PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding

Saining Xie[1(✉)], Jiatao Gu[1], Demi Guo[1], Charles R. Qi[1], Leonidas Guibas[2], and Or Litany[2]

[1] Facebook AI Research, Menlo Park, USA
xiesaining@gmail.com
[2] Stanford University, Stanford, USA

Abstract. Arguably one of the top success stories of deep learning is transfer learning. The finding that pre-training a network on a rich source set (*e.g.*, ImageNet) can help boost performance once fine-tuned on a usually much smaller target set, has been instrumental to many applications in language and vision. Yet, very little is known about its usefulness in 3D point cloud understanding. We see this as an opportunity considering the effort required for annotating data in 3D. In this work, we aim at facilitating research on 3D representation learning. Different from previous works, we focus on high-level scene understanding tasks. To this end, we select a suit of diverse datasets and tasks to measure the effect of unsupervised pre-training on a large source set of 3D scenes. Our findings are extremely encouraging: using a unified triplet of architecture, source dataset, and contrastive loss for pre-training, we achieve improvement over recent best results in segmentation and detection across 6 different benchmarks for indoor and outdoor, real and synthetic datasets – demonstrating that the learned representation can generalize across domains. Furthermore, the improvement was similar to supervised pre-training, suggesting that future efforts should favor scaling data collection over more detailed annotation. We hope these findings will encourage more research on unsupervised pretext task design for 3D deep learning.

Keywords: Unsupervised learning · Point cloud recognition · Representation learning · 3D scene understanding

1 Introduction

Representation learning is one of the main driving forces of deep learning research. In 2D vision, the finding that pre-training a network on a rich source

D. Guo, C. R. Qi, L. Guibas and O. Litany—Work done while at Facebook AI Research.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58580-8_34) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12348, pp. 574–591, 2020.
https://doi.org/10.1007/978-3-030-58580-8_34

set (*e.g.* ImageNet classification) can help boost performance once fine-tuned on the usually much smaller target set, has been key to the success of many applications. A particularly important setting, is when the pre-training stage is unsupervised, as this opens up the possibility to utilize a practically infinite train set size. Unsupervised pre-training has been remarkably successful in natural language processing [13,47], and has recently attracted increasing attention in 2D vision [3,3,8,8,26,26,27,38,40,40,64,81].

In the past few years, the field of 3D deep learning has witnessed much progress with an ever-increasing number of 3D representation learning schemes [1,9,12,15,16,21,22,34,62,69,75]. However, it still falls behind compared to its 2D counterpart as evidently, in all 3D scene understanding tasks, ad-hoc training *from scratch* on the target data is still the dominant approach. Notably, all existing representation learning schemes are tested either on single objects or low-level tasks (e.g. registration). This status quo can be attributed to multiple reasons: 1) Lack of large-scale and high-quality data: compared to 2D images, 3D data is harder to collect, more expensive to label, and the variety of sensing devices may introduce drastic domain gaps; 2) Lack of unified backbone architectures: in contrast to 2D vision where architectures such as ResNets were proven successful as backbone networks for pre-training and fine-tuning, point cloud network architecture designs are still evolving; 3) Lack of a comprehensive set of datasets and high-level tasks for evaluation.

The purpose of this work is to move the needle by initiating research on *unsupervised pre-training* with *supervised fine-tuning* in deep learning for 3D scene understanding. To do so, we cover four important ingredients: 1) Selecting a large dataset to be used at pre-training; 2) identifying a backbone architecture that can be shared across many different tasks; 3) evaluating two unsupervised objectives for pre-training the backbone network; and 4) defining an evaluation protocol on a set of diverse downstream datasets and tasks.

Specifically, we choose ScanNet [11] as our source set on which the pre-training takes place, and utilize a sparse residual U-Net [9,49] as the backbone architecture in all our experiments and focus on the point cloud representation of 3D data. For the pre-training objective, we evaluate two different contrastive losses: Hardest-contrastive loss [10], and PointInfoNCE – an extension of InfoNCE loss [40] used for pre-training in 2D vision. Next, we choose a broad set of target datasets and downstream tasks that includes: semantic segmentation on S3DIS [2], ScanNetV2 [11], ShapeNetPart [71] and Synthia 4D [50]; and object detection on SUN RGB-D [31,53,55,65] and ScanNetV2. Remarkably, our results indicate improved performance across all datasets and tasks (See Table 1 for a summary of the results). In addition, we found a relatively small advantage to pre-training with supervision. This implies that future efforts in collecting data for pre-training should favor scale over precise annotations.

Our contributions can be summarized as follows:

– We evaluate, for the first time, the transferability of learned representation in 3D point clouds to high-level scene understanding.

- Our results indicate that *unsupervised pre-training* improves performance across downstream tasks and datasets, while using a single unified architecture, source set and objective function.
- Powered by unsupervised pre-training, we achieve a new state-of-the-art performance on 6 different benchmarks.
- We believe these findings would encourage a change of paradigm on how we tackle 3D recognition and drive more research on 3D representation learning.

2 Related Work

Representation Learning in 3D. Deep neural networks are notoriously data hungry. This renders the ability to transfer learned representations between datasets and tasks extremely powerful. In 2D vision it has led to a surge of interest in finding optimal pretext unsupervised tasks [3,5,8,10,14,18,26,27,38–41,64,77,78,81]. We note that while many of these tasks are *low-level* (*e.g.* pixel or patch level reconstruction), they are evaluated based on their transferability to *high-level* tasks such as object detection. Being much harder to annotate, 3D tasks are potentially the biggest beneficiaries of unsupervised- and transfer-learning. This was shown in several works on single object tasks like reconstruction, classification and part segmentation [1,16,21,22,34,51,62,69]. Yet, generally much less attention has been devoted to representation learning in 3D that extends beyond the single-object level. Further, in the few cases that did study it, the focus was on low-level tasks like registration [12,15,75]. In contrast, here we wish to push forward research in 3D representation learning by focusing on transferability to more high-level tasks on more complex scenes.

Deep Architectures for Point Cloud Processing. In this work we focus on learning useful representation for point cloud data. Inspired by the success in 2D domain, we conjecture that an important ingredient in enabling such progress is the evident standardization of neural architectures. Canonical examples include VGGNet [54] and ResNet/ResNeXt [25,66]. In contrast, point cloud neural network design is much less mature, as is apparent by the abundance of new architectures that have been recently proposed. This has multiple reasons. First, is the challenge of processing unordered sets [37,45,48,74]. Second, is the choice of neighborhood aggregation mechanism which could either be hierarchical [16,32,33,46,76], spatial CNN-like [29,35,57,68,79], spectral [58,60,72] or graph-based [52,59,63,67]. Finally, since the points are discrete samples of an underlying surface, continuous convolutions have also been considered [4,61,70]. Recently Choy et al. proposed the Minkowski Engine [9], an extension of submanifold sparse convolutional networks [20] to higher dimensions. In particular, sparse convolutional networks facilitate the adoption of common deep architectures from 2D vision, which in turn can help standardize deep learning for point cloud. In this work, we use a unified UNet [49] architecture built with Minkowski Engine as the backbone network in all experiments and show it can gracefully transfer between tasks and datasets.

3 PointContrast Pre-training

In this section, we introduce our unsupervised pre-training pipeline. First, to motivate the necessity of a new pre-training scheme, we conduct a pilot study to understand the limitations of existing practice (pre-training on ShapeNet) in 3D deep learning (Sect. 3.1). After briefly reviewing an inspirational local feature learning work *Fully Convolutional Geometric Features (FCGF)* (Sect. 3.2), we introduce our unsupervised pre-training solution, *PointContrast*, in terms of pretext task (Sect. 3.3), loss function (Sect. 3.4), network architecture (Sect. 3.5) and pre-training dataset (Sect. 3.6).

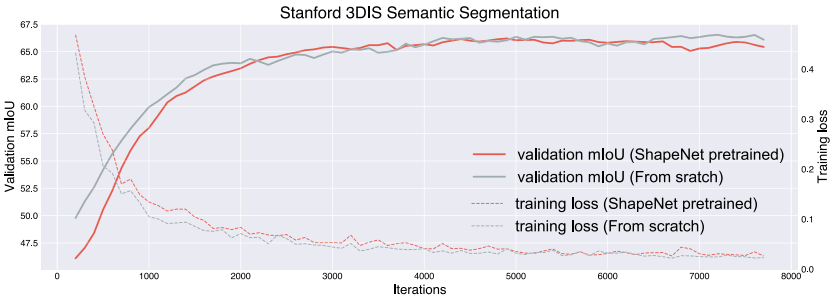

Fig. 1. Training from scratch *vs.* fine-tuning with ShapeNet pretrained weights.

3.1 Pilot Study: Is Pre-training on ShapeNet Useful?

Previous works on unsupervised 3D representation learning [1,16,21,22,34,62,69] mainly focused on ShapeNet [7], a dataset of single-object CAD models. One underlying assumption is that by adopting ShapeNet as the ImageNet counterpart in 3D, features learned on *synthetic single objects* could transfer to other real-world applications. Here we take a step back and reassess this assumption by studying a straightforward supervised pre-training setup: we simply pre-train an encoder network on ShapeNet with *full supervision*, and fine-tune it with a U-Net on a downstream task (S3DIS semantic segmentation). Based on results in 2D representation learning, we use full supervision here as an upper bound to what could be gained from pre-training. We train a sparse ResNet-34 model (details to follow in Sect. 3.5) for 200 epochs. The model achieves a high validation accuracy of 85.4% on ShapeNet classification task. In Fig. 1, we show the downstream task training curves for (a) training from scratch and (b) fine-tuning with ShapeNet pretrained weights. Critically, one can observe that ShapeNet pre-training, even in the supervised fashion, *hampers* downstream task learning. Among many potential explanations, we highlight two major concerns:

– **Domain gap between source and target data:** Objects in ShapeNet are synthetic, normalized in scale, aligned in pose, and lack scene context. This makes pre-training and fine-tuning data distributions drastically different.

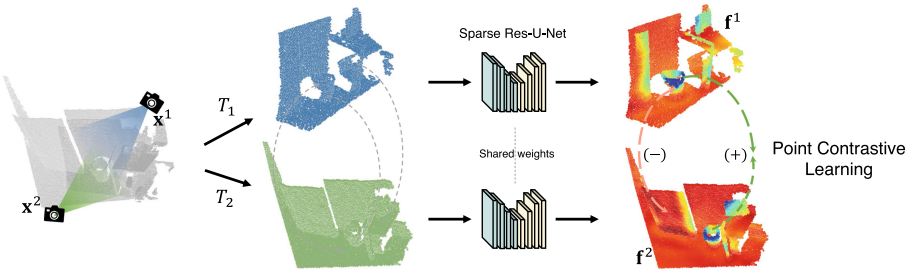

Fig. 2. PointContrast: pretext task for 3D pre-training.

Table 1. Summary of downstream fine-tuning tasks. Compared to the baseline learning paradigm of training from scratch, which is dominant in 3D deep learning, our unsupervised pre-training method PointContrast boosts the performance across the board when finetuning on a diverse set of high-level 3D understanding tasks. $*$ indicates results trained using only 1% of the training data.

PointContrast: downstream tasks for fine-tuning					
Datasets	Real/Synth.	Complexity	Env.	Task	Rel.gain
S3DIS	Real	Entire floor, office	Indoor	Segmentation	(+2.7%) mIoU
SUN RGB-D	Real	Medium-sized cluttered rooms	Indoor	Detection	(+3.1%) mAP0.5
ScanNetV2	Real	Large rooms	Indoor	Segmentation Detection	(+1.9%) mIoU (+2.6%) mAP0.5
ShapeNet	Synth.	Single objects	Indoor & outdoor	Classification	(+4.0%) Acc.*
ShapeNetPart	Synth.	Object parts	Indoor & outdoor	Segmentation	(+2.2%) mIoU*
Synthia 4D	Synth.	Street scenes, driving envs.	Outdoor	Segmentation	(+3.3%) mIoU

– **Point-level representation matters:** In 3D deep learning, the local geometric features, *e.g.* those encoded by a point and its neighbors, have proven to be discriminative and critical for 3D tasks [45,46]. Directly training on *object instances* to obtain a global representation might be insufficient.

This led us to rethink the problem: if the goal of pre-training is to boost performance across many real world tasks, exploring pre-training strategies on single objects might offer limited potential. (1) To address the domain gap concern, it might be beneficial to directly pre-train the network on complex scenes with multiple objects, to better match the target distributions; (2) to capture point-level information, we need to design a pretext task and corresponding network architecture that is not only based on instance-level/global representations, but instead can capture dense/local features at the point level.

3.2 Revisiting Fully Convolutional Geometric Features (FCGF)

Here we revisit a previous approach FCGF [10] designed to learn geometric features for *low-level* tasks (*e.g.* registration) as our work is mainly inspired

by FCGF. FCGF is a deep learning based algorithm that learns local feature descriptors on correspondence datasets via metric learning. FCGF has two major ingredients that help it stand out and achieve impressive results in registration recall: (1) **a fully-convolutional design** and (2) **point-level metric learning**. With a fully-convolutional network (FCN) [36] design, FCGF operates on the entire input point cloud (*e.g.* full indoor or outdoor scenes) without having to crop the scene into patches as done in previous works; this way the local descriptors can aggregate information from a large number of neighboring points (up to the extent of receptive field size). As a result, point-level metric learning becomes natural. FCGF uses a U-Net architecture that has full-resolution output (*i.e.* for N points, the network outputs N associated feature vectors), and positive/negative pairs for metric learning are defined at the point level.

Despite having a fundamentally different goal in mind, FCGF offers inspirations that might address the pretext task design challenges: A fully-convolutional design will allow us to pre-train on the target data distributions that involve complex scenes with a large number of points, and we could define the pretext task directly on points. Under this perspective, we pose the question: *Can we repurpose FCGF as the pretext task for high-level 3D understanding?*

Algorithm 1 General Framework of PointContrast

Input: Backbone architecture NN; Dataset $X = \{\mathbf{x}_i \in \mathbb{R}^{N\times 3}\}$; Point feature dimension D;
Output: Pre-trained weights for NN.
for *each point cloud* $\mathbf{x}$ *in* X **do**
- From $\mathbf{x}$, generate two views $\mathbf{x}^1$ and $\mathbf{x}^2$.
- Compute correspondence mapping (matches) M between points in $\mathbf{x}^1$ and $\mathbf{x}^2$.
- Sample two transformations $\mathbf{T}_1$ and $\mathbf{T}_2$.
- Compute point features $\mathbf{f}^1, \mathbf{f}^2 \in \mathbb{R}^{N\times D}$ by $\mathbf{f}^1 = \mathrm{NN}(\mathbf{T}_1(\mathbf{x}^1))$ and $\mathbf{f}^2 = \mathrm{NN}(\mathbf{T}_2(\mathbf{x}^2))$.
- Backprop. to update NN with contrastive loss $\mathcal{L}_c(\mathbf{f}^1, \mathbf{f}^2)$ on the matched points.

3.3 PointContrast as a Pretext Task

FCGF focuses on local descriptor learning for low-level tasks only. In contrast, a good pretext task for pre-training aims to learn *network weights* that are universally applicable and useful to many high-level 3D understanding tasks. To take the inspiration of FCGF and create such pretext tasks, there are several design choices that need to be revisited. In terms of *architecture*, since inference speed is a major concern in registration tasks, the network used in FCGF is very light-weight; Contrarily, the success of pre-training relies on over-parameterized networks, as clearly evidenced in other domains [8,13]. In terms of *dataset*, FCGF uses domain-specific registration datasets such as 3DMatch [75] and KITTI odometry [17], which lack both scale and generality. Finally, in terms of *loss design*, contrastive losses explored in FCGF are tailored for registration and it is interesting to explore other alternatives.

In Algorithm 1, we summarize the overall pretext task framework explored in this work. We name the framework *PointContrast*, since the high-level strategy of this pretext task is, contrasting—at the point level—between two transformed point clouds. Conceptually, given a point cloud $\mathbf{x}$ sampled from a certain distri-

bution, we first generate two views $\mathbf{x}^1$ and $\mathbf{x}^2$ that are aligned in the same world coordinates. We then compute the correspondence mapping M between these two views. If $(i, j) \in M$ then point $\mathbf{x}_i^1$ and point $\mathbf{x}_j^2$ are a pair of matched points across two views. We then sample two random geometric transformations T_1 and T_2 to further transform the point clouds in two views. The transformation is what could make the pretext task challenging as the network needs to learn certain *equivariance* with respect to the geometric transformation imposed. In this work, we mainly consider rigid transformation including rotation, translation and scaling. Further details are provided in Appendix. Finally, a contrastive loss is defined over points in two views: we minimize the distance for matched points and maximize the distance of unmatched points. This framework, though coming from a very different motivation (metric learning for geometric local descriptors), shares a strikingly similar pipeline with recent contrastive-based methods for 2D unsupervised visual representation learning [8,23,64]. The key difference is that most work for 2D focuses on contrasting between instances/images, while in our work the contrastive learning is done densely at the point level.

3.4 Contrastive Learning Loss Design

Hardest-Contrastive Loss. The first loss function, hardest-contrastive loss we try, is borrowed from the best-performing loss design proposed in FCGF [10], which adopts a hard negative mining scheme in traditional margin-based contrastive learning formulation,

$$\mathcal{L}_c = \sum_{(i,j)\in\mathcal{P}} \left\{ [d(\mathbf{f}_i, \mathbf{f}_j) - m_p]_+^2 / |\mathcal{P}| + 0.5 [m_n - \min_{k\in\mathcal{N}} d(\mathbf{f}_i, \mathbf{f}_k)]_+^2 / |\mathcal{N}_i| + 0.5 [m_n - \min_{k\in\mathcal{N}} d(\mathbf{f}_j, \mathbf{f}_k)]_+^2 / |\mathcal{N}_j| \right\}$$

Here $\mathcal{P}$ is a set of matched (positive) pairs of points $\mathbf{x}_i^1$ and $\mathbf{x}_j^2$ from two views $\mathbf{x}^1$ and $\mathbf{x}^2$, and $\mathbf{f}_i^1$ and $\mathbf{f}_j^2$ are associated point features for the matched pair. $\mathcal{N}$ is a randomly sampled set of non-matched (negative) points which is used for the hardest negative mining, where the hardest sample is defined as the closest point in the $\mathcal{L}_2$ normalized feature space to a positive pair. $[x]_+$ denotes function $\max(0, x)$. $m_p = 0.1$ and $m_n = 1.4$ are margins for positive and negative pairs.

PointInfoNCE Loss. Here we propose an alternative loss design for PointContrast. InfoNCE proposed in [40] is widely used in recent unsupervised representation learning approaches for 2D visual understanding. By modeling the contrastive learning framework as a dictionary look-up process [23], InfoNCE poses contrastive learning as a classification problem and is implemented with a Softmax loss. Specifically, the loss encourages a query q to be similar to its positive key k^+ and dissimilar to, typically many, negative keys k^-. One challenge in 2D is to scale the number of negative keys [23].

However, in the domain of 3D, we have a different problem: usually the real-world 3D datasets are much smaller in terms of instance count, but the number of points for each instance (*e.g.* a indoor or outdoor scene) can be huge, *i.e.* 100K+ points even from one RGB-D frame. This unique property of 3D data property, together with the original motivation to modelling point level information, inspire us to propose the following PointInfoNCE loss:

$$\mathcal{L}_c = -\sum_{(i,j)\in\mathcal{P}} \log \frac{\exp(\mathbf{f}_i \cdot \mathbf{f}_j/\tau)}{\sum_{(\cdot,k)\in\mathcal{P}} \exp(\mathbf{f}_i \cdot \mathbf{f}_k/\tau)}$$

Here $\mathcal{P}$ is the set of all the positive matches from two views. In this formulation, we only consider points that have at least one match and do not use additional non-matched points as negatives. For a matched pair $(i, j) \in \mathcal{P}$, point feature $\mathbf{f}_i^1$ will serve as the query and $\mathbf{f}_j^2$ will serve as the positive key k^+. We use point feature $\mathbf{f}_k^2$ where $(\cdot, k) \in \mathcal{P}$ and $k \neq j$ as the set of negative keys. In practice, we sample a subset of 4096 matched pairs from $\mathcal{P}$ for faster training.

Compared to hardest-contrastive loss, the PointInfoNCE loss has a simpler formulation with less hyperparatmers. Perhaps more importantly, due to the large number of negative distractors, it is more robust against *mode collapsing* (features collapsed to a single vector) than the hardest-contrastive loss. In our experiments, we find that hard-contrastive loss is unstable and hard to train: the representation often collapses with extended training epochs (which is also observed in FCGF [10]).

3.5 A Sparse Residual UNet as Shared Backbone

We use a Sparse Residual UNet (SR-UNet) architecture in this work. It is a 34-layer UNet [49] architecture that has an encoder network of 21 convolution layers and a decoder network of 13 convolution/deconvolution layers. It follows the 2D ResNet basic block design and each conv/deconv layer in the network are followed by Batch Normalization (BN) [30] and ReLU activation. The overall UNet architecture has 37.85M parameters. We provide more information and a visualization of the network in Appendix. The SR-UNet architecture was originally designed in [9] that achieved significant improvement over prior methods on the challenging ScanNet semantic segmentation benchmark. In this work we explore if we can use this architecture as a unified design for both the pre-training task and a diverse set of fine-tuning tasks.

3.6 Dataset for Pre-training

For local geometric feature learning approaches, including FCGF [10], training and evaluation are typically conducted on domain and task specific datasets such as KITTI odometry [17] or 3DMatch [75]. Common registration datasets are typically constrained in either scale (training samples collected from just dozens of scenes), or generality (focusing on one specific application scenario, *e.g.* indoor

scenes or LiDAR scans for self-driving car), or both. To facilitate future research on 3D unsupervised representation learning, in our work we utilize the ScanNet dataset for pre-training, aiming to address the scale issue. ScanNet is a collection of ∼1500 indoor scenes. Created with a light-weight RGB-D scanning procedure, ScanNet is currently the largest of its kind.[1]

Here we create a point cloud pair dataset on top of ScanNet for the pretraining framework shown in Fig. 2. Given a scene $\mathbf{x}$, we extract pairs of partial scans $\mathbf{x}^1$ and $\mathbf{x}^2$ from different views. More precisely, for each scene, we first sub-sample RGB-D scans from the raw ScanNet videos every 25 frames, and align the 3D point clouds in the same world coordinates (by utilizing estimated camera poses for each frame). Then we collect point cloud pairs from the sampled frames and require that two point clouds in a pair have at least 30% overlap. We sample a total number of 870K point cloud pairs. Since the partial views are aligned in ScanNet scenes, it is straight-forward to compute the correspondence mapping M between two views with nearest neighbor search.

Although ScanNet only captures indoor data distributions, as we will see in Section 4.4, surprisingly it can generalize to other target distributions. We provide additional visualizations for the pre-training dataset in Appendix.

4 Fine-Tuning on Downstream Tasks

The most important motivation for representation learning is to learn features that can transfer well to different downstream tasks. There could be different evaluation protocols to measure the usefulness of the learned representation. For example, probing with a linear classifier [19], or evaluating in a semi-supervised setup [26]. The *supervised fine-tuning* strategy, where the pre-trained weights are used as the initialization and are further refined on the target downstream task, is arguably the most practically meaningful way of evaluating feature transferability. With this setup, good features could directly lead to performance gains in downstream tasks.

Under this perspective, in this section we perform extensive evaluations of the effectiveness of PointContrast framework by fine-tuning the pre-trained weights on multiple downstream tasks and datasets. We aim to cover a diverse suit of high-level 3D understanding tasks of different natures such as semantic segmentation, object detection and classification. In all experiments we use the same backbone network, pre-trained on the proposed ScanNet pair dataset (Sect. 3.6) using both PointInfoNCE and Hardest-Constrastive objectives.

4.1 ShapeNet: Classification and Part Segmentation

Setup. In Sect. 3.1 we have observed that weights learned on supervised ShapeNet classification are not able to transfer well to scene-level tasks. Here we explore the opposite direction: Are PointContrast features learned on ScanNet

[1] Admittedly, ScanNet is still much smaller in scale compared to 2D datasets.

Table 2. ShapeNet classification. Top: classification accuracy with limited labeled training data for finetuning. Bottom: classification accuracy on the least represented classes in the data (tail-classes). In all cases, PointContrast boosts performance. Relative improvement increases with scarcer training data and on less frequent classes.

Evaluating on all 55 classes	1% data	10% data	100% data
Trained from scratch	62.2	77.9	85.1
PointContrast (Hardest-Contrastive)	**66.2** (+4.0)	**79.0** (+1.1)	**85.7** (+0.6)
PointContrast (PointInfoNCE)	**65.8** (+3.6)	**78.8** (+0.9)	**85.7** (+0.6)
Using 100% training data	10 tail classes	30 tail classes	All 55 classes
Train from scratch	65.0	70.9	85.1
PointContrast (Hardest-Contrastive)	**70.9** (+5.9)	**72.9** (+2.0)	**85.7** (+0.6)
PointContrast (PointInfoNCE)	**67.8** (+2.8)	**72.0** (+1.1)	**85.7** (+0.6)

Table 3. ShapeNet part segmentation. Replacing the backbone architecture with SR-UNet already boosts performance. PointContrast pre-training further adds a significant gain, and outshines where labels are most limited.

Methods	IoU (1% data)	IoU (5% data)	IoU (100% data)
SO-Net [34]	64.0	69.0	–
PointCapsNet [80]	67.0	70.0	–
Multitask Unsupervised [22]	68.2	77.7	–
Train from scratch	71.8	79.3	84.7
PointContrast (Hardest-Contrastive)	**74.0** (+2.2)	**79.9** (+0.6)	**85.1** (+0.4)
PointContrast (PointInfoNCE)	**73.1** (+1.3)	**79.9** (+0.6)	**85.1** (+0.4)

useful for tasks on ShapeNet? To recap, ShapeNet [7] is a dataset of synthetic 3D objects of 55 common categories. It was curated by collecting CAD models from online open-sourced 3D repositories. In [71], part annotations were added to a subset of ShapeNet models segmenting them into 2–5 parts. In order to provide a comparison with existing approaches, here we utilize the ShapeNetCore dataset (SHREC 15 split) for classification, and the ShapeNet part dataset for part segmentation, respectively. We uniformly sample point clouds of 1024 points from each model for classification and 2048 points for part segmentation. Albeit containing overlapping indoor object categories with ScanNet, this dataset is substantially different as it is synthetic, and contains only single objects. We also follow recent works on 3D unsupervised representation learning [22] to explore a more challenging setup: using a very small percentage (*e.g.* 1%–10%) of training data to fine-tune the pre-trained model.

Results. As shown in Table 2 and Table 3, for both datasets, the effectiveness of pre-training are correlated with the availability of training data. In the ShapeNet classification task (Table 2), pre-training helps most where less training data is available, achieving a 4.0% improvement over the training-from-scratch baseline with the hardest-negative objective. We also note that ShapeNet is a class-imbalanced dataset and the minority (tail) classes are very infrequent. When

using 100% of the training data, pre-training provides a class-balancing effect, as it boosts performance more on underrepresented (tail) classes. Table 3 shows a similar effects of pre-training on part segmentation performance. Notably, using SR-UNet backbone architecture already boosts performance; yet, pre-training is able to provide further gains, especially when training data is scarce.

4.2 S3DIS Segmentation

Setup. Stanford Large-Scale 3D Indoor Spaces (S3DIS) [2] dataset comprises 3D scans of 6 large-scale indoor areas collected from 3 office buildings. The scans are represented as point clouds and annotated with semantic labels of 13 object categories. Among the datasets used here for evaluation S3DIS is probably the most similar to ScanNet. Transferring features to S3DIS represents a typical scenario for fine-tuning: the downstream task dataset is similar yet much smaller than the pre-training dataset. For the commonly used benchmark split ("Area 5 test"), there are only about 240 samples in the training set. We follow [9] for pre-processing, and use standard data augmentations. See Appendix for details.

Results. Results are summarized in Table 4. Again, merely switching the SR-UNet architecture, training from scratch already improves upon prior art. Yet, fine-tuning the features learned by PointContrast achieves markedly better segmentation results in mIoU and mAcc. Notably, the effect persists across both loss types, achieving a 2.7% mIoU gain using Hardest-Contrastive loss and an on-par improvement of 2.1% mIoU for the PointInfoNCE variant.

Table 4. Stanford Area 5 Test (Fold 1) (S3DIS). Replacing the backbone network with SR-UNet improves upon prior art. Using PointContrast adds further significant boost with a mild preference for Hardest-contrastive over the PointInfoNCE objective. See Appendix for more methods in comparison.

Methods	mIoU	mAcc
PointNet [45]	41.1	49.0
PointCNN [35]	57.3	63.9
MinkowskiNet32 [9]	65.4	71.7
Train from scratch	68.2	75.5
PointConstrast (Hardest-Contrastive)	**70.9**	**77.0**
PointConstrast (PointInfoNCE)	70.3	76.9

4.3 SUN RGB-D Detection

Setup. We now attend to a different high-level 3D understanding task: object detection. Compared to segmentation tasks that estimate point labels, 3D

object detection predicts 3D bounding boxes (localization) and their corresponding object labels (recognition). This calls for an architectural modification as the SR-UNet architecture does not directly output bounding box coordinates. Among many different choices [28,42,44,73], we identify the recently proposed VoteNet [43] as a good candidate for three main reasons. First, VoteNet is designed to work directly on point clouds with no additional input (e.g. images). Second, VoteNet originally uses PointNet++ [46] as the backbone architecture for feature extraction. Replacing this with a SR-UNet requires a minimal modification, keeping the proposal pipeline intact. In particular, we reuse the same hyperparameters. Third, VoteNet is the current state-of-the-art method that uses geometric features only, rendering an improvement markedly useful. We evaluate the detection performance on the SUN RGB-D dataset [55], a collection of single view RGB-D images. The train set contains 5K images annotated with amodal, 3D oriented bounding boxes for objects from 37 categories.

Results. We summarize the results in Table 5. We find that by simply switching in the backbone network, our baseline results (training from scratch) with the SR-UNet architecture achieves worse results (-1.4% mAP@0.25). This may be attributed to the fact that VoteNet design and hyperparamter settings were tailored to its PointNet++ backbone. However, PointContrast gracefully closes the gap by showing a +3.1% gain on mAP@0.5, which also sets a new state-of-the-art in this metric. The performance gain with harder evaluation metric (mAP@0.5) suggests that the PointContrast pre-training can greatly help localization.

Table 5. SUN RGB-D detection results. PointContrast demonstrates a substantial boost compared to training from scratch. We observe a larger improvement in localization as manifested by the ΔmAP being larger for @0.5 than @0.25.

Methods	Input	mAP@0.5	mAP@0.25
VoteNet [43]	Geo	–	57.0
VoteNet [43]	Geo+Height	32.9	57.7
Train from scratch	Geo	31.7	55.6
PointContrast(Hardest-Contrastive)	Geo	34.5	**57.5**
PointContrast(PointInfoNCE)	Geo	**34.8**	**57.5**

Table 6. Segmentation results on the 4D Synthia test set. All networks here are SR-UNet with 3D kernels, trained on individual 3D frames without temporal modeling.

Methods	mIoU	mAcc
MinkowskiNet32 [9]	78.7	91.5
Train from scratch	79.8	91.5
PointContrast (Hardest-contrastive)	82.6	**93.7**
PointContrast (PointInfoNCE)	**83.1**	**93.7**

4.4 Synthia4D Segmentation

Setup. Synthia4D [50] is a large synthetic dataset designed to facilitate the training of deep neural networks for visual inference in driving scenarios. Photo-realistic renderings are generated from a virtual city, allowing dense and precise annotations of 13 semantic classes, together with pixel-accurate depth. We follow the train/val/test split as prescribed by [9] in the clean setting. In the context of this work, Synthia4D is especially interesting since it is probably the most distant from our pre-training set (outdoor v.s. indoor, synthetic v.s. real). We test the segmentation performance using 3D SR-UNet on a per-frame basis.

Results. PointContrast pre-training brings substantial improvement over the baseline model trained from scratch (+2.3% mIoU) as seen in Table 6. PointInfoNCE performs noticeably better than the hardest-contrastive loss. With unsupervised pre-training, the overall results are much better than the previous state-of-the-art reported in [9]. Note that in [9] it has been shown that adding the temporal learning (*i.e.* using a 4D network instead of a 3D one) brings additional benefit. To use 3D pre-trained weights for a 4D network with an additional temporal dimension, we can simply inflate the convolutional kernels, following the standard practice in 2D video recognition [6]. We leave it as a future work.

Table 7. Segmentation results on ScanNet validation set. PointContrast boosts performance on the "in-domain" transfer task where the pre-training and fine-tuning datasets come from a common source, showing the usefulness of pre-training even when labels are available.

Methods	mIoU	mAcc
Train from scratch	72.2	80.7
PointContrast(Hardest-Contrastive)	73.3	81.0
PointContrast(PointInfoNCE)	**74.1**	**81.6**

Table 8. 3D object detection results on ScanNet validation set. Similarly to in-domain *segmentation* task, here as well PointContrast boost performance on *detection*, setting a new best result over prior art. See Appendix for more methods in comparison.

Methods	Input	mAP@0.5	mAP@0.25
DSS [28,56]	Geo+RGB	6.8	15.2
3D-SIS [28]	Geo+RGB (5 Views)	22.5	40.2
VoteNet [43]	Geo+Height	33.5	58.6
Train from scratch	Geo	35.4	56.7
PointContrast(Hardest-Contrastive)	Geo	37.3	**59.2**
PointContrast(PointInfoNCE)	Geo	**38.0**	58.5

4.5 ScanNet: Segmentation and Detection

Setup. Although typically the source dataset for pre-training and the target dataset for fine-tuning are different, because of the specific multi-view contrastive learning pipeline for pre-training, it is likely that PointContrast can learn different representations (*e.g.* invariance/equivariance to rigid transformations or robustness to noise) compared to directly training with supervision. Thus it is interesting to see whether the pre-trained weights can further improve the results on ScanNet itself. We use ScanNet semantic segmentation and object detection tasks to test our hypothesis. For the segmentation experiment, we use the SR-UNet architecture to directly predict point labels. For the detection experiment, we again follow VoteNet [43] and simply switch the original backbone network with the SR-UNet without other modifications to the detection head (See Appendix for details).

Results. Results are summarized in Table 7 and Table 8. Remarkably, on both detection and segmentation benchmark, models pre-trained with PointContrast outperform those trained from scratch. Notably, PointInfoNCE objective performs better than the Hardest-contrastive one, achieving a relative improvement of +1.9% in terms of segmentation mIoU and 2.6%+ in terms of detection mAP@0.5. Similar to SUN RGB-D detection, here we also observe that PointContrast features help most for localization as indicated by the larger margin of improvement for mAP@0.5 than mAP@0.25.

4.6 Analysis Experiments and Discussions

In this section we show additional experiments to provide more insights on our pre-training framework. We use S3DIS segmentation for the experiments below.

Supervised Pre-training. While the focus of this work is unsupervised pre-training, a natural baseline is to compare against supervised pre-training. To this end, we use the training-from-scratch baseline for the segmentation task on ScanNetV2 and finetune the network on S3DIS. This yields an mIoU of 71.2%, which is only 0.3% better than PointContrast unsupervised pre-training. We deem this a very *encouraging signal* that suggests that the gap between supervised and unsupervised representation learning in 3D has been mostly closed (*cf.* years of effort in 2D). One might argue that this is due to the limited quality and scale of ScanNet, but even at this scale the amount of labor involved in annotating thousands of rooms is large. The outcome of this, complements the conclusion we had so far: not only should we put resources into creating large-scale 3D datasets for pre-training; but if facing a trade-off between scaling the data size and annotating it, we should favor the former.

Fine-Tuning vs From-Scratch Under Longer Training Schedule. Recent study in 2D vision [24] suggests that simply by training from scratch for more epochs might close the gap from ImageNet pre-training. We conduct additional

experiment to train the network from scratch with 2× and 3× schedules on S3DIS, relative to the 1× schedule of our default setup (10K iterations with batch size 48). We found that validation mIoU does not improve with longer training. In fact, the model exhibits overfitting due to the small dataset size, achieving 66.7% and 66.1% mIoU at 20K and 30K iteration, respectively. This suggests that potentially many of the 3D datasets could fall into the "breakdown regime" [24] where network pre-training is essential for good performance.

Holistic Scene as a Single View for PointContrast. To show that the multi-view design in PointContrast is important, we try a different variant where instead of having partial views $\mathbf{x}^1$ and $\mathbf{x}^2$, we directly use the reconstructed point cloud $\mathbf{x}$ (a full scene in ScanNet) PointContrast. We still apply independent transformations T_1 and T_2 to the same $\mathbf{x}$. We tried different variants and augmentations such as random cropping, point jittering, and dropout. We also tried different transformations for T_1 and T_2 of different degrees of freedom. However, with the best configuration we can get a validation mIoU on S3DIS of 68.35, which is just slightly better than the training from scratch baseline of 68.17. This suggests that the multi-view setup in PointContrast is critical. Potential reasons include: much more abundant and diverse training samples; natural noise due to camera instability as good regularization, as also observed in [75].

Acknowledgements. O.L. and L.G. were supported in part by NSF grant IIS-1763268, a Vannevar Bush Faculty Fellowship, and a grant from the SAIL-Toyota Center for AI Research.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. arXiv preprint arXiv:1707.02392 (2017)
2. Armeni, I., et al.: 3D semantic parsing of large-scale indoor spaces. In: ICCV (2016)
3. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: NeurIPS (2019)
4. Boulch, A.: ConvPoint: continuous convolutions for point cloud processing. Comput. Graph. **88**, 24–34 (2020)
5. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: ECCV (2018)
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
7. Chang, A.X., et al.: ShapeNet: an information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709 (2020)
9. Choy, C., Gwak, J., Savarese, S.: 4D spatio-temporal convnets: Minkowski convolutional neural networks. In: CVPR (2019)
10. Choy, C., Park, J., Koltun, V.: Fully convolutional geometric features. In: ICCV (2019)

11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
12. Deng, H., Birdal, T., Ilic, S.: PPF-FoldNet: unsupervised learning of rotation invariant 3D local descriptors. In: ECCV (2018)
13. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
14. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: ICCV (2015)
15. Elbaz, G., Avraham, T., Fischer, A.: 3D point cloud registration for localization using a deep neural network auto-encoder. In: CVPR (2017)
16. Gadelha, M., Wang, R., Maji, S.: Multiresolution tree networks for 3D point cloud processing. In: ECCV (2018)
17. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
18. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR (2018)
19. Goyal, P., Mahajan, D., Gupta, A., Misra, I.: Scaling and benchmarking self-supervised visual representation learning. In: ICCV (2019)
20. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: CVPR (2018)
21. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3D surface generation. In: CVPR (2018)
22. Hassani, K., Haley, M.: Unsupervised multi-task feature learning on point clouds. In: ICCV (2019)
23. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
24. He, K., Girshick, R., Dollár, P.: Rethinking ImageNet pre-training. In: ICCV (2019)
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
26. Hénaff, O.J., Razavi, A., Doersch, C., Eslami, S., Oord, A.V.D.: Data-efficient image recognition with contrastive predictive coding. arXiv preprint arXiv:1905.09272 (2019)
27. Hjelm, R.D., et al.: Learning deep representations by mutual information estimation and maximization. In: ICLR (2019)
28. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: CVPR (2019)
29. Hua, B.S., Tran, M.K., Yeung, S.K.: Pointwise convolutional neural networks. In: CVPR (2018)
30. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
31. Janoch, A. et al.: A category-level 3d object dataset: putting the kinect to work. In: Fossati, A., Gall, J., Grabner, H., Ren, X., Konolige, K. (eds.) Consumer Depth Cameras for Computer Vision. Advances in Computer Vision and Pattern Recognition. Springer, London (2013). https://doi.org/10.1007/978-1-4471-4640-7_8
32. Klokov, R., Lempitsky, V.: Escape from cells: deep Kd-networks for the recognition of 3D point cloud models. In: ICCV (2017)
33. Lei, H., Akhtar, N., Mian, A.: Spherical convolutional neural network for 3D point clouds. arXiv preprint arXiv:1805.07872 (2018)
34. Li, J., Chen, B.M., Hee Lee, G.: SO-Net: self-organizing network for point cloud analysis. In: CVPR (2018)

35. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: PointCNN: convolution on X-transformed points. In: NeurIPS (2018)
36. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
37. Maron, H., Litany, O., Chechik, G., Fetaya, E.: On learning sets of symmetric elements. arXiv preprint arXiv:2002.08599 (2020)
38. Misra, I., van der Maaten, L.: Self-supervised learning of pretext-invariant representations. In: CVPR (2020)
39. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 69–84. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_5
40. Oord, A.V.D., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
41. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: CVPR (2016)
42. Qi, C.R., Chen, X., Litany, O., Guibas, L.J.: ImVoteNet: boosting 3D object detection in point clouds with image votes. In: CVPR (2020)
43. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3D object detection in point clouds. In: ICCV (2019)
44. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum PointNets for 3D object detection from RGB-D data. In: CVPR (2018)
45. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3D classification and segmentation. In: CVPR (2017)
46. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
47. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
48. Ravanbakhsh, S., Schneider, J., Poczos, B.: Deep learning with sets and point clouds. arXiv preprint arXiv:1611.04500 (2016)
49. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
50. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
51. Sauder, J., Sievers, B.: Self-supervised deep learning on point clouds by reconstructing space. In: NeurIPS (2019)
52. Shen, Y., Feng, C., Yang, Y., Tian, D.: Mining point cloud local structures by kernel correlation and graph pooling. In: CVPR (2018)
53. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
54. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
55. Song, S., Lichtenberg, S.P., Xiao, J.: Sun RGB-D: a RGB-D scene understanding benchmark suite. In: CVPR (2015)
56. Song, S., Xiao, J.: Deep sliding shapes for amodal 3D object detection in RGB-D images. In: CVPR (2016)

57. Su, H., et al.: SPLATNet: sparse lattice networks for point cloud processing. In: CVPR, pp. 2530–2539 (2018)
58. Te, G., Hu, W., Zheng, A., Guo, Z.: RGCNN: regularized graph CNN for point cloud segmentation. In: ACM Multimedia (2018)
59. Verma, N., Boyer, E., Verbeek, J.: FeastNet: feature-steered graph convolutions for 3D shape analysis. In: CVPR (2018)
60. Wang, C., Samari, B., Siddiqi, K.: Local spectral graph convolution for point set feature learning. In: ECCV (2018)
61. Wang, S., Suo, S., Ma, W.C., Pokrovsky, A., Urtasun, R.: Deep parametric continuous convolutional neural networks. In: CVPR (2018)
62. Wang, Y., Solomon, J.M.: Deep closest point: learning representations for point cloud registration. In: ICCV (2019)
63. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM TOG **38**(5), 1–12 (2019)
64. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018)
65. Xiao, J., Owens, A., Torralba, A.: SUN3D: a database of big spaces reconstructed using SFM and object labels. In: ICCV (2013)
66. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (2017)
67. Xie, S., Liu, S., Chen, Z., Tu, Z.: Attentional ShapeContextNet for point cloud recognition. In: CVPR, pp. 4606–4615 (2018)
68. Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y.: SpiderCNN: deep learning on point sets with parameterized convolutional filters. In: ECCV (2018)
69. Yang, Y., Feng, C., Shen, Y., Tian, D.: FoldingNet: point cloud auto-encoder via deep grid deformation. In: CVPR (2018)
70. Yang, Z., Litany, O., Birdal, T., Sridhar, S., Guibas, L.: Continuous geodesic convolutions for learning on 3D shapes. In: arXiv preprint arXiv:2002.02506 (2020)
71. Yi, L., et al.: A scalable active framework for region annotation in 3D shape collections. In: SIGGRAPH Asia (2016)
72. Yi, L., Su, H., Guo, X., Guibas, L.J.: SyncSpecCNN: synchronized spectral CNN for 3D shape segmentation. In: CVPR (2017)
73. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.: GSPN: generative shape proposal network for 3d instance segmentation in point cloud. In: CVPR (2019)
74. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: NeurIPS (2017)
75. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3DMatch: learning local geometric descriptors from RGB-D reconstructions. In: CVPR (2017)
76. Zeng, W., Gevers, T.: 3DContextNet: KD tree guided hierarchical learning of point clouds using local and global contextual cues. In: ECCV (2018)
77. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
78. Zhang, R., Isola, P., Efros, A.A.: Split-brain autoencoders: unsupervised learning by cross-channel prediction. In: CVPR (2017)
79. Zhang, Z., Hua, B.S., Yeung, S.K.: ShellNet: efficient point cloud convolutional neural networks using concentric shells statistics. In: ICCV (2019)
80. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3D point capsule networks. In: CVPR (2019)
81. Zhuang, C., Zhai, A.L., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings. In: ICCV (2019)