Self-Supervised Pretraining of 3D Features on *any* Point-Cloud

Zaiwei Zhang[1,2*] Rohit Girdhar[1] Armand Joulin[1] Ishan Misra[1]
[1]Facebook AI Research [2]The University of Texas at Austin

Abstract

Pretraining on large labeled datasets is a prerequisite to achieve good performance in many computer vision tasks like image recognition, video understanding etc. *However, pretraining is not widely used for 3D recognition tasks where state-of-the-art methods train models from scratch. A primary reason is the lack of large annotated datasets because 3D data labelling is time-consuming. Recent work shows that self-supervised learning is useful to pretrain models in 3D but requires multi-view data and point correspondences. We present a simple self-supervised pretraining method that can work with single-view depth scans acquired by varied sensors, without 3D registration and point correspondences. We pretrain standard point cloud and voxel based model architectures, and show that joint pretraining further improves performance. We evaluate our models on 9 benchmarks for object detection, semantic segmentation, and object classification, where they achieve state-of-the-art results. Most notably, we set a new state-of-the-art for object detection on ScanNet (69.0% mAP) and SUNRGBD (63.5% mAP). Our pretrained models are label efficient and improve performance for classes with few examples.*

1. Introduction

Pretraining visual features on large labeled datasets is a pre-requisite to achieve good performance when access to annotations is limited [27, 46, 52, 87]. More recently, self-supervised pretraining has become a popular alternative to supervised pretraining especially for tasks where annotations are time-consuming, such as detection and segmentation in images [9, 36, 37, 56, 93] or tracking in videos [41]. In 3D vision too, annotations are difficult to acquire. Labeling a 3D scene composed of thousands of 3D points is time-consuming and can take around 22 minutes per scene [18].

This cumbersome annotation process results in a lack of large annotated 3D datasets. However, acquiring 3D data in the form of single-view depth maps has become easier than ever due to the advent of consumer grade depth sensors, *e.g.*, in phones [24, 73, 83]. While these depth maps can be leveraged to pretrain self-supervised 3D features, there is surprisingly little work that can be applied. Recent work [105] applies self-supervised pretraining to 3D models but uses multi-view depth scans with point correspondences. Since 3D sensors only acquire single-view depth scans, multi-view depth scans and point correspondences are typically obtained via 3D reconstruction. Unfortunately, even with good sensors, 3D reconstruction can fail easily for a variety of reasons such as non-static environments, fast camera motion or odometry drift [16].

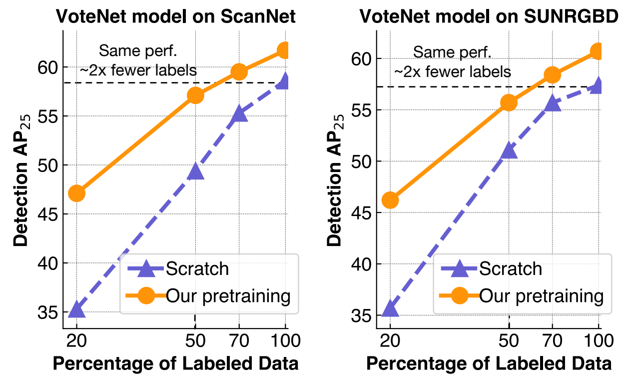

Figure 1: Label-efficiency of our self-supervised pretraining. We finetune detection models from scratch or using our pretraining as initialization. Our pretraining which uses unlabeled single-view 3D data, outperforms training from scratch, and achieves the same detection performance with about half the detection labels.

In this paper, we introduce a simple contrastive framework, **DepthContrast**, to representations from single-view depth scans. From a practical perspective, self-supervised learning from single-view depth scans is more broadly applicable for 3D data. It is also an interesting scientific question whether just using single-view information can provide benefits for self-supervised learning in 3D. Our approach is based on the Instance Discrimination method by Wu *et al*. [103] applied to depth maps. We side-step the need of registered point clouds or correspondences, by considering each depth map as an instance and discriminating between them, even if they come from the same scene. Since different 3D applications require different 3D scene representa-

[*]Work done during an internship at Facebook.

tions such as voxels for segmentation [17], point clouds for detection [64], we use our method for both voxels and point clouds. We jointly learn features by considering voxels and point clouds of the same 3D scene as data augmentations that are processed with their associated networks [93].

Our contributions can be summarized as follows:

- We show that single-view 3D depth scans can be used for self-supervised learning.
- Our single-view representations perform comparably, or in some settings better than their multi-view counterparts, showing that single-view depth scans are indeed powerful for learning features.
- Our method is applicable across different model architectures, indoor/outdoor 3D data, single/multi-view 3D data. We also show that it can be used to pretrain high capacity 3D architectures which otherwise overfit on tasks like detection and segmentation.
- We show that *joint* training of different input representations like points and voxels is important for learning good representations, and a naive application of contrastive learning may not yield good results.
- We show performance improvements over *nine* downstream tasks, and set a new state-of-the-art for *two* object detection tasks (ScanNet and SUNRGBD). Our models are efficient few-shot learners.

2. Related Work

Our method builds on the work from the self-supervised learning literature, with 3D data as an application. In this section, we give an overview of the recent advances in both self-supervision and 3D representations.

Self-supervised learning for images. Self-supervised learning is a well studied problem in machine learning and computer vision [53, 60, 69, 72, 95]. There are many classes of methods for learning representations - clustering [7, 8, 43], GANs [20, 55], pretext tasks [19, 59, 97] *etc*. Recent advances [9, 13, 30, 36, 37, 47, 56, 94] have shown that self-supervised pretraining is a viable alternative to supervised pretraining for 2D recognition tasks. Our work builds upon contrastive learning [34, 61] where models are trained to discriminate between each instance [21] with no explicit classifier [103]. These instance discrimination methods can be extended to multiple modalities [57, 62, 93]. Our method extends the work of Wu *et al*. [103] to multiple 3D input formats following Tian et al. [93] using a momentum encoder [36] instead of a memory bank.

Self-supervised learning for 3D data. Most methods on self-supervised learning focus on single 3D object representation with different applications to reconstruction, classification or part segmentation [1, 2, 25, 33, 35, 44, 49, 74, 99, 110]. Recently, Xie *et al*. [105] proposed a self-supervised method to build representations of scene level point clouds. Their method relies on the complete 3D reconstruction of a scene with point-wise correspondences between the different views of a point cloud. These point-wise correspondences requires post-processing the data by registering the different depth maps into a single 3D scene. Their method can only be applied to static scenes that have been registered, which greatly limits the applications of their work. We show a simple self-supervised method that learns state-of-the-art representations from *single-view* 3D data, and can also be applied to multi-view data.

Representations of 3D scenes. There are multiple ways to represent 3D information in different vectorized forms such as point-clouds, voxels or meshes. Point-cloud based models [66, 68] are widely used in classification and segmentation tasks [6, 40, 45, 66, 68, 91, 98, 100, 106, 107], 3D reconstruction [23, 90, 110] and 3D object detection [63, 64, 77, 98, 112, 112, 114]. Since many 3D sensors acquire data in terms of 3D points, point clouds are a convenient input for deep networks. However, since using convolution operations on point-clouds directly is difficult [31, 66], voxelized data is another popular input representation. 3D convolutional models [3, 17, 28, 31, 38, 50, 70, 85, 92] are widely used in 3D scene understanding [32, 82, 109, 116]. There are also efforts to combine different 3D input representations [29, 76, 96, 113–115]. In this work, we propose to jointly pretrain two architectures for points and voxels, that are PointNet++ [68] for points and Sparse Convolution based U-Net [17] for voxels.

3D transfer tasks and datasets. We use shape classification, scene segmentation, and object detection as the recognition tasks for transfer learning. Shape classification techniques [11, 54, 66–68, 86] are widely evaluated on the ModelNet [102] dataset, which we use. It contains synthetic 3D data and each sample contains exactly one object. We also evaluate on complete 3D scenes using the more general 3D scene understanding task. Scene-centric datasets can be broadly divided into indoor scens [5, 10, 18, 39, 58, 63, 75, 81, 84, 104], and outdoor (self-driving focussed) scenes [26, 71, 88]. We use these datasets and evaluate the performance of our methods on the indoor detection [12, 22, 64, 65, 114], scene segmentation [17, 68, 92, 101, 108], and outdoor detection tasks [15, 48, 76–78, 109, 111].

3. DepthContrast

We present our approach to unsupervised 3D representation learning. DepthContrast can learn from either unprocessed single-view or multi-view depth maps. Our method, illustrated in Fig 2, is based on the instance discrimination framework from Wu *et al*. [103] with a momentum encoder [36]. We also show an extension of DepthContrast built upon [93] that learns representations across 3D input formats like points and voxels, and across 3D architectures.

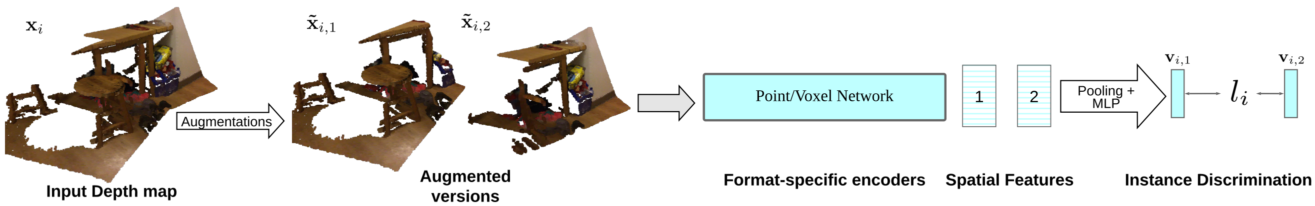

Figure 2: Approach Overview. We propose DepthContrast - a simple 3D representation learning method that uses large amounts of unprocessed single/multi-view depth maps. Given a depth map we construct two augmented versions using data augmentation and represent them with selected input formats (point coordinates or voxels). We use format-specific encoders to get spatial features which are pooled and projected to obtain global features $\mathbf{v}$. The global features are used to setup an instance discrimination task and pretrain the encoders.

3.1. Instance Discrimination

Given a dataset $\mathcal{D} = \{\mathbf{X}\}_{i=1}^N$ containing N samples $\mathbf{X}$, we wish to learn a function $g(\mathbf{X})$ that produces useful representations $\mathbf{v} = g(\mathbf{X})$ of the input sample. As shown in Fig 2, our method uses 3D data where $\mathbf{X}$ can be represented by point coordinates or voxels[1]. We apply a data augmentation t sampled randomly from a large set of augmentations $\mathcal{T}$, to obtain an augmented sample $\widetilde{\mathbf{X}} = t(\mathbf{X})$. The augmented sample is input to a deep network g that extracts unit-norm global features $\mathbf{v} = g(\widetilde{\mathbf{X}})$ by pooling over the 3D spatial coordinates. We setup an instance discrimination problem where the features $\mathbf{v}_{i,1}$ and $\mathbf{v}_{i,2}$ obtained from two data augmented versions of sample i must be similar to each other, and different from features $\mathbf{v}_j$ obtained using K other (negative) samples j in the dataset. We use a contrastive loss [34, 61, 80] to achieve this goal:

$$l_i = -\log \frac{\exp(\mathbf{v}_{i,1}^\top \mathbf{v}_{i,2}/\tau)}{\exp(\mathbf{v}_{i,1}^\top \mathbf{v}_{i,2}/\tau) + \sum_{j \neq i}^K \exp(\mathbf{v}_{i,1}^\top \mathbf{v}_j/\tau)}, \quad (1)$$

where τ is the temperature that controls the smoothness of the softmax distribution. This loss encourages features from different augmentations of the same scene to be similar, while being dissimilar to features of other scenes. Thus, it learns features that focus on discriminative regions of a scene that make it different from other scenes in the dataset.

Minimal assumptions on input data. Our method makes minimal assumptions about the input $\mathbf{X}$, *i.e.*, it is an unprocessed single-view depth map. It does not require careful sampling of overlapping multi-view 3D inputs [105] or object centric depth maps [35, 44]. These minimal assumptions enable us to learn from large scale single-view 3D depth maps in § 4 and outdoor 3D depth maps obtained from different sensors without relying on 3D calibration in § 5.3.

Momentum encoder. As using a large number of negatives is important for contrastive learning [13, 36, 56, 103], we use the method of He *et al.* [36] where the features of the other augmentation $\mathbf{v}_{i,2}$ and negative samples $\mathbf{v}_j$ in Eq 1 are obtained using a momentum encoder and a queue respectively. This allows us to use a large number K of negative samples without increasing the training batch size.

[1]Points in a depth map are a set, but for simplicity we denote them as a matrix. Our method does not rely on any specific ordering of the points.

3.2. Extension to Multiple 3D Input Formats

Multiple input formats are commonly used to represent 3D data - point clouds, voxels, meshes *etc*. Different input formats can be easily transfered from one to another and have their specific deep learning architectures and applications. Our self-supervised method can be naturally extended to accommodate these input formats and architectures. For each input format f, we denote the corresponding input sample as $\mathbf{X}^f$, the format-specific encoder network as g^f, and the extracted feature as $\mathbf{v}^f$. Extending Eq 1, we can minimize a single objective that performs instance discrimination within and across input formats a, b:

$$l_i^{ab} = -\log \frac{\exp(\mathbf{v}_{i,1}^{a\top} \mathbf{v}_{i,2}^b/\tau)}{\exp(\mathbf{v}_{i,1}^{a\top} \mathbf{v}_{i,2}^b/\tau) + \sum_{j \neq i}^K \exp(\mathbf{v}_{i,1}^{a\top} \mathbf{v}_j^b/\tau)}. \quad (2)$$

When the input formats a, b are identical, this objective reduces to the *within format* loss of Eq 1, and when $a \neq b$ this objective aligns the feature representations $\mathbf{v}^f = g^f(\mathbf{X}^f)$ obtained *across formats* f using different network architectures g^f. As illustrated in Fig 3, we use two popular input formats - point clouds and voxels, and train these format-specific models with a single joint loss function

$$L_i = \underbrace{l_i^{ab} + l_i^{ba}}_{\text{across format}} + \underbrace{l_i^{aa} + l_i^{bb}}_{\text{within format}}. \quad (3)$$

Similar techniques have been explored in the context of different modalities of data, *e.g.*, color and grayscale im-

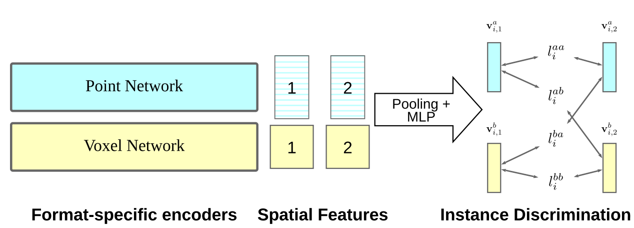

Figure 3: Multiple 3D Input Formats We extend our approach to joint training with point and voxel input formats.

ages [93], audio and video [57, 62] *etc*. While these methods use different modalities, our extension uses the *same 3D data* and only changes the input format.

3.3. Model Architecture

We describe the model architecture used for our input format-specific encoders. Both encoders operate on the same augmented input 3D data, and differ only in the way the input is represented. We provide the full, layer-wise architecture details in the supplemental material.

Point input. We use PointNet++ [64] as the backbone network which takes XYZ coordinates as input. Our network takes in 20K points, and produces C dimensional per-point features for 1024 points after aggregation. We obtain the scene level 256 dimensional feature $\mathbf{v}$ in Eq 2 by global max pooling to these last layer features, followed by a two layer MLP as in [13] and L2 normalization.

Voxel input. We use a sparse convolution U-Net model [17] as the backbone for the voxel 3D input. The network takes a 3D occupancy grid and corresponding RGB values as the input representation of the 3D data. We use a voxel size of $5cm$ to voxelize the input data following [17].To obtain the scene level 256 dimensional feature $\mathbf{v}$ (Eq 2), similar to the point input, we apply global max-pooling to the last layer feature, followed by a two layer MLP and L2 normalization.

3.4. Data Augmentation for 3D

Data augmentation is as an essential component of our framework. We first adopt standard pointcloud data argumentation methods proposed in [64], which are random point up/down sampling, random flip in xy axis, and random rotation. However, after adding these methods, it is still easy for the network to distinguish different training instances. Thus, we add two new data augmentation methods: random cuboid and random drop patches. Inspired by the random crop in 2D images [89], we define a random cuboid augmentation that extracts random cuboids from the input point cloud. Cuboids are sampled using a random scale $[0.5, 1.0]$ of the original scene, and a random aspect ratio $[0.75, 1.0]$. We also drop (erase) cuboids to force the network learn local geometric features. The dropped cuboid is randomly cropped with 0.2 of the scene scale. The performance boost from each augmentation is analyzed in § 5. For voxelized inputs, in addition to all the point augmentations, we use the augmentations from [17].

3.5. Implementation Details

We use 130K negatives for contrastive learning in Eq 3 and a momentum of 0.9 for the momentum encoder following [36]. As noted in § 3.3, we follow Chen *et al*. [13] and use an additional non-linear projection and L2 normalization to obtain the features $\mathbf{v}$. The features $\mathbf{v}$ are 128 dimensional and we use a temperature value of 0.1 while computing the non-parametric softmax in Eq 1. We use a standard SGD optimizer with momentum 0.9, cosine learning rate scheduler [51] starting from 0.12 to 0.00012 and train the model for 1000 epochs with a batch size of 1024.

Dataset	**Stats**	**Task**	**Gain of DepthContrast**
Self-supervised Pretraining			
ScanNet-vid [18]	190K single-view depth maps (Indoor)		
Redwood-vid [16]	370K single-view depth maps (Indoor/Outdoor)		
Transfer tasks			
ScanNet [18]	1.2K train, 312 val (Indoor)	Det.	**+3.6% mAP**
		Seg.	**+0.9% mIOU†**
SUNRGBD [81]	5.2K train, 5K val (Indoor)	Det	**+3.3% mAP**
S3DIS [4]	199 train, 67 val (Indoor)	Det	**+12.1% mAP**
		Seg.	**+2.4% mIOU**
Synthia [71]	19.8K train, 1.8K val (Synth.)	Seg.	**+2.4% mIOU**
Matterport3D [10]	1.4K train, 232 val (Indoor)	Det.	**+3.9% mAP**
Shape Classification [102]	9.8K train, 2.4K val (Synth.)	Cls.	**+3.1% Acc†**

Det.: Object Detection, Seg: Semantic Segmentation
Cls: Classification, Synth.: Synthetic, †Results in supplemental.

Table 1: Pretraining datasets and transfer tasks used in this paper. We use two different pretraining datasets without post-processing like 3D registration, camera calibration. We use 8 different transfer tasks for evaluation where our DepthContrast pretraining gives consistent gains (last column) over scratch pretraining. Additonally, we show evaluation results on LiDAR data in § 5.3.

4. Experiments

We evaluate DepthContrast pretraining by transfer learning, *i.e*., fine-tuning the learned representation on downstream tasks and datasets. As Table 1 shows, we use a diverse set of 3D understanding tasks like object classification, semantic segmentation, and object detection. We first study a single input 3D format and a single network architecture in § 4.1. We show DepthContrast's performance on multiple downstream tasks, even when compared to multi-view methods [105]; further improvements by scaling amount of pretraining data and model capacity; as well as its benefits in few-shot downstream tasks with limited labeled data. Finally, in § 4.2, we show the benefits of our pre-training across different 3D input formats.

Pretraining Details. We use single-view depth map videos from the popular ScanNet [18] dataset and term it as ScanNet-vid. ScanNet-vid contains about 2.5 million RGB-D scans for more than 1500 indoor scenes. Following the train/val split from [64], we extract around 190K RGB-D scans (2FPS) from about 1200 video sequences in the train set. We do not use camera calibration or 3D registration methods and operate directly on single-view depth maps. We use our data augmentation described in § 3.4 and use the training objectives from § 3. Additional details are provided in § 3.5 and the supplemental material.

Downstream Tasks. We evaluate our pretrained model by transfer learning and finetune it on different downstream datasets and tasks summarized in Table 1. We use diverse downstream datasets - full scenes/object cen-

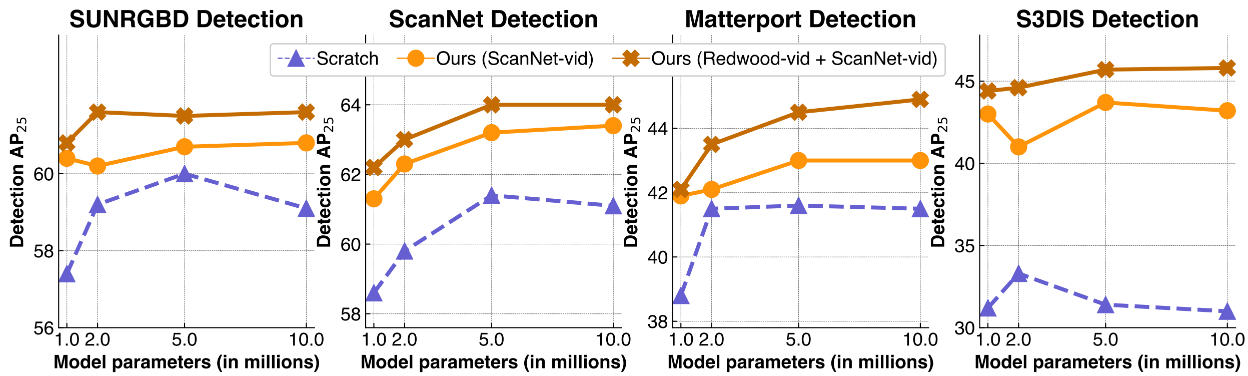

Figure 4: Scaling the model size and pretraining data. We increase the model capacity of the PointNet++ model by increasing the width by $\{2\times, 3\times, 4\times\}$. When training from scratch, increasing the model capacity increases the performance but ultimately leads to overfitting. Overfitting is more pronounced on small datasets like S3DIS. Our DepthContrast pretraining on ScanNet-vid improves the performance for larger models and reduces overfitting. We increase the pretraining data by combining the readily available single-view depth maps from ScanNet-vid and Redwood-vid. DepthContrast's performance improves significantly when using both large data and large models.

Initialization	ScanNet	SUNRGBD	Matterport3D	S3DIS
Scratch	58.6	57.4	38.8	31.2
Supervised	-	59.1 (+1.7)	41.7 (+2.9)	**48.5** (+17.3)
DepthContrast (Ours)	**61.3** (+2.7)	**60.4** (+3.0)	**41.9** (+3.1)	43.3 (+12.1)
PointContrast [105]	59.2(+2.5)	57.5(+1.9)	-	-

Table 2: Detection AP_{25} using VoteNet [64]. We evaluate different pretrained models - random initialization, supervised VoteNet on ScanNet, our self-supervised DepthContrast using the point input format, and self-supervised with PointContrast. We provide in green the improvements over a detector trained from scratch. Note that PointContrast uses a UNet backbone. DepthContrast outperforms the scratch model on all benchmarks and is better than the detection-specific supervised pretraining on two datasets.

tric; single/multi-view; real/synthetic; indoor/outdoor. On these diverse datasets, we use three major tasks, which are classification, semantic segmentation, and object detection. These tasks test different aspects of the pretrained model - while object detection and semantic segmentation use local features, classification is performed on global features.

4.1. Pretraining with Point Input Format

Setup. We pretrain a PointNet++ model using the instance discrimination objective in Eq 1 on the single-view depth maps from ScanNet-vid. We study the transfer performance of the pretrained model on object detection using the VoteNet [64] framework that uses a PointNet++ backbone.

Baselines. *Scratch* - Training from scratch or random initialization is standard practice in VoteNet [64] and serves as a baseline for comparing other pretraining methods. *Supervised* - We introduce a supervised pretraining baseline by pretraining a PointNet++ backbone on the ScanNet detection task. As the supervised baseline is pretrained specifically on object detection, it serves as a strong baseline. *PointContrast* - We compare with a PointContrast [105] pretrained model that uses strictly more information (multi-view) than our model (single-view) and serves as an important upper bound. We note that the architecture of this model is different, and as reported in their work [105], PointContrast performs poorly with single-view data.

In Table 2 we report the detection results by finetuning the VoteNet model with different backbone initializations. We use the implementation of [64] for finetuning and report the detection performance using the mean Average Precision at IoU=0.25 (AP_{25}) metric. Scratch training provides competitive results on the larger detection datasets like ScanNet and SUNRGBD [42, 79, 81, 104], however, its performance on the smaller S3DIS dataset is low. In comparison, supervised pretraining provides large gains in the detection performance across all datasets. DepthContrast outperforms training from scratch on all the four datasets, and **improves performance by 12.1% mAP** on the small S3DIS dataset that has only 200 labeled training samples. We further analyze label efficiency of our model in § 4.1.4. Interestingly, despite using no labels during pretraining, DepthContrast is better than the detection-specific supervised pretraining for two datasets (SUNRGBD and Matterport3D). Compared to PointContrast, our model achieves a similar gain over the scratch baseline. This shows that our single-view DepthContrast can learn representations that are at par with a multi-view method for object detection.

4.1.1 Training Higher Capacity Models

We now apply DepthContrast to higher capacity models. Following standard practice in 2D self-supervised learning [47], we increase the capacity of PointNet++ model by multiplying the channel width of all the layers by $\{2, 3, 4\}$. We pretrain all models on the ScanNet-vid dataset and measure their transfer performance in Fig 4. Training large models from scratch provides some benefit, but quickly leads to reduced or plateauing performance. We observe

Method	ScanNet		SUNRGBD	
	AP_{25}	AP_{50}	AP_{25}	AP_{50}
F-PointNet [65]	54.0	-	-	-
VoteNet [64]	58.6	33.5	57.7	32.9
H3DNet [114]	67.2	48.1	60.1	39.0
HGNet [12]	61.3	34.4	61.6	34.4
3D-MPA [22]	64.2	49.2	-	-
PointContrast (VoteNet) [105]	59.2	38.0	57.5	34.8
DepthContrast (VoteNet)	64.0	42.9	61.6	35.5
DepthContrast (H3DNet)	**69.0**	**50.0**	**63.5**	**43.4**

Table 3: Transfer using state-of-the-art detection frameworks. We use our pretrained model (PointNet++ 3× on Redwood-vid +ScanNet-vid) and transfer it using two state-of-the-art detection frameworks - H3DNet [114] and VoteNet [64]. Our DepthContrast pretraining outperforms all prior work and sets a new state-of-the-art on both ScanNet and SUNRGBD detection datasets.

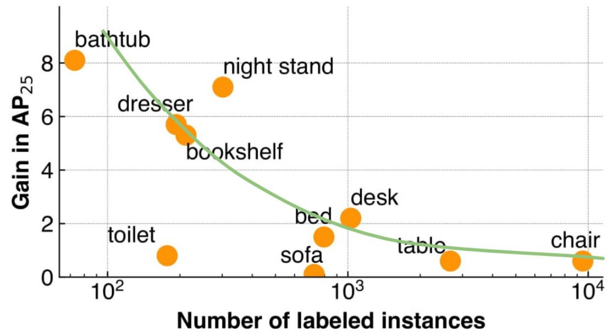

Figure 5: Pretraining benefits long tail classes. We analyze the gain of our pretraining across different classes for SUNRGBD object detection. The training data has a long tailed distribution where the least frequent classes occur 50× less than the most frequent classes. Our pretraining improves performance for classes with fewer labeled instances by $4-8\%$. (Trending line in green.)

overfitting on the small datasets like S3DIS where increasing the model capacity does not improve performance. However, our self-supervised pretraining on ScanNet-vid reduces this overfitting and performance improves or stays the same for larger models. This suggests that *pretraining is crucial* for training large 3D detection models and DepthContrast can provide an easy way to train such models.

4.1.2 Using More Single-view Pretraining Data

We increase the pretraining data by using readily available single-view 3D data from the Redwood-vid dataset [16]. Redwood-vid contains over 23 million depth scans from RGB-D videos taken in both indoor and outdoor settings. As this dataset is extremely large, we use a subset of 2500 video sequences consisting of 10 categories and extract 370K RGB-D scans. Since the Redwood-vid dataset *does not contain* camera extrinsic parameters, multi-view methods like PointContrast [105] cannot be used to such dataset.

Combining the Redwood-vid and ScanNet-vid datasets allows us to triple our pretraining data. We pretrain all models on this combined dataset and report their performance (AP_{25}) in Fig 4. DepthContrast's performance improves with both model capacity and number of pretraining samples across all four detection datasets. The higher capacity models show a larger improvement in performance particularly on the smaller S3DIS dataset. These results highlight that DepthContrast can leverage large amounts of readily available single-view data to train high capacity 3D models. Compared to multi-view methods [105], this makes DepthContrast more broadly applicable.

4.1.3 State-of-the-art Detection Frameworks

We use two state-of-the-art detection frameworks - H3DNet [114] and VoteNet [64] and study the benefit of using our pretrained model. We use our PointNet++ 3× model pretrained on the combined Redwood-vid and ScanNet-vid dataset and transfer it using these detection frameworks. The detection results in Table 3 show that our pretrained model achieves state-of-the-art performance on SUNRGBD and ScanNet. In particular, as the gains are larger on stricter mAP at IoU=0.5, our pretrained models result in detection models that are better at localization.

4.1.4 Label Efficiency of Pretrained Models

Pretraining allows models to be finetuned with small amount of labeled data. In Table 2, we observe that small labeled datasets benefit more from pretraining. We study the label efficiency of DepthContrast pretrained models by varying the amount of labeled data used for finetuning. While varying the data, we draw 3 independent samples and report average results. We use the PointNet++ models pretrained on ScanNet-vid (§ 4.1) and report the detection performance in Fig 1. DepthContrast pretraining provides large gains in performance at every setting. On both the ScanNet and SUNRGBD datasets, our model with just 50% *samples gets the same performance* as training from scratch with the full dataset. When using 20% samples for finetuning, our pretrained models provide a **gain of over 10% mAP**. This shows that our pretraining is label efficient and can improve performance especially on tasks with limited supervision.

Does pretraining benefit tail classes? 3D detection datasets like SUNRGBD and ScanNet exhibit a long tailed distribution where many 'tail' classes have few training instances. In SUNRGBD, the 'tail' classes like `bathtub`, `toilet`, `dresser` have less than 200 training instances, while classes like `chair` have over 9000 instances. Fig 5 shows the gain of our pretrained model over the scratch model across object classes on SUNRGBD. Our pretraining improves the performance of classes with fewer instances, *i.e.*, the tail classes, by $4-8\%$ AP. This suggests that DepthContrast pretraining is especially effective in few-shot set-

Loss	Point Transfer		Voxel Transfer	
	SUNRGBD	ScanNet	S3DIS	Synthia
Scratch	57.4	58.6	68.2	78.9
Within Format only	60.4 (+3.0)	61.3 (+1.7)	66.5 (-2.7)	80.1 (+1.2)
Across format only	60.0 (+2.6)	61.1 (+2.5)	69.9 (+1.7)	81.2 (+2.3)
Both (Ours)	**60.7** (+3.3)	**62.2** (+3.6)	**70.6** (+2.4)	**81.3** (+2.4)
PointContrast [105]	59.2(+2.5)	57.5(+1.9)	**70.9**(+2.7)	**83.1**(+3.3)

Table 4: Multiple input formats. We study the importance of training 3D representations jointly using multiple input formats - points and voxels. We vary the within format and across format loss terms in Eq 3. We report detection mAP@0.25 on the point transfer tasks and segmentation mIOU for the voxel transfer tasks. We observe that performing instance discrimination across the input formats (third row) greatly improves over the within format loss term. Note that PointContrast is trained with ScanNet multi-view scans.

tings and can partially address the long tailed label distributions of current 3D scene understanding benchmarks.

4.2. Pretraining with Multiple Input Formats

We pretrain DepthContrast using both the point and voxel input formats and use two format-specific encoders - PointNet++ for points and UNet for voxels.

Baselines. As explained in Eq 3, when using multiple 3D input formats, we can define two loss terms - a within format loss and an across format loss. To analyze which loss terms matter for pretraining, we consider three variants - *(1) Within format* which independently trains format-specific models for each input format and is a straightforward application of instance discrimination to 3D; *(2) Across format* which trains the format-specific models jointly using the second term of Eq 3; *(3) Ours* which trains the format-specific models jointly using our combined loss function. *PointContrast* - Similar to § 4.1, we use a pretrained PointContrast [105] UNet model trained with multi-view data. This model is trained with multi-view point correspondences to enable it to learn better point features. Since PointContrast uses strictly more information than our single-view method, it serves as an important upper bound.

Setup. We evaluate the pretrained models by transfer learning. As in § 4.1, we finetune the pretrained point input format PointNet++ models on SUNRGBD and ScanNet detection using VoteNet. We finetune the voxel UNet models on segmentation using the framework from Spatio-Temporal Segmentation [17] which uses a UNet backbone network. The results are summarized in Table 4.

Compared to training from scratch, the within format pretraining only provides a benefit for the point input format PointNet++ models. For the voxel models, this pretraining does not improve consistently over training from scratch, which is in line with observations from recent work [105]. This shows that a naive application of instance discrimination to 3D representation learning *may not* yield good pretrained models. The across format loss improves performance for both the point and voxel models, suggesting the

Task	VoteNet [64]	+Rand. Cuboid	+Rand. Drop
Shape Classification Linear (Accuracy)	80.6	**85.4**	85.0
SUNRGBD Detection (mAP)	58.6	59.5	**60.7**

Table 5: Data augmentation. We vary the data augmentation used for pretraining DepthContrast point models and report their transfer performance. The standard data augmentation used in supervised learning (VoteNet) is not sufficient to learn good self-supervised models. Our proposed Random Cuboid and Random Drop augmentations improve performance.

benefit of using multiple input formats. Our proposed joint loss provides the best transfer performance. The gains are particularly significant on the voxel format model which improves by 4% **over the within format loss**. In the supplemental material, we show that this benefit of joint training over the within format loss also holds across different pretraining data and architectures.

Compared to the multi-view PointContrast upper bound [105], our results on the voxel transfer task are slightly worse. PointContrast uses multi-view point correspondences to enforce point-level supervision during pretraining. This enables their model to learn point features that are more suitable for point prediction tasks like segmentation. However, despite not relying on multi-view information, DepthContrast pretraining still provides competitive performance. We believe these results are encouraging given the broad applicability of DepthContrast to vast amounts of single-view data captured by modern sensors. We note that our UNet architecture is different from [105] since their architecture underfit on our pretraining task.

5. Analysis

In this section we present a series of experiments designed to understand DepthContrast better. We first pretrain point format (PointNet++) models on the ScanNet-vid dataset following the settings from § 4.1. We use two transfer tasks for evaluation - (1) object detection on SUNRGBD using VoteNet [64] where we finetune the full model and test the quality of the pretraining; (2) object classification on Shape Classification dataset [102] where we keep the model fixed and only train linear classifiers on fixed features, thus testing the quality of the learned representations [30, 47]. Finally, we also evaluate DepthContrast's generalizability to outdoor 3D data.

5.1. Importance of Data Augmentation

Data augmentations play an important role for self-supervised representation learning and have been studied extensively in the case of 2D images [9, 14, 56, 93, 94]. However, the impact of data augmentation for 3D representation learning is less well understood. Thus, we analyze the effect of our proposed augmentations from § 3.4 on transfer

Task	Scratch	Pretraining ScanNet (Multi-view)	ScanNet-vid (Single-view)	Redwood-vid (Single-view)
Shape Classification Linear (Accuracy)	50.7	85.1	85.0	86.4
SUNRGBD Detection (mAP)	57.4	60.5	60.7	60.4

Table 6: Single-view or multi-view 3D data. We study whether our pretraining is sensitive to single-view or multi-view data. We use ScanNet and ScanNet-vid which are multi-view and single-view versions of the same dataset [18], and Redwood-vid [16] which is a single-view only 3D dataset. Our pretrained model is robust to 3D preprocessing, and using single-view or multi-view data gives similar performance.

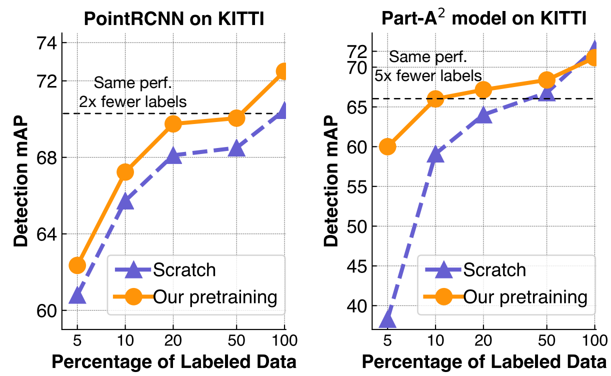

Figure 6: Using outdoor LiDAR data. We finetune detection models from scratch or using our pretraining and report mAP (with 40 recall positions) on the `cyclist` class at moderate difficulty level of the KITTI val split. Our models are pretrained using unlabeled outdoor data from the Waymo dataset and outperform scratch training using either point (left) or voxel (right) inputs.

performance. We train different DepthContrast point models with the same training setup and only vary the data augmentation used. Our results are summarized in Table 5.

The widely used VoteNet [64] augmentations perform worse than our proposed augmentations. Our augmentations lead to both a better feature representation: a gain of 5% accuracy on Shape Classification [102], and a better pretrained model: 2% mAP on SUNRGBD detection. We also consistently observe gains from our improved data augmentation on all the downstream tasks from § 4 which underscores the importance of designing good data augmentation.

5.2. Impact of Single-view or Multi-view 3D Data

We now study whether pretraining on reconstructed multi-view 3D scenes impacts the downstream performance. We use the ScanNet [18] dataset which contains multi-view 3D data obtained by 3D registration of the ScanNet-vid depth maps. As another single-view dataset, we pretrain on the Redwood-vid dataset from § 4.1.2. We pretrain DepthContrast point models on these datasets and compare their performance by transfer learning in Table 6.

The transfer performance is similar when models are pretrained on ScanNet-vid or ScanNet. Since ScanNet-vid and ScanNet only differ in the 3D preprocessing involved, the result suggests that DepthContrast is not sensitive to single-view or multi-view input data. This is not surprising given that our objective does not rely on multi-view information. Pretraining on the single-view Redwood-vid dataset also gives similar performance suggesting that DepthContrast is robust to different data distributions during pretraining. All the DepthContrast models outperform the scratch model.

5.3. Generalization to Outdoor LiDAR data

We test DepthContrast's generalization to outdoor LiDAR data by pretraining on the Waymo Open Dataset [88] where we extract 79K single-view scans from the videos. We use the same data augmentation parameters from § 3.4 and only modify random cuboid to work on the full scale of the Z dimension of the scene. We use the standard LiDAR-specific model architectures as our format-specific encoders - PointnetMSG [77] for point clouds and Spconv-UNet [78] for voxels. Similar to § 3.5, we obtain features from these models after global max pooling and a two layer MLP. The models are optimized jointly with Eq 3 using both within and across-format losses. For transfer learning, we use the standard KITTI [26] object detection benchmark, and PointRCNN [77] and Part-A^2 [78] for down-stream models. We report results on the `cyclist` class since it has fewer examples in the training set compared to the other classes. We provide results for other classes and finetuning details in the supplemental material. Similar to § 4.1.4, while varying the fraction of pretraining data, we report average performance across 3 independent runs. Fig 6 shows that our pretrained models outperform training from scratch especially when finetuning on fewer training samples. For Spconv-Unet, we achieve a **20% gain with 5% of labeled data**. This suggests that DepthContrast pretraining generalizes across multiple input formats, and our proposed data augmentation generalizes to different depth sensors and scene types.

6. Conclusion

We propose DepthContrast- an easy to implement self-supervised method that works across model architectures, input data formats, indoor/outdoor, single/multi-view 3D data. DepthContrast pretrains high capacity models for 3D recognition tasks, and leverages large scale 3D data that may not have multi-view information. We show state-of-the-art performance on detection and segmentation benchmarks, outperforming all prior work on detection. We provide crucial insights that make our simple implementation work well - training jointly with multiple input data formats and novel data augmentations. We hope DepthContrast helps future work in 3D self-supervised learning.

References

[1] Idan Achituve, Haggai Maron, and Gal Chechik. Self-supervised learning for domain adaptation on point-clouds. *arXiv preprint arXiv:2003.12641*, 2020.

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 40–49. PMLR, 2018.

[3] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. In *Computer Graphics Forum*, volume 29, pages 753–762. Wiley Online Library, 2010.

[4] Iro Armeni, Sasha Sax, Amir Roshan Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *CoRR*, abs/1702.01105, 2017.

[5] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[6] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018.

[7] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[9] 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.

[10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. Matterport license available at http://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf.

[11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[12] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 392–401, 2020.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[14] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[15] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9775–9784, 2019.

[16] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016.

[17] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[18] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[20] J. Donahue, P. Krahenbühl, and T. Darrell. Adversarial feature learning. In *International Conference on Learning Representations (ICLR)*, 2016.

[21] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(9):1734–1747, 2016.

[22] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9031–9040, 2020.

[23] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[24] Cat Franklin. Apple unveils new ipad pro with breakthrough lidar scanner and brings trackpad support to ipados. https://www.apple.com/, 2020.

[25] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[26] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[27] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. *arXiv preprint arXiv:1905.00561*, 2019.

[28] Rohit Girdhar, David Ford Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[29] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019.

[30] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan

Misra. Scaling and benchmarking self-supervised visual representation learning. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.
[31] Ben Graham. Sparse 3d convolutional neural networks. *arXiv preprint arXiv:1505.02890*, 2015.
[32] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018.
[33] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 216–224, 2018.
[34] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
[35] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 8160–8171, 2019.
[36] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.
[37] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.
[38] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (TOG)*, 37(6):1–12, 2018.
[39] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *International Conference on 3D Vision (3DV)*, 2016.
[40] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 984–993, 2018.
[41] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020.
[42] 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*, pages 141–165. Springer, 2013.
[43] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.
[44] Longlong Jing, Yucheng Chen, Ling Zhang, Mingyi He, and Yingli Tian. Self-supervised modal and view invariant feature learning. *arXiv preprint arXiv:2005.14169*, 2020.
[45] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 863–872, 2017.
[46] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6, 2019.
[47] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.
[48] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019.
[49] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9397–9406, 2018.
[50] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 820–830, 2018.
[51] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
[52] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.
[53] J. Masci, U. Meier, D. Cires, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, pages 52–59, 2011.
[54] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015.
[55] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.
[56] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019.
[57] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. *https://arxiv.org/abs/2004.12943*, 2020.
[58] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.
[59] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
[60] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.
[61] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[62] Mandela Patrick, Yuki M Asano, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020.

[63] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2019.

[64] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. *arXiv preprint arXiv:1904.09664*, 2019.

[65] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[66] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[67] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[68] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[69] Marc'Aurelio Ranzato, Fu-Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

[70] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3577–3586, 2017.

[71] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[72] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *AI-STATS*, pages 448–455, 2009.

[73] Samsung. What is depthvision camera on galaxy s20+ and s20 ultra? https://www.samsung.com/, 2020.

[74] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12962–12972, 2019.

[75] Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. Pigraphs: learning interaction snapshots from observations. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.

[76] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[77] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019.

[78] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *arXiv preprint arXiv:1907.03670*, 2019.

[79] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[80] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[81] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[82] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.

[83] Scott Stein. Lidar on the iphone 12 pro. https://www.cnet.com/, 2020.

[84] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.

[85] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018.

[86] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[87] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[88] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. This publication was made using the Waymo Open Dataset, provided by Waymo LLC under license terms available at waymo.com/open.

[89] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[90] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas

Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017.

[91] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018.

[92] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, pages 537–547. IEEE, 2017.

[93] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[94] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

[95] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[96] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.

[97] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

[98] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019.

[99] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 3523–3532, 2019.

[100] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[101] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

[102] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1912–1920, 2015.

[103] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[104] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the IEEE international conference on computer vision*, pages 1625–1632, 2013.

[105] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[106] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4606–4615, 2018.

[107] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018.

[108] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020.

[109] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[110] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 206–215, 2018.

[111] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1951–1960, 2019.

[112] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019.

[113] Zaiwei Zhang, Zhenxiao Liang, Lemeng Wu, Xiaowei Zhou, and Qixing Huang. Path-invariant map networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11084–11094, 2019.

[114] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. *arXiv preprint arXiv:2006.05682*, 2020.

[115] Zaiwei Zhang, Zhenpei Yang, Chongyang Ma, Linjie Luo, Alexander Huth, Etienne Vouga, and Qixing Huang. Deep generative modeling for scene synthesis via hybrid representations. *ACM Transactions on Graphics (TOG)*, 39(2):1–21, 2020.

[116] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.