Improving Semantic Analysis on Point Clouds via Auxiliary Supervision of Local Geometric Priors

Lulu Tang*, Ke Chen*, *Member, IEEE,* Chaozheng Wu, Yu Hong, Kui Jia†, *Member, IEEE,* and Zhi-Xin Yang†, *Member, IEEE*

Abstract—Existing deep learning algorithms for point cloud analysis mainly concern discovering semantic patterns from global configuration of local geometries in a supervised learning manner. However, very few explore geometric properties revealing local surface manifolds embedded in 3D Euclidean space to discriminate semantic classes or object parts as additional supervision signals. This paper is the first attempt to propose a unique multi-task geometric learning network to improve semantic analysis by auxiliary geometric learning with local shape properties, which can be either generated via physical computation from point clouds themselves as self-supervision signals or provided as privileged information. Owing to explicitly encoding local shape manifolds in favor of semantic analysis, the proposed geometric self-supervised and privileged learning algorithms can achieve superior performance to their backbone baselines and other state-of-the-art methods, which are verified in the experiments on the popular benchmarks.

Index Terms—Geometric properties, point clouds, semantic analysis, self-supervised learning, privileged learning.

I. INTRODUCTION

Point clouds collecting a set of order-less points to represent 3D geometry of objects have been verified as a powerful shape representation in a number of recent works [1], [2], [3], [4], [5], [6], [7], [8]. Semantic analysis on a point set aims to categorizing the points globally into semantic classes (*e.g.* plane, chairs, mugs) [3], [4], [6], [9], [10], [11] or locally into object parts [3], [6], [10] according to their topological configuration. Such a problem plays a vital role in many applications, especially those demanding visual perception and interaction between machines and surrounding environment such as augmented reality, robotics and automatic driving. Semantic patterns of point clouds can be discovered from global configuration of local geometric patterns, but it is
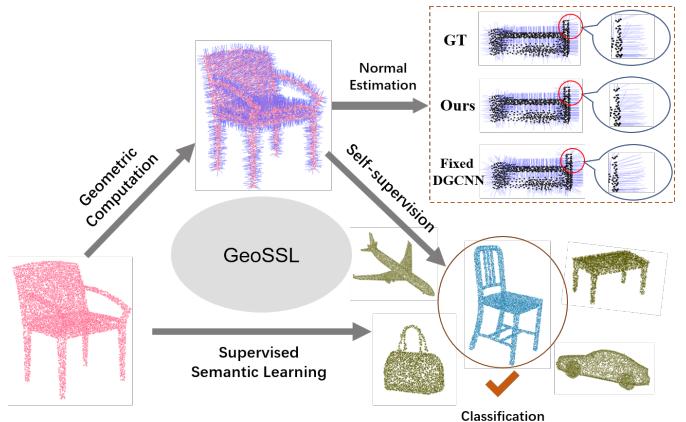
Fig. 1. A flow chart of the proposed geometric self-supervised learning (GeoSSL): Geometric properties generated by physical computation are considered as self-supervised signals to support supervised semantic shape analysis. Owing to additionally fitting geometric properties, the backbone methods (*e.g.* PointNet++ [4], DGCNN [6] in our experiments) can be improved for semantic analysis on point sets.

challenging to discover and exploit such local geometries due to inherently missing point-wise connectivity in their neighborhood.

A number of recent works have been proposed to feature learning on point sets, via either designing locally-connected convolutional/pooling layers on irregular non-Euclidean points such as PointNet [3], PointCNN [5], Dynamic Graph CNN (DGCNN) [6], and GeoNet [2], or hierarchically aggregating features revealing geometric patterns across scales, *e.g.* PointNet++ [4], SO-Net [10]. These existing methods in a supervised learning manner utilize pre-defined annotations to implicitly learn a global topology and local geometries sensitive to semantic classes. Very few work pays an attention to explicitly constraining 3D neural classifiers with auxiliary regressing onto local geometric properties.

Local geometric properties such as point-wise normal vectors, curvatures, and tangent spaces *etc.* are the primitive properties of local point groupings that reveal local geometric manifolds. For example, for computing a normal of a point, the typical solution is to first fit a plane via a set of its nearest neighboring points and obtain the normal of the plane, which indicates point-wise geometric properties describing local connectivity across nearby points. Some works [12], [13] design a deep network to directly estimate these geometric properties from point clouds. However, local geometric properties can be freely obtained by physical computation with no price

for additional efforts on manually annotation, especially for massively amounts of auxiliary data that are usually produced by computer aided design (CAD).

Point-wise geometric properties, in most of existing works [4], [5], are combined with their corresponding point coordinates together as a type of rich point-base feature representation, which are set as input and then fed into deep networks directly for semantic analysis. Alternatively, geometric properties can be served as auxiliary self-supervision signals, inspired by the recent success of self-supervised learning in visual recognition [14], [15], [16], [17], [18], [19], [20], which generate supervision signals from data itself to avoid expensive manual annotations and then learns a proxy loss for network optimization. Moreover, high-quality local properties preserving finer geometric details can be more accurate in view of more dense sampling of points, which can be provided as privileged supervision signals only available during training.

Existing geometric learning methods concern on discovering semantic patterns from global shape, which consists of local geometric patterns. It remains an open problem whether capturing local geometric patterns have any positive effects on semantic analysis of its global configuration. This paper is the first attempt to design a novel geometric learning method to explicitly fit local geometric properties in either a self-supervised or a privileged-supervised learning manner as an additional optimization goal to support semantic analysis on point sets. Fig. 1 shows the main difference between the proposed geometric learning and conventional supervised classifier. Specifically, our deep model shares the low-level feature encoding layers and has two branches for semantic analysis (*e.g.* 3D object classification, part/scene segmentation) and geometric properties estimation tasks respectively in a multi-task learning style.

The core idea in our work is an auxiliary-supervised learning mechanical, which can boost the performance of general tasks, like classification and segmentation. Moreover, it is also a multi-task framework, since an additional geometrical loss function is needed. Our method is an orthogonal idea, which can be integrated into different baseline models. Meanwhile, various of different integration methods can also be explored. Therefore, the novelty of this work is to discover an objective law that can benefit the entire community, that is, adding geometric constraints to the 3D deep learning(3DDL) network can improve the performance of different 3DDL tasks.

The main contributions of this paper are as follows.

- This work for the first time explores geometric properties of point-based surface, perceiving the underlying local connectivity, as auxiliary supervision signals to improve 3D semantic analysis.
- A novel geometric self-supervised learning method is proposed to jointly encode feature discriminative for semantic analysis on point sets and also well fitting local geometric properties in a multi-task learning manner.
- Beyond geometric properties via physical computation, high-quality geometric properties as privileged information can further boost performance on semantic analysis.

Experimental evaluation on three public benchmarks can demonstrate our motivation to exploit local geometric patterns

to improve learning semantic patterns of point clouds, with consistently achieving superior performance to its backbone competitor DGCNN [6] and other state-of-the-art methods in 3D object classification and part/scene segmentation.

The remainder of this paper is structured as follows. Section II reviews related works with semantic analysis on point cloud. Section III describes the proposed methodology. Section IV demonstrates the detail of our experiments and discusses the results in this work. The conclusions are presented in Section V. Source codes and pre-trained models can be downloaded at https://github.com/Necole123/GeoSSL.

II. RELATED WORKS

Semantic analysis of point clouds – Most traditional features on point cloud are handcrafted towards specific tasks, such as wave kernel signature(WKS) [21], local reference frame(LRF)[22], point feature histograms (PFH)[23] and so on. Those point features are often encoded with certain statistical properties or transformed to its 2D counterpart, and are designed to be invariant to certain transformations. In contrast to deep learning based techniques, these hand-crafted point features do not generalize well across different domains. As a pioneer, the PointNet [3] starts the trend of designing deep networks for operating on irregular point-based surface, with the permutation invariance of points encoded by point-wise manipulation in multi-layer perceptrons (MLPs) and a symmetric function for accumulating features. Its following work – Pointnet++ [4] hierarchically aggregates multi-scale features to inherently capturing different modes of semantic patterns. However, both PointNet and PointNet++ only implicitly model semantic concept aware geometric patterns in local regions via deep feature encoding, but miss considering neighborhood information of points to benefit semantic analysis. Recently, the SO-Net [10] explicitly regularizes spatial correlation across points via k-NN search on 2D projection of 3D points during feature encoding, while GeoNet [2] implicitly incorporates local connectivity via an autoencoder and a geodesic matching into extra point-wise features for further fusion. An alternative solution for analyzing point clouds are recently-proposed geometric deep learning methods, such as spectral networks [24], which apply convolution operation on graphs representing irregular distributed structure of points. Its follow-uppers concern on either reducing computational cost by replacing Laplacian eigen decomposition with a polynomial [25], [26] and rational [27] spectral alternatives, or improving its generalization capabilities [28], [29], [30]. Recently, DGCNN is proposed by Wang *et al.* [6] to discover local geometric manifold of each point by an edge convolution operation on a dynamic k-NN graph, which is iteratively updated by the nearest neighbours. Such a DGCNN model achieves the state-of-the-art performance on semantic analysis, which is thus adopted in our methods as the backbone CNN model. The key difference between our methods and the DGCNN baseline lies in incorporating an extra branch (as shown in Fig. 2) to learn local geometric patterns with self-supervision or privileged supervision signals. Superior performance of our methods can be achieved and illustrated in Tables I and VI of Sec. IV.

Geometric analysis of point clouds – Geometric analysis on point clouds aims to obtaining point-wise geometric properties such as the normal and curvature. A typical solution for obtaining local geometric properties of a point is direct computation based on principle component analysis (PCA) [31] within a local region, *e.g.* a plane best fitting the point and its k-nearest neighbours. Such a method is simple but sensitive to noises and generation strategies of local regions. A number of advanced geometric computation techniques [32], [33] are developed to improve robustness against the aforementioned challenges, but remain impractical due to their poor generalization. On the other hand, geometric shape analysis can be learning-based, *i.e.* learning a regression mapping from point sets to point-wise geometric properties. A recent deep learning based PCPNet [12] performs robustly against noises and shape variation under a wide variety of settings, with sufficiently large-scale training data. Our goal of this paper is to directly mine local geometric patterns to additionally support semantic analysis on point-based shape via an auxiliary supervised mapping onto geometric properties. In our proposed multi-task network, more robust estimation on geometric properties can be achieved than fixed backbone baseline (See Fig. 7 and 8), with also improving classification accuracy for semantic analysis (See Table I).

Deep self-supervised learning – Deep learning has gained significant successes in visual recognition [34], [35], [36], [37] and semantic shape analysis [3], [4], [6], [9], [10], [11] but heavily hinges on large-scale labelled training samples. Data augmentation becomes a simple yet effective pre-processing step to alleviate the demand for sufficient data to fit network parameters, especially for the larger network capacity than size of training samples. For avoiding label acquisition for some supervision-starved tasks and using vast numbers of unlabelled data, self-supervised learning is considered as a powerful alternative to relax the impractical requirement about large-scale labelled data available, via generating supervision labels from data itself. In other words, the self-supervised learning paradigm is typically formulated into a *pretext* learning task, such as motion segmentation in videos [38], and relative positions [39], exemplars [40] in the image domain. In light of this, the *target* task can be solved through transferring knowledge from self-supervised learning on a proxy loss. Inspired by the concept in self-supervised learning, this paper for the first time develops a novel geometric self-supervised learning (GeoSSL) to exploit local geometric patterns discovered by self-supervised learning to improve semantic analysis of point clouds. With local geometric regularization on deep feature encoding for semantic analysis, experiment results of the proposed GeoSSL can beat its direct competitor – DGCNN (the backbone net) as well as other comparative methods (see Table I).

Learning with privileged information – Information only available during training is referred to privileged information, which has been exploited in classification [41], [42], regression [43] and ranking [44]. For image based semantic analysis, text [44], attributes [44], bounding boxes [44], head pose [43], and gender [43] have been exploited as privileged information

to boost performance, but this paper is the first work, to the best of our knowledge, in geometric learning with high-quality properties from more densely sampled points as privileged information. Similar to the aforementioned GeoSSL method, our geometric privileged learning (GeoPL) employs the identical multi-task network structure, and the only difference between GeoSSL and GeoPL lies in the quality of geometric properties to discover local patterns of 3D geometry to support semantic classification and segmentation. Experimental verification in this paper demonstrates that our model with privileged geometric properties performs better than the state-of-the-art methods in Table I as well as its self-supervised variant.

III. METHODOLOGY

A. Supervised Semantic Learning

Existing deep algorithms on point clouds focus on analysing semantic patterns of 3D geometry, in view of only semantic labels available in 3D object classification [6] or part segmentation [5]. Given a pair of 3D observation in the representation of a point cloud $\boldsymbol{P}$ and its semantic label y, the typical network architecture of supervised semantic learning frameworks such as PointNet [3], PointNet++ [4], PointCNN [5] and DGCNN [6] consists of several feature encoding layers (*e.g.* convolutional layers, MLP layers or a hybrid of both). Take the DGCNN [6] (the backbone network of the proposed GeoSSL and GeoPL) as an example, which is shown in gray box of Figure 2. The DGCNN introduces edge convolution operation on a directed graph representation for local connectivity of points. In details, a directed k-Nearest Neighbour (kNN) graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ models correlation across closest vertices, where $\mathcal{V} = 1, 2, \ldots, k$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes its vertices and edges. A parametric mapping function on edges $f_\theta(\mathcal{V}_i, \mathcal{V}_j) = f_\theta(\mathcal{V}_i, \mathcal{V}_j - \mathcal{V}_i)$ is adopted for capturing global and local shape patterns, where θ is the parameters to be optimized in each edge convolution layer. In this sense, the output of edge convolution on the k-NN graph on each vertex is calculated by aggregating k edge features, which is thus invariant to the total size of points in the set.

Shared parts of the DGCNN is made up of three MLP based edge convolution blocks and a fully-connected layer to encode each point into a 1024-dimensional feature, and task-specific layers for object classification and part segmentation respectively. On one hand, another multi-layer perception, the output dimension of hidden layers in each MLP based decoder fixed to $\{512, 256, C\}$, where C denotes the size of object classes, is added to the shared parts of the DGCNN for semantic object classification. On the other hand, the shared parts of the DGCNN is followed by a multi-layer perception with $\{256, 128, P\}$, where P denotes the size of object part classes in part segmentation. However, such a model cannot provide supervision signals to incorporate local geometric structural information, which encourages us to design a novel network for improving semantic analysis by learning primitive geometric properties of points in their local neighbourhood.

B. Generation of Local Geometric Properties

Given a point set $\boldsymbol{P}$, point-wise geometric properties can be either measured or calculated directly. A typical solution
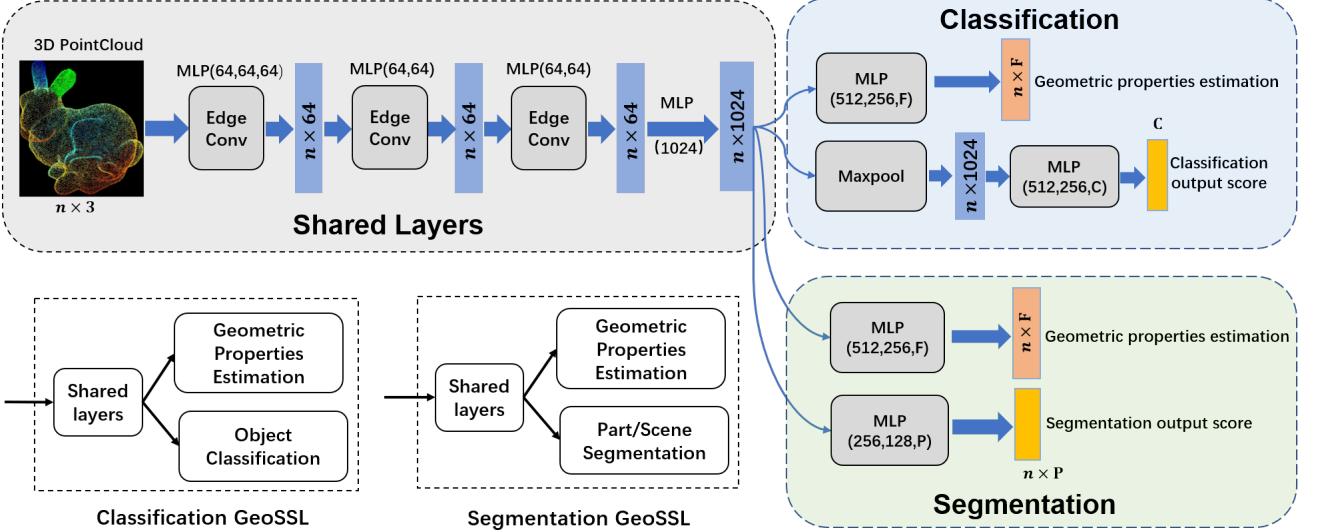
Fig. 2. The proposed networks are based on the DGCNN architecture, which aims to estimating local geometric properties and further augmenting semantic analysis of point clouds. In Geometric Self-supervised Learning (**GeoSSL**), the classification network (GeoSSL$_{cls}$) takes n points as input, and shares the first three Edge convolution layers and one MLP layer, which is then divided into two task branches. The top one is the branch to estimate geometric properties, which consists of three fully-connected layers followed with a mean square loss on local geometric properties, while the bottom branch of GeoSSL$_{cls}$ estimates classification scores with the cross-entropy loss. The segmentation network (GeoSSL$_{seg}$) shares most of network architecture as GeoSSL$_{cls}$, and the only difference lies in the bottom branch to output the segmentation score on each point. Note that, Geometric Privileged Learning (**GeoPL**) employs the same network, but feeding with high quality of geometric properties as supervision signals in the top branch.

of generating ith point's normal is first to find out its k-nearest neighbors $G = \{\boldsymbol{p}'_1, \boldsymbol{p}'_2, \ldots, \boldsymbol{p}'_k\}$ and then calculate the covariance matrix $\boldsymbol{C}$ as

$$\boldsymbol{C} = \sum_G \boldsymbol{r}\boldsymbol{r}^T, \tag{1}$$

where $\boldsymbol{p}$ denotes points in the cloud and $\boldsymbol{r} = \boldsymbol{p}_i - \boldsymbol{p}'_j$, $j = \{1, 2, \ldots, k\}$. Eigenvectors $\boldsymbol{e}$ and eigenvalues λ of $\boldsymbol{C}$ can be obtained via spectral decomposition [45]. The eigenvector corresponding to the minimal eigenvalue defines the estimated surface normal $\boldsymbol{n}_{p_i}$ of point $\boldsymbol{p}_i$, as defined in [46]. Similarly, the second-order geometric property – curvature can also be calculated based on eigen decomposition on covariance matrix $\boldsymbol{C}$ [45]. Particularly, the ratio of the minimal eigenvalue and the sum of all the eigenvalues can be used to estimate the change of geometric curvature. In mathematics, for i-th point $\boldsymbol{p}_i$, the change of curvature u_i can be approximated as the following

$$u_{p_i} = \frac{\lambda_{\min}}{\sum \lambda}, \tag{2}$$

where $\lambda_{\min}$ denotes the minimal eigenvalue of $\boldsymbol{C}$. Additionally, for i-th point $\boldsymbol{p}_i$, the curvature u_{p_i} can also be computed by the normal vectors of that point and its neighbors as

$$u_{p_i} = \frac{1}{k} \sum_{j=1}^{k} \left\| \boldsymbol{n}_{p_i} - \boldsymbol{n}_{neighour(j,p_i)} \right\|. \tag{3}$$

Although geometric properties can be directly computed from point clouds, they can also be estimated via supervised regression learning algorithms [12], [13].

Normal and curvature approximating local geometric patterns of the shape are vital in semantic analysis, which

encourages a number of work [3], [4] to combine such pointwise geometric properties with their corresponding coordinates, which are then fed into a supervised learning model as feature input. However, very few works consider normal and curvature of points as auxiliary supervision signals to improve analyzing semantic patterns owing to feature encoding local manifold structure and superior robustness against noisy point sets, especially when the model is trained on clean data. Beyond point-wise normal and curvature by computational self-generation from point clouds, more accurate and high quality geometric properties can be provided as privileged information available only during training, *e.g.* via physical computation from more dense points.

C. Multi-Task Geometric Learning

In view of lack of local connectivity across order-less points, our motivation is to design an auxiliary task (regression learning with geometric properties) to explicitly incorporate local neighborhood information underlying surface manifolds. To this end, we propose a multi-task geometric learning network to simultaneously learn semantic and geometric patterns for 3D object classification and part segmentation, whose pipelines are visualized in Fig. 2. Given input and output pairs for an ordinary supervised learning network, *i.e.* a point cloud P and its semantic class labels y, geometric properties $\boldsymbol{g}$ can be generated by physical computation in Sec. III-B as extra self-supervision signals or provided as privileged information extracted from high quality point clouds, *i.e.* Geometric Self-Supervised Learning (GeoSSL) and Geometric Privileged Learning (GeoPL) respectively. It is noted that, regardless of qualities of auxiliary labels, the proposed networks have an identical network structure for classification or segmentation.

Training pairs for our multi-task geometric learning network are thus $\{P, \boldsymbol{g}, y\}^{i=1,2,...,N}$, where $\boldsymbol{g} = (\boldsymbol{n}, u)_i^{i=1,2,...,N} \in \mathbb{R}^4$ denotes point-wise geometric properties and N is the size of training samples.

Based on the backbone DGCNN depicted in Sec. III-A, the proposed geometric learning consists of the shared layers and the application-specific block (blue or green boxes in Fig. 2), which shares the first three Edge convolution blocks followed by one MLP layer and is divided into two task-specific branches. The top branch is an auxiliary task to regress point-wise local geometric properties, while the bottom one is the original tasks of semantic analysis (*i.e.* classification, part/scene segmentation). To jointly optimizing both branches, we introduce a combinational loss function as the following, which utilizes the mean square loss $L_{reg}(\boldsymbol{g}, \hat{Y}_{reg})$ to control the quality of normal/curvature estimation in geometric learning branch and the cross-entropy loss $L_{task}(y, \hat{Y}_{task})$ for task-specific semantic analysis on point sets as:

$$\underset{\theta_s, \theta_{task}, \theta_{reg}}{arg \, min} \; L_{task}(y, \hat{Y}_{task}) + \lambda L_{reg}(\boldsymbol{g}, \hat{Y}_{reg}), \quad (4)$$

where $\hat{Y}_{task}$ and $\hat{Y}_{reg}$ denote output of two branches in the proposed model, and $\theta = \{\theta_s, \theta_{task}, \theta_{reg}\}$ are weighting parameters of the proposed geometric learning model. θ_s denotes shared weights in the lower shared layers, and $\{\theta_{task}, \theta_{reg}\}$ are weights for the classification/segmentation and the geometry regression branch, respectively. λ is a trade-off parameter between two loss terms.

The key merit of the aforementioned cost function lies in that it brings additional object function to discover geometric patterns missing by existing supervised point cloud classifiers trained by semantic labels only. During training, we adopt the mean square loss for L_{reg} and the cross-entropy loss for L_{task}. It is noted that the regression loss is not limited to the mean square, and we select it owing to its solid performance on estimation of geometric properties. Specifically, we have explored the Euclidean distance and Cosine similarity for the oriented normal vector, the unoriented normal Euclidean distance and RMS angle difference between the estimated normal and ground truth normal in our experiments. Without the loss of generality, we also employ the mean square loss for supervising geometric curvature. As a result, with both normal and curvature, the loss function L_{reg} can be written as

$$L_{reg} = \frac{1}{m} \sum_{i=1}^{m} \|\boldsymbol{n}_i - \hat{\boldsymbol{n}}_i\|^2 + \frac{1}{m} \sum_{i=1}^{m} (u_i - \hat{u}_i)^2 \quad (5)$$

where $\boldsymbol{n}_i$ and $\hat{\boldsymbol{n}}_i$ denote the ground truth normal (self-generated in GeoSSL or privileged provided in GeoPL) and the predicted normal, and u_i and $\hat{u}_i$ denote the ground truth curvature and the predicted curvature.

IV. EXPERIMENTS

We evaluate the proposed geometric learning algorithms (*i.e.* GeoSSL and GeoPL) introduced in Sec. III on three popular semantic analysis tasks, *i.e.* 3D object classification, part segmentation and scene segmentation.

Datasets and Settings – Evaluation on 3D object classification was conducted on the commonly used ModelNet40 benchmark
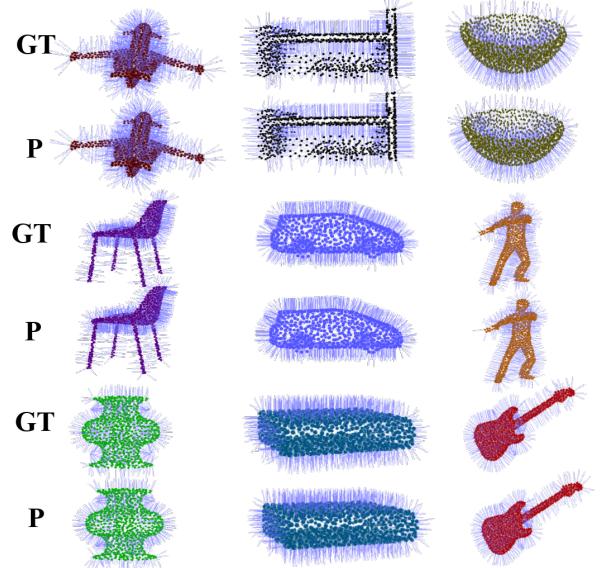

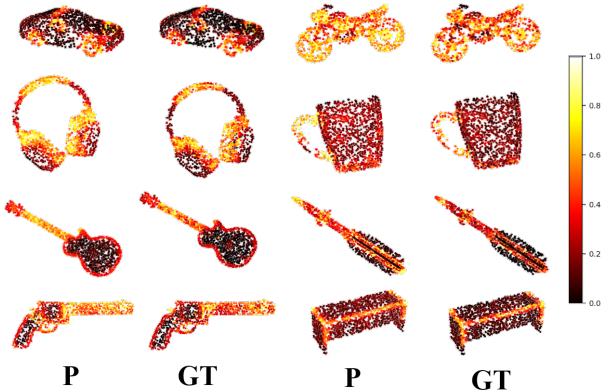Fig. 3. Visualization of predicted normal with GeoSSL$_{dgcnn}$.

Fig. 4. Visualization of predicted curvatures with GeoSSL$_{dgcnn}$ with a color bar on the right hand side. The darker, the smaller value of curvature.

[47], which contains 12,311 CAD models belonging to 40 pre-defined categories. In our experiments, we split the dataset into two parts, *i.e.* 9,843 for training and 2,468 for testing. We followed the same experimental settings as in [3], [6]. Specifically, 1024 points are sampled from mesh faces by farthest point sampling, and are normalized into a unit sphere. We evaluated our model architectures for part segmentation on the ShapeNet part dataset [48], containing 16,880 3D shapes from 16 object categories, annotated with 50 parts in total. We followed the data split as [5], *i.e.* 14006 for training and 2874 for testing. Part category labels are assigned to each point in the point cloud, which consists of 2048 points uniformly sampled from mesh surfaces of training samples. It is worth mentioning here that we assume that each object contains less than six parts. S3DIS [49] dataset is adopted on evaluation of our method for scene segmentation. Unlike the samples in the ModelNet40 and ShapeNet, which are made by 3D modeling tools, the S3DIS samples are collected from real scans of

indoor environments. In details, this dataset contains 3D scans from Matterport scanners in 6 areas within 271 rooms. Each point in the scan is annotated with one semantic label from 13 categories.

Performance Metrics – For the classification task, we use mean accuracy (mA) as our evaluation metric widely adopted in recent work [3], [4], [6].

$$mA = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{i=1}^{N} I_k(\hat{y}_i)}{\sum_{i=1}^{N} I_k(y_i)} \tag{6}$$

where K is the number of categories in test set, N is the number of testing samples. $I_k(y)$ is the indicator function, when $y = k$, $I_k(y) = 1$, otherwise $I_k(y) = 0$. y_i and $\hat{y}_i$ are the true label and predicted label of the ith sample respectively.

In the part segmentation task, Intersection-over-Union (IoU) is used to evaluate our model and other comparative methods, following the same evaluation protocol as the DGCNN [6], the IoU of a shape is obtained by averaging the IoUs of different parts involving in that shape, while the mean IoU (mIoU) is calculated by averaging the IoUs of all the testing samples.

$$IOU_c^i = \frac{|P_{\hat{j}=c}^i \cap P_{j=c}^i|}{|P_{\hat{j}=c}^i \cup P_{j=c}^i|} \tag{7}$$

$$IOU^i = \frac{1}{C} \sum_{c=1}^{C} IOU_c^i \tag{8}$$

$$mIOU = \frac{1}{N} \sum_{i=1}^{N} IOU^i \tag{9}$$

where C is the number of categories need to be split in each sample. N is the number of testing samples. $P_{j=c}^i$ indicates the subset of category c in point cloud P^i. $P_{\hat{j}=c}^i$ represents the subset that be correctly divided into category c in P^i, then IOU_c^i is the IOU of the cth category in P^i, and IOU^i is the IOU of point cloud P^i.

In the scene segmentation task, mean Intersection-over-Union (mIoU) and overall accuracy(OA) are utilized for evaluating our method as follows

$$OA = \frac{|\{p|p \in \mathcal{P} \cap \hat{y}_p = y_p\}|}{|\mathcal{P}|} \tag{10}$$

where $|\mathcal{P}|$ is the numbers of point in evaluate dataset $\mathcal{P}$, and $|\{p|p \in \mathcal{P} \cap \hat{y}_p = y_p\}|$ is the number of correctly segmented points.

Implementation Details – To efficiently use the geometric cues, we pre-train independently the shared layers and geometric leaning branch (the top branches in Fig. 2) with generated geometry properties on the ShapeNetCore dataset, which is similar to the DGCNN architecture for part segmentation with the only change lying in the last layer to output 4 continuous values. Model parameters learned by such a network are then used to initialize the shared layers both in GeoSSL$_{cls}$ and GeoSSL$_{seg}$. The learning rates of the GeoSSL$_{cls}$ and GeoSSL$_{seg}$ are set as 0.01 and 0.001 respectively, and are

TABLE I
COMPARISONS OF CLASSIFICATION ACCURACY ON THE MODELNET40.
NOTE THAT, THE DGCNN AND DGCNN+ HERE DENOTE THE DGCNN IN
[6] WITHOUT AND WITH THE SPATIAL TRANSFORMER RESPECTIVELY.
OUR GEOSSL AND GEOPL ADOPT THE FORMER AS THEIR BACKBONE.

Methods	Mean Class Accuracy	Overall Accuracy
VoxNet [50]	83.0	85.9
PointNet [3]	86.0	89.2
PointNet++ [4]	-	90.7
SO-NET [10]	87.3	90.9
PointCNN [5]	-	92.2
DGCNN [6]	88.2	91.2
DGCNN+ [6]	90.2	92.2
GeoSSL$_{dgcnn}$ (ours)	90.3	92.9
GeoPL$_{dgcnn}$(ours)	**90.8**	**93.5**

decreased with an exponential function by every 20 epochs. The overall training epochs in our experiments are 200.

A. Comparison with State-of-the-Art

3D object classification – Comparative evaluation in 3D object classification on the ModelNet40 are shown in Table I. We can see that our GeoSSL$_{dgcnn}$ achieves superior performance to its direct competitor DGCNN [6] as well as other state-of-the-art methods. In light of the identical input and output as well as the backbone CNN model, performance gain can only be explained by auxiliary incorporation of local geometric properties into the DGCNN. We also evaluate our geometric privileged learning (GeoPL) for classification on the ModelNet40 with privileged geometric properties only available during training, whose normal and curvature are generated from more dense point-based surface and thus more accurate than those directly computed from sparse points. For example, we can generate privileged normal and curvature from a dense point cloud consisting of 10000 points used in our experiment compared to ordinary one with 1024 points. Experiment results in Table I show significantly better performance than other comparative algorithms given accurate geometric properties, which further verifies the effectiveness of our concept on improve semantic analysis via exploiting local geometric priors.

3D Part segmentation – The part segmentation network is evaluated on the ShapeNet Part benchmark, whose results on Intersection-over-Union (IoU) are illustrated in Table II. Evidently, regardless of the network structure, *e.g.* PointNet++ [4], PointCNN [5] or DGCNN [6], the proposed GeoSSL can consistently perform better than the backbone competitors. Specifically, PointNet++ [4] achieves the better performance compared to our GeoSSL but demands high quality point-wise geometric properties as input, which can be impractical for accurate point-wise normals available in the real world. We re-implement PointNet++, PointCNN and DGCNN by following the settings in original works, but slightly change the input or network architectures, whose results are reported[1] and noted as the backbone in each block of Table II. It is noted that the input of original PointNet++ is coordinates combined

---

[1]Our Implementation is slightly worse than the reported results in the original works.

TABLE II
COMPARISONS OF PART SEGMENTATION RESULTS ON THE SHAPENET
PART DATASET WITH MEAN IOU (%).

Methods	Mean IoU
PointNet [3]	83.7
SO-NET [10]	84.9
PointNet++ [4]	85.1
PointNet++ (backbone)	84.3
GeoSSL$_{pointnet++}$ (ours)	84.8
PointCNN [5]	86.1
PointCNN (backbone)	85.3
GeoSSL$_{pointcnn}$ (ours)	85.6
DGCNN+ [6]	85.1
DGCNN (backbone)	84.5
GeoSSL$_{dgcnn}$ (ours)	85.7

with normal, while our backbone PointNet++ only utilizes coordinates as input, which needs to be consistent with our proposed GeoSSL. For backbone DGCNN, we slightly change the network architecture by removing the spatial transformer module in the original DGCNN. Because we need to calculate the geometric properties by the neighbor information of each point. While spatial transformer module may destroy the local surface structure. Therefore, Our GeoSSL and GeoPL adopt the DGCNN without spatial transformer as their backbone.

In view of the identical network structure to capture semantic properties in the GeoSSL and its backbone baselines, performance gain can only be explained by exploiting local fine-detailed geometries of objects, which can demonstrate our motivation again. More results about part segmentation results are illustrated in Fig. 5 and Fig.6, from which we can see that the segmentation results of our method are very close to the ground truth and always better than its backbone.

Indoor Scene Segmentation – We also apply our GeoSSL to the semantic scene segmentation task, which replaces object part labels in part segmentation by semantic object classes in the scene. We conduct experiments on the S3DIS[49], which is collected from real scans of indoor environments. For a fair comparison, we follow the same setting as the DGCNN, where each room is sliced into 1×1 square-meters block, and 4096 points are sampled for each block. Based on the sampled points, we then calculate point-wise geometric properties (*i.e.* normal, curvature) using the method in Sec. III-B. Finally, we use the 6-fold cross validation over the 6 areas, and report the mean of evaluation results. We compare the proposed method with the state-of-the-art methods on the S3DIS, whose results are shown in Table III. We can conclude that our method consistently achieves superior segmentation performance to its direct competitor DGCNN [6], yet outperforms most of state-of-the-art methods except for PointCNN [5] and SPGraph [51]. Note that, the concept of our method is generic, which can be applied to other specific backbone CNN models, which achieves state-of-the-art scene segmentation performance, such as PointCNN [5] and SPGraph [51].

B. More Results and Discussions

Ablation studies on geometric properties – Evaluation on combination of different geometric properties is shown in Ta-
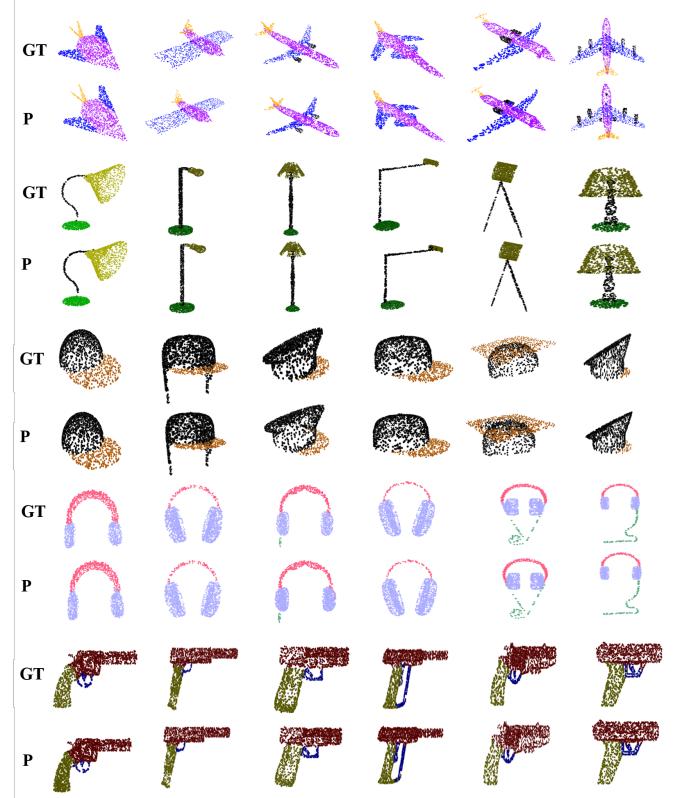

Fig. 5. Visualization of part segmentation results with GeoSSL$_{dgcnn}$, where GT denotes ground truth label, and P means predicted result.

TABLE III
SEGMENTATION COMPARISONS ON S3DIS IN MEAN IOU (MIOU, %) AND
OVERALL ACCURACY (OA, %).

Methods	Mean IoU	Overall Accuracy
PointNet(baseline) [3]	20.1	53.2
PointNet [3]	47.6	78.5
PointCNN [5]	65.4	88.1
G+RCU [52]	49.7	81.1
SGPN [53]	50.4	-
RSNet [54]	56.5	-
SPGraph [51]	62.1	85.5
DGCNN+ [6]	56.1	84.1
DGCNN$_{(our baseline)}$	54.5	83.6
GeoSSL$_{dgcnn}$	59.1	86.3

ble IV. In DGCNN [6], geometric properties are concatenated as additional feature input, while our GeoSSL exploits them as self-supervision signals of an auxiliary task. We observe that all methods with geometric properties either as input feature or as self-supervision signals can boost classification performance, which demonstrates our motivation to employ local geometric properties can reveal rich local geometries of 3D semantic classes. Moreover, geometric properties as self-supervision signals (in the right column) can consistently perform better than that as feature (in the middle column). The main reason is that our GeoSSL takes the form of multi-task learning, where self-supervision serves an auxiliary task to regularize learning of the main, supervised task. This is different from some alternatives, *e.g.* pre-training based self-
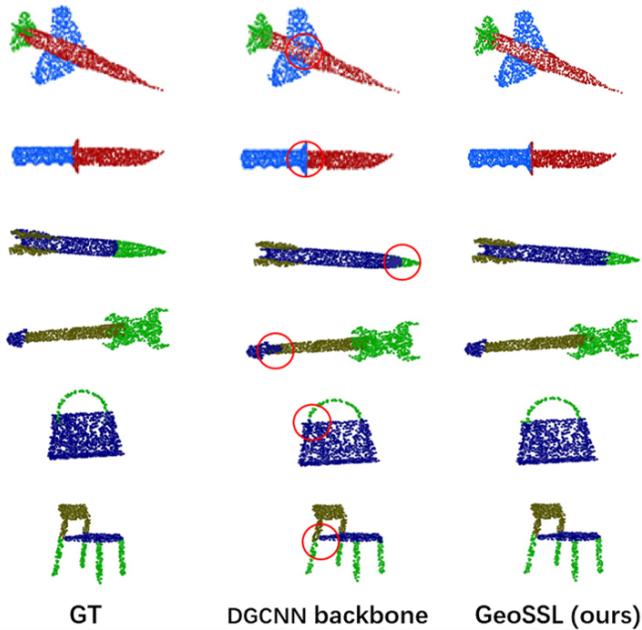
Fig. 6. Comparisons with segmentation results by the proposed GeoSSL$_{\text{dgcnn}}$ (ours) and backbone DGCNN.

TABLE IV
POINTS (P) WITH VS. WITHOUT GEOMETRIC PROPERTIES INCLUDING NORMAL ($\boldsymbol{n}$) AND CURVATURE ($\boldsymbol{u}$) ON MEAN CLASSIFICATION ACCURACY (%).

Methods	DGCNN[6]	GeoSSL$_{\text{dgcnn}}$
P	91.2	92.2
$P + \boldsymbol{n}$	91.7	92.5
$P + \boldsymbol{u}$	91.4	92.3
$P + \boldsymbol{n} + \boldsymbol{u}$	91.9	**92.9**

supervision methods, where features are learned via self-supervision alone, and are subsequently used for supervised tasks. Given large capacities of deep networks, GeoSSL regularizes feature learning (via self-supervised prediction learning of local geometric properties), reduces their potentials of over-fitting, and thus improves generalization of the learned features for the supervised tasks. Moreover, the combination with normal and curvature can be preferred as self-supervision signals in view of exploiting both first and second order geometric smoothness in point sets.

TABLE V
COMPARISON OF THE COSINE SIMILARITY FOR NORMAL ESTIMATION WITH INVOLVED METHODS

Methods	Cosine Similarity
DGCNN [6]	0.99
DGCNN$_{fixed}$	0.97
GeoSSL$_{dgcnn}$	0.99

Effects of learning geometric patterns in typical supervised semantic learning – We are interested in whether the learned feature in supervised semantic learning on point clouds can be used to estimate geometric properties. As a result, we conduct an experiment for normal estimation to compare the following
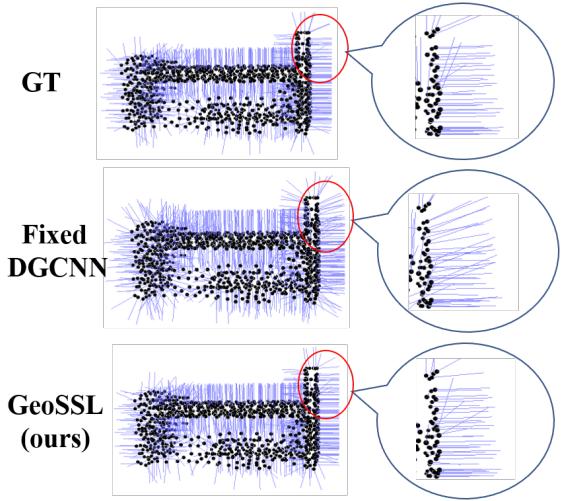

Fig. 7. Comparisons with learned geometric properties by the proposed GeoSSL$_{\text{dgcnn}}$ (ours) and DGCNN$_{fixed}$.

TABLE VI
CLASSIFICATION PERFORMANCE (%) OF GEOSSL WITH OTHER BASELINE CNN MODELS ON THE MODELNET40.

Methods	Overall Accuracy
Pointnet++ [4]	90.7
PointCNN [5]	92.2
DGCNN+ [6]	92.2
GeoSSL$_{\text{pointnet++}}$	91.7
GeoSSL$_{\text{pointcnn}}$	92.8
GeoSSL$_{\text{dgcnn}}$	**92.9**

models: the first setting is to train the DGCNN for normal estimation from scratch, denoted as DGCNN in Table V; the second setting is another DGCNN, whose network parameters of lower layers are shared by the DGCNN pre-trained on the ModelNet40 for classification, which are then fixed during training with tuning the other parameters in higher layers (we denote it as Fixed DGCNN (DGCNN$_{fixed}$). The results are illustrated in Table V for a comparative purpose on the Cosine Similarity metric, which reveals an angle difference between the predict normal and the ground truth normal, *i.e.* the larger its value, the better. We also illustrate qualitative difference between our method and DGCNN$_{fixed}$ in Fig. 7, which shows that the proposed method can predict more accurate normal than its competitor. Quantitative comparisons with normal estimation errors can be found in Fig. 8. Both Table V, Fig. 7 and 8 show that the DGCNN$_{fixed}$ gain the worse performance in comparison with the DGCNN and GeoSSL$_{dgcnn}$. It implies that existing point cloud analysis methods with only semantic supervision labels pay less attention on whether the networks can learn local geometric patterns. Our method with geometric self-supervised learning can benefit each other task simultaneously, which captures local geometric patterns to further augment semantic recognition tasks.

Evaluation across CNN backbone models – Evaluative results on different CNN baselines (*i.e.* PointNet++ [4], DGCNN [6], and PointCNN [5]) are illustrated in Table VI. We can evidently find out that, our proposed methods can consistently
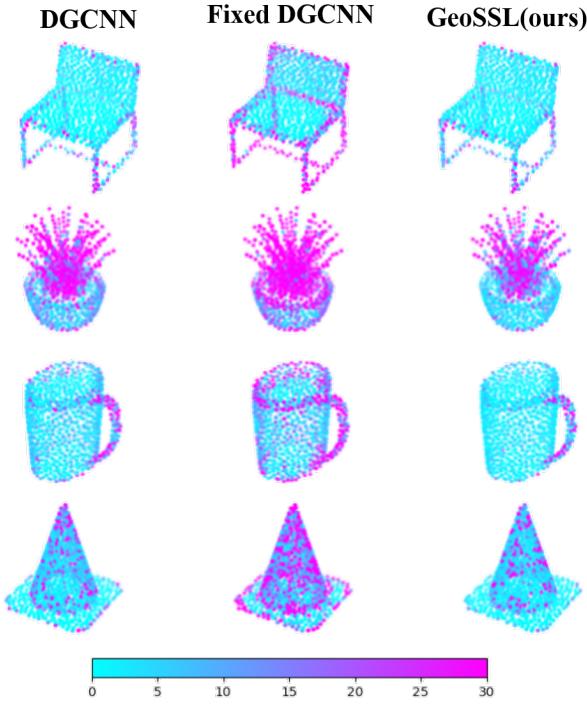
DGCNN   **Fixed DGCNN**   **GeoSSL(ours)**

Fig. 8. Quantitative comparisons of normal estimation for GeoSSL$_{dgcnn}$ and DGCNN, the colors of points correspond to angular difference (estimation error) between predicted normal and ground truth normal, which are mapped to a heat-map ranging from 0-30 degrees. The small its value, the better.

outperform their baseline models. It further confirms that the nature of auxiliary geometric learning on improving semantic point cloud recognition.

TABLE VII
COMPARISONS OF VARIOUS OF MULTI-TASK ON THE SHAPENET PART.

Methods	Classification Accuracy(%)	Segmentation IoU(%)
DGCNN+ [6]	98.8	85.1
GeoSSL$_{cls+seg}$	98.9	84.4
GeoSSL$_{cls+reg}$	**99.4**	-
GeoSSL$_{seg+reg}$	-	**85.7**

Evaluation on multi-task learning architecture – To this end, we additionally conducted experiments on the ShapeNet Part dataset. The network architecture used here is the same as in Fig. 2, the only difference lies in the task setting. Comparison results are shown in Table VII, where we evaluate different options of combining two tasks in a multi-task learning framework. As can be seen from Table VII, when simply combining classification and segmentation tasks in a multi-task manner, denoted as MTNet$_{cls+seg}$. The classification performance (98.9%) of the MTNet$_{cls+seg}$ is only slightly better than its baseline DGCNN (98.8%), but even worse than its baseline DGCNN on segmentation performance (84.4%). Different from that, our models with an auxiliary fitting on geometric properties achieve superior results to the DGCNN and MTNet$_{cls+seg}$ both on classification (GeoSSL$_{cls+reg}$) and segmentation (GeoSSL$_{cls+seg}$) tasks, which further demon-

strates performance gain of our method can be credited to additional regression learning branch.

Evaluation on estimation of geometric properties – Fig. 3 and 4 visualize the predicted normals and curvatures with the proposed GeoSSL respectively. From which we can see that estimation performance of our method are very close to the ground truth. Furthermore, when neural networks are trained on clean point sets, they could predict more accurate normals than those obtained by geometric computation, especially for noisy testing set. This could be attributed to their capability to learn statistical regularities from training data. For verification, we train a DGCNN based normal estimation network using clean training sets of 1024 points from the ModelNet40; for testing, we add Gaussian permutations to 2468 instances of point sets, where the noise level for each point is $\sigma = 0.01$ (clean point sets are normalized in a unit sphere). Geometric computation produces an averaged error of 1.0015 against GT normals (measured in the Cosine distance, ranging in $[0, 2]$), and our trained neural model gives a lower one of 0.7332, which verifies our claim the learning based method with clean data can predict more accurate geometric properties.

TABLE VIII
EFFECT OF DIFFERENT λ PROPORTION OF TWO LOSS IN GEOSSL$_{DGCNN}$.
THE SMALLER λ IS, THE LESS EFFECT OF LOCAL GEOMETRIC LEARNING
AFFECTS.

Setting (λ)	1	e-1	e-2	e-3	e-4	e-5
Accuracy (%)	87.1	89.3	**92.9**	92.3	91.9	91.7

Evaluation on ratio between losses – In our classification settings, λ is an important parameter to determine the proportion of two loss function (*i.e.* the regression loss for fitting local geometries and the classification/segmentation loss). We hold out 20% of training data as the validation set. We observe that the trade-off parameter λ varies across different network architectures and different tasks, but when λ is set as between $[10^{-3}, 10^{-2}]$, our model can steadily perform well. As a result, we select either 0.01 or 0.001 for λ in our experiments. Specifically, Table VIII illustrates the trend of classification accuracy with λ varying on the ModelNet40 with GeoSSL$_{dgcnn}$. When $\lambda = $ e-2, it can reach the best classification performance.

TABLE IX
EVALUATION ON TRANSFERRING KNOWLEDGE FOR 3D CLASSIFICATION
USING THE GEOSSL$_{DGCNN}$.

Experiment Setting	Accuracy
Random initialization	92.5
Pre-trained on the ShapeCore	**92.9**

Effects of pre-training with auxiliary data – An experiment to evaluate the effects of auxiliary data on pre-training is conducted by pre-training the proposed models on the ShapeNetCore dataset. Results in Table IX show that moderate improvement on the pre-trained models can be achieved over the identical network with random initialization, which encourages us to adopt pre-training for boosting performance.

V. Conclusion

This paper, for the first time, systematically introduces self-supervised learning into 3D point cloud semantic analysis, which is a generic method to readily replace its backbone with any other deep geometric learning. Rather than employing geometric properties as additional feature input, our network utilizes them as auxiliary supervision signals, which can consistently improve performance on semantic analysis. Given accurate privileged local shape information, our method can further be boosted to 93.5% mean classification accuracy on the ModelNet40.

References

[1] M. Liu, "Robotic online path planning on point cloud," *IEEE T. Cybern.*, vol. 46, no. 5, pp. 1217–1228, 2015.

[2] T. He, H. Huang, L. Yi, Y. Zhou, C. Wu, J. Wang, and S. Soatto, "Geonet: Deep geodesic networks for point cloud analysis," *arXiv preprint arXiv:1901.00680*, 2019.

[3] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[4] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[5] 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*, 2018, pp. 828–838.

[6] 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, 2019.

[7] Y. Wen, J. Lin, K. Chen, and K. Jia, "Geometry-aware generation of adversarial and cooperative point clouds," *arXiv preprint arXiv:1912.11171*, 2019.

[8] K. Wang, K. Chen, and K. Jia, "Deep cascade generation on point sets," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 3726–3732.

[9] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *Advances in neural information processing systems*, 2016, pp. 82–90.

[10] J. Li, B. M. Chen, and G. Hee Lee, "So-net: Self-organizing network for point cloud analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9397–9406.

[11] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 206–215.

[12] P. Guerrero, Y. Kleiman, M. Ovsjanikov, and N. J. Mitra, "Pcpnet learning local shape properties from raw point clouds," in *Comput. Graph. Forum*, vol. 37, no. 2, 2018, pp. 75–85.

[13] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "Nesti-net: Normal estimation for unstructured 3d point clouds using convolutional neural networks," *arXiv preprint arXiv:1812.00709*, 2018.

[14] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2051–2060.

[15] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," *arXiv preprint arXiv:1901.09005*, 2019.

[16] C. Gan, B. Gong, K. Liu, H. Su, and L. J. Guibas, "Geometry guided convolutional neural networks for self-supervised video representation learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5589–5597.

[17] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9359–9367.

[18] D. Novotny, S. Albanie, D. Larlus, and A. Vedaldi, "Self-supervised learning of geometrically stable features through probabilistic introspection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3637–3645.

[19] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3636–3645.

[20] X. Wang, K. He, and A. Gupta, "Transitive invariance for self-supervised visual representation learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1329–1338.

[21] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *2011 IEEE international conference on computer vision workshops (ICCV workshops)*. IEEE, 2011, pp. 1626–1633.

[22] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3d local surface description and object recognition," *International journal of computer vision*, vol. 105, no. 1, pp. 63–86, 2013.

[23] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.

[24] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," in *International Conference on Learning Representations (ICLR2014)*, 2014, pp. http–openreview.

[25] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[27] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *arXiv preprint arXiv:1705.07664*, 2017.

[28] M. Dominguez, R. Dhamdhere, A. Petkar, S. Jain, S. Sah, and R. Ptucha, "General-purpose deep point cloud feature extractor," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1972–1981.

[29] X. Zhou, F. Shen, L. Liu, W. Liu, L. Nie, Y. Yang, and H. T. Shen, "Graph convolutional network hashing," *IEEE T. Cybern.*, 2018.

[30] S. Pan, R. Hu, S. Fung, G. Long, J. Jiang, and C. Zhang, "Learning graph embedding with adversarial training methods," *IEEE T. Cybern.*, pp. 1–13, 2019.

[31] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, *Surface reconstruction from unorganized points*. ACM, 1992, vol. 26, no. 2.

[32] Q. Mérigot, M. Ovsjanikov, and L. J. Guibas, "Voronoi-based curvature and feature estimation from point clouds," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 6, pp. 743–756, 2010.

[33] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. R. Zhang, "Edge-aware point set resampling," *ACM Trans. Graph.*, vol. 32, no. 1, p. 9, 2013.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[35] Y. Zhang, K. Jia, and Z. Wang, "Part-aware fine-grained object categorization using weakly supervised part detection network," *IEEE Trans. Multimedia*, 2019.

[36] L. Wu, Y. Wang, X. Li, and J. Gao, "Deep attention-based spatially recursive networks for fine-grained visual recognition," *IEEE T. Cybern.*, vol. 49, no. 5, pp. 1791–1802, 2018.

[37] S. Li, K. Jia, Y. Wen, T. Liu, and D. Tao, "Orthogonal deep neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2019.

[38] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2701–2710.

[39] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.

[40] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[41] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Netw.*, vol. 22, no. 5, pp. 544–557, 2009.

[42] M. Lapin, M. Hein, and B. Schiele, "Learning using privileged information: SVM+ and weighted SVM," *Neural Netw.*, vol. 53, pp. 95–108, 2014.

[43] H. Yang and I. Patras, "Privileged information-based conditional regression forest for facial feature detection," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. IEEE, 2013, pp. 1–6.

[44] V. Sharmanska, N. Quadrianto, and C. H. Lampert, "Learning to rank using privileged information," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 825–832.

[45] K.-H. Bae and D. D. Lichti, "A method for automated registration of unorganised point clouds," *ISPRS-J. Photogramm. Remote Sens.*, vol. 63, no. 1, pp. 36–54, 2008.

[46] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, "Tangent convolutions for dense prediction in 3d," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3887–3896.

[47] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[48] 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, 2016.

[49] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1534–1543.

[50] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *International Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 922–928.

[51] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," *arXiv preprint arXiv:1711.09869*, 2017.

[52] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3d semantic segmentation of point clouds," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 716–724.

[53] W. Wang, R. Yu, Q. Huang, and U. Neumann, "Sgpn: Similarity group proposal network for 3d point cloud instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2569–2578.

[54] Q. Huang, W. Wang, and U. Neumann, "Recurrent slice networks for 3d segmentation of point clouds," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2626–2635.