Exploring Data-Efficient 3D Scene Understanding with Contrastive Scene Contexts

Ji Hou[1] Benjamin Graham[2] Matthias Nießner[1] Saining Xie[2]
[1]Technical University of Munich [2]Facebook AI Research

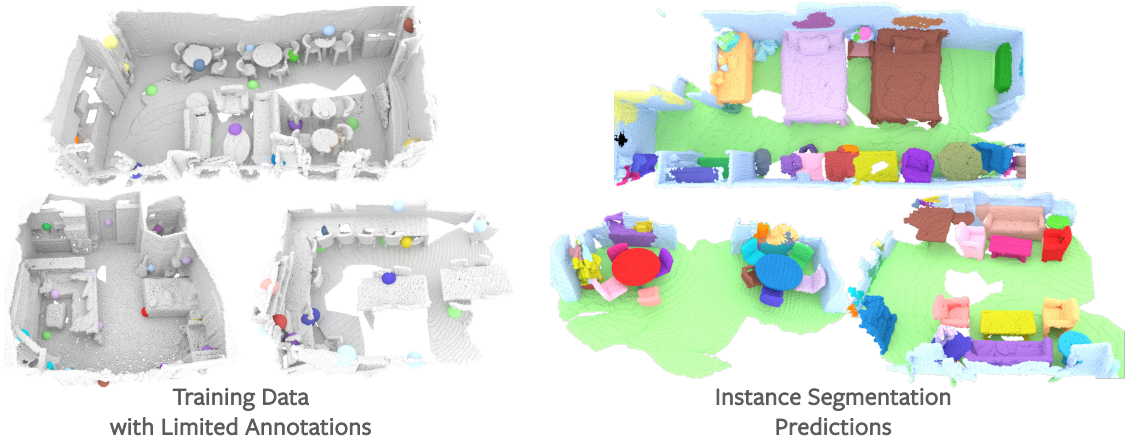

Figure 1: How many point labels are necessary to train a 3D instance segmentation model on point clouds? It turns out not too many! With the help of unsupervised pre-training, only 20 labelled points per scene (*less than 0.1% of the total points*) are used to fine-tune an instance segmentation model on ScanNet. **Left**: Train samples; only colored points (enlarged for better visibility) are labeled. **Right**: Predictions in validation set and different colors represent different instances.

Abstract

*The rapid progress in 3D scene understanding has come with growing demand for data; however, collecting and annotating 3D scenes (*e.g. *point clouds) are notoriously hard. For example, the number of scenes (*e.g. *indoor rooms) that can be accessed and scanned might be limited; even given sufficient data, acquiring 3D labels (*e.g. *instance masks) requires intensive human labor. In this paper, we explore data-efficient learning for 3D point cloud. As a first step towards this direction, we propose Contrastive Scene Contexts, a 3D pre-training method that makes use of both point-level correspondences and spatial contexts in a scene. Our method achieves state-of-the-art results on a suite of benchmarks where training data or labels are scarce. Our study reveals that exhaustive labelling of 3D point clouds might be unnecessary; and remarkably, on ScanNet, even using 0.1% of point labels, we still achieve 89% (instance segmentation) and 96% (semantic segmentation) of the baseline performance that uses full annotations* [1].

[1]Code is available at GitHub

1. Introduction

Recent advances in deep learning on point clouds, such as those obtained from LiDAR or depth sensors, together with a proliferation of public, annotated datasets [9, 13, 53, 32, 65, 2, 40, 56], have led to swift progress in 3D scene understanding. However, compared to large-scale 2D scene understanding on images [14, 38, 23], the scale of 3D scene understanding—in terms of the amount and diversity of data and annotations, the model size, the number of semantic categories, and so on—still falls behind. We argue that one major bottleneck is the fact that collecting and annotating diverse 3D scenes are significantly more expensive. Unlike 2D images that comfortably exists on the Internet, collecting real world 3D scene datasets usually involves traversing the environment in real life and scanning with 3D sensors. Therefore, the number of indoor scenes that can be scanned might be limited. What is more concerning is that, even given sufficient data acquisition, 3D semantic labelling (*e.g.* bounding boxes and instance masks) requires complex pipelines [13] and labor-intensive human effort.

In this work, we explore a new learning task in 3D, *i.e.* data-efficient 3D scene understanding, which focuses on the

problem of learning with limited data or supervision[2]. We note that the importance of data-efficient learning in 3D is two-fold. One concerns the status quo: given limited data we have right now, *can we design better methods that perform better?* The other one is more forward-looking: *is it possible to reduce the human labor for annotation*, with a goal of creating 3D scene datasets on a much larger scale?

To formally study the problem, we first introduce a suite of scene understanding benchmarks that encompasses two complementary settings for data-efficient learning: (1) *limited scene reconstructions (LR)* and (2) *limited annotations (LA)*. The first setting concerns the scenario where the bottleneck is the number of scenes that can be scanned and reconstructed. The second one focuses on the case where in each scene, the budget for labeling is constrained (*e.g.* one can only label a small set of points). For each setting, the evaluation is done on a diverse set of scene understanding tasks including object detection, semantic segmentation and instance segmentation.

For data-efficient learning in 2D [28], representation learning, *e.g.* pre-training on a rich source set and fine-tuning on a much smaller target set, often comes to the rescue; in 3D, representation learning for data-efficient learning is even more wanted but long overdue. With this perspective, we focus on studying data-efficient 3D scene understanding through the lens of representation learning.

Only recently, PointContrast [66] demonstrates that network weights pre-trained on 3D partial frames can lead to a performance boost when fine-tuned on 3D semantic segmentation and object detection tasks. Our work is inspired by PointContrast. However, we observe that the simple contrastive-learning based pretext task used in [66] only concerns point-level correspondence matching, which completely disregards the spatial configurations and contexts in a scene. In Section 3, we show that this design limits the scalibility and transferability; we further propose an approach that integrates the spatial information into the contrastive learning framework. The simple modification can significantly improve the performance over PointContrast, especially on complex tasks such as instance segmentation.

Our exploration in data-efficient 3D scene understanding provides some surprising observations. For example, on ScanNet, even using 0.1% of point labels, we are still able to recover 89% (instance segmentation) and 96% (semantic segmentation) of the baseline performance that uses full annotations. The results imply that exhaustive labelling of 3D point clouds might not be necessary. In both scenarios of *limited scene reconstructions (LR)* and *limited annotations (LA)*, our pre-trained network, when used as the initialization for supervised fine-tuning, offers consistent improvement across multiple tasks and datasets. In the scenario of *LA*, we also show that an active labeling strategy can be enabled by clustering the pre-trained point features.

In summary, the contributions of our work include:

- A systematic study on data-efficient 3D scene understanding with a comprehensive suite of benchmarks.
- A new 3D pre-training method that can gracefully transfer to complex tasks such as instance segmentation and outperform the state-of-the-art results.
- Given the pre-trained network, we study practical solutions for data-efficient learning in 3D through fine-tuning as well as an active labeling strategy.

2. Related Work

3D Scene Understanding. Research in deep learning on 3D point clouds have been recently shifted from synthetic, single object classification [47, 46, 48] to the challenge of large-scale, real-world scene understanding. A variety of datasets [2, 13, 54, 18, 56] and algorithms have been proposed for 3D object detection [45, 44, 43, 24], semantic segmentation [46, 57, 63, 58, 20, 12] and instance segmentation [60, 30, 71, 36, 61, 69, 31, 15, 34, 33]. In the past year, sparse convolutional networks [20, 12] stand out as a promising approach to standardize deep learning for point clouds, due to its computational efficiency and state-of-the-art performance for 3D scene understanding tasks [12, 25, 34]. In this work, we also adopt a sparse U-Net [49] backbone for our exploration.

3D Representation Learning. Compared to 2D vision, the limits of big data are far from being fully explored in 3D. In 2D representation learning, for example, transfer learning from a rich source data (*e.g.* ImageNet [14]) to a (typically smaller) target data, has become a dominant framework for many applications [19]. In contrast, 3D representation learning has not been widely adopted and most 3D networks are trained from scratch on the target data directly. Recently, unsupervised pre-training has made great progress and drawn significant attention in 2D [42, 3, 39, 28, 64, 59, 29, 27, 10, 8, 21]. Following suit, recent works attempt to adapt the 2D pretext tasks to 3D, but mostly focus on single object classification tasks on ShapeNet [1, 17, 70, 22, 37, 62, 26, 51, 50]. Our work is mostly inspired by a recent contrastive-learning based method PointContrast [66], which first demonstrates the effectiveness of unsupervised pre-training on a diverse set of scene-level understanding tasks. As we will show in the later sections, the simple point-level pre-training objective in PointContrast ignores the spatial contexts of the scene (such as relative poses of objects, and distances between them) which limits its transferability for complex tasks such as instance segmentation. PointContrast also focuses on

[2]Sometimes a distinction is drawn between *data-efficiency* and *label-efficiency*, to separate the scenarios of limited amount of data samples and limited supervision; here, we use *data-efficiency* to encompass both cases.

downstream tasks with 100% data and labels, while we systematically explore a new data-efficient paradigm that has practical importance.

Data-Efficient Learning. Data-efficient learning concerns the problem of learning with limited training examples or labels. This capability is known in cognitive science to be a distinctive characteristic of humans [6]. In contrast, training deep neural networks is not naturally data-efficient, as it typically relies on large amount of annotated data. Among many potential solutions towards this goal, representation learning (commonly through transfer learning) is arguably the most promising one. A good representation "*entangles the different explanatory factors of variation behind the data*" [5] and thus makes the downstream prediction easier (and less data-hungry). This concept has been validated successfully in natural language processing [7] and to some extent in 2D image classification [28]. Pursuing this direction in 3D is even more desirable, considering the potential benefit in reducing the labor of data collection and annotation. Existing work focuses on mostly single CAD model classification or part segmentation [72, 52, 41, 11, 26, 16, 68]. To the best of our knowledge, our work is the first to systematically explore data-efficient learning in a real-world, large-scale 3D scene understanding (on semantic/instance segmentation and detection) setup.

3. Contrastive Scene Contexts for Pre-training

In this section, we first briefly revisit the PointContrast framework [66], and discuss the shortcomings and remedies. We then introduce our pre-training algorithm.

Revisiting PointContrast. The pre-training objective for PointContrast is to achieve point *equivariance* with respect to a set of random geometric transformations. Given a pair of overlapping partial scans, a contrastive loss for pre-training is defined over the point features. The objective is to minimizes the distance for matched points (positive pairs) and maximize the distance between unmatched ones (negative pairs). Despite the fact that strong spatial contexts exist among objects in a scene, this objective does not capture any of the spatial information: the negative pairs could be sampled from arbitrary locations across many scenes in a mini-batch. We hypothesize that this leads to some limitations: 1) the spatial contexts (*e.g.* relative pose, direction and distance), which could be pivotal for complex tasks such as instance segmentation, are entirely discarded from pre-training; 2) the scalibility of contrastive learning might be hampered; PointContrast cannot utilize a large number of negative points, potentially because that contrasting a pair of spatially distant and unrelated points would contribute little to learning. In fact, PointContrast uses only a random sampling of 1024 points per scene for pre-training, and it has been shown that results do not improve with more sampled points [66]. We also confirm this behavior with experiments later this section.

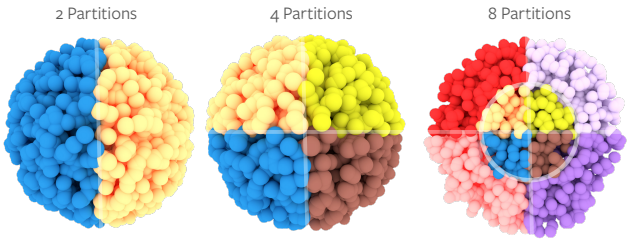

Figure 2: **Illustration of Scene Contexts.** We visualize the 2,4 and 8 spatial partitions for Scene Contexts. The anchor point is in the center. For 2 and 4 partitions, only relative angles are sufficient. For 8 partitions (a cross-section is shown), both relative angles and distances are needed.

Contrastive Scene Contexts. We hope to integrate spatial contexts into the pre-training objective. There are many ways to achieve the goal, and here we take inspiration from the classic *ShapeContext* local descriptor [4, 35, 67] for shape matching. The *ShapeContext* descriptor partitions the space into spatially inhomogeneous cells, and encodes the spatial contexts about the shape at each point by computing a histogram over the number of neighboring points in each cell. We call our method *Contrastive Scene Contexts* because at a high level, our method also aims to capture *the distribution over relative locations in a scene*. We partition the scene point cloud into multiple regions, and instead of having a single contrastive loss for the entire point set sampled in a mini-batch, we perform contrastive learning in each region separately, and aggregate the losses in the end.

Concretely, given a pair of partial frame point clouds $\mathbf{x}$ and $\mathbf{y}$ from the same scene, we have correspondence mapping $(i, j) \in M_{\mathbf{xy}}$ available, where i is the index of a point $\mathbf{x}_i \in \mathcal{R}^3$ in frame $\mathbf{x}$ and j is the index of a matched point $\mathbf{y}_j \in \mathcal{R}^3$ in frame $\mathbf{y}$. Similar to PointContrast, we sample N pairs of matched points as positives. However, in our method, for each anchor point $\mathbf{x}_i$, the space is divided into multiple partitions and other points are assigned to different partitions based on their relative angles and distances to i.

The distance and angle information needed for scene context partition at anchor point $\mathbf{x}_i$ is as follows,

$$\mathcal{D}_{ik} = \sqrt{\sum_{d=1}^{3} (\mathbf{x}_i^d - \mathbf{x}_k^d)^2} \tag{1}$$

$$\mathcal{A}_{ik} = arctan2(\mathcal{D}_{ik}) + 2\pi \tag{2}$$

where $\mathcal{D}$ is the relative distance matrix. $\mathcal{D}_{ik}$ stores the distance between point i and point k and $\mathcal{A}$ is the relative angle matrix, where A_{ik} stores the relative angle between point i and point k. In Equation (1) d represents the 3D dimension. With $\mathcal{D}$ and $\mathcal{A}$, a *ShapeContext*-like spatial partitioning function can be easily constructed on-the-fly. In Fig-

ure 2, we show a visual illustration of how the space partitioning works. Computing 2 or 4 partitions only requires cutting the space according to relative angles based on $\mathcal{A}$; while the 8 or more partitions also require the extent of the inner regions using $\mathcal{D}$. We always partition the space uniformly along the relative angles and distances. Note that the partitioning is *relative to the anchor point* i.

Suppose there are P partitions, we denote the spatial partition functions as $par_p(\cdot)$, where $p \in \{1, \dots, P\}$. Function $par_p(\cdot)$ takes the anchor point i as input, and return a set of points as negatives. A PointInfoNCE loss $\mathcal{L}_p$ is independently computed for each partition:

$$\mathcal{L}_p = -\sum_{(i,j)\in M} \log \frac{\exp(\mathbf{f}_i^1 \cdot \mathbf{f}_j^2/\tau)}{\sum_{(\cdot,k)\in M, k\in par_p(i)} \exp(\mathbf{f}_i^1 \cdot \mathbf{f}_k^2/\tau)} \quad (3)$$

Details of Equation (3) and other implementation details can be found in Appendix. The final loss is computed by aggregating all partitions $\mathcal{L} = \frac{1}{|P|}\sum_p \mathcal{L}_p$.

Analysis. We first show that by integrating the scene contexts into the objective, our pre-training method can benefit more from a larger point set. We conduct an analysis experiment by varying the number of scene context partitions and the number of points sampled for computing the contrastive loss. We pre-train our model for a short schedule (20K iters). We then fine-tune the pre-trained weights on S3DIS instance segmentation benchmark [2]. Results are shown in Figure 3, the green line represents a variant with *no spatial partitioning*; the left-most point represents PointContrast[3]. Similar to the observation in [66], without scene contexts, increasing the number of sampled points does not improve the performance; with more partitions, increasing # sampled points leads to a consistent boost in performance (up to 4096 points). We use 8 partitions as empirically it works best. This shows that our method leads to better *scalability* as more points can be utilized for pre-training.

We achieve state-of-the-art instance segmentation results in terms of mAP@0.5 (Table. 1) using a simple bottom-up clustering mechanism with voting loss (details in Appendix). We do not use any special modules such as Proposal Aggregation [15] or Scoring Network [34]. We observe a 2.9% absolute improvement over PointContrast pre-training, which brings the improvement over train-from-scratch baseline to 4.1%. This substantial margin demonstrates the effectiveness of Contrastive Scene Contexts on instance segmentation tasks. We provide more results comparing against PointContrast in Section 5.3.

4. Data-Efficient 3D Scene Understanding

To formally explore data-efficient 3D scene understanding, in this section, we propose two different learning paradigms and relevant benchmarks that are associated with

[3]Not exactly identical since the matched points are sampled per scene in this experiment, rather than from the whole mini-batch as in PointContrast; we have verified that this nuance does not influence the conclusion.

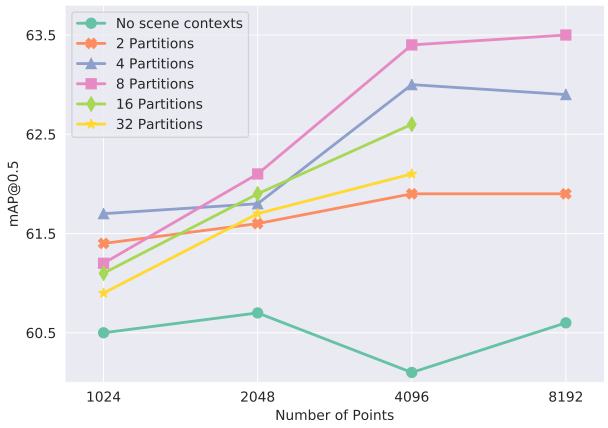

Figure 3: **Analysis Experiment.** Varying the number of partitions and sampled points for pre-training; Results are reported on the S3DIS instance segmentation task [2]. Using scene context partitions has enabled constrastive learning to utilize more points for better performance.

Methods	mAP@0.5
ASIS [61]	55.3
3D-BoNet [69]	57.5
PointGroup [34]	57.8
3D-MPA [15]	63.1
Train from scratch	59.3
PointContrast (PointInfoNCE) [66]	60.5 (+1.2)
Contrastive Scene Contexts	**63.4** (+4.1)

Table 1: **Fine-tuning results for instance segmentation on S3DIS [2].** A simple clustering-based model with *Contrastive Scene Contexts* pre-trained backbone performs significantly better than the train-from-scratch baseline and PointContrast pre-training [66].

two complementary settings that can occur in real world application scenarios: (1) *limited scene reconstructions (LR)* and (2) *limited annotations (LA)*. The first setting mainly concerns the scenario where the bottleneck of data collection is the *number of scenes* that can be scanned and reconstructed. The second one focuses on the case where in each scene, the budget for labeling is limited (*e.g.* one can only label a small set of points). Since 3D point labeling is human intensive, this represents a practical scenario where a data-efficient learning strategy can greatly reduce the annotation cost. An overview is presented in Figure 4, and details of individual benchmarks are described below.

4.1. Limited Annotations (LA)

In this benchmark, we explore 3D scene understanding with a limited budget for point cloud annotations. We consider a diverse set of tasks including semantic segmentation, instance segmentation and object detection. Specifically, for instance segmentation and semantic segmentation, the

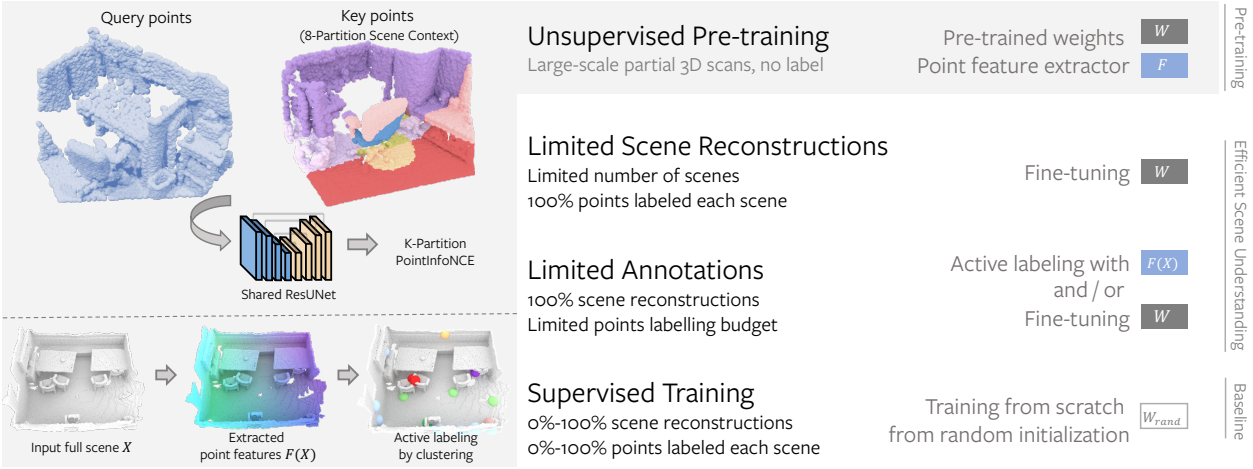

Figure 4: **Overview of Data-Efficient 3D Scene Understanding. Left**: Unsupervised pre-training with *Contrastive Scene Contexts*. The outputs of pre-training are 1) a pre-trained U-Net **F** (that can be used as an offline feature extractor) and 2) its associated weights **W**. **Right**: After pre-training, different learning scenarios can be applied for the downstream tasks such as learning with limited scene reconstructions (*LR*) or limited annotations (*LA*). In the case of *LR*, the pre-trained weights **W** are used as network initialization for fine-tuning. In the case of *LA*, all the scene reconstructions can be used but only a limited annotation budget is available, *e.g.* 20 points can be annotated (semantic labels) per scene. Again, **W** can be used as network initialization for fine-tuning; optionally the feature extractor **F** can be used in an active labeling strategy to decide which points to annotate. Baselines are standard supervised learning where models are trained from scratch.

annotation budget is in terms of the *number of points* for labelling. This is practically useful: if an annotator only needs to label the semantic labels for 20 points, it will only require a few minutes to label a full room. Our benchmark considers four different training configurations on ScanNet including using $\{20, 50, 100, 200\}$ labeled points per scene. For object detection, the annotation budget is with respect to the *number of bounding boxes* to label in each scene. Our benchmark considers four different training configurations including $\{1, 2, 4, 7\}$ labeled bounding boxes. Our base dataset is ScanNetV2 [13] which has 1201 scenes for training. We evaluate the model performance on standard ScanNetV2 validation set of 312 scenes that has full labels.

4.2. Limited Scene Reconstructions (LR)

For current 3D scene datasets, it is common for annotators to carry commodity depth cameras and record 3D videos at private houses or furniture stores. It might be unrealistic to enter a large number of homes and obtain detailed scanning. In this case, the number of scenes might be the bottleneck and the training has to be done on limited amount of scene reconstructions. We simulate this scenario by random sampling a subset of ScanNetV2 training set. Our benchmark has four configurations $\{1\%, 5\%, 10\%, 20\%\}$ (100% represents the entire ScanNet train set) for semantic segmentation and instance segmentation; and $\{10\%, 20\%, 40\%, 80\%\}$ for object detection. During test time, evaluation is on all scenes in the validation set.

5. Experimental Results

In this section, we present our experimental results on the data-efficient 3D scene understanding benchmarks: **ScanNet-LA** with limited annotations and **ScanNet-LR** with limited scene reconstructions. In both scenarios, we compare our method against the baseline of training from scratch, and report results on semantic/instance segmentation and object detection. We also compare our models with the state-of-the-art method in the last part of the section.

Experiments Setup For pre-training, we use SGD optimizer with learning rate 0.1 and a batch-size of 32. The learning rate is decreased by a factor of 0.99 every 1000 steps. The model is trained for 60K steps. The fine-tuning experiments on instance segmentation and semantic segmentation are trained with a batch-size of 48 for a total of 10K steps. The initial learning rate is 0.1, with polynomial decay with power 0.9. For all experiments, we use data parallel on 8 NVIDIA V100 GPUs. For object detection experiments, we fine-tune the model with a batch-size of 32 for 180 epochs. The initial learning rate is set to 0.001 and decayed by a factor of 0.1 at epoch 80, 120 and 160. For all the experiments, we use the same Sparse Res-UNet [66] as the backbone. For both training and testing, the voxel size for Sparse ConvNet is set to 2.0 cm. We use Sparse ConvNet implemented by MinkowskiEngine [12].

5.1. Limited Annotations

As introduced in Section 4, the *Limited Annotation (LA)* benchmark covers two different annotation types: *Limited*

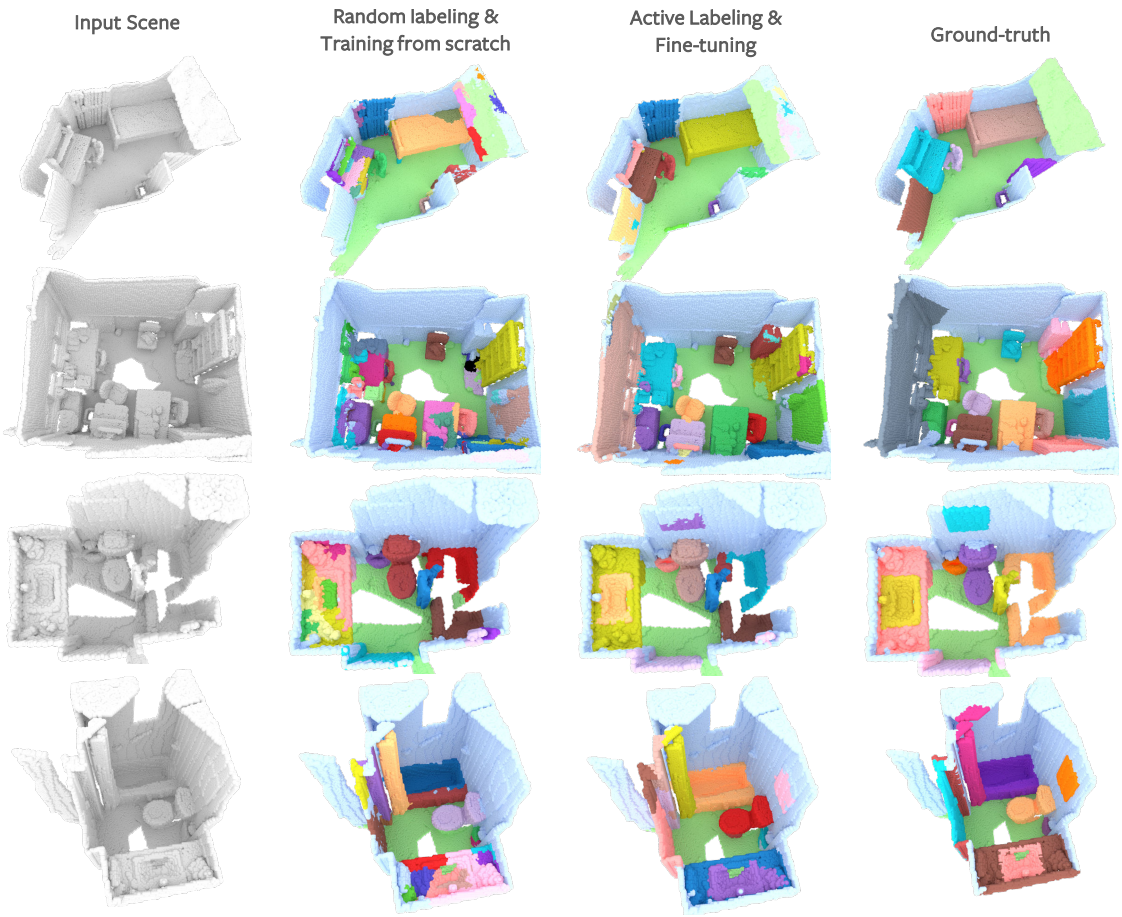

Figure 5: **Qualitative Instance Segmentation Results (ScanNet-LA).** With our pre-trained model as initialization for fine-tuning, together with an active labeling process, our approach (trained with 20 labeled points per scene) generates high-quality instance masks. Different color represents instance index only (same instances might not share the same color).

Point Annotations for semantic and instance segmentation and *Limited Bounding Box Annotations* for detection. The pre-trained network (and its weights) can be used as initialization for fine-tuning, or integrate in an active labeling strategy, which we describe below.

Active labeling. Since we focus on the scenario of having limited annotation budget, it is natural to consider an *active learning* strategy during the data annotation process; *i.e.* one can interactively query an annotator to label some data points that can help most for subsequent training. The core idea of our approach is to perform a **balanced sampling** on the **feature space**, so that the selected points will be the most representative and exemplary ones in a scene. Our pre-trained network extracts dense features at each point of the to-be-annotated point cloud, by simply performing a forward pass. We then perform k-means clustering in this feature space to obtain K cluster centroids. We select the K centroids as the points to be provided to the annotators for labeling. We also present two baseline strategies including a simple **random sampling** strategy where K points are randomly selected to be labeled, and a similar **k-means sampling** strategy on raw (RGB+XYZ) inputs, rather than on the pre-trained features.

We note that although our experiments are simulated based on the already collected ScanNet dataset, our pre-trained feature extractor and the labeling strategy are readily useful in a real-world data annotation pipeline.

Results. In Figure 6 we show that compared to the naive from-scratch baselines, our proposed pre-training framework can lead to much improved performance. It is interesting to see that, for both semantic segmentation and instance segmentation, even *without* fine-tuning, the *active labeling* strategy alone provides point labels that make the trained model perform significantly better, compared to *random sampling* or *k-means sampling* baseline strategies, yielding a $>$10% absolute improvement in terms of mAP@0.5 and mIoU when the training data has only 20 point labels.

The fact that *active labeling* strategy performs on

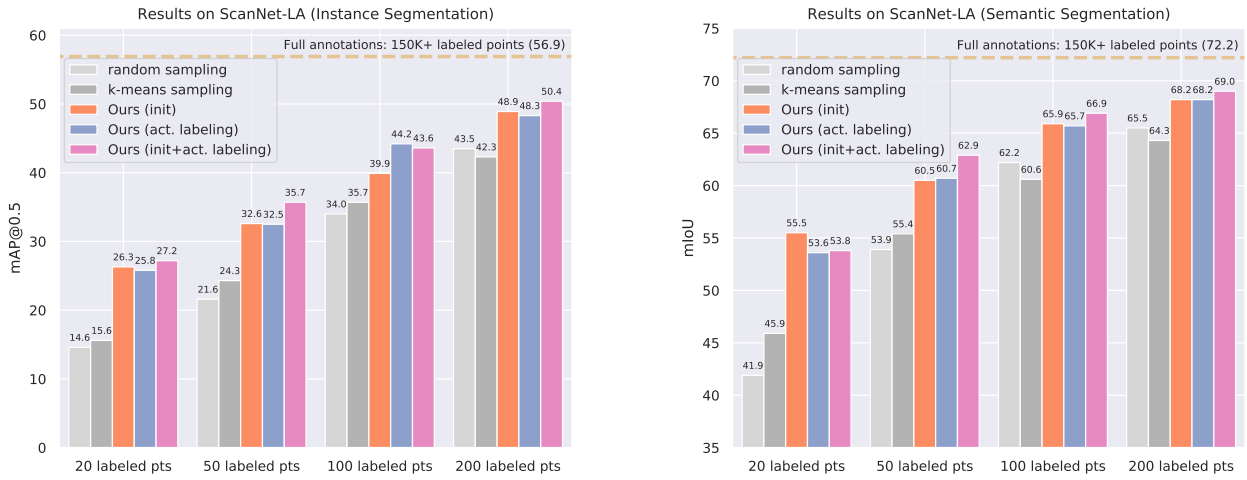

Figure 6: **3D Instance and Semantic Segmentation with Limited Point Annotations (ScanNet-LA).** Ours (init) denotes the network initialization by our pre-trained model. Ours (act. labeling) denotes the active selection of annotated points by our pre-trained model. Ours (init+act. labeling) denotes using our model as both network initialization and active labeling. We additionally mark the upper bound of using all 150K annotated points (in average) per scene as the dash line.

No. of Boxes	VoteNet (scratch)	VoteNet (ours)
all	35.4	**39.3** (+3.9)
1	27.5	**30.3** (+2.8)
2	30.9	**32.4** (+1.5)
4	32.5	**34.6** (+2.1)
7	33.4	**35.9** (+2.5)

Table 2: **Object detection results using Limited Bounding Box Annotations on ScanNet.** The metric is mAP@0.5. "Ours" denotes the fine-tuning results with our pre-trained model. We list the upper-bound performance using all annotated bounding boxes (in average about 13 bounding boxes per scene) as a reference in the first row.

par with the more common pre-training and fine-tuning paradigm, suggests that finding exemplary points to label is crucial for data-efficient learning. Of course, in real applications both active labeling and fine-tuning can be used jointly, and we indeed observe a further (though admittedly smaller) boost in performance by 1) active sampling points to label and then 2) fine-tuning with the pre-trained weights.

Overall, with the help of our Contrastive Scene Contexts pre-training, even using around 0.1% of point labels (*e.g.* 200 labeled points out of 150K total points per scene), we are still able to achieve 50.4% mAP@0.5 for instance segmentation, and 69.0% mIoU for semantic segmentation. This indicates a recovery of 89% and 96% of baseline performance that uses 100% of the annotations. We show additional qualitative comparison in Figure 5.

Limited Bounding Box Annotations. For object detection, we use VoteNet [44] as the detector framework; following [66], we replace PointNet [46] with our Sparse ResUNet. For this part, we do not use any active labeling strategy as the labeling cost for bounding boxes are much smaller. We random sample $\{1, 2, 4, 7\}$ bounding boxes per scene and train the detector. In Table 2, we observe that our pre-training also consistently improves over the baseline VoteNet, and the performance gap does not diminish when more box annotations are available.

5.2. Limited Scene Reconstructions

In this section, we report the experimental results for another scenario of data-efficient 3D scene understanding, when there is a shortage of scene reconstructions. For instance segmentation and semantic segmentation tasks, we random sample subsets of ScanNet scenes of different sizes. We sample $\{1\%, 5\%, 10\%, 20\%\}$ of the entire 1201 scenes in the training set (which corresponds to 12, 60, 120, and 240 scenes, respectively). For object detection, we find it very difficult to train the detector when the scenes are too scarce. Thus we sample $\{10\%, 20\%, 40\%, 80\%\}$ subsets. For each configuration, we randomly sample 3 subsets and report the averaged results to reduce variance. We also use the official ScanNetV2 validation set for evaluation.

Network fine-tuned with our pre-trained model again shows a clear gap compared to the training from scratch baseline (Table 3). We achieve competitive results (50.6% mAP@0.5 for instance segmentation and 64.6% mIoU for semantic segmentation) using only 20% of the total scenes.

Similar behavior can be observed on the object detection task on ScanNet, and the difference between with and without our pre-training is more pronounced in Table 4: the detector can barely produce any meaningful results when the data is scarce (*e.g.* 10% or 20%) and trained from scratch. However, fine-tuning with our pre-trained weights, VoteNet

Data Pct.	Instance Seg.		Semantic Seg.	
	Scratch	Ours	Scratch	Ours
100%	56.9	**59.4** (+2.5)	72.2	**73.8** (+1.6)
1%	9.9	**13.2** (+3.3)	26.0	**28.9** (+2.9)
5%	31.9	**36.3** (+4.4)	47.8	**49.8** (+2.0)
10%	42.7	**44.9** (+2.2)	56.7	**59.4** (+2.7)
20%	48.1	**50.6** (+2.5)	62.9	**64.6** (+1.7)

Table 3: **3D semantic and instance segmentation results with Limited Scene Reconstructions (ScanNet-LR).** Metric is mAP@0.5 for instance segmentation and mIoU for semantic segmentation. "Scratch" denotes the training from scratch baseline, and "Ours" denotes the fine-tuning results using our pre-trained weights. Results using 100% of the data during training are listed in the first row.

can perform significantly better (*e.g.* improve the mAP@0.5 by more than 16% with 20% training data).

Data Pct.	VoteNet (scratch)	VoteNet (ours)
100%	35.4	**39.3** (+3.9)
10%	0.3	**8.6** (+8.3)
20%	4.6	**20.9** (+16.3)
40%	22.0	**29.2** (+7.2)
80%	33.7	**36.7** (+3.0)

Table 4: **Object detection results with Limited Scene Reconstructions on ScanNet.** Metric is mAP@0.5. We show constantly improved results over training from scratch, especially so when 10% or 20% of the data are available. Results using all scenes are listed in the first row.

5.3. Additional Comparisons to PointContrast

As Contrastive Scene Contexts is closely related to PointContrast [66], we provide additional results in this section, including comparisons on the data-efficient ScanNet benchmarks (Table 5) as well as on other datasets and benchmarks (Table 6). Our pre-training method outperforms [66] in almost every benchmark setting, sometimes by a big margin. These results further render the importance of integrating scene contexts in contrastive learning. Notably, our pre-training method on S3DIS achieves 72.2% mIoU which outperforms, for the first time, the *supervised* pre-training result reported in [66].

5.4. Analysis on Active Labeling: Cluttered Scenes

To better explain our active labeling strategy and show that it can work in scenes with heavy occlusion and clutter, we filter out a ScanNet subset of 200 cluttered scenes that has multiple objects per one square meter area. Compared to naive k-means sampling, active labeling performs even better on cluttered scenes. In Figure 7, we visualize a cluttered scene and sampled points (bottom); we also show quantitatively (top) our strategy covers more distinct objects and thus has a balancing effect.

Settings	Task (Metric)	SC	PC [66]	Ours
LA (200 points)	ins (mAP@0.5)	43.5	44.5 (+1.0)	**48.9** (+5.4)
LA (200 points)	sem (mIoU)	65.5	67.8 (+2.3)	**68.2** (+2.7)
LA (7 bboxes)	det (mAP@0.5)	33.4	34.9 (+1.5)	**35.9** (+2.5)
LR (240 scenes)	ins (mAP@0.5)	48.1	48.4 (+0.3)	**50.6** (+2.5)
LR (240 scenes)	sem (mIoU)	62.9	63.0 (+0.1)	**64.6** (+1.7)
LR (960 scenes)	det (mAP@0.5)	33.7	36.3 (+2.6)	**36.7** (+3.0)

Table 5: **Comparisons to PointContrast for data-efficient 3D scene understanding on ScanNet.** We compare our method with PointContrast (PC) and training from scratch (SC) in various tasks. Our method constantly achieves better results in both Limited Point Annotations (LA) and Limited Scene Reconstructions (LR) scenarios.

Datasets	Task (Metric)	SC	PC [66]	Ours
S3DIS	ins (mAP@0.5)	59.3	60.5 (+1.2)	**63.4** (+4.1)
S3DIS	sem (mIoU)	68.2	70.3 (+2.1)	**72.2** (+4.0)
SUN RGB-D	det (mAP@0.5)	31.7	34.8 (+3.1)	**36.4** (+4.7)
ScanNet	ins (mAP@0.5)	56.9	58.0 (+1.1)	**59.4** (+2.5)
ScanNet	sem (mIou)	72.2	**74.1** (+1.9)	73.8 (+1.6)
ScanNet	det (mAP@0.5)	35.4	38.0 (+2.6)	**39.3** (+3.9)

Table 6: **Downstream fine-tuning results on other benchmarks.** Contrastive Scene Contexts (Ours) achieve better or on par results compared to PointContrast (PC) [66] on instance segmentation (ins), semantic segmentation (sem) and object detection (det) across multiple datasets.

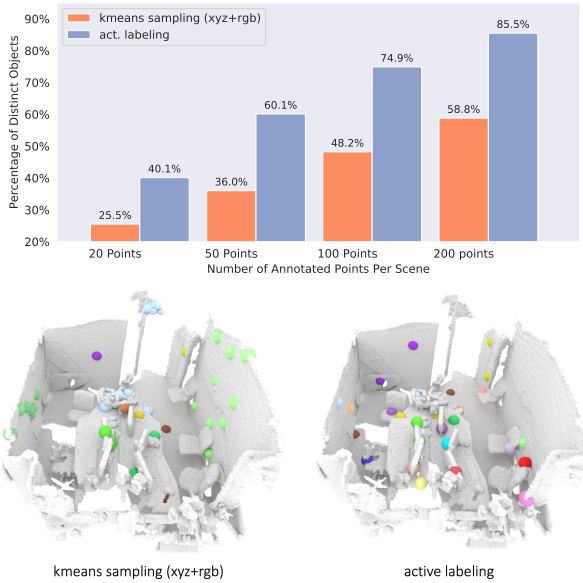

Figure 7: **Top**: object coverage percentage—more distinct objects are covered with active labeling; **Bottom**: Visualization of sampled points in a cluttered scene.

6. Conclusion

In this work, we focus on data-efficient 3D scene understanding through a novel unsupervised pre-training algorithm that integrates the scene contexts in the contrastive

learning framework. We show the possibility of using extremely few data or annotations to achieve competitive performance leveraging representation learning. Our results and findings are very encouraging and can potentially open up new opportunities in 3D (interactive) data collection, unsupervised 3D representation learning, and large-scale 3D scene understanding.

Acknowledgments Work done during Ji's internship at FAIR. Matthias Nießner was supported by ERC Starting Grant *Scan2CAD (804724)*. The authors would like to thank Norman Müller, Manuel Dahnert, Yawar Siddiqui and Angela Dai and anonymous reviewers for their constructive feedback.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. *ICML*, 2018. 2

[2] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *ICCV*, 2016. 1, 2, 4, 14, 15

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019. 2

[4] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 24(4):509–522, 2002. 3

[5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *TPAMI*, 35(8):1798–1828, 2013. 3

[6] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 3

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 3

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2

[9] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020. 2

[11] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-Net: Branched autoencoder for shape co-segmentation. In *CVPR*, 2019. 3

[12] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 2, 5, 12

[13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 1, 2, 5, 15

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2

[15] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In *CVPR*, 2020. 2, 4, 15

[16] Matheus Gadelha, Aruni RoyChowdhury, Gopal Sharma, Evangelos Kalogerakis, Liangliang Cao, Erik Learned-Miller, Rui Wang, and Subhransu Maji. Label-efficient learning on point clouds using approximate convex decompositions. *ECCV*, 2020. 3

[17] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3D point cloud processing. In *ECCV*, 2018. 2

[18] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2

[19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[20] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 2

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020. 2

[22] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. In *CVPR*, 2018. 2

[23] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1

[24] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3D single-shot object detection. *ECCV*, 2020. 2

[25] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D instance segmentation. In *CVPR*, 2020. 2

[26] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *ICCV*, 2019. 2, 3

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2

[28] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *ICML*, 2020. 2, 3

[29] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua

Bengio. Learning deep representations by mutual information estimation and maximization. *ICLR*, 2019. 2

[30] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *CVPR*, 2019. 2

[31] Ji Hou, Angela Dai, and Matthias Nießner. RevealNet: Seeing Behind Objects in RGB-D Scans. In *CVPR*, 2020. 2

[32] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer depth cameras for computer vision*, 2013. 1

[33] Haiyong Jiang, Feilong Yan, Jianfei Cai, Jianmin Zheng, and Jun Xiao. End-to-End 3D Point Cloud Instance Segmentation Without Detection. In *CVPR*, 2020. 2

[34] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *CVPR*, 2020. 2, 4, 12

[35] Marcel Körtgen, Marcin Novotni, and Reinhard Klein. 3D shape matching with 3D shape contexts. In *In The 7th Central European Seminar on Computer Graphics*. Citeseer, 2003. 3

[36] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *ICCV*, 2019. 2

[37] Jiaxin Li, Ben M Chen, and Gim Hee Lee. SO-Net: Self-organizing network for point cloud analysis. In *CVPR*, 2018. 2

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[39] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 2

[40] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding. In *CVPR*, 2019. 1

[41] Sanjeev Muralikrishnan, Vladimir G Kim, and Siddhartha Chaudhuri. Tags2Parts: Discovering semantic regions from shape tags. In *CVPR*, 2018. 3

[42] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[43] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3D object detection in point clouds with image votes. In *CVPR*, 2020. 2

[44] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3D object detection in point clouds. *ICCV*, 2019. 2, 7

[45] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3D object detection from RGB-D data. In *CVPR*, 2018. 2

[46] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. *CVPR*, 2017. 2, 7

[47] Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas Guibas. Volumetric and multi-view cnns for object classification on 3D data. In *CVPR*, 2016. 2

[48] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 2

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[50] Aditya Sanghi. Info3D: Representation learning on 3D objects using mutual information maximization and contrastive learning. *ECCV*, 2020. 2

[51] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *NeurIPS*, 2019. 2

[52] Gopal Sharma, Evangelos Kalogerakis, and Subhransu Maji. Learning point embeddings from shape repositories for few-shot segmentation. In *2019 International Conference on 3D Vision (3DV)*, pages 67–75. IEEE, 2019. 3

[53] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGB-D images. *ECCV*, 2012. 1

[54] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *CVPR*, 2015. 2

[55] Shuran Song and Jianxiong Xiao. Sliding shapes for 3D object detection in depth images. In *ECCV*. 2014. 15

[56] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1, 2

[57] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3D point clouds. In *3DV*, 2017. 2

[58] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. KPConv: Flexible and deformable convolution for point clouds. In *CVPR*, 2019. 2

[59] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *ECCV*, 2020. 2

[60] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, 2018. 2

[61] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, 2019. 2, 4

[62] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *ICCV*, 2019. 2

[63] Wenxuan Wu, Zhongang Qi, and Li Fuxin. PointConv: Deep convolutional networks on 3D point clouds. In *CVPR*, 2019. 2

[64] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2

[65] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013. 1

[66] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3D point cloud understanding. *ECCV*, 2020. 2, 3, 4, 5, 7, 8, 12, 15

[67] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *CVPR*, 2018. 3

[68] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, 2020. 3

[69] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3D instance segmentation on point clouds. In *NeurIPS*, 2019. 2, 4

[70] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, 2018. 2

[71] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas Guibas. GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. In *CVPR*, 2019. 2

[72] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J Guibas, and Hao Zhang. AdaCoSeg: Adaptive shape co-segmentation with group consistency loss. In *CVPR*, 2020. 3

Appendix

In this supplemental document, we describe the details of our implementation in Section A. We show more visualizations of our models on semantic segmentation and object detection tasks with extremely scarce data for training in Section B. Detailed per-category results on data-efficient benchmark as well as on full data are showed in Section C.

A. Implementation Details

Data Preprocessing. Following [66], we subsample the partial frames by every 25 frames. We find pairs of frames within each scene by computing their overlaps. In detail, every single frame is transformed to world coordinates. We iterate every pair of frames to calculate how many points are overlapped by 2.5cm threshold. For example, for each point in frame A, if we can find another point in frame B within 2.5cm in the transformed coordinate system (world), then those 2 points are stored as a correspondence pair. When 2 frames have at least 30% overlaps of points, those 2 frames are saved for training. We save and use both the xyz coordinates and rgb color for pre-training.

PointInfoNCE Loss. Here we explain the details of the PointInfoNCE loss (Equation 3 in the main paper).

$$\mathcal{L}_p = -\sum_{(i,j)\in M} \log \frac{\exp(\mathbf{f}_i^1 \cdot \mathbf{f}_j^2/\tau)}{\sum_{(\cdot,k)\in M, k\in par_p(i)} \exp(\mathbf{f}_i^1 \cdot \mathbf{f}_k^2/\tau)}$$

M denotes the set of all the corresponding matches from two frames. Denote the point features from two frames $\mathbf{f}^1$ and $\mathbf{f}^2$ respectively. In this formulation, we use the points that have at least one match as negative, and non-matched points are discarded. For a matched pair $(i,j) \in M$, point feature $\mathbf{f}_i^1$ serves as the query and $\mathbf{f}_j^2$ serves as the positive key. Point feature $\mathbf{f}_k^2$ where $\exists(\cdot,k) \in M$, $k \in par_p(i)$ and $k \neq j$ are used as the set of negative keys. In practice, we sample a subset of matched pairs from M for training.

Active Labelling. We first use our pre-trained network to make a forward pass on all the voxels of each scene in the training data, and save the 96-dim penultimate layer features at each voxel. Then we back-project the features at each voxel to the raw point cloud using nearest neighbour search. We run a k-means clustering algorithm on the features and xyz coordinates of the point cloud on each scene to get k centroids, where k is the number of points we propose to annotator to label. We run k-means for 50 iterations.

Clustering Algorithm in Instance Segmentation. We adapt the code of breadth first search from PointGroup [34]. Clustering only happens in the test time. In the test time, we cluster on points that are shifted by learned directional and distance vectors. Directional and distance vectors are learned by voting-center loss in the training time. We use 3cm-ball as threshold for every point to search its neighbouring points at each iteration. Within the ball, the points are grouped into one instance when they have the same semantic label. We don't use the ScoreNet proposed in PointGroup, so that we don't have additional network for training. We simply average the scores of semantic prediction of the points belonging to the same instance.

B. More Visualizations

We show more visualizations of semantic segmentation and object detection predictions from our model trained with extremely scarce annotations. We show the semantic segmentation on ScanNet validation set with our model trained on 20 labelled points per scene in Figure 9. We also demonstrate the object detection results on ScanNet validation set predicted by our model trained on 1 bounding box annotated per scene in Figure 8.

C. Per-Category Results

In this section, we demonstrate detailed per-category performance as supplement of data-efficient benchmark. Instance segmentation on ScanNet-LA (Limited Scene Annotations, 200 labelled points for training) is showed in Table 7; semantic segmentation of per-category performance on ScanNet-LA is showed in Table 8; object detection on Limited Bounding Boxes Annotations is showed in Table 9.

We further show the detailed per-category performance as supplement of Table. 6 in the main paper on full data. Instance segmentation and semantic segmentation results on S3DIS are showed in Table 10 and Table 11; object detection on SUN-RGBD result is showed in Table 12; instance segmentation and object detection on ScanNet validation set are showed in Table 13 and Table 14.

D. Different Backbones.

We use Sparse Residual U-Net (SR-UNet-34, also used in [12]) as backbone architecture. 3D-MPA also uses a Sparse Residual U-Net backbone, and the performance gap is due to the additional head modules (e.g., Proposal Consolidation) which is orthogonal to our pre-training method. To show our algorithm is generic and agnostic to the specific backbone, we perform experiments with different backbones, including SR-UNet-18A and PointNet++. Models pre-trained with our method yield significant better results; see Tab. 15.

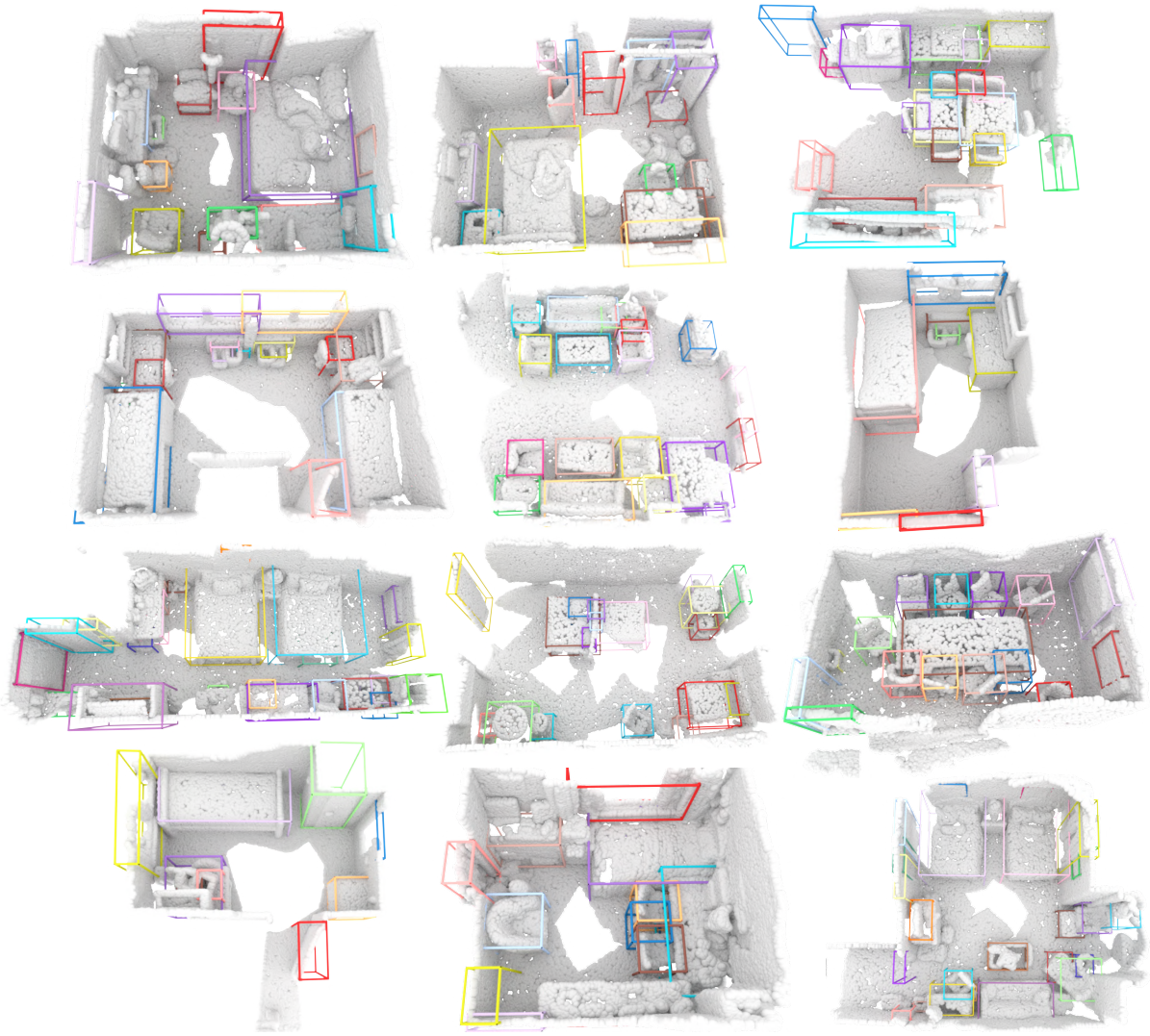

Figure 8: **Object Detection Results (Limited Bounding Box Annotations).** With our pre-trained model as initialization for fine-tuning, our approach generates high-quality detection predictions. Here our model is trained with 1 bounding box annotated per scene.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
Scratch	31.8	72.4	56.0	52.7	**55.9**	36.6	25.3	47.6	14.7	11.3	10.1	**36.4**	34.5	57.5	90.0	33.7	**80.3**	35.8	43.5
PointContrast	39.2	71.2	**63.1**	**71.4**	48.4	36.9	20.5	45.2	18.2	8.1	13.9	32.4	31.5	**64.1**	97.0	42.3	54.9	**40.1**	44.5
Ours	**43.7**	**75.2**	62.9	65.7	50.5	**43.4**	**27.4**	**52.9**	**26.9**	**19.7**	**14.4**	34.4	**39.9**	61.9	**97.4**	**49.4**	75.3	39.0	**48.9**

Table 7: **Instance Segmentation with Limited Point Annotations (ScanNet-LA)**. We use mAP@0.5 as metric and demonstrate per-category performance over 18 classes on data-efficient benchmark (200 labelled points for training per scene).

	wall	floor	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
Scratch	81.6	**96.1**	57.5	79.5	88.1	82.2	67.1	55.9	54.4	76.3	24.3	59.9	52.9	67.9	39.8	55.9	86.9	58.2	82.4	42.1	65.5
PointContrast	83.0	96.0	**61.1**	**79.5**	89.5	81.9	**71.6**	57.1	**57.0**	73.0	22.6	62.0	**58.8**	**69.1**	44.4	63.6	91.5	**59.4**	**85.2**	48.5	67.8
Ours	**84.0**	95.9	60.2	79.0	**89.5**	**83.8**	69.6	**60.2**	56.7	**80.6**	**26.1**	**63.9**	55.6	63.5	**45.1**	**63.7**	**91.9**	56.9	84.7	**52.6**	**68.2**

Table 8: **Semantic Segmentation with Limited Point Annotations (ScanNet-LA)**. We evaluate mean IoU over 20 classes on data-efficient benchmark (200 labelled points per scene for training).

Figure 9: **Semantic Segmentation Results (ScanNet-LA).** With our pre-trained model as initialization for fine-tuning, together with an active labeling process, our approach generates high-quality semantic segmentation predictions. Here our model is fine-tuned with 20 labeled points per scene.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
Scratch	5.4	**71.9**	64.2	59.8	37.3	17.1	6.8	32.3	0.4	**16.8**	**33.7**	**26.7**	29.2	3.3	**87.9**	20.6	70.2	17.9	33.4
PointContrast	10.3	71.8	**71.1**	61.2	43.1	**21.6**	**9.4**	34.7	**2.3**	6.8	25.7	21.2	32.6	**17.1**	84.1	20.4	74.6	20.0	34.9
Ours	**10.9**	69.5	70.2	**62.1**	44.3	18.2	9.0	**39.8**	1.0	9.2	32.9	25.3	**35.6**	10.3	78.9	**26.5**	**81.0**	**21.2**	**35.9**

Table 9: **Object Detection with Limited Bounding Box Annotations** . We evaluate mAP@0.5 over 18 classes on data-efficient benchmark (7 annotated bounding boxes for training per scene).

	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	**avg**
Scratch	46.8	89.5	72.5	0.0	**38.2**	72.5	89.5	88.0	39.3	34.7	72.7	85.7	59.3
PointContrast	66.0	**93.0**	73.0	0.0	18.6	72.8	88.3	**91.4**	42.3	**29.5**	63.6	88.0	60.5
Ours	**74.4**	88.0	**76.5**	**0.0**	32.4	**74.6**	**96.4**	91.0	**45.0**	28.8	**63.6**	**90.5**	**63.4**

Table 10: **Instance Segmentation on Stanford Area 5 Test [2]**. We evaluate mAP@0.5 over 12 classes.

	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter	**avg**
Scratch	91.5	98.6	84.1	0.0	33.0	56.9	63.9	90.1	81.7	72.5	76.5	77.9	59.6	68.2
PointContrast	93.3	**98.7**	85.6	**0.1**	**45.9**	54.4	67.9	91.6	80.1	**74.7**	78.2	81.5	62.3	70.3
Ours	**95.1**	98.4	**86.3**	0.0	40.7	**60.8**	**85.2**	**91.8**	**81.9**	73.9	**78.9**	**82.8**	**62.4**	**72.2**

Table 11: **Semantic Segmentation on Stanford Area 5 Test [2]**. We evaluate mIoU over 13 classes.

	bed	table	sofa	chair	toilet	desk	dresser	night stand	book	bathtub	**avg**
Scratch	47.8	19.6	48.1	54.6	60.0	6.3	15.8	27.3	5.4	32.1	31.7
PointContrast [66]	50.5	19.4	51.8	**54.9**	57.4	**7.5**	**16.2**	37.0	5.9	**47.6**	34.8
Ours	**55.3**	**20.3**	**53.8**	53.6	**65.9**	6.1	15.5	**38.0**	**9.1**	46.5	**36.4**

Table 12: **Object Detection on SUN RGB-D [55]**. We use mAP@0.5 as metric and show per-category AP@0.5 over 10 classes.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
Scratch	49.0	70.0	87.4	66.5	71.1	47.4	39.6	53.0	30.8	**32.8**	30.8	41.7	48.6	60.1	**99.9**	68.4	75.3	52.4	56.9
PointContrast	49.4	72.1	87.2	**71.7**	67.0	**49.0**	40.7	**57.8**	**35.6**	24.0	30.2	**49.9**	**53.0**	65.2	98.3	61.7	80.5	50.8	58.0
Ours	**50.8**	**74.1**	**88.7**	61.4	**67.2**	48.0	**42.0**	57.0	33.8	32.5	**42.9**	47.4	49.5	**68.9**	98.2	**71.3**	**80.5**	**54.7**	**59.4**

Table 13: **Instance Segmentation on ScanNetV2 [13] Validation Set**. We evaluate the mean average precision with IoU threshold of 0.5 over 18 classes.

	cab	bed	chair	sofa	tabl	door	wind	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	**avg**
Scratch	9.9	70.5	70.0	60.5	43.4	**21.8**	10.5	**33.3**	0.8	15.4	33.3	26.6	**39.3**	9.7	74.7	23.7	75.8	18.1	35.4
PointContrast	13.1	**74.7**	**75.4**	**61.3**	44.8	19.8	12.9	32.0	0.9	**21.9**	31.9	**27.0**	32.6	17.5	**87.4**	23.2	80.8	**26.7**	38.0
Ours	**15.1**	74.3	71.9	60.2	**46.4**	21.2	**15.0**	32.5	**1.1**	9.4	**36.6**	21.3	37.3	**47.5**	84.3	**26.2**	**86.8**	21.2	**39.3**

Table 14: **Object Detection on ScanNetV2 Validation Set**. We use mAP@0.5 as metric and show per-category performance over 18 classes.

	Task	Dataset	Backone	mAP@0.5
scratch	ins	S3DIS	SR-UNet-18A	58.6
ours (pre-trained)	ins	S3DIS	SR-UNet-18A	**62.8**
scratch	det	ScanNet	PointNet++	33.5
ours (pre-trained)	det	ScanNet	PointNet++	**39.2**

Table 15: Pre-training with different backbones; 100% of available train data is used; we would expect larger deltas with smaller train set.

E. ScanNet Benchmark

We report validation results to directly compare with PointContrast which also evaluates on the val set. Additionally, we submitted our model to the ScanNet Benchmark (test set); see Tab. 16. Our method significantly outperforms 3D-MPA, despite not leveraging the special 3D-MPA proposal module.

	AP	AP@50	AP@25
3D-MPA [15]	35.5	61.1	73.7
ours (pre-trained)	**40.5**	**64.8**	**79.1**

Table 16: ScanNet **test** set: similar to S3DIS, we outperform 3D-MPA.