PointGLR: Unsupervised Structural Representation Learning of 3D Point Clouds

Yongming Rao, *Student Member, IEEE*,
Jiwen Lu, *Senior Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

Abstract—This work explores the use of global and local structures of 3D point clouds as a free and powerful supervision signal for representation learning. Local and global patterns of a 3D object are closely related. Although each part of an object is incomplete, the underlying attributes about the object are shared among all parts, which makes reasoning about the whole object from a single part possible. We hypothesize that a powerful representation of a 3D object should model the attributes that are shared between parts and the whole object, and distinguishable from other objects. Based on this hypothesis, we propose a new framework to learn point cloud representations by bidirectional reasoning between the local structures at different abstraction hierarchies and the global shape. Moreover, we extend the unsupervised structural representation learning method to more complex 3D scenes. By introducing structural proxies as the intermediate-level representations between local and global ones, we propose a hierarchical reasoning scheme among local parts, structural proxies, and the overall point cloud to learn powerful 3D representations in an unsupervised manner. Extensive experimental results demonstrate that the unsupervised representations can be very competitive alternatives of supervised representations in discriminative power, and exhibit better performance in generalization ability and robustness. Our method establishes the new state-of-the-art of unsupervised/few-shot 3D object classification and part segmentation. We also show our method can serve as a simple yet effective regime for model pre-training on 3D scene segmentation and detection tasks. We expect our observations to offer a new perspective on learning better representations from data structures instead of human annotations for point cloud understanding.

Index Terms—Point cloud, unsupervised learning, representation learning, 3D object recognition, 3D scene understanding

1 Introduction

Facilitating machines to understand the 3D world is crucial to many important real-world applications, such as autonomous driving, augmented reality, and robotics. One core problem on 3D geometric data such as point clouds is learning powerful representations that are discriminative, generic, and robust. To tackle this problem, current state-of-the-art methods on point cloud analysis [2], [37], [39], [44], [55], [63], [69], [71], [75] are established with the help of extensive human-annotated supervised information. However, manually labeled data require high costs of human labor and may limit the generalization ability of the learned models. Therefore, unsupervised learning is an attractive direction to obtain generic and robust representations for 3D object understanding.

Learning useful representations from unlabeled data is a fundamental and challenging problem for point cloud analysis. While several efforts have been devoted to learn representations of a point cloud without human supervision [1], [11], [20], [26], [37], [42], [67], [76], [80], these methods are mainly based on self-supervision signals provided by generation or reconstruction tasks, including self-reconstruction [1], [11], [20], [37], [67], [76], [80], local-to-global reconstruction [26], [42] and distribution estimation [1], [37]. These methods have proven to be effective in capturing structural and low-level information of point clouds but usually fail to learn high-level semantic information. Therefore, unsupervised models still perform far behind the state-of-the-art supervised models. The goal of this work is to explore an unsupervised learning algorithm that can learn both structural information and semantic knowledge to promote the quality of learned representations.

Different from images where local patches are noisy and usually independent from the whole image (for example, given a patch of a dog, we cannot identify whether this image is about animals or the people nearby), the underlying semantic and structural information are shared in all parts of a 3D object. This distinct property of 3D objects makes reasoning the whole object from any part possible. Based on this observation, we hypothesize that a powerful representation of a 3D object should model the underlying attributes that are shared between parts and the whole object and distinguishable from other objects. As shown in Fig. 1, given a point cloud of the tail of an airplane, a good representation of the tail should reflect the type of the corresponding airplane. Meanwhile, the representation of the whole airplane should contain all the necessary details to infer the local structures of this airplane.

• *The authors are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: raoyongmimg95@gmail.com, {lujiwen, jzhou}@tsinghua.edu.cn.*

Manuscript received 26 May 2021; revised 24 February 2022; accepted 13 March 2022. Date of publication 16 March 2022; date of current version 6 January 2023.
This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grants 62125603 and U1813218, and in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI).
(Corresponding author: Jiwen Lu.)
Recommended for acceptance by K. Schindler.
Digital Object Identifier no. 10.1109/TPAMI.2022.3159794

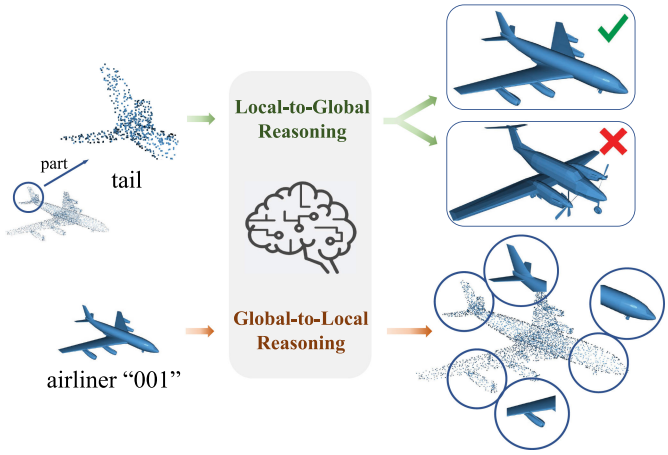

Fig. 1. *Illustration of our main idea.* We propose to learn representations in an *unsupervised* manner from data structures by training the networks to solve two problems: reasoning the whole object from a single part and reasoning detailed structures from the global representation.

In this paper, we propose a new scheme called *PointGLR* for unsupervised point cloud representation learning by bidirectional reasoning between local representations at different abstraction hierarchies in a network and the global representation of a 3D object. Our method is simple yet effective and can be applied to a wide range of deep learning methods for point cloud understanding. While most existing unsupervised learning methods focus on exploiting structural information by learning various autoencoders, our method aims to capture the underlying semantic knowledge shared between local structures and global shape in 3D point clouds. Specifically, the proposed Global-Local Reasoning consists of two sub-tasks: 1) local-to-global reasoning: we formulate the problem of capturing shared attributes between local parts and global shape as a self-supervised metric learning problem, where local features are encouraged to be closer to the global feature of the same object than features of other objects, such that the distinct semantic information of each object can be extracted by local representations; 2) global-to-local reasoning: we further use the self-supervised tasks including self-reconstruction and normal estimation to learn global features that contain necessary structural information of 3D objects. Benefiting from the bidirectional reasoning framework, we can simply combine the global and local features to obtain unsupervised representations with both semantic and structural knowledge of the 3D point cloud.

While the global and local patterns of a 3D object are closely related, the structural relations in 3D scenes are more complex and thus directly reasoning between the global and local representations of a 3D scene is difficult. To extend our method to real-world 3D scenes, we design a new hierarchical global-local reasoning scheme (*PointHGLR*). By introducing structural proxies as the intermediate-level representations between local and global ones, the model is required to perform bidirectional reasoning hierarchically among local parts, structural proxies, and the overall point cloud. Specifically, we first divide the whole scene into several local regions and the structural proxies are obtained by aggregating the local representations in each local region. Based on the structural proxies, two bidirectional reasoning sub-tasks are designed: 1) predicting the corresponding structural proxies from the local representation and recovering detailed point clouds from multiple structural proxies and 2) predicting the global representation from the structural proxy and reconstructing the coarse skeleton of the whole scene from the global representation. Different from recent progress on 3D model pre-training like PointContrast [74], our method introduces no assumption on the training data and models while [74] is only designed for multi-view point clouds and sparse convolution backbones [8]. These properties make our method suitable for *all types of* point clouds and *off-the-shelf* backbone models.

Extensive experiments on both synthetic and real-world 3D understanding datasets demonstrate that the unsupervised representations can be very competitive alternatives of supervised representations in discriminative power, and exhibit better performance in generalization ability and robustness. Our method establishes the new state-of-the-art of unsupervised/few-shot 3D object classification and part segmentation. We show the unsupervised models can consistently outperform their supervised counterparts. With our unsupervised learning method, we show a simple and light-weight SSG PointNet++ [55] model can achieve very competitive results with supervised methods (92.2% classification accuracy on ModelNet40 [72]). By simply increasing the channel width, we further obtain 93.0% and 87.2% single view accuracy on ModelNet40 and ScanObjectNN [66] benchmarks respectively, surpassing the state-of-the-art unsupervised and supervised methods, while the supervised version of this model suffers from overfitting. We also show our method can serve as a simple yet effective regime for model pre-training on 3D scene segmentation and detection tasks without using extra training data. Code is available at https://github.com/raoyongming/PointGLR.

This paper is an extended version of our conference paper [56]. We make several new contributions: 1) we design a new hierarchical global-local reasoning scheme, which extends our method to learn 3D scene representations in an unsupervised manner; 2) we provide more in-depth analysis and visualization of our method. We evaluate our method on two new tasks: 3D object part segmentation and few-shot object classification to show the generalization ability of the unsupervised representations. We also develop a new few-shot learning benchmark for 3D point cloud to compare with prevalent few-shot learning algorithms; 3) we conduct extensive experiments on 3D scene understanding tasks including 3D object detection and semantic segmentation to verify the effectiveness of our new method.

2 Related Work

In this section, we briefly review recent advancements in three related topics: deep learning on 3D point clouds, unsupervised representation learning, and model pre-training.

2.1 Deep Learning on 3D Point Clouds

Recent years have witnessed rapid development on 3D point cloud analysis thanks to the deep learning techniques that are designed to consume 3D point clouds directly [39], [44], [54], [55], [69]. PointNet [37] pioneers this line of works and designs a deep network that can handle unordered and unstructured 3D points by independently learning on each point and fusing point features with max pooling. Though

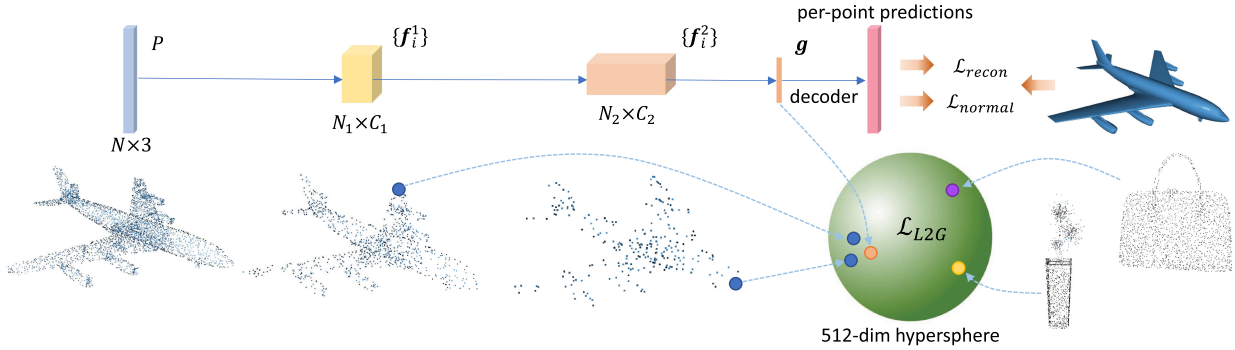

Fig. 2. *The overall framework of our unsupervised feature learning approach.* The representations are learned by connecting local structures and global shapes. We map the local representations at different levels and global representations to the shared feature space and use a self-supervised metric learning objective to mine semantic knowledge from data. By further incorporating self-reconstruction and normal estimation tasks, powerful representations that contain rich semantic and structural information can be learned.

efficient, PointNet fails to capture local structures, which have proven to be crucial to the success of CNNs. PointNet++ [55] is proposed to mitigate this issue by developing a hierarchical grouping architecture to extract local features progressively at different abstraction levels. The subsequent works such as PointCNN [39], PointConv [71] and RelationShape CNN [44] also focus on local structures of point cloud and further improve the quality of captured features. Since only the relation between local and global features is needed, our method is suited for all these PointNet++ variants. While recent works push state-of-the-art of point cloud deep learning by promoting the capacity of networks, this work offers a new route to learn powerful representations in an *unsupervised* fashion, without any human annotations.

2.2 Unsupervised Representation Learning

Unsupervised learning has been an important group of methods in computer vision since the earliest day [19], which aims to learn transformations of the data that make the subsequent downstream problem easier to solve [5]. Classical deep methods for unsupervised learning such as autoencoders [31], generative adversarial networks [23] and autoregressive models [48] learn representations by faithfully reconstructing the input data, which focus on low-level variations in data and are not very useful for downstream tasks like classification. Recent works on self-supervised learning present a powerful family of models that can learn discriminative representations with rich semantic knowledge. This group of methods design various problem generators such that models need to learn useful information from data in order to solve these generated problems [3], [14], [15], [29], [64]. In this work, we also follow this line and propose to learn point cloud representations by solving the global-local bidirectional reasoning problem.

There are several prior attempts on learning representations of a point cloud without human supervision [1], [11], [20], [26], [37], [42], [67], [76], [80]. These methods discover useful information in the 3D point cloud by performing data reconstruction, which has proven to be effective in learning structural information. However, because of lacking effective semantic supervision, previous methods limit the networks' ability in downstream tasks. Our method resolves this issue by incorporating semantic supervision with structural supervision. With the exploration of high-level semantic knowledge, our method is able to learn discriminative representations like the supervised methods while maintaining the robustness and generalization of unsupervised representations.

2.3 Model Pre-Training

Model pre-training has become a popular practice in many deep learning applications, including computer vision [7], [28], [74] and natural language processing [50]. In 2D computer vision tasks, a widely used pipeline is first conducting model pre-training on ImageNet [12] with full supervision, and then fine-tuning the pre-trained model on downstream tasks like detection [22] and semantic segmentation [45]. Recently, unsupervised pre-training based on contrastive learning [7], [28] has shown impressive results on downstream tasks. Compared to 2D visual understanding, there has been less exploration on 3D tasks. Most existing 3D pre-training methods either focus on benefiting the tasks at single object level, including classification, reconstruction and part segmentation [20], [27], [76], or some low-level tasks like registration [11], [16], [78]. Modeling pre-training for more complex 3D scene understanding tasks like detection and segmentation has not been studied only until a recent work *PointContrast* [74], which exploits the point correspondence across multiple 3D point clouds from different views to learn the representations by performing the emerging contrastive representation learning method [7], [28]. Different from [74] that requires multi-view sensor data to learn correspondence, we show that model pre-training using no extra data other than the input point cloud can also yield significant and consistent performance improvement on downstream tasks.

3 Approach

The core of 3D point cloud understanding is to learn discriminative, generic, and robust representations that can capture the underlying shape. To achieve this goal in an unsupervised manner, we propose to learn point cloud representations by solving a bidirectional reasoning problem between the local structures and the global shape. The overall framework of our method is presented in Fig. 2.

3.1 Hierarchical Point Cloud Feature Learning

We begin by reviewing the hierarchical point cloud feature learning framework first proposed in PointNet++ [55], on which our method is built.

Consider a set of 3D points $P \subset \mathbb{R}^3$ with N elements, in which each point p_i is represented by a 3D coordinate. To learn features based on these 3D coordinates, PointNet [54] proposes to use a symmetric function f that is invariant to point permutations to transfer point set into feature space

$$f(P) = \mathcal{A}(h(p_1), h(p_2), \ldots, h(p_N)), \tag{1}$$

where h is a multi-layer perceptron network that processes each point independently and shares parameters for all points and $\mathcal{A}$ is a symmetric aggregation function like max pooling to summarize features from each point. Since each point is processed independently by h, the structural information among points is captured only by the aggregation function $\mathcal{A}$. Therefore, PointNet lacks the ability to capture local context. To address this issue, PointNet++ and its variants [39], [44], [71] use a hierarchical structure to learn point cloud feature progressively at different abstraction levels. Specifically, at the ℓth level, the point set is abstracted by using iterative furthest point sampling [55] to produce a new set $P^\ell \subset P^{\ell-1}$ with fewer points and we can extract the local geometrical feature $\mathbf{f}_i^\ell$ by applying a small PointNet on the local point subset around the centroid for each point $p_i^\ell \in P^\ell$. The global representation of the point cloud $\mathbf{g}$ is then obtained by applying another small PointNet model on the points and features at the highest abstraction level.

Almost all previous works [2], [37], [39], [44], [55], [63], [69], [71], [75] on supervised point cloud learning employ an end-to-end training paradigm, where the representations are learned directly from the annotated labels. Although having achieved promising performance, these methods neglect the intrinsic semantic and structural information contained in the point clouds themselves. In this work, we focus on exploring this property of point clouds and provide a very competitive alternative for point cloud representation learning.

To discover the structure and semantic information from data without human annotations, we propose two problems for the networks to solve: *local-to-global reasoning* and *global-to-local reasoning*, which aim to learn semantic and structural knowledge in an *unsupervised* fashion respectively.

3.2 Local-to-Global Reasoning

Humans are able to recognize many objects even when only a small part of the object is presented. This fact inspires us to exploit the relation between local parts and global shape as a free and plentiful supervisory signal for training a rich representation for point cloud understanding. Therefore, the goal of local-to-global reasoning is to mine the shared semantic knowledge among different abstraction hierarchies of point clouds. Since the global representation usually can better capture the semantic information of 3D objects than local representations, local-to-goal reasoning operates by predicting the global representation from local ones. To evaluate the predictions, we formulate the prediction as a self-supervised metric learning problem and use a multi-class N-pair loss [59] to supervise the prediction task. To learn the distinct semantic information for each object, we treat the global representation of the current object as the *positive* sample and use the global representations of other objects as the *negative* samples inspired by the idea of instance discrimination [73]. In the following, we describe the details of the local-to-global reasoning.

Prediction Networks. Since the local features $\{\mathbf{f}_i^\ell\}$ and global feature $\mathbf{g}$ have different numbers of channels, we cannot directly measure the similarity of them. Thus, we first use prediction networks $\{\boldsymbol{\phi}^\ell, \ell = 1, 2, \ldots, L\}$ and φ to embed them into a shared feature space, respectively. The prediction networks can be implemented as multi-layer perceptron (MLP) networks and are shared at each abstraction level.

Self-Supervised Metric Learning. A straightforward method to optimize the predictions is to minimize the absolute overall differences between $\boldsymbol{\phi}^\ell(\mathbf{f}_i^\ell)$ and $\varphi(\mathbf{g})$, i.e., minimize $\sum_{i,\ell} ||\boldsymbol{\phi}^\ell(\mathbf{f}_i^\ell) - \varphi(\mathbf{g})||$. However, this objective may lead to degenerate representations that map all inputs to a constant value. Therefore, we choose to supervise the *relative* quality of the predictions with an unsupervised metric learning task. Specifically, for each embedded local representation $\mathbf{f}_i^\ell$, we enforce its embedding to be closer to the embedded global representation of the same object than any other object. The local-to-global reasoning objective can be written as

$$\mathcal{L}_{\text{L2G}}^{i,\ell} = \log\left(1 + \sum_{\mathbf{g}_k \neq \mathbf{g}} \exp(s\boldsymbol{\phi}^\ell(\mathbf{f}_i^\ell)^\top \varphi(\mathbf{g}_k) - s\boldsymbol{\phi}^\ell(\mathbf{f}_i^\ell)^\top \varphi(\mathbf{g})\right), \tag{2}$$

and

$$\begin{aligned}\mathcal{L}_{\text{L2G}} &= \frac{1}{M}\sum_{i,\ell} \mathcal{L}_{\text{L2G}}^{i,\ell} \\ &= -\frac{1}{M}\sum_{i,\ell} \log \frac{\exp(s\boldsymbol{\phi}^\ell(\mathbf{f}_i^\ell)^\top \varphi(\mathbf{g}))}{\sum_k \exp(s\boldsymbol{\phi}^\ell(\mathbf{f}_i^\ell)^\top \varphi(\mathbf{g}_k))},\end{aligned} \tag{3}$$

where $\{\mathbf{g}_k, k = 1, 2, \ldots, m\}$ are the global representations of different point sets in the mini-batch with batch size m and M is the number of local features. Inspired by the studies on metric learning for face recognition [13], [41], [68] that perform metric learning on features on a hypersphere, we normalize the outputs of prediction networks before computing similarities and use a constant value $s = 64$ [13] to re-scale the features. Empirically, our experiments show that forcing features to be distributed on a hypersphere with a radius of s will significantly stabilize the training process and improve the discriminative ability of the learned features.

Discussions. The proposed local-to-global reasoning is connected to mutual information maximization methods [3], [29], [32], [64] for unsupervised image representation learning. The multi-class N-pair loss can be viewed as a variant of InfoNCE [49]. Therefore, minimizing the $\mathcal{L}_{\text{G2L}}$ maximizes the lower bound of the mutual information between local representations and the global representation. From this perspective, our method captures the underlying semantic knowledge of a 3D object by maximizing the mutual information of features at different hierarchies. Unlike previous works that perform adversarial learning between the mutual information estimator and the feature encoder [32] or maximizes the mutual information of seen patches and unseen patches [29], different views of images [3] or different modalities of images [64] , our work explores the distinct property of point clouds by connecting local and global structures of a 3D

object. Furthermore, our local-to-global loss offers a metric learning view of InfoNCE, which is different from previous works that are based on Noise-Contrastive Estimation [47]. Benefiting our modifications inspired by metric learning and face recognition methods, we observe that our loss is more effective and stable than previous methods on point cloud understanding tasks in our experiments. Our method is also related to recent self-supervised learning methods without negative samples like BYOL [24] and DINO [6]. The proposed local-to-global reasoning can be viewed as a self-distillation framework [6] where we use the knowledge learned by the deep and global representations to teach the shallow and local representations. Different from self-supervised methods in 2D [6], [24] that pass the global and local views of the same image to the whole model to obtain the global and local representations, we directly use the features from different hierarchies to efficiently form the global and local representations. Extending our method to 2D domain may be an interesting future direction to improve the training efficiency of the image self-supervised learning algorithms.

3.3 Global-to-Local Reasoning

Since discovering knowledge that is helpful for downstream tasks from unlabeled data is usually quite intractable, local-to-global reasoning may not necessarily lead to useful representations. This fact is also pointed out by studies on mutual information maximization methods [64], [65], where evidence shows that larger mutual information may not guarantee a better performance for downstream tasks [65]. Intuitively, since the local-to-global reasoning only supervises the local representations to be close to the global one, the quality of the global representation is critical. That is, if the global representation is well initialized, decent supervision to local representations will be offered, thus creating a *virtuous circle* for the learning of local and global features. On the contrary, the learning process may lead to unpredictable results for the bad initial state of the global representation. To avoid this issue, we propose an auxiliary global-to-local reasoning task to supervise the networks for learning useful representations corporately. Specifically, we employ two low-level generation tasks, including self-reconstruction and normal estimation as two self-supervision signals, such that global representation needs to capture the basic structural information of point clouds.

Self-Reconstruction. Self-reconstruction, or point autoencoding, is a widely used technique for unsupervised point cloud representation learning [1], [11], [20], [37], [67], [76], [80]. To perform self-reconstruction, we adopt the folding-based [76] decoder D to deform the canonical 2D grid onto 3D coordinates of a point cloud conditioned on the global representation $\mathbf{g}$. The reconstruction error is defined as Chamfer Distance [17]

$$\begin{aligned}\mathcal{L}_{\text{recon}} &= ChamferDist(D(\mathbf{g}), P) \\ &= \frac{1}{N}\left(\sum_{p\in P}\min_{x\in D(\mathbf{g})}\|x-p\|_2 + \sum_{x\in D(\mathbf{g})}\min_{p\in P}\|x-p\|_2\right). \quad (4)\end{aligned}$$

Normal Estimation. Normal estimation is a more challenging task that requires a higher-level understanding of the underlying surface information of a 3D shape. Different from previous works [44] that pursue the estimation precision, we use this task as a supervisory signal to improve the global representation. Thus, we simply concatenate the 3D coordinates with the global representation and employ a shared light-weight MLP σ to produce the estimated normals. The cosine loss is used to measure the estimation error

$$\mathcal{L}_{\text{normal}} = 1 - \frac{1}{N}\sum_{i}\cos\left(\sigma([p_i, \varphi(\mathbf{g})]), p_i^{\text{normal}}\right). \quad (5)$$

Combining the local-to-global reasoning and the global-to-local reasoning, we arrive at the global-local bidirectional reasoning objective

$$\mathcal{L}_{\text{GLR}} = \mathcal{L}_{\text{L2G}} + \mathcal{L}_{\text{G2L}} = \mathcal{L}_{\text{L2G}} + \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{normal}}. \quad (6)$$

3.4 Hierarchical Reasoning With Structural Proxies

Compared to 3D objects, which have closely related local and global patterns to make the above-mentioned bidirectional reasoning feasible, the geometric and semantic structures in 3D scenes are far more complex. Our experimental study shows directly applying the above method may not lead to satisfactory results. To better extend our method to more complex 3D scenes, we devise an intermediate level called structural proxy to bridge the local and global representations. We construct the structural proxies by dividing the whole scene into M_{proxy} regions and the representation of each region can be defined as the aggregation of local representations in the region

$$\mathbf{f}_i^{\text{proxy}} = \mathcal{A}(\omega(\mathbf{f}_{n_1}^L), \ldots, \omega(\mathbf{f}_{n_m}^L)), \quad (7)$$

where $\mathbf{f}_{n_1}^L, \ldots, \mathbf{f}_{n_m}^L$ are m local representations from the points in the ith region $\mathcal{N}_i$, $\mathcal{A}$ is the max pooling operation to summarize the representations, and ω is a point-wise network like MLP. Note that in order to encode the high-level semantic patterns in the region, we directly summarize the highest level representations $\{\mathbf{f}_i^L\}$ in the feature extractor to compose the structural proxies. We form the local region $\mathcal{N}_i$ by first finding M_{proxy} region centers using iterative furthest point sampling and then dividing all points to each region by assigning them to the closest center point.

After obtaining the structural proxies, we extend the above global-local reasoning method to more general hierarchical reasoning. The hierarchical global-local reasoning (HGLR) objective can be written as

$$\mathcal{L}_{\text{HGLR}} = \mathcal{L}_{\text{GLR}}^{\text{Local}\leftrightarrow\text{Proxy}} + \mathcal{L}_{\text{GLR}}^{\text{Proxy}\leftrightarrow\text{Global}}, \quad (8)$$

where the local-proxy reasoning performs the proposed bidirectional reasoning loss between the local representations in the local region and the corresponding proxy, and the proxy-global reasoning performs the proposed bidirectional reasoning between the structural proxies and the global representation of the 3D scene. More specifically, the bidirectional reasoning between local representations and structural proxies is defined as

$$\mathcal{L}_{\text{GLR}}^{\text{Local}\leftrightarrow\text{Proxy}} = \sum_{i,\ell}\mathcal{L}_{\text{L2P}}^{\ell} + \mathcal{L}_{\text{P2L}}, \quad (9)$$

where

$$\mathcal{L}_{\text{L2P}}^{\ell} = -\frac{1}{M}\sum_{j}\sum_{\mathbf{f}_i^{\ell}\in\mathcal{N}_j}\log\frac{\exp(s\boldsymbol{\phi}^{\ell}(\mathbf{f}_i^{\ell})^{\top}\boldsymbol{\varphi}^{\text{proxy}}(\mathbf{f}_j^{\text{proxy}}))}{\sum_k \exp(s\boldsymbol{\phi}^{\ell}(\mathbf{f}_i^{\ell})^{\top}\boldsymbol{\varphi}^{\text{proxy}}(\mathbf{f}_k^{\text{proxy}}))}, \tag{10}$$

and

$$\mathcal{L}_{\text{P2L}} = \sum_{j} ChamferDist(D_{\text{L2P}}(\mathbf{f}_j^{\text{proxy}}), \mathcal{N}_j). \tag{11}$$

Here we use $\boldsymbol{\phi}^{\ell}$ and $\boldsymbol{\varphi}^{\text{proxy}}$ to map the local representations from level ℓ and structural proxy of region j to the hypersphere. $\mathbf{f}_k^{\text{proxy}}$ represent the structural proxies from other regions of the scene and regions of other scenes. D_{L2P} is a FoldingNet [76] model that inversely maps the structural proxy to the point clouds in the corresponding region.

Similarly, we define the bidirectional reasoning objective between the structural proxies and the global feature of the scene as

$$\mathcal{L}_{\text{GLR}}^{\text{Proxy}\leftrightarrow\text{Global}} = \mathcal{L}_{\text{P2G}} + \mathcal{L}_{\text{G2P}}, \tag{12}$$

where

$$\mathcal{L}_{\text{P2G}}^{\ell} = -\frac{1}{M_{\text{proxy}}}\sum_{j}\log\frac{\exp(s\boldsymbol{\phi}^{\text{proxy}}(\mathbf{f}_j^{\text{proxy}})^{\top}\boldsymbol{\varphi}(\mathbf{g}))}{\sum_k \exp(s\boldsymbol{\phi}^{\text{proxy}}(\mathbf{f}_j^{\text{proxy}})^{\top}\boldsymbol{\varphi}(\mathbf{g}_k))}, \tag{13}$$

and

$$\mathcal{L}_{\text{G2P}} = ChamferDist(D_{\text{G2P}}(\mathbf{g}), P_{\text{coarse}}). \tag{14}$$

Correspondingly, $\boldsymbol{\phi}^{\text{proxy}}$ and $\boldsymbol{\varphi}$ are the projections from structural proxies and the global feature to the shared space. D_{G2P} is the FoldingNet decoder to recover the coarse version of the input point cloud P_{coarse}.

The overall framework of the proposed hierarchical global-local reasoning is illustrated in Fig. 3.

3.5 Point Cloud Analysis With GLR

Point cloud representations can be learned in an *unsupervised* manner by enforcing networks to solve the proposed global-local reasoning (GLR) problems, where the representations can be used in various downstream point cloud analysis applications varying from single object analysis and 3D scene understanding. In this subsection, we detail how to apply the unsupervised model to various downstream tasks.

3D Object Classification. The most widely used metric for evaluating the quality of unsupervised representations is the linear separability of the classification task [1], [7], [26], [28], where a supervised linear SVM [9] model or single-layer neural network is trained on unsupervised representations to measure the test accuracy. For PointNet++ [55] model and its variants, we use the aggregated representation for the classification task, which is obtained by summarizing embedded global and local representations

$$\mathbf{f} = [\mathcal{A}(\{\boldsymbol{\phi}^1(\mathbf{f}_i^1)\}), \ldots, \mathcal{A}(\{\boldsymbol{\phi}^L(\mathbf{f}_i^L\})), \boldsymbol{\varphi}(\mathbf{g})], \tag{15}$$

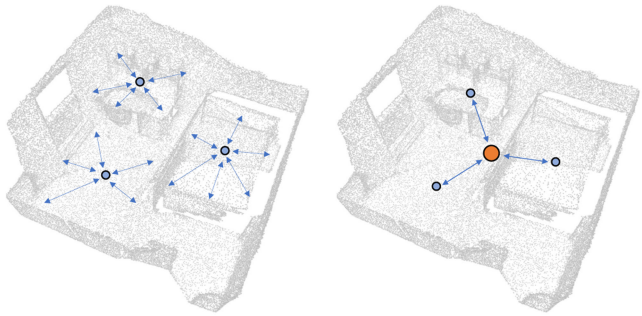

Fig. 3. *Hierarchical reasoning with structural proxies for 3D scenes.* Given a point cloud of a 3D scene, we perform hierarchical bidirectional reasoning by constructing the global representation (orange point) as well as the structural proxies of the scene (blue points). The reasoning task is divided into two sub-tasks: 1) bidirectional reasoning between the local features and structural proxies (left), and 2) bidirectional reasoning between the structural proxies and the global representation (right).

where we use a max pooling operation $\mathcal{A}$ to aggregate local features of each abstraction level from 1 to L and concatenate these features with the global feature.

Apart from the above unsupervised learning setting, we also evaluate our method based on a *hybrid learning* framework where GLR serves as an auxiliary loss to further improve the robustness of representations. In this setting, the supervised global representation can be viewed as a good initialization for the proposed GLR framework.

For the 3D object classification task, we consider two baseline models: PointNet++ [55] and Relation-Shape CNN (RSCNN) [44]. Note that for both baseline models, we use the Single-Scale Grouping (SSG) [55] as the point grouping module, which is more than 3× smaller than Multi-Scale Grouping (MSG) [55] module used in the original PointNet++ model. Besides, we divide the MLP used in each set abstraction layer into two fully connected layers and use them before and after aggregation operation, respectively. Our experiments show this modification can reduce computation and improve performance while keeping the number of parameters unchanged. For unsupervised learning setting, we train a linear SVM [9] on unsupervised representations of the training data and report the classification accuracy on the test set. For supervised learning and hybrid learning settings, we use the aforementioned aggregated representation for fair comparison and employ a two-layer classifier where dropout technique [61] with a ratio of 50% is used for each layer. We follow the network architecture configuration of [44], where our basic models of both PointNet++ and RSCNN have three hierarchical levels and the numbers of output channels are 128, 512, 1,024 respectively. We fix the channel of embedding features as 512 in all our experiments. Detailed model configurations can be found in Supplementary Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3159794.

3D Object Part Segmentation. We evaluate the quality of the learned local features on the 3D object part segmentation task. Given a model pre-trained by PointGLR, we extract the local features from different abstraction levels and upsample them hierarchically using three nearest neighborhoods interpolation following [55]. The dense local features of the point cloud are then formed by concatenating the upsampled

features from different levels. Note that this process does not introduce any extra parameters, thus we do not need to fine-tune the pre-trained model for dense prediction tasks. To measure the part segmentation performance of the pre-trained features, we follow the protocol described in [26], where a single linear layer is trained to map the pre-trained features to the segmentation categories. To show the linear separability of the features, we fix the feature extraction model during training the linear layer.

3D Object Few-Shot Classification. Few-shot classification task is usually considered as a useful tool to show the generalization ability of the learned model, which is widely studied in the image recognition literature [18], [36], [58], [62]. Previous work on few-shot classification can be divided into metric based methods [58], [62] and optimization based methods [18], [36]. Different from them, we show that our unsupervised representation learning method with a linear SVM classifier is also a strong baseline for point cloud few-shot classification. In most few-shot learning literature, the dataset is divided into a meta training set and a meta test set. To fairly compare with other few-shot learning methods, we perform *unsupervised* representation learning on the support set and train the SVM classifier on the training set of each few-shot task during testing. For baseline methods, we follow the original implementation for image classification and only replace the backbone with our modified PointNet++ model for the point cloud classification task. The detailed configurations of the few-shot classification task and the technical details of baseline few-shot learning methods can be found in Supplementary Material, available online.

3D Scene Detection & Segmentation. We adopt the *unsupervised pre-training + supervised fine-tuning* pipeline to verify the effectiveness of our method on 3D scene semantic segmentation and object detection tasks. Specifically, the pipeline consists of two stages. In this first stage, we pre-train the model by performing the proposed hierarchical reasoning task on the training set without using the annotation. In the second stage, we fine-tune the model on the training set. Although our pre-training process does not require any extra data, we show the model initialized with the pre-trained weights can lead to better performance on the final task. For 3D object detection task, we use the widely used VoteNet [51] and H3DNet [79] as our baseline methods. For the 3D semantic segmentation task, we use the high-performance Sparse Residual U-Net model [8] as our baseline.

4 Experiments

We extensively evaluate our method on both object level and scene level 3D point cloud understanding benchmark datasets including ModelNet10/40 [72], ShapeNet [72], ScanObjectNN [66] and ScanNet [10]. We start by evaluating our method on the discriminative power, generalization ability, and robustness across datasets and comparing with the state-of-the-art unsupervised and supervised methods on the 3D object classification task in Section 4.1. We also provide detailed experiments to analyze our method on model design and complexity. Then, we extend our method on four different tasks including 3D object part segmentation (Section 4.2), 3D object few-shot classification (Section 4.3) and 3D scene object detection and semantic segmentation (Section 4.4).

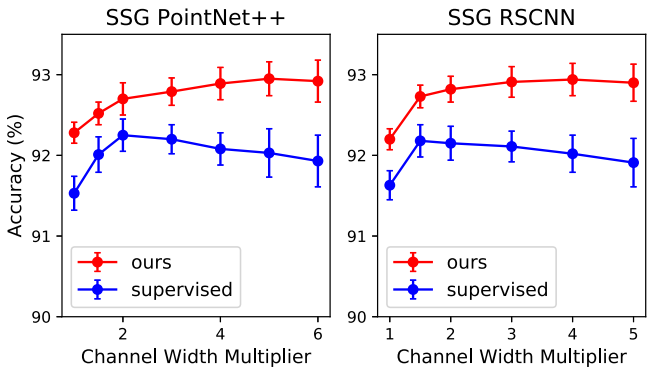

Fig. 4. ModelNet40 classification accuracy (%) of our unsupervised models and their supervised counterparts. We report the median ± standard deviation across 3 identical runs.

Finally, we visualize the learned representations to have an intuitive understanding of our method in Section 4.5. The following describes the details of the experiments, results, and analysis.

4.1 3D Object Classification

4.1.1 Setups

We test our method on ModelNet40/10 [72] and ScanObjectNN [66] benchmarks to compare with the state-of-the-art. ModelNet40 and ModelNet10 comprise 9832/3991 training objects and 2468/908 test objects in 40 and 10 classes respectively, where the points are sampled from CAD models. ScanObjectNN [66] is a real-world dataset, where 2902 3D objects are extracted from scans. To conduct cross dataset evaluation, we use the "object-only" split in all our experiments. ScanNet [10] is also used in our cross dataset evaluation experiments, where we follow the practice in [39] to obtain point cloud from indoor scenes according to the instance segmentation labels. In all our experiments, we sample 1,024 points for each point cloud for training and evaluation and all our results are measured using a single view without using the multi-view voting trick to better show the performance of different models. Surface normal information was used to provide unsupervised signals for our models trained on ModelNet and we did not use it as input. For the models trained on ScanObjectNN and ScanNet, we only used the self-reconstruction loss for global-to-local reasoning.

4.1.2 Comparisons With the Supervised Counterparts

We first compare our method with the supervised baselines as presented in Fig. 4, where we report the classification accuracy on ModelNet40 using the basic models (1×) and wider models (1.5 to 6× channel width). Note that we used the same network architecture and training settings for our models and their counterparts for a fair comparison. Clearly, our unsupervised models with different channel widths consistently outperform the supervised counterparts. By increasing the model capacity, our models can achieve better performance and reach the highest accuracy using 5× PointNet++ and 4× RSCNN backbones. In the following experiments, we denote the basic 1× models and the best models as "Small" and "Large" models respectively. Besides, we further compared three different training strategies: unsupervised learning,

TABLE 1
Classification Accuracy (%) of Three Different Training Strategies on ModelNet40

Backbone	Unsupervised	Supervised	Hybrid
PN++ (Small)	92.28	91.53	**92.50**
PN++ (Large)	**92.96**	92.03	92.68
RSCNN (Small)	92.20	91.63	**92.32**
RSCNN (Large)	**92.94**	92.02	92.79

supervised learning, and hybrid learning, which are presented in Table 1. We see hybrid learning can outperform both supervised and unsupervised models when the networks are small, but the unsupervised method achieves the best performance when large networks are used. We conjecture that the supervised models are prone to overfitting more severely to the training set. All these results reveal that our unsupervised representations are more discriminative and generalizable than their supervised counterpart.

4.1.3 Comparisons With the Unsupervised State-of-the-Arts

To show the effectiveness of the proposed global-local reasoning method, we compared several variants of our models with the state-of-the-art unsupervised representation learning methods in Table 2. Except for point-based methods, we also compare with some advanced voxel and view-based methods. Note that we only use ModelNet40 as the training data, while some methods are trained on larger ShapeNet [72] dataset. Nevertheless, our models outperform all other methods by a large margin. As can be observed, our small PointNet++ model surpasses state-of-the-art methods and our large model significantly advances the best point cloud model (MAP-VAE) by 2.81% on ModelNet40.

4.1.4 Comparisons With the Supervised State-of-the-Arts

More notably, our *unsupervised* method can even achieve very competitive results compared to state-of-the-art supervised methods. We compared our method with the supervised methods on both the widely used synthetic dataset ModelNet and the recently proposed real-world dataset ScanObjectNN. Our unsupervised representations were trained on ModelNet40 and a linear SVM is then trained on the target dataset to produce predictions. The results are summarized in Tables 3 and 4. Surprisingly, our unsupervised learned representations can outperform all other state-of-the-arts methods in the single-view setting[1] on both datasets. Since only a linear classifier is applied, these results demonstrate that our representations are much more discriminative than the supervised representations on the test set. Moreover, we observe that our representations can achieve very strong results on ScanObjectNN without finetuning. As the categories in ModelNet and ScanObjectNN are different, this evidence indicates that our method can discover semantic knowledge shared in different kinds of objects.

1. Here we borrow the concept of "view" from image recognition literature, where the number of views represents the number of augmented inputs (e.g., rotated or scaled point clouds) used during testing.

TABLE 2
Comparisons of the Classification Accuracy (%) of Our Method Against the State-of-the-Art *unsupervised* 3D Representation Learning Methods on ModelNet40 and ModelNet10

Method	Input	Accuracy	
		MN40	MN10
TL Network [21]	voxel	74.40	-
VConv-DAE [57]	voxel	75.50	80.50
3DGAN [70]	voxel	83.30	91.00
VSL [40]	voxel	84.50	91.00
VIPGAN [25]	views	91.98	94.05
LGAN† [1]	points	85.70	95.30
LGAN [1]	points	87.27	92.18
FoldingNet† [76]	points	88.40	94.40
FoldingNet [76]	points	84.36	91.85
MRTNet† [20]	points	86.40	-
3D-PointCapsNet [80]	points	88.90	-
MAP-VAE [26]	points	90.15	94.82
Ours w/ PN++ (Small)	points	92.28	94.93
Ours w/ PN++ (Large)	points	**93.96**	**95.62**
Ours w/ RSCNN (Small)	points	92.20	94.56
Ours w/ RSCNN (Large)	points	92.94	95.44

† *indicates that the model is trained on ShapeNet.*

4.1.5 Cross Dataset Evaluation

To further explore the generalization ability of the learned representations, we conducted extensive cross dataset evaluation experiments on ModelNet, ScanObjectNN, and ScanNet, which are varying in categories and sources. Our experiments were conducted based on the unsupervised representations of the PointNet++ large model and we compared the results with the supervised version of this model. Specifically, we trained the features using supervised or

TABLE 3
Comparisons of the *Single-View* Classification Accuracy (%) of Our Method Aganist the State-of-the-Art *supervised* Point Cloud Models on *ModelNet40*

Method	#Points	Supervised	Acc.
PointNet [54]	1k	✓	89.2
PointNet++ [55]	1k	✓	90.5
SO-Net [38]	1k	✓	92.5
PointCNN [39]	1k	✓	92.5
DGCNN [69]	1k	✓	92.9
DensePoint [43]	1k	✓	92.8
RSCNN [44]	1k	✓	92.9
PointNet++ [55] (vote)	1k	✓	90.7
DensePoint [43] (vote)	1k	✓	93.2
RSCNN [44] (vote)	1k	✓	93.6
DGCNN [69]	2k	✓	93.5
PointNet++ [55] (vote, nor)	5k	✓	91.9
SO-Net [38] (nor)	5k	✓	93.4
KPConv [63]	~ 6.8k	✓	92.9
PN++ (Large)	1k	✓	92.0
Ours w/ PN++ (Large)	1k	✗	**93.0**
RSCNN (Large)	1k	✓	92.0
Ours w/ RSCNN (Large)	1k	✗	92.9

We also list results that use more points, normal information ("nor") or/and multi-view voting trick ("vote") in gray as references. Besides, we show the supervised baselines of our models.

TABLE 4
Comparisons of the *Single-View* Classification Accuracy (%) of Our Method Aganist the State-of-the-Art *supervised* Point Cloud Models on *ScanObjectNN*

Method	Supervised	Accuracy
3DmFV [4]	✓	73.8
PointNet [54]	✓	79.2
SpiderCNN [75]	✓	79.5
PointNet++ [55]	✓	84.3
DGCNN [69]	✓	86.2
PointCNN [39]	✓	85.5
Ours w/ PN++ (Large)	✗	**87.2**
Ours w/ RSCNN (Large)	✗	86.9

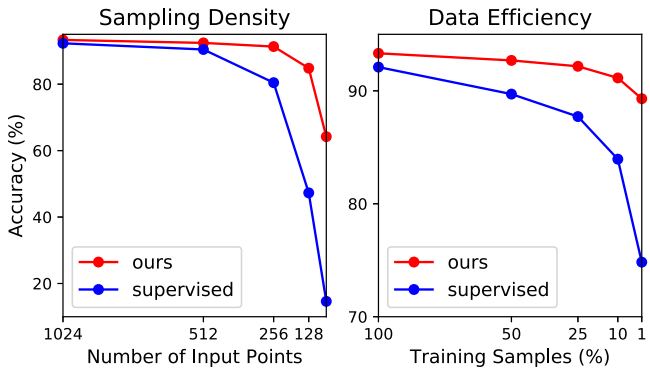

Fig. 5. The *robustness* of our method on sampling density and the number of training samples compared to the supervised baseline.

TABLE 5
Cross Dataset Evaluation

Task	Sup.	Ours	Δ
ModelNet10 → ModelNet30	85.45	92.34	+6.89
ModelNet30 → ModelNet10	91.32	95.47	+4.15
ModelNet40 → ScanObjectNN	65.92	87.22	+21.30
ScanObjectNN → ModelNet40	78.76	90.80	+12.04
ModelNet40 → ScanNet	77.31	89.23	+11.92
ScanNet → ModelNet40	80.38	91.32	+10.94
ScanObjectNN → ScanNet	84.31	87.96	+3.63
ScanNet → ScanObjectNN	82.44	85.43	+2.99

We evaluate generalization ability of unsupervised and supervised representations to unseen datasets. We report the classification accuracy (%) measured using a linear SVM trained on the target dataset. (Sup.: supervised).

unsupervised learning methods on the source dataset and used a linear SVM trained on the target dataset to perform classification. The results are presented in Table 5, where we used the rest 30 categories in ModelNet40 apart from 10 categories in ModelNet10 to form the ModelNet30 dataset. We see the unsupervised representations have much stronger transferability than the supervised counterparts and our models generalize well to various unseen data since we learn from *data structures instead of labels*. Our method can maintain strong performance even in cross data evaluation, which reflects the unsupervised representations can be *generic* representations of 3D objects cross datasets.

4.1.6 Robustness Analysis

The robustness of our method on sampling density and the number of training samples is shown in Fig. 5. For the former, we tested the model trained with 1,024 points with sparser points of 1,024, 512, 256, 128, and 64. Note that different from previous works [44], [55], we did not perform random input dropout during training. For the latter, we trained the representations with randomly sampled 100%, 50%, 25%, 10%, and 1% ModelNet40 training set and trained the linear classifier on the whole set. We used the PointNet++ large model in this experiment. Generally, we see our models are much more robust than their supervised versions. Notably, our method can maintain decent performance even when using only 10% (983 samples) and 1% (98 samples) training samples and achieve 91.4% and 89.3% accuracy on ModelNet40 respectively.

TABLE 6
Ablation Study

Model	$\mathcal{L}_{\text{L2G}}$	$\mathcal{L}_{\text{recon}}$	Agg.	$\mathcal{L}_{\text{normal}}$	SN	Acc.
A		✓				86.77
B	✓					90.02
C	✓	✓				90.96
D	✓	✓	✓			91.69
E	✓	✓	✓	✓		92.22
F	✓	✓	✓	✓	✓	**92.30**

We report the classification accuracy (%) on ModelNet40. ($\mathcal{L}_{\text{L2G}}$: local-to-global reasoning, $\mathcal{L}_{\text{recon}}$: self-reconstruction, Agg: multi-level feature aggregation in Eq. (15), $\mathcal{L}_{\text{normal}}$: normal estimation, SN: training on ShapeNet.)

4.1.7 Ablation Study

To examine the effectiveness of our designs, we conducted a detailed ablation study based on the small PointNet++ network. The results are summarized in Table 6. The baseline model A can be viewed as a variant of FoldingNet [76], which was trained by self-reconstruction loss only and gets a low classification accuracy of 86.77%. We see the model trained by the proposed local-to-global reasoning task (model B) can significantly improve the baseline model by 3.25%. This convincingly verifies its effectiveness. Then, when incorporating these two losses, the accuracy can be further improved to 90.96%. We also observe a 0.73% improvement by aggregating local and global representations (model D). Our full model can be obtained by adding normal estimation supervision (model E), which achieves a notable 92.22% accuracy on ModelNet40 with a very lightweight network. In addition, we also investigated the training set size by adding more training data (model F) from ShapeNet [72], but obtained a slight improvement in accuracy (0.08%). We conjecture that ModelNet is large enough for learning a good representation. Thus we conducted most of the experiments on ModelNet.

4.1.8 Complexity Analysis

Table 7 shows the model complexity in theoretical computation cost (in FLOPs) and actual inference throughput on GPU of our models and several state-of-the-art methods. We see our large model requires considerable computation cost but maintains an acceptable actual cost on GPU due to the simplicity of the SSG model. These results reveal that increasing channel width can achieve a better trade-off on

TABLE 7
Complexity Analysis

Model	FLOPs	Thrput.	Acc.
MSG PN++ [55]	1.68G	113pc/s	90.5
SSG RSCNN [44]	**0.30G**	634pc/s	**92.2**
Our PN++ (Small)	0.31G	**731pc/s**	**92.2**
MSG PN++ [55] (12 votes)	14.15G	9pc/s	90.7
SSG RSCNN [44] (10 votes)	**2.95G**	63pc/s	92.7
Our PN++ (Large)	5.65G	**194pc/s**	**93.0**

We report the FLOPs and GPU inference throughput with batch size 16. Measured on NVIDIA GTX 1080Ti GPU. (pc/s: point clouds per second, Thrput.: Throughput).

speed and accuracy compared to voting. For computational cost-sensitive applications, we think our learned model can provide strong supervision to train lighter models for real-time applications by model distillation [30] or generating pseudo labels [35], which is an interesting direction for future research.

4.2 3D Object Part Segmentation

We evaluate the quality of the unsupervised local features on the widely used ShapeNet Part dataset [54]. This dataset is a widely used benchmark to evaluate 3D part segmentation performance, which consists of 16,681 objects from 16 categories. Each object is densely labeled by 2-6 part labels. In our experiments, we follow the previous work [26] to train a linear classifier on the unsupervised features. The feature extraction model is trained on the objects from the training set. We reported the standard evaluation metrics including mean IoU across all part classes and IoU for each category. The linear classifier is trained using Adam [34] optimizer with a base learning rate of 0.001 and 0 weight decay for 100 epochs. The learning rate is decayed by 0.975 every one epoch.

We show the results and compare them with previous methods in Table 8. We see our method significantly improves the previous methods on both mIoU ($+4.0\%$) and per-point classification accuracy ($+2.2\%$) metrics. Our method also achieves the best performance on the most categories (all 16 categories on mIoU and 14 out of 16 on per-point classification accuracy). These results indicate that our method also has clear advantages in learning discriminative local features without any supervision. We also notice that although our method largely improves the previous state-of-the-art, there is a large margin between unsupervised models and fully supervised models (e.g., PointNet++ [55] achieves 85.1% mIoU on this benchmark), which suggests that supervised models suffer less from overfitting on the dense prediction task and there is still substantial room to improve the unsupervised local features.

4.3 3D Object Few-Shot Classification

We extend our method to tackle the few-shot point cloud classification task. Few-shot classification requires the model to rapidly generalize to new tasks with only one or few annotated examples, which is a realistic and challenging scenario to analyze the generalization ability of the learned features. We design a 3D object few-shot classification benchmark based on ModelNet40 [72], where we randomly divide the 40 categories into 30 training categories and 10 test categories. The split is fixed in our experiments to better compare the different algorithms. There are 10,076 point clouds in the meta training set and 2,235 point clouds in the meta test set. Following the common practice in few-shot learning, we test our method and the baseline methods under K-way M-shot settings, where $K \in \{5, 10\}$ and $M \in \{1, 5, 10\}$. We select four prevalent few-shot learning methods as our baseline methods, including two metric-based methods RelationNet [62] and ProtoNet [58] and two optimization-based methods MAML [18] and MetaOptNet [36]. In our experiments, we use the same modified PointNet++ backbone model for a fair comparison. The final accuracy is computed as the average performance of 20 i.i.d. experiments, where 20 samples are randomly sampled from the unseen test set for each category (i.e., $20K$ samples are used to measure accuracy in each experiment). We also report the standard deviation of the 20 experiments as references.

The results of our methods and the four few-shot learning methods are listed in Table 9. We see although our method did not use the labels of the training set, the proposed method can outperform widely used few-shot learning algorithms by a large margin. Specifically, on the 5-way few-shot classification benchmark, our method improves the best baseline methods by 4.3%, 5.0%, and 4.0% in the $\{1, 5, 10\}$-shot settings respectively. On the 10-way few-shot classification benchmark, we beat the best baselines by 5.0%, 3.2%, and 4.8% in the $\{1, 5, 10\}$-shot settings respectively. From the benchmark, we also see the metric-based methods generally perform better in the 1-shot setting while optimization methods work better when the number of labeled samples increases. Our method consistently outperforms both kinds of methods in different settings. These

TABLE 8
Part Segmentation Results on ShapeNet Part Benchmark

	Method	Mean	Aero	Bag	Cap	Car	Chair	Ear	Guitar	Knife	Lamp	Laptop	Motor	Mug	Pistol	Rocket	Skate	Table
mIoU	LGAN [1]	57.0	54.1	48.6	62.6	43.2	68.3	58.3	74.3	68.4	53.4	82.6	18.6	75.1	54.7	37.2	46.7	66.4
	MAP-VAE [26]	67.9	62.7	67.0	72.9	58.4	77.0	67.3	84.8	77.0	60.8	90.8	35.8	87.7	64.2	44.9	60.3	74.7
	Ours	**71.9**	70.1	72.1	78.5	61.9	85.9	67.8	85.2	77.0	76.0	94.6	36.2	87.8	69.1	45.0	66.4	76.7
ACC	LGAN [1]	78.2	74.9	84.3	77.0	71.1	78.2	78.3	84.4	78.2	69.0	86.8	67.9	90.4	81.9	68.4	82.2	78.2
	MAP-VAE [26]	87.4	83.5	93.7	86.1	83.2	87.0	88.0	93.1	86.6	79.3	94.8	77.3	98.8	90.5	77.1	93.2	86.2
	Ours	**89.6**	85.4	93.9	90.6	85.3	92.0	90.0	93.9	87.3	86.6	97.6	77.9	98.7	92.1	78.2	92.3	91.3

We report the mIoU (%) for each object category and the per-point classification accuracy (%) of PointGLR and previous unsupervised methods. The supervised linear classifier is used to predict per-point semantic labels.

TABLE 9
Few-Shot Classification Results on ModelNet40

Type	Method	5-way Accuracy			10-way Accuracy		
		1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
Metric	RelationNet [62]	65.2 ± 9.0	80.2 ± 5.0	84.3 ± 4.8	54.7 ± 7.2	71.2 ± 2.8	74.2 ± 2.1
	ProtoNet [58]	66.8 ± 7.9	83.0 ± 4.9	87.2 ± 4.5	55.5 ± 6.9	74.1 ± 2.7	76.2 ± 1.9
Optimization	MAML [18]	62.9 ± 10.2	81.7 ± 7.4	86.9 ± 6.2	49.8 ± 8.2	72.4 ± 3.9	76.1 ± 2.5
	MetaOptNet [36]	66.4 ± 9.4	83.3 ± 6.0	87.9 ± 4.5	54.8 ± 7.1	74.6 ± 3.2	76.7 ± 1.8
Unsupervised	Ours	$\mathbf{71.1 \pm 9.2}$	$\mathbf{88.3 \pm 5.9}$	$\mathbf{91.9 \pm 4.4}$	$\mathbf{60.5 \pm 6.8}$	$\mathbf{77.8 \pm 2.9}$	$\mathbf{81.5 \pm 1.9}$

We report few-shot classification accuracy (%) of the four baseline models and our method. Our feature extraction model and linear SVM classifier are trained without annotations on the support set and trained on the training split of the query set in a supervised manner respectively. The baseline models are trained following the standard supervised training procedure.

results suggest learning better representations with unsupervised learning is a strong baseline for point cloud few-shot classification. Apart from better representations, we think our method also provides a practical tool to combine deep learning and conventional machine learning methods like SVM, which are usually more data-efficient compared to deep learning models. Since our combination of deep models and SVM classifier only requires training SVM on the query set, we also observed our method is much faster measured by training time when novel categories occur.

4.4 3D Scene Understanding

The proposed hierarchical global-local reasoning framework (PointHGLR) enables us to learn representations for 3D scenes in an unsupervised manner. To verify the effectiveness of the proposed method, we adopt the unsupervised pre-training and supervised fine-tuning pipeline on a widely used 3D scene understanding benchmark ScanNet [10]. ScanNet is a richly annotated dataset of 3D reconstructed meshes of indoor scenes. This dataset contains 1,513 scanned and reconstructed real scenes, where we mainly consider 18 different categories of objects with various sizes and shapes. Currently, it is the largest one that was created with a lightweight RGB-D scanning procedure. We split the whole dataset into two subsets with 1,201 and 312 scenes for training and testing following [10], [51].

3D Object Detection. We first conducted experiments on the 3D object detection task. We consider two different backbone models in this experiment: standard PointNet++ [51] and its wider version described in [33]. We show the effectiveness of our method by the improvement upon VoteNet [51] and H3DNet [33]. Both the models take as input 40,000 points. Following the network configurations in these two works, we extract the features from different abstraction levels to obtain $\{256, 512, 1024, 2048\}$-point features as the local features. We set M_{proxy} to 32 in all our experiments. We downsample the input point clouds to 1,024 points using furthest point sampling [55] to form P_{coarse}. For the downstream object detection task, we follow the basic experiment settings in [51] and [33]. We directly load the pre-trained models as the weight initialization and keep other training settings unchanged. We evaluate the performance by mAP with a 3D IoU threshold of 0.25 and 0.5. Please refer to the original papers for more details with regard to the experiment settings.

The main results are presented in Table 10, where we compare several state-of-the-art 3D detection methods and the recent unsupervised pre-training method PointContrast [74]. We see our method consistently and significantly improves the two baseline methods. We improve the VoteNet by 2.1% on both mAP_{25} and mAP_{50}. Benefiting from our unsupervised pre-training method, we improve the recent H3DNet by 1.2% and 3.1% on mAP_{25} and mAP_{50} respectively, which establishes the new state-of-the-art on this dataset. Notably, our method achieves better performance compared to the original PointContrast method that is based on Sparse Residual U-Net. To better compare with PointContrast, we also implement their method based on the VoteNet. It can be observed that PointContrast pre-training cannot produce a significantly better performance on VoteNet, which leads to 0.1% worse mAP_{25} and 0.5% better mAP_{50}. Note that PointContrast requires raw depth video frames during pre-training, which are not always available in 3D scene datasets.

We also conducted several analysis experiments on the effects of the proposed structural proxy. The results are shown in Table 11. We first test the original PointGLR (i.e., $M_{\text{proxy}} = 0$), we see that directly applying our method to the 3D scene can improve the baseline method by 0.9% and 0.8% on mAP_{25} and mAP_{50} respectively, which brings less improvement compared to the newly proposed hierarchical reasoning framework. We also tested the effects of M_{proxy} varying from 16 to 64. We found $M_{\text{proxy}} = 32$ will lead to the best performance.

TABLE 10
3D Object Detection Results on ScanNet Validation Set

	Input	mAP_{25}	mAP_{50}
DSS[60]	Geo + RGB	15.2	6.8
F-PointNet[53]	Geo + RGB	19.8	10.8
GSPN[77]	Geo + RGB	30.6	17.7
3D-SIS [33]	Geo + 5 views	40.2	22.5
PointContrast [74]	Geo only*	58.5	38.0
VoteNet [52]	Geo only	58.6	33.5
PointContrast [74] + VoteNet	Geo only*	58.5	34.0
Ours + VoteNet	Geo only	60.7	35.6
H3DNet [79]	Geo only	67.2	48.1
Ours + H3DNet	Geo only	**68.4**	**51.2**

*We show mean of average precision (mAP) with 3D IoU threshold 0.25 and 0.5. *indicates extra data are used during pre-training.*

TABLE 11
Effects of the Structural Proxy

	mAP_{25}	mAP_{50}
VoteNet [52]	58.6	33.5
PointGLR ($M_{\text{proxy}} = 0$)	59.5	34.8
PointHGLR ($M_{\text{proxy}} = 16$)	60.2	35.2
PointHGLR ($M_{\text{proxy}} = 32$)	**60.7**	**35.6**
PointHGLR ($M_{\text{proxy}} = 64$)	60.5	35.3

We analysis the effects of the number of proxies on ScanNet. We use the original VoteNet model as the baseline method and test the 3D object detection performance when different numbers of proxies are used.

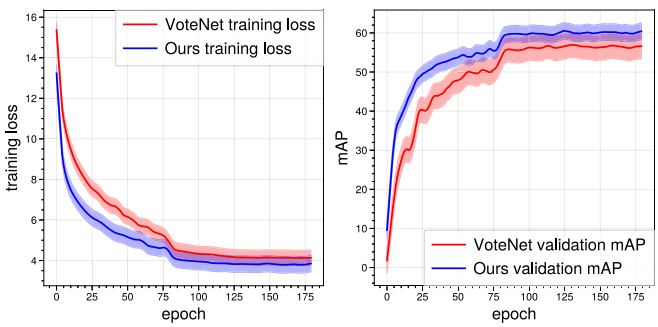

Fig. 6. *Effects of unsupervised pre-training.* We plot the 3D detection training loss and the validation mAP@0.25 of VoteNet with/without unsupervised pre-training on ScanNet.

To have an intuitive understanding of the effects of unsupervised pre-training, we show the learning curve of our method and the baseline VoteNet in Fig. 6. We observe that our pre-training weights significantly help improve the learning speed and stabilize the training process. The model with pre-training weights can achieve lower training loss and better validation mAP, which clearly demonstrates the effectiveness of the proposed method.

3D Scene Semantic Segmentation. We also extend our method to the 3D scene semantic segmentation task. We use the high-performance Sparse Residual U-Net (MinkowskiNet34) [8] as our baseline. Our experiments are based on sparse convolution library Minkowski Engine 0.5 [8]. We use the features with stride $\{1, 2, 4, 8\}$ as the local features. To enable a larger batch size, we use the 5cm version of MinkowskiNet34 as our base model. For the downstream semantic segmentation task and the PointContrast baseline, we follow the original settings in their papers and official implementation.

The results are shown in Table 12. We see our method can largely improve the base model by 2.3% and 1.3% in mIoU and mAcc respectively. We also see our method can outperform PointContrast without using any extra data, which clearly shows the effectiveness of our method. These results also demonstrate that our method works well on both PointNet++ style networks and sparse convolutional networks.

4.5 Visualization

In this section, we provide the visualization of our models. We analyze the unsupervised features by showing the feature distributions of different samples and the relation between local and global features.

TABLE 12
3D Semantic Segmentation Results on ScanNet Validation Set

	mIoU	mAcc
MinkowskiNet34 [8]	66.6	74.8
PointContrast [74] + MinkowskiNet34 *	68.2	75.9
Ours + MinkowskiNet34	**68.9**	**76.1**

*We report the mIOU and mAcc by taking average across all semantic classes. *indicates extra data are used during pre-training.*

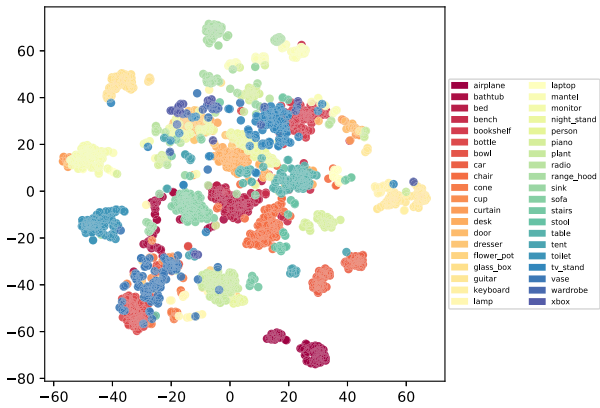

Fig. 7. Visualization of unsupervised representations on the test set of ModelNet40 using t-SNE. Best viewed in color.

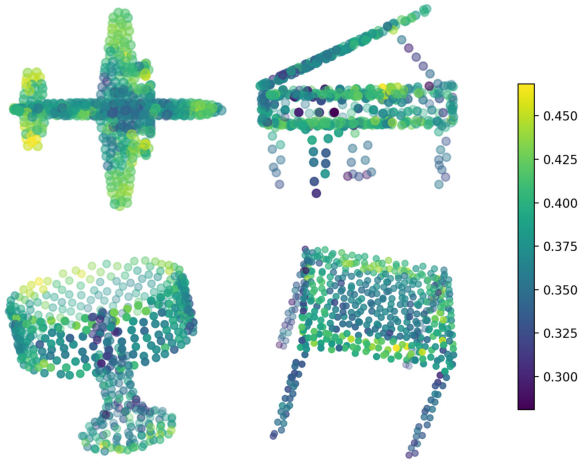

Fig. 8. Visualization of the similarity scores between the local features from the first abstraction level and the global feature on the test set of ModelNet40. Best viewed in color.

Feature Distribution. To have an intuitive understanding of our models, we visualized the unsupervised learn features on the test set of ModelNet40 in Fig. 7. The features are mapped to 2D space by applying t-SNE [46]. We see features from different categories are naturally separated without explicit supervision, which reflects the strong discriminative power of our representations.

Global-Local Relation. To show the effectiveness of our global-local reasoning method and understand the relation between the global feature and local parts, we visualize the similarity scores between the local features from the first abstraction level and the global feature on the test set of ModelNet40 in Fig. 8. In general, we see that the local features are close to the global feature (similarity >0.25) and the more distinguishable regions usually have higher similarity scores.

5 Conclusion

We have proposed a new scheme for unsupervised representation learning of 3D point clouds by bidirectional global-local reasoning and hierarchical global-local reasoning. Comprehensive experimental studies have demonstrated our unsupervised representations can surpass their supervised counterparts and achieve state-of-the-art performance on several widely used 3D object classification benchmarks. We have also shown the effectiveness of our method on various object-level and scene-level 3D understanding tasks including 3D object part segmentation, point cloud few-shot classification, 3D scene object detection, and semantic segmentation. We expect our method to open a new door for learning better point cloud representations from data structures instead of human annotations.

References

[1] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 40–49.

[2] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," 2018, *arXiv:1803.10091*.

[3] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," 2019, *arXiv:1906.00910*.

[4] Y. Ben-Shabat, M. Lindenbaum, and A. Fischer, "3DmFV: Three-dimensional point cloud classification in real-time using convolutional neural networks," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3145–3152, Oct. 2018.

[5] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[6] M. Caron *et al.*, "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[8] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal ConvNets: Minkowski convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3075–3084.

[9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[10] 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/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5828–5839.

[11] H. Deng, T. Birdal, and S. Ilic, "PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 602–618.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[13] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4690–4699.

[14] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2015, pp. 1422–1430.

[15] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2017, pp. 2051–2060.

[16] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2472–2481.

[17] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3D object reconstruction from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 605–613.

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[19] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Proc. Competition Cooperation Neural Nets*, 1982, pp. 267–285.

[20] M. Gadelha, R. Wang, and S. Maji, "Multiresolution tree networks for 3D point cloud processing," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 103–118.

[21] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a predictable and generative vector representation for objects," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 484–499.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[23] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[24] J.-B. Grill *et al.*, "Bootstrap your own latent–A new approach to self-supervised learning," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21271–21284.

[25] Z. Han, M. Shang, Y.-S. Liu, and M. Zwicker, "View inter-prediction GAN: Unsupervised representation learning for 3D shapes by learning global shape memories to support local view predictions," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 8376–8384.

[26] Z. Han, X. Wang, Y.-S. Liu, and M. Zwicker, "Multi-angle point cloud-VAE: Unsupervised feature learning for 3D point clouds from multiple angles by joint self-reconstruction and half-to-half prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10441–10450.

[27] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8159–8170.

[28] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[29] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord, "Data-efficient image recognition with contrastive predictive coding," 2019, *arXiv:1905.09272*.

[30] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[31] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[32] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–24.

[33] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4416–4425.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[35] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2013, Art. no. 2.

[36] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 657–10 665.

[37] C.-L. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov, "Point cloud GAN," 2018, *arXiv:1810.05795*.

[38] J. Li, B. M. Chen, and G. Hee Lee, "SO-net: Self-organizing network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9397–9406.

[39] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 828–838.

[40] S. Liu, L. Giles, and A. Ororbia, "Learning a hierarchical latent-variable model of 3D shapes," in *Proc. Int. Conf. 3D Vis.*, 2018, pp. 542–551.

[41] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 212–220.

[42] X. Liu, Z. Han, X. Wen, Y.-S. Liu, and M. Zwicker, "L2G auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 989–997.
[43] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan, "DensePoint: Learning densely contextual representation for efficient point cloud processing," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5239–5248.
[44] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8895–8904.
[45] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[46] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.
[47] A. Mnih and K. Kavukcuoglu, "Learning word embeddings efficiently with noise-contrastive estimation," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2265–2273.
[48] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," 2016, *arXiv:1601.06759*.
[49] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[50] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 2227–2237.
[51] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9277–9286.
[52] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3D object detection in point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9277–9286.
[53] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 918–927.
[54] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.
[55] 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. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
[56] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5376–5385.
[57] A. Sharma, O. Grau, and M. Fritz, "VConv-DAE: Deep volumetric shape learning without object labels," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 236–250.
[58] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.
[59] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.
[60] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 808–816.
[61] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[62] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1199–1208.
[63] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6410–6419.
[64] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2019, *arXiv:1906.05849*.
[65] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and L. Mario, "On mutual information maximization for representation learning," 2019, *arXiv:1907.13625*.
[66] M. A. Uy, Q.-H. Pham, B.-S. Hua, T. Nguyen, and S.-K. Yeung, "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1588–1597.
[67] D. Valsesia, G. Fracastoro, and E. Magli, "Learning localized generative models for 3D point clouds via graph convolution," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–15.
[68] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5265–5274.
[69] 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, 2019, Art. no. 146.
[70] 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 *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 82–90.
[71] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9621–9630.
[72] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.
[73] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3733–3742.
[74] S. Xie, J. Gu, D. Guo, C. R. Qi, L. J. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3D point cloud understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 574–591.
[75] 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.*, 2018, pp. 90–105.
[76] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point cloud auto-encoder via deep grid deformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 206–215.
[77] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "GSPN: Generative shape proposal network for 3D instance segmentation in point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3942–3951.
[78] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning local geometric descriptors from RGB-D reconstructions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 199–208.
[79] Z. Zhang, B. Sun, H. Yang, and Q. Huang, "H3DNet: 3D object detection using hybrid geometric primitives," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 311–329.
[80] Y. Zhao, T. Birdal, H. Deng, and F. Tombari, "3D point capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1009–1018.

Yongming Rao (Student Member, IEEE) received the BEng degree from the Department of Electronic Engineering, Tsinghua University, China, in 2018. Currently, he is a fourth-year PhD student with the Department of Automation, Tsinghua University, China, advised by Prof. Jiwen Lu. His research interests include computer vision and deep learning. He has authored more than 20 conference papers in CVPR/ICCV/ECCV/NeurIPS and four journal papers in *IEEE Transactions on Pattern Analysis and Machine Intelligence/International Journal of Computer Vision*. He serves as a reviewer for several international conferences and journals, where he was recognized as the outstanding reviewer of ECCV 2020, CVPR 2021 and ICME 2019. He is a recipient of the CCF-CV Academic Emerging Award, in 2019.

Jiwen Lu (Senior Member, IEEE) received the BEng degree in mechanical engineering and the MEng degree in electrical engineering from the Xian University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the PhD degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an associate professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition. He was/is a member of the Image, Video and Multidimensional Signal Processing Technical Committee, Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, a member of the Multimedia Systems and Applications Technical Committee and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society, respectively. He serves as the general co-chair for the International Conference on Multimedia and Expo (ICME) 2022, the program co-chair for the International Conference on Multimedia and Expo 2020, the International Conference on Automatic Face and Gesture Recognition (FG) 2023, and the International Conference on Visual Communication and Image Processing (VCIP) 2022. He serves as the co-editor-of-chief for *Pattern Recognition Letters*, an associate editor for *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Biometrics, Behavior, and Identity Sciences*, and *Pattern Recognition*. He was a recipient of the National Natural Science Funds for Distinguished Young Scholar. He is a fellow of IAPR.

Jie Zhou (Senior Member, IEEE) received the BS and MS degrees both from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From then to 1997, he served as a postdoctoral fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a full professor with the Department of Automation, Tsinghua University, China. His research interests include computer vision, pattern recognition, and image processing. In recent years, he has authored more than 300 papers in peer-reviewed journals and conferences. Among them, more than 100 papers have been published in top journals and conferences such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, and CVPR. He is an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence* and two other journals. He received the National Outstanding Youth Foundation of China Award. He is an IAPR fellow.