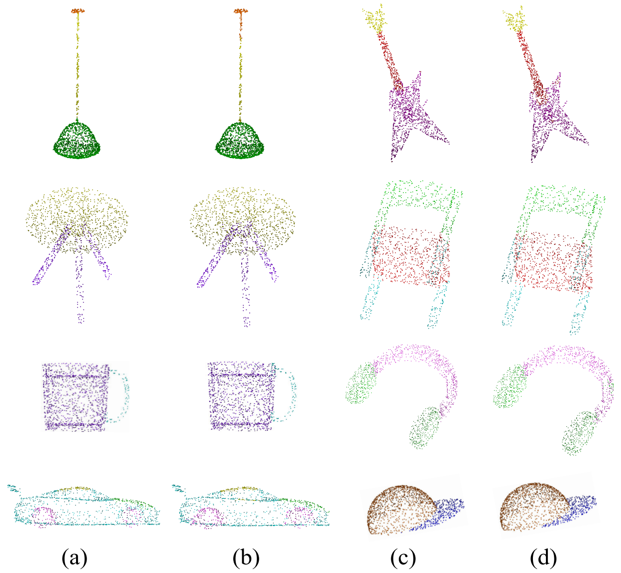


FIGURE 10. Qualitative results of part segmentation on ShapeNet dataset. (a) and (c) are ground truth, while (b) and (d) are corresponding prediction results.

TABLE 5. Quantitative results on ModelNet40 dataset. '–' denotes value not available. Best results are in bold. "vox", "pnt" and "nor" denote voxel, coordinates of point and normal vector, respectively.

Methods	input	points	OA	mAcc
3DShapeNets [11]	vox	-	84.7	77.3
OctNet [42]	vox	-	86.5	83.8
PointNet [16]	pnt	1k	89.2	86.2
DPAM [27]	pnt	1k	91.9	89.9
PointNet++ [17]	pnt, nor	5k	91.9	-
DGCNN [24]	pnt	1k	92.2	90.2
PointWeb [21]	pnt, nor	1k	92.3	89.4
PointConv [22]	pnt, nor	1k	92.5	-
A-CNN [45]	pnt	1k	92.6	90.3
Point2Node [29]	pnt	1k	93.0	-
PointASNL [50]	pnt, nor	1k	93.2	-
Ours	pnt, nor	1k	**93.3**	**90.5**

indicating the potential of this method in part segmentation by fully exploring spatial relations. The visualization of some segmentation results is shown in Fig. 10. The results of our method is much close to the ground truth.

D. SHAPE CLASSIFICATION ON ModelNet40 DATASET

Tab. 5 shows the shape classification results of our model compared with other competitive methods. Our method performs better than other methods in terms of two metrics. Although the value is not much higher, the OA of our method is 0.7% better than that of A-CNN [45], which ranks second in mAcc. Besides, our method can improve the accuracy of each class while achieving the highest overall accuracy. It indicates that our model can effectively capture and accurately classify various shape information.

TABLE 6. Comparison of time and space complexity of proposed model on different datasets. "M" stands for million.

Method	dataset	batch size	#params	#points	forward time(ms)	
					per-batch	per-sample
PointMTL	S3DIS	8	83.1M	4096	764	95.5
	ScanNetv2	4	82.9M	8192	2370	592.5
	ShapeNet	8	80.6M	2048	844	105.5
	ModelNet40	8	80.8M	1024	596	74.5

TABLE 7. Performance contribution of each proposed module. Best results are in bold.

	Ablation Studies	MTE	GSF	SIC			mIoU	Δ
				Euclidean	Cosine	HSIC		
(a)	Baseline						62.30	0.00
(b)	+MTE	√					63.02	+0.72
(c)	+GSF	√	√				63.43	+1.13
(d)		√	√	√			63.62	+1.32
(e)	+SIC	√	√		√		63.97	+1.67
(f)		√	√			√	**65.15**	**+2.85**
(g)	-GSF	√				√	64.16	+1.86

E. MODEL COMPLEXITY

In Tab. 6, we show the time and space complexity of PointMTL on different datasets in terms of the network parameters and forward time. It should be noted that since the network removes the decoder in the shape classification on ModelNet40, the number of parameters and the forward time are relatively minimum. In addition, the model parameters on S3DIS and ScanNetv2 datasets are larger than the other two, which is due to the model has one more layer of encoder and decoder in the semantic segmentation than in other tasks. For the ScanNetv2 dataset, the forward speed is slower than that of the S3DIS dataset because the network has to process 8192 points in one inference.

F. ABLATION STUDY

To demonstrate the impact of the proposed modules, we conduct the following ablation studies. All the experiments below are evaluated on Area 5 of the S3DIS dataset. The results on other datasets have similar trends.

1) EFFECTS OF MTE AND GSF

To evaluate the benefits of the proposed MTE and GSF, we conduct the experiments with/without them. The quantitative results are shown in Tab. 7. (a)-(c) show the results of mIoU after adding the MTE and GSF, which increase by 0.72% and 0.41%, respectively. This illustrates the effectiveness of the modules and demonstrates that MTE and GSF can fully learn and fuse local and global dependencies from multiple transforms. In addition, it is noted that the mIoU of (g) decreased by 0.99% compared to the complete model. This value is higher than 0.41%, which shows that GSF can capture richer representations with the help of SIC.

TABLE 8. Performance comparison of different sampling and aggregation strategies. Best results are in bold.

	Methods	OA	mIoU
Sampling Strategies	Random Sampling	87.76	63.01
	Farthest Point Sampling	**88.92**	**65.15**
Aggregation Strategies	Direct Summation	87.97	63.83
	Concatenate+MLP	88.28	64.12
	General Gated Strategy	88.61	64.76
	Ours	**88.92**	**65.15**

2) EFFECTS OF SIC

The SIC module is designed to enlarge differences of transforms and guarantee more diverse features. To verify the effectiveness, we either remove it or replace it with Euclidean distance and Cosine similarity. In Tab. 7, (d) $\sim$ (f) shows the corresponding experimental results. It can be seen that the SIC module has a great contribution to the network performance, with a 1.72% improvement in mIoU. In addition, the mIoU obtained by HSIC is 1.53% and 1.18% higher than that obtained by Euclidean distance and Cosine similarity, respectively. This demonstrates that HSIC can better measure information differences and overcome the deficiencies of measuring only in absolute distance (i.e., Euclidean distance) or direction (i.e., Cosine similarity).

3) EFFECTS OF DIFFERENT SAMPLING STRATEGIES

In order to explore the effect of different sampling strategies on network performance, we compare random sampling (RS) and farthest point sampling (FPS) we used. In Tab. 8, the results of OA and mIoU in random sampling are 1.16% and 2.14% lower than those of FPS, respectively. The results demonstrate that compared with the complex uniform sampling by FPS, RS discards key features in exchange for computing efficiency.

4) EFFECTS OF DIFFERENT AGGREGATION STRATEGIES

In order to aggregate the features from multiple transforms, we use an improved gated strategy in the MTE module. To further explore its effects, we compare it with direct summation, MLP after concatenating or general gating strategy (i.e., The results are obtained by the weighted sum of different gates and corresponding features without adding any other coefficient constraints). The comparison results are shown in Tab. 8. It can be seen that the other three methods have better performance than summing features directly. This indicates that the linear combination of features can be interfered by useless information to a large extent. In addition, our method achieves the best performance, which is 1.03% and 0.39% higher than the other two non-linear combined methods on mIoU, respectively. This result effectively proves that our method can better enhance useful information and suppress the useless information.

5) EFFECTS OF DIFFERENT NUMBER OF TRANSFORMS

We expect that multiple transforms can learn diverse features that are different from the original features. Therefore, we explore the effects of multiple transforms. As shown in Tab. 9, the performance of our model is gradually enhanced as the number of transforms increases. But when it increases to a certain number (here is 4), the redundancy and overlap of features appears, resulting in a decrease in the segmentation performance.

TABLE 9. The results of our model with different numbers of transforms and nearest neighbors. "*N*" denotes the number of neighbors and "*T*" denotes the number of transforms. Best results are in bold.

	Numbers	OA	mIoU
Transforms ($N = 21$)	2	88.34	63.34
	3	**88.92**	**65.15**
	4	88.56	64.68
	5	88.29	64.07
	6	88.04	63.98
Neighbors ($T = 3$)	12	88.38	64.52
	18	88.63	64.96
	21	**88.92**	**65.15**
	32	88.36	64.40
	40	88.15	64.21

6) EFFECTS OF DIFFERENT NUMBER OF NEIGHBORS

We use the KNN algorithm [36] to search for neighbor points. We conduct experiments on different numbers of neighbors to verify the effect, as shown in Tab. 9. Compared with fewer neighbors, the increase of neighbor points can obtain larger local receptive field and richer features. However, a large number of neighbors will decrease the performance. The main reason is that choosing more neighbor points by Euclidean distance will destroy the geometric structure of the object, thus introducing other perturbations.

VI. CONCLUSION

In this paper, we propose a novel Multi-Transform Learning framework (PointMTL) for feature representation of 3D point clouds. In order to extract features from multiple transforms, we introduce a MTE module to encode the information in local regions. In addition, we propose a GSF module to further extract global information. To guarantee diverse representations, we also introduce a SIC strategy to enlarge the differences of multiple transforms and reduce the feature redundancies. Extensive experiments on four datasets from three different tasks have demonstrated the effectiveness of our method, achieving a considerable performance over other methods.

ACKNOWLEDGMENT

(Yifan Jian and Yuwei Yang contributed equally to this work.)

REFERENCES

[1] H. Zhang, J. Wang, T. Fang, and L. Quan, "Joint segmentation of images and scanned point cloud in large-scale street scenes with low-annotation cost," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4763–4772, Nov. 2014.

[2] P. Zhang, W. Liu, Y. Lei, H. Wang, and H. Lu, "RAPNet: Residual atrous pyramid network for importance-aware street scene parsing," *IEEE Trans. Image Process.*, vol. 29, pp. 5010–5021, 2020.

[3] P. Zhang, W. Liu, Y. Lei, and H. Lu, "Semantic scene labeling via deep nested level set," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 9, 2020, doi: 10.1109/TITS.2020.2995730.

[4] P. Zhang, W. Liu, D. Wang, Y. Lei, H. Wang, and H. Lu, "Non-rigid object tracking via deep multi-scale spatial-temporal discriminative saliency maps," *Pattern Recognit.*, vol. 100, Apr. 2020, Art. no. 107130.

[5] C. Ma, W. An, Y. Lei, and Y. Guo, "BV-CNNs: Binary volumetric convolutional networks for 3D object recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2017, vol. 1, no. 2, p. 4.

[6] S. Guo, W. Huang, L. Wang, and Y. Qiao, "Locally supervised deep hybrid model for scene recognition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 808–820, Feb. 2017.

[7] B. Shuai, H. Ding, T. Liu, G. Wang, and X. Jiang, "Toward achieving robust low-level and high-level scene parsing," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1378–1390, Mar. 2019.

[8] P. Zhang, W. Liu, Y. Lei, H. Wang, and H. Lu, "Deep multiphase level set for scene parsing," *IEEE Trans. Image Process.*, vol. 29, pp. 4556–4567, 2020.

[9] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 945–953.

[10] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[11] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[12] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5648–5656.

[13] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu, "Multi-view 3D object retrieval with deep embedding network," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5526–5537, Dec. 2016.

[14] E. Kalogerakis, M. Averkiou, S. Maji, and S. Chaudhuri, "3D shape segmentation with projective convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3779–3788.

[15] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong, "O-CNN: Octree-based convolutional neural networks for 3D shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Jul. 2017.

[16] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5099–5108.

[18] B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 984–993.

[19] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 820–830.

[20] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "SpiderCNN: Deep learning on point sets with parameterized convolutional filters," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 87–102.

[21] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5565–5573.

[22] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9621–9630.

[23] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4558–4567.

[24] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Nov. 2019.

[25] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3693–3702.

[26] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, "Graph attention convolution for point cloud semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10296–10305.

[27] J. Liu, B. Ni, C. Li, J. Yang, and Q. Tian, "Dynamic points agglomeration for hierarchical point sets learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7546–7555.

[28] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10433–10441.

[29] W. Han, C. Wen, C. Wang, X. Li, and Q. Li, "Point2Node: Correlation learning of dynamic-node for point cloud feature modeling," 2019, *arXiv:1912.10775*. [Online]. Available: http://arxiv.org/abs/1912.10775

[30] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11108–11117.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[32] J. Li, Z. Tu, B. Yang, M. R. Lyu, and T. Zhang, "Multi-head attention with disagreement regularization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 2897–2903.

[33] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2019, pp. 37–42.

[34] H. Du and J. Qian, "Hierarchical gated convolutional networks with multi-head attention for text classification," in *Proc. 5th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2018, pp. 1170–1175.

[35] D. C. Garcia, T. A. Fonseca, R. U. Ferreira, and R. L. de Queiroz, "Geometry coding for dynamic voxelized point clouds using octrees and multiple contexts," *IEEE Trans. Image Process.*, vol. 29, pp. 313–322, 2020.

[36] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[37] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1534–1543.

[38] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.

[39] L. Yi, V. G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.

[40] P.-H. Chen, H.-C. Yang, K.-W. Chen, and Y.-S. Chen, "MVSNet++: Learning depth-based attention pyramid features for multi-view stereo," *IEEE Trans. Image Process.*, vol. 29, pp. 7261–7273, 2020.

[41] R. Klokov and V. Lempitsky, "Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.

[42] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3577–3586.

[43] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 71.1–71.12, 2018.

[44] M. Feng, L. Zhang, X. Lin, S. Z. Gilani, and A. Mian, "Point attention network for semantic segmentation of 3D point clouds," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107446.

[45] A. Komarichev, Z. Zhong, and J. Hua, "A-CNN: Annularly convolutional neural networks on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7421–7430.

[46] D. Thanou, P. A. Chou, and P. Frossard, "Graph-based compression of dynamic 3D point cloud sequences," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1765–1778, Apr. 2016.

[47] S. Chen, C. Duan, Y. Yang, D. Li, C. Feng, and D. Tian, "Deep unsupervised learning of 3D point clouds via graph topology inference and filtering," *IEEE Trans. Image Process.*, vol. 29, pp. 3183–3198, 2020.

[48] L. Landrieu and M. Boussaha, "Point cloud oversegmentation with graph-structured deep metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7440–7449.

[49] X. Wang, J. He, and L. Ma, "Exploiting local and global structure for point cloud semantic segmentation with contextual point representations," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 4573–4583.

[50] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5589–5598.

[51] Y. Lei, Z. Zhou, P. Zhang, Y. Guo, Z. Ma, and L. Liu, "Deep point-to-subspace metric learning for sketch-based 3D shape retrieval," *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106981.

[52] X. Peng, J. Feng, J. T. Zhou, Y. Lei, and S. Yan, "Deep subspace clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5509–5521, Dec. 2020.

[53] P. Zhang, W. Liu, H. Wang, Y. Lei, and H. Lu, "Deep gated attention networks for large-scale street-level scene segmentation," *Pattern Recognit.*, vol. 88, pp. 702–714, Apr. 2019.

[54] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? An analysis of BERT's attention," 2019, *arXiv:1906.04341*. [Online]. Available: http://arxiv.org/abs/1906.04341

[55] Y. Liu, L. Liu, P. Wang, P. Zhang, and Y. Lei, "Semi-supervised crowd counting via self-training on surrogate tasks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 242–259.

[56] C. Tao, S. Gao, M. Shang, W. Wu, D. Zhao, and R. Yan, "Get the point of my utterance! Learning towards effective responses with multi-head attention mechanism," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 4418–4424.

[57] P.-Y. Huang, X. Chang, and A. Hauptmann, "Multi-head attention with diversity for learning grounded multilingual multimodal representations," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1461–1467.

[58] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for RGB-D indoor semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3029–3037.

[59] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M.-H. Yang, "Gated fusion network for single image dehazing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3253–3261.

[60] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2393–2402.

[61] P. Zhang, W. Liu, Y. Lei, and H. Lu, "HyperFusion-Net: Hyper-densely reflective feature fusion for salient object detection," *Pattern Recognit.*, vol. 93, pp. 521–533, Sep. 2019.

[62] W. Liu, Y. Song, D. Chen, S. He, Y. Yu, T. Yan, G. P. Hancke, and R. W. H. Lau, "Deformable object tracking with gated fusion," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 3766–3777, Aug. 2019.

[63] X. Li, H. Zhao, L. Han, Y. Tong, and K. Yang, "GFF: Gated fully fusion for semantic segmentation," 2019, *arXiv:1904.01803*. [Online]. Available: http://arxiv.org/abs/1904.01803

[64] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. Int. Conf. Algorithmic Learn. Theory*. Berlin, Germany: Springer, 2005, pp. 63–77.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[66] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "SEGCloud: Semantic segmentation of 3D point clouds," in *Proc. Int. Conf. 3D Vis. (3DV)*, Oct. 2017, pp. 537–547.

[67] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2589–2597.

[68] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3D segmentation of point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2626–2635.

[69] Z. Zhang, B.-S. Hua, and S.-K. Yeung, "ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1607–1616.

[70] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," 2017, *arXiv:1704.06857*. [Online]. Available: http://arxiv.org/abs/1704.06857

[71] Y. Shen, C. Feng, Y. Yang, and D. Tian, "Mining point cloud local structures by kernel correlation and graph pooling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4548–4557.

[72] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2530–2539.

[73] D. Rethage, J. Wald, J. Sturm, N. Navab, and F. Tombari, "Fully-convolutional point networks for large-scale point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 596–611.

[74] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 452–468.

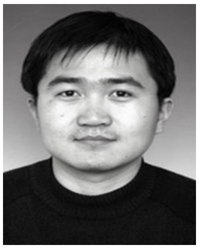
[75] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional ShapeContextNet for point cloud recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4606–4615.

YIFAN JIAN received the B.E. and M.E. degrees in control theory and control engineering from North China Electric Power University (NCEPU), Beijing, China. His main research interests include point cloud recognition, semantic segmentation, and deep learning algorithm.

YUWEI YANG received the bachelor's degree from Sichuan University (SCU), China, in 2020, where he is currently pursuing the M.S. degree in information and communication engineering. His current research interests include 3D point clouds and semantic segmentation.

ZHI CHEN received the Ph.D. degree in nuclear power science and engineering from the Nuclear Power Institute of China (NPIC). Since 1997, he has been working as an Engineer with NPIC. He is currently the Academic Leader with the Nuclear Power Control of the Science and Technology on Reactor System Design Technology Laboratory. He is also the Deputy Chief Designer of the small modular reactor ACP100 of CNNC. His main research interests include deep learning, image identification, intelligent control system, and I&C design.

XIANGUO QING received the bachelor's degree in electrical engineering automation from Sichuan University (SCU), in 1995. His current research interests include image processing, intelligent instrument control system design, and machine learning.



YANG ZHAO received the B.E. and M.E. degrees in precision instruments and machinery from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2007. His main research interests include pattern recognition, object detection, and overall structure design of intelligent control systems.

LIANG HE received the B.E. and M.E. degrees in electrical engineering from Chongqing University (CQU), Chongqing, China, in 2006. His main research interests include machine learning, electrical intelligent control, and fault diagnosis of mechanical equipment.

XUEKUN CHEN received the M.E. degree in control theory and control engineering from the Harbin Institute of Technology (HIT), Harbin, China. His main research interests include intelligent instrument control, data mining, and deep learning algorithm.

WEI LUO received the B.E. degree in electronic and information engineering and the M.E. degree in nuclear technology and application from the University of South China (USC), Hengyang, China. His main research interests include intelligent electronic science and technology, nuclear measurement, and deep learning algorithm.