



# Image2Point: 3D Point-Cloud Understanding with 2D Image Pretrained Models

Chenfeng Xu<sup>1\*</sup>, Shijia Yang<sup>1\*</sup>, Tomer Galanti<sup>2</sup>, Bichen Wu<sup>3\*\*</sup>, Xiangyu Yue<sup>1</sup>, Bohan Zhai<sup>1</sup>, Wei Zhan<sup>1</sup>, Peter Vajda<sup>3</sup>, Kurt Keutzer<sup>1</sup>, and Masayoshi Tomizuka<sup>1</sup>

<sup>1</sup> University of California, Berkeley

<sup>2</sup> Massachusetts Institute of Technology

<sup>3</sup> Meta Reality Labs

**Abstract.** 3D point-clouds and 2D images are different visual representations of the physical world. While human vision can understand both representations, computer vision models designed for 2D image and 3D point-cloud understanding are quite different. Our paper explores the potential of transferring 2D model architectures and weights to understand 3D point-clouds, by empirically investigating the feasibility of the transfer, the benefits of the transfer, and shedding light on why the transfer works. We discover that we can indeed use the same architecture and pretrained weights of a neural net model to understand both images and point-clouds. Specifically, we transfer the image-pretrained model to a point-cloud model by copying or inflating the weights. We find that finetuning the transformed image-pretrained models (FIP) with minimal efforts — only on input, output, and normalization layers — can achieve competitive performance on 3D point-cloud classification, beating a wide range of point-cloud models that adopt task-specific architectures and use a variety of tricks. When finetuning the whole model, the performance gets further improved. Meanwhile, FIP improves data efficiency, reaching up to 10.0 top-1 accuracy percent on few-shot classification. It also speeds up the training of point-cloud models by up to 11.1x for a target accuracy (e.g., 90 % accuracy). Lastly, we provide an explanation of the image to point-cloud transfer from the aspect of *neural collapse*. The code is available at: <https://github.com/chenfengxu714/image2point>.

**Keywords:** Point-cloud, Pre-training, Finetuning, Neural collapse

## 1 Introduction

Point-cloud is an important visual representation for 3D computer vision. It is widely used in a variety of applications, including autonomous driving [3,6,81], robotics [1,53,76], augmented and virtual reality [61,62,72], *etc.* However, a point-cloud represents visual information in a significantly different way from a 2D

---

\* Equal contribution

\*\* Corresponding Author

image. Specifically, a point-cloud consists of a set of unordered points lying on the object’s surface, with each point encoding its spatial  $x, y, z$  coordinates and potentially other features such as intensity. In contrast, a 2D image organizes visual features as a dense 2D RGB pixel array.

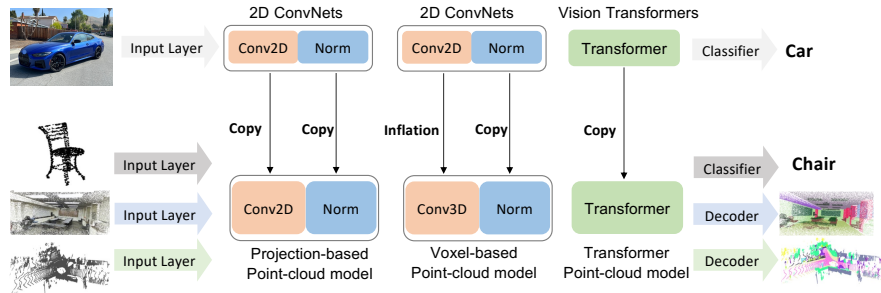
Due to the representation differences, 2D image and 3D point-cloud understanding are treated as two separate problems. 2D image models and point-cloud models are designed to have different architectures and are trained on different types of data. No efforts have tried to directly transfer models from images to point-clouds.

Intuitively, both 3D point-clouds and 2D images are visual representations of the physical world. Their low-level representations are drastically different, but they can represent the same underlying visual concept. Furthermore, human vision has no problem understanding both representations. To connect images and point-clouds, previous works attempted to generate pseudo point-clouds by estimating the depth of mono/stereo images [24,67,80]. However, depth estimation from a single image is a challenging problem in computer vision, which requires large-scale dense depth labels [58]. Estimating depth from stereo images is easier but requires strict calibrated and synchronized stereo cameras, which limits the data scale. Therefore, it is interesting to ask whether we could use large-scale image models that were pretrained using supervised classification datasets (*e.g.*, *ImageNet1K/ImageNet21K* classification) for point-cloud understanding.

Remarkably, the answer to the question above is positive. As we show in this work, 2D image models trained on image datasets can be transferred to understand 3D point-clouds with minimal effort. As illustrated in Figure 1, given the commonly-used image-pretrained models, such as 2D ConvNets [27] and vision transformers [17], we can easily convert them into various kinds of point-cloud models. In particular, a pretrained 2D ConvNet or vision transformer can be easily extended into a projection-based, voxel-based, or transformer-based point-cloud model via copying weights or inflating weights [9].

In this paper, we primarily focus on 3D ConvNets inflated from 2D pre-trained models. With the transformed point-cloud model (*e.g.*, inflated 3D ConvNets), we add linear input and output layers to the network; and on a target point-cloud dataset, we only finetune the input and output layers, and batch normalization layers, while keeping the pretrained model weights untouched. We call such partially-finetuned-image-pretrained models as *FIP-IO+BN* (finetuning input, output, and BN layers). As we show, *FIP-IO+BN* can achieve competitive performance up to 90.8% top-1 accuracy on the ModelNet 3D Warehouse dataset, on top of ResNet50, outperforming previous point-cloud models that adopt task-specific model architectures and tricks.

Even though incorporating pretrained models is useful for tackling downstream tasks, point-cloud models are typically trained from scratch. Based on our discovery, we further investigate fully-finetuned-image-pretrained models (termed as *FIP-ALL*). We observe that *FIP-ALL* brings significant improvement on top of different kinds of point-cloud models transformed from image-pretrained models. Besides, we also find that it generalizes to PointNet++ [55] which is pre-trained on



**Fig. 1.** We investigate the feasibility of pretrained 2D image models transferring to 3D point-cloud models. For example, with filter inflation and finetuning the input, output (classifier for classification task and decoder for semantic segmentation task), and normalization layers, the transformed 2D ConvNets are capable of dealing with point-cloud classification, indoor, and driving scene segmentation.

images by ourselves. Specifically, FIP-ALL outperforms the training-from-scratch by a large margin on top of PointNet++, SimpleView, ViT-B-16, and ViT-L-16, respectively. In addition to the performance gain, FIP-ALL exhibits superior data efficiency with up to 10.0% accuracy improvement in few-shot classification on the ModelNet 3D Warehouse dataset. Compared with training-from-scratch, FIP-ALL also dramatically speeds up the training by using 11.1 times fewer epochs to reach the same validation accuracy (*e.g.*, 90% accuracy).

Finally, we theoretically explore the relationship between transferring knowledge between tasks of different modalities and neural collapse to shed light on why the transfer works. The analysis is based on extending the framework proposed in [19] and is provided in Appendix.

## 2 Related Work

### 2.1 Point-Cloud Processing Models

In this section, we list the most prominent approaches for processing point-clouds.

The **3D convolution-based method** is one of the mainstream point-cloud processing approaches which efficiently processes point-clouds based on voxelization. In this approach, voxelization is used to rasterize point-clouds into regular grids (called voxels). Then, we can apply 3D convolutions to the processed point-cloud. However, enamors empty voxels make lots of unnecessary computations. Sparse convolution is proposed to apply on the non-empty voxels [13,18,64,66,78,85], largely improving the efficiency of 3D convolutions.

The **projection-based method** attempts to project a 3D point-cloud to a 2D plane and uses 2D convolution to extract features [5,38,63,69,70,71,75]. Specifically, bird-eye-view projection [37,79] and spherical projection [50,70,71,75] have made great progress in outdoor point-cloud tasks.

Another approach is the **point-based method**, which directly processes the point-cloud data. The most classic methods, PointNet [54] and PointNet++ [55],

consume points by sampling the center points, group the nearby points, and aggregate the local features. Many works further develop advanced local-feature aggregation operators that mimic the 3D convolution operation to structured data [32,36,40,41,42,44,66,76].

## 2.2 Pretraining in 2D and 3D Computer Vision

**Pretraining in 2D computer vision** is an effective approach using supervised [17,21], self-supervised [23,33], and contrastive learning [2,8,10,12,26,29]. After pretraining on a large amount of data, a 2D model requires less computational and data resources for finetuning in order to obtain competitive performance on downstream tasks [7,11,28,34].

**Pretraining in 3D computer vision** has been studied similarly as pretraining in 2D vision: both self-supervised and contrastive pretraining [31,65,73] show promising results. 3D point-clouds are difficult to annotate, and there is no large-scale annotated dataset available. To address this, previous works have tried to use model pretraining to improve data efficiency [77]. Recent works [30,83] explored using contrastive learning on point-clouds. Our work does not rely on long-time pretraining. Instead, we can directly take large amounts of open-sourced image-pretrained models for a variety of point-cloud tasks.

## 2.3 Cross-Modal Transfer Learning

**Cross-modal transfer learning** takes advantage of data from various modalities [15,45,47,52,74]. For example, [43] proposed pixel-to-point knowledge transfer (PPKT) from 2D to 3D which uses aligned RGB and RGB-D images during pretraining. Our work does not rely on joint image-point-cloud pretraining. Instead, we directly transfer an image-pretrained model to a point-cloud model with the simplest pretraining-finetuning scheme.

Some of the previous works for video and medical images [9,60] have adopted the method of simply extending a pretrained 2D convolutional filter along time or depth direction for transferring to 3D models. However, the domain gaps between point-clouds and images are much more than that of videos/medical images and images. Between language and image modalities, transfer learning with minimal finetuning also shows a competitive performance [46,57].

## 2.4 Neural Collapse

Neural collapse (NC) [25,51] is a recently discovered phenomenon in deep learning. It has been observed that during the training of deep overparameterized neural networks for standard classification tasks, the penultimate layer’s features associated with training samples belonging to the same class concentrate around their class means. Essentially, [51] observed that the ratio of the within-class variances and the distances between the class means converge to zero. In addition to that, it has also been observed that asymptotically the class means (centered at

their global mean) are not only linearly separable, but are also maximally distant and located on a sphere centered at the origin up to scaling, and furthermore, that the behavior of the last-layer classifier (operating on the features) converges to that of the nearest-class-mean decision rule.

Recently, [19] studied the relationship between **neural collapse and transfer learning**. They studied a transfer learning setting, where we intend to solve a target (classification) task, where only a limited amount of samples is available, so a model is pretrained and transferred from a source (classification) task. They showed that neural collapse extends beyond training and generalizes also to unseen test samples and new classes. In addition, it was shown that in the presence of neural collapse in the new classes, training a linear classifier on top of the learned penultimate layer requires only a few samples to generalize well. However, their empirical and theoretical analysis assumes that the source and target classes are i.i.d. samples (*e.g.*, a random split of the classes in ImageNet). This implies that the two tasks share the same modality. Therefore, we suggest training an adaptor (*e.g.*, a linear layer) along with retraining the normalization parameters as part of the transfer process. Intuitively, the adaptor takes samples of the second modality and translates them to representations that are interpretable by the pretrained model, such that it produces feature embeddings that are clustered into classes. In Appendix B, we extend the framework in [19] to the case where the source and target tasks are of different modalities and theoretically analyze it.

### 3 Converting a 2D Image Model to a 3D Point-Cloud Model

In this paper, we primarily focus on the 3D sparse-convolution-based method to process point-clouds, since it can be extended to a wide range of point-cloud tasks. The other point-cloud models we use in this paper are byproducts of copying the weights of 2D image models, for example, 2D ConvNets [27] or vision transformers [17]. In this section, we provide an in-depth introduction to how we transform the 2D ConvNets into 3D sparse ConvNets by inflation [9].

*Inflating a 2D ConvNet into a 3D sparse ConvNet.* As discussed in Section 2.1, we consider a set of points, where each point is represented by its 3D coordinates and additional features such as its intensity and RGB. We then voxelize/quantize these points into voxels according to their 3D space coordinates, following [13]. A voxel’s feature is inherited from the point that lies within the voxel. If there are multiple points associated with the same voxel, we average all points’ features and assign the mean to the voxel. If there is no point in the voxel, then we simply set the voxel’s feature to 0. With sparse convolution, the computation on empty voxels can be skipped.

Given a pretrained 2D ConvNet, we convert it to a 3D ConvNet that takes 3D voxels as input. The key element of this procedure is to convert 2D convolution filters to 3D, *i.e.*, constructing 3D filters with the weights directly inherited from 2D filters. A 2D convolutional filter can be represented with a 4D tensor of shape

$[M, N, K, K]$ , representing output dimension, input dimension, and two spatial kernel sizes, respectively. A 3D convolutional filter has an extra dimension, and its shape is  $[M, N, K, K, K]$ . To better illustrate, we ignore the output and input dimensions and only consider a spatial slice of the 2D filter with shape  $[K, K]$ . The simplest way to convert this 2D filter to 3D is to repeat the 2D filter  $K$  times along a third dimension. This operation is the same as the *inflation* technique used by [9] to initialize a video model with a pretrained 2D ConvNet.

Besides convolution, other operations such as downsampling, BN, and non-linear activation can be easily migrated to 3D. Our 3D model inherits the architecture of the original 2D ConvNet, but we also add a linear layer as the input layer and an output layer depending on the target task. For classification, we use a global average pooling layer followed by one fully connected layer to get the final prediction. For semantic segmentation, the output layer is a U-Net style decoder [59]. The architecture of the input/output layers is described in more detail in Appendix B.7.

*A note on image-to-video transfer.* It is noteworthy to mention that although inflation is commonly used in video domains, image-to-point-cloud transfer is fundamentally different. Even though videos and point-clouds are both 3D data, they are represented with completely different visual modalities with different distributions. Intrinsically, 3D point-clouds are represented as a sparse set of points lying on object surfaces and parameterized by  $xyz$ -coordinates, while videos are dense RGB arrays, where the two spatial arrays represent RGB images and the temporal array reflects how images evolve through time. Point-clouds are translation and rotation invariant or equi-variant, while for videos, the spatial and temporal dimensions are not interchangeable. In this paper, we surprisingly find that with simple operations such as inflation, the image-pretrained models can be directly used for point-cloud understanding under the situation that image and point-cloud are drastically different. The detailed experiments showing the feasibility and utility, and the discussion of why it works from the aspect of neural collapse are illustrated in Section 4 and Section 5, respectively.

## 4 Empirical Evaluation

To explore the image to point-cloud transfer, we study three settings: **(1)** finetuning input, output, and batch normalization layers (FIP-IO+BN), **(2)** finetuning the whole pretrained network (FIP-ALL), and optionally **(3)** partially-finetuned-image-pretrained model, only finetuning input and output layers (FIP-IO). Under the three settings, we extensively explore the feasibility of transferring the image-pretrained model for point-cloud understanding and its benefits. The entire empirical evaluation is organized as four questions: **(1)** Can we transfer pretrained-image models to recognize point-clouds? (Section 4.1) **(2)** Can image-pretraining benefit the performance of point-cloud recognition? (Section 4.2) **(3)** Can image-pretrained models improve the data efficiency on point-cloud recognition? (Section 4.3) **(4)** Can image-pretrained models accelerate training point-cloud models? (Section 4.4)

*Datasets.* We evaluate the transferred models on ModelNet 3D Warehouse classification [72], S3DIS indoor segmentation [1], and SemanticKITTI outdoor segmentation [3] tasks. ModelNet 3D Warehouse is a CAD model classification dataset that consists of point-clouds with 40 categories, and CAD models come from 3D Warehouse [62]. In this benchmark, we only utilize  $x, y, z$  coordinates as features. S3DIS is a dataset collected from real-world indoor scenes and includes 3D scans of Matterport Scanners from 6 areas. It provides point-wise annotations for indoor objects like chair, table, and bookshelf, *etc.* SemanticKITTI dataset from KITTI Vision Odometry [20] is a driving scene dataset. It provides dense point-wise annotations for the complete 360 degrees field-of-view of the deployed automotive lidar, which is currently one of the most challenging datasets.

ResNet [27] series is used mostly throughout our experiments. Depending on the experiments, ResNets are pretrained on Tiny-ImageNet, ImageNet-1K, ImageNet-21K [16], and Fractal database (FractalDB) [34]. Our pretrained models are directly downloaded from various sources, with detailed links provided in the Appendix. To study the benefits of using pretrained image models, we also utilize PointNet++ [55], ViT [17], and SimpleView [22] as our baselines.

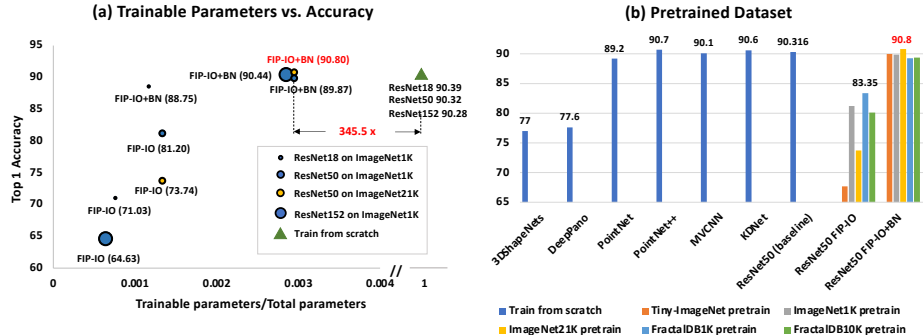
#### 4.1 Can we transfer pretrained-image models to recognize point-clouds?

To evaluate the feasibility of transferring pretrained 2D image models to 3D point-cloud tasks, we conduct experiments on top of the ResNet series since there are abundant open-source pretrained ResNet available. In particular, we convert 2D ConvNets into 3D ConvNets using the procedure described in Section 3. We hypothesize that, if a pretrained 2D image model is capable of understanding point-clouds directly, we can see a non-trivial performance by only finetuning input and output layers of the transferred model. Further, as we gradually relax the frozen parameters, finetuning BN parameters as well, the transferred model can achieve better performance, even surpassing training-from-scratch.

We conduct two groups of experiments with FIP-IO and FIP-IO+BN, with the results shown in Figure 2. The first is to evaluate the performance as the trainable parameters gradually increase. As shown in Figure 2 (a), training **no more than 0.3 % (345.5x fewer)** of the whole parameters, the image pretraining even beats the training-from-scratch (100 % trainable parameters). Specifically, ResNet152 FIP-IO+BN with ImageNet1K pretraining improves training-from-scratch by 0.16 points, and ResNet50 FIP-IO+BN with ImageNet21K pretraining improves 0.48 points. Meanwhile, FIP-IO reaches a non-trivial performance. ResNet50 FIP-IO pretrained on ImageNet1K achieves 81.20 % top-1 accuracy, only 9.12 points worse than training-from-scratch with approximately 0.1 % trainable parameters.

Furthermore, to investigate the effect of different datasets, as shown in the right figure of Figure 2, we inflate ResNet50 pretrained from different image datasets, including Tiny-ImageNet, ImageNet1K, ImageNet21K, FractalDB1K, and FractalDB10K, then evaluate on the ModelNet 3D Warehouse.

We discover that, even if we only finetune the input and output layers while keeping the image-pretrained weights frozen, the FIP-IO pretrained from



**Fig. 2.** a) the left figure shows the trainable parameters ratio *w.r.t* top-1 accuracy on ModelNet 3D Warehouse dataset. b) the right figure shows the performance of FIP-IO and FIP-IO+BN on top of **ResNet50** pretrained on different datasets.

**Table 1.** ModelNet 3D Warehouse classification results (top-1 accuracy %) of fully-finetuned-image-pretrained models (FIP-ALL) based on different pretrained models. We include 2021 SOTAs, such as RSMix [39], Point Transformer (Point-Trans) [84], DRNet [56], and PointCutMix [82], for comparison.

Method	ResNet18	ResNet50	ResNet152	ResNet101×2
From Scratch	90.39	90.32	90.28	90.03
FIP-ALL on ImageNet1K	90.52 (+0.13)	90.92 (+0.60)	91.09 (+0.81)	90.52 (+0.49)
FIP-ALL on ImageNet21K	-	91.05 (+0.73)	-	-
Method	PointNet++(SSG) ViT-B-16	ViT-L-16	SimpleView	
From Scratch	90.34	84.27	83.48	
FIP-ALL on ImageNet1K	91.22 (+0.88)	-	93.8 (+0.50)	
FIP-ALL on ImageNet21K	-	87.77 (+3.50)	87.66 (+4.18)	
Method	RSMix	Point-Trans	DRNet	PointCutMix
From Scratch	93.5	93.7	93.1	93.4

ImageNet1K, FractalDB1K, and FractalDB10K achieves competitive performance. Specifically, ResNet50 FIP-IO with ImageNet1K pretraining outperforms 3D ShapeNet [72] and DeepPano [61], which were the state-of-the-arts in 2015, by 4.2 and 3.6 points respectively in top-1 accuracy on ModelNet 3D Warehouse. More importantly, with ImageNet21K pretrained model, ResNet50 FIP-IO+BN surpasses training-from-scratch by 0.48 points, even beating a variety of well-known methods including PointNet [54], MVCNN [63], *etc.*

Notably, we find out the answer to "Can we transfer pretrained-image models to recognize point-clouds?": Yes. The pretrained 2D image models can be directly used for recognizing point-clouds. Surprisingly, the pretraining dataset is not restricted to natural but also synthetic images like those in FractalDB1K/10K.



**Table 2.** Indoor scene and outdoor scene segmentation results (mIoU %) of fully-finetuned-image-pretrained Model (FIP-ALL). In this table, all image-pretrained models are pretrained on ImageNet1K.

Method	S3DIS (mIoU %)		SemanticKITTI (mIoU %)	
	PointNet++(SSG)	ResNet18	HRNetV2-W48	ResNet18
From Scratch	52.45	55.09	44.12	64.75
FIP-ALL on ImageNet1K	55.01 (+ <b>2.56</b> )	56.62 (+ <b>1.53</b> )	47.53 (+ <b>3.41</b> )	65.57 (+ <b>0.82</b> )

**Table 3.** Comparison with PointContrast [73] on the ModelNet 3D Warehouse. PointContrast provides two different pretrained models with using PointInfoNCE loss and Hardest Contrastive loss, respectively.

From scratch	PointInfoNCE	Hardest Contrastive	ImageNet1K pretrain (Ours)
89.95	90.24 (+ <b>0.29</b> )	90.15 (+ <b>0.20</b> )	90.88 (+ <b>0.93</b> )

## 4.2 Can image-pretraining benefit point-cloud recognition?

From the previous subsection, we find unexpectedly that the image-pretrained model can be directly used for point-cloud understanding. In this subsection, we investigate whether the image-pretrained model is helpful to improve the performance of point-cloud tasks. We use different baselines, including voxelization-based method (simply ResNet), point-based method (PointNet++ [55]), projection-based method (SimpleView [22]), and current popular transformer-based method (ViT-B-16 and ViT-L-16 [17]), and fully finetune them on three point-cloud datasets: classification on ModelNet 3D Warehouse, scene segmentation on S3DIS and SemanticKITTI, as shown in Table 1 and Table 2.

For PointNet++, we use ImageNet1K to pretrain: we break each image into pixels and regard it as a point-cloud. For ViT, we directly use the open-source pretrained model and finetune it on ModelNet 3D Warehouse. All the implementation details are illustrated in Appendix A.

Table 1 presents performance on ModelNet 3D Warehouse dataset. We observe that FIP-ALL improves all baselines steadily and significantly. Besides, pretraining brings more improvements to deeper models. For example, ResNet18 can only be improved by 0.13% top-1 accuracy, but pretraining on ImageNet1K leads to 0.81 points top-1 accuracy improvement on top of ResNet152. Moreover, larger pretrained datasets also lead to better performance. Specifically, ResNet50 FIP-ALL from ImageNet21K can reach 91.05% top-1 acc, with 0.73 points improvement over training-from-scratch. Such FIP-ALL significantly outperforms a series of well-known methods such as [35,40,54,55,63,68].

We also explore FIP-ALL on different architectures, as shown in the second group of Table 1. In particular, FIP-ALL on top of PointNet++, ViT-B-16, ViT-L-16, and SimpleView with image dataset pretraining improve the training-from-scratch by 0.88, 3.50, 4.18, 0.50 points, respectively. Especially for the current superior baseline in image recognition, ViT-B-16 and ViT-L-16, the

**Table 4.** Few-shot experiments on top of different ResNets on the ModelNet 3D Warehouse dataset. We conduct 3 trials for each setting and results are as mean  $\pm$  std.

Few-shot	ResNet18	ResNet50 (from scratch/FIP-ALL)	ResNet152
10-shot	72.2 $\pm$ 0.8/73.2 $\pm$ 0.6 (+1.0)	71.7 $\pm$ 0.7/74.1 $\pm$ 0.8 (+2.4)	69.8 $\pm$ 1.1/73.9 $\pm$ 0.4 (+4.1)
5-shot	63.7 $\pm$ 1.6/66.6 $\pm$ 0.8 (+2.9)	62.4 $\pm$ 1.1/66.0 $\pm$ 2.2 (+3.6)	59.4 $\pm$ 0.8/66.5 $\pm$ 0.9 (+7.1)
1-shot	26.8 $\pm$ 4.4/36.8 $\pm$ 0.6 (+10.0)	28.1 $\pm$ 0.4/34.1 $\pm$ 0.2 (+6.0)	23.3 $\pm$ 4.3/33.2 $\pm$ 1.3 (+9.9)

improved performance is quite significant, revealing the huge potential of using image-pretrained models for point cloud recognition.

For the challenging indoor and outdoor scene segmentation, using ImageNet1K pretrained models (FIP-ALL on ImageNet1K) also improve the training-from-scratch consistently, as shown in Table 2. PointNet++ (resp. ResNet18) pretrained on ImageNet1K outperforms the training-from-scratch by 2.56 points (resp. 1.53 points) mIoU on S3DIS dataset. For SemanticKITTI, we utilize the commonly used projection-based method with 2D ConvNet HRNet. With ImageNet1K pretraining, we observe 3.41 points mIoU improvement, a large margin in such a challenging task. Since HRNetV2-W48 has rich pretrained models, we finetune Cityscapes pretrained HRNetV2-W48 and observe this enhances more (5.25% mIoU improvement over training from scratch). Even for the ResNet18 with a high from-scratch performance of 64.75% mIoU, the ImageNet1K pretraining can also bring 0.82 points mIoU improvement.

Finally, we compare the performance gain with the well-known point-cloud self-supervised method PointContrast [73], as presented in Table 3. We use the same model architecture and finetuning recipe, and the only difference is the pretraining weights. Note that the model architecture used in PointContrast does not have corresponding open-sourced image-pretrained weights, so we pretrain it by ourselves on ImageNet1K, with the standard ImageNet training recipe provided by Pytorch. We can observe that image-pretraining on ImageNet1K significantly boosts the training-from-scratch by 0.93 points, surpassing the PointContrast by at least 0.64 points.

Therefore, the answer to "Can image-pretraining benefit point-cloud recognition" is: Yes. Image-pretraining can indeed improve point-cloud recognition, generalize to a wide range of backbones, and benefit multiple challenging tasks.

### 4.3 Can image-pretrained models improve the data efficiency on point-cloud recognition?

Data efficiency is essential in point-cloud understanding due to the huge labor of collecting and annotating point-cloud data. In this subsection, we investigate whether the image-pretrained model can help to improve the data efficiency by conducting few-shot setting experiments, including 1-shot, 5-shot, and 10-shot.

In detail, for each class (ModelNet 3D Warehouse involves 40 classes), we randomly choose a few point-clouds as training data and still evaluate on the whole test set. We compare the results between training-from-scratch and FIP-ALL

**Table 5.** Semi-supervised distillation experiments on top of ResNet34 on the ModelNet 3D Warehouse dataset.

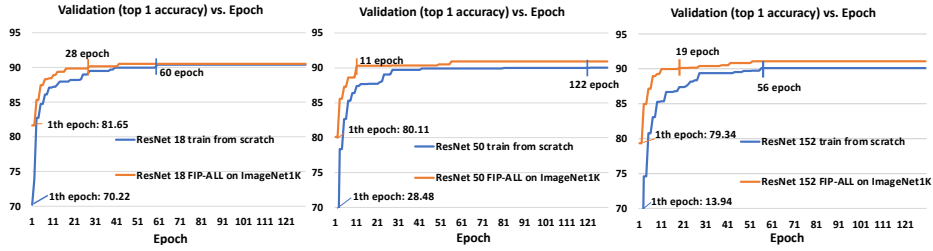
Few-shot	From scratch	PointInfoNCE	Hardest Contrastive	ImageNet1K pretrain (Ours)
10-shot	72.2	74.6 (+2.4)	74.6 (+2.4)	74.9 (+2.7)
5-shot	61.9	65.1 (+3.2)	65.9 (+4.0)	66.0 (+4.1)
1-shot	29.2	39.0 (+9.8)	37.2 (+8.0)	41.1 (+11.9)

pretrained on the ImageNet1K dataset. The experimental results are shown in Table 4. We observe that FIP-ALL dramatically surpasses training-from-scratch on the low data regime (1-shot): pretraining on ImageNet1K brings 10.0, 6.0, and 9.9 points top-1 accuracy improvement for ResNet18, ResNet50, and ResNet152, respectively. For 5-shot and 10-shot settings, using ImageNet1K pretraining can still consistently improve the performance.

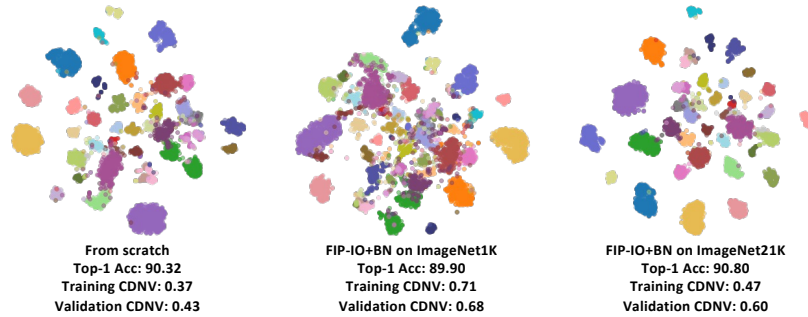
Furthermore, inspired by previous work [11] which proposed *big self-supervised models are strong semi-supervised learners* in 2D image recognition, we borrow the idea and propose *an image-pretrained model is also a strong semi-supervised learner in point-cloud recognition*. We also compared the image-pretrained model with the self-supervised pretrained model in this experiment. Specifically, we first take pretrained models from the previous self-supervised pretraining method PointContrast [73]. PointContrast provides two ScanNet [14] pretrained models of architecture ResNet34 trained with hardest-contrastive loss and PointInfoNCE loss. Then, we finetune PointContrast on 1/5/10 shot of the labeled ModelNet 3D Warehouse dataset and regard it as a teacher model. Finally, we distill the teacher model to a randomly initialized student model. In detail, we pass in the rest of unlabeled ModelNet 3D Warehouse dataset and 1/5/10 shot of the labeled dataset into the teacher model to generate pseudo labels. We use softmax MSE loss as consistency loss between student model outputs and pseudo labels. When the data instance is labeled, we add an additional cross entropy loss as a class criterion between student output and the label.

To show the effectiveness of the image-pretrained model, we repeat the above experiment, only replacing self-supervised pretrained models with ResNet34 ImageNet1K pretrained models. Results are reported in Table 5. We observe that image-pretrained ResNet34 consistently outperforms PointContrast, and improves the baseline by a large margin with 11.9, 4.1, and 2.7 points on 1-shot, 5-shot, and 10-shot, respectively. The results in Table 5 show that an image-pretrained model is indeed a strong semi-supervised learner in point-cloud recognition.

However, in both Table 4 and Table 5, we observe that as the amount of training data increases, the performance increases. Therefore, our answer to "Can image-pretrained models improve the data efficiency on point-cloud recognition?" is: Yes. Image-pretrained models can improve the data efficiency on point-cloud recognition, especially on low data regime. Although when the training data increases, performance gain becomes marginal.



**Fig. 3.** The curves of validation accuracy w.r.t training epoch. We compare the results between training-from-scratch and FIP-ALL on the ImageNet1K, on top of ResNet18, ResNet50, and ResNet152, respectively.



**Fig. 4.** tSNE visualization and class-distance normalized variance of fine-tuned models on train and validation split of ModelNet 3D Warehouse dataset. FIP-IO+BN on ImageNet1K/21K are the same models in Figure 2.

#### 4.4 Can image-pretrained models accelerate point-cloud training?

We also investigate whether the image-pretrained model can accelerate training on the point-clouds. The results are shown in Figure 3.

We discover that, after training only one epoch on ModelNet 3D Warehouse dataset, FIP-ALL pretrained on ImageNet1K achieves very impressive performance, yet the performance of training-from-scratch is still low. For instance, after the first epoch, ResNet50 (resp. ResNet152) with training from scratch achieves 28.48% (resp. 13.94%) top-1 accuracy while ResNet50 (resp. ResNet152) with ImageNet1K pretraining reaches 80.11% (resp. 79.34%) top-1 accuracy. Moreover, to reach 90% top-1 accuracy, a non-trivial performance, FIP-ALL significantly accelerates the training by 2.14x (28 vs. 60 epoch), 11.1x (11 vs. 122 epoch), 2.95x (19 vs. 56 epoch) over training-from-scratch, on top of ResNet18, ResNet50, and ResNet152, respectively.

Therefore, our answer to “Can image-pretrained models accelerate point-cloud training?” is still positive. The image-pretrained models can significantly accelerate the training speed of point-cloud tasks.

## 5 Neural Collapse in Cross-Modal Transfer

In this section, we provide an explanation of why the image to point-cloud transfer works based on the recently observed phenomenon called neural collapse [25,51]. [19] in depth studied the relationship between neural collapse and transfer learning between two classification tasks of the same modality (image domain). Similar to this work, we focus on transferring pretrained models between domains of different modalities, *i.e.*, from images to point-clouds.

As illustrated in Section 4, we can transfer image-pretrained models to the point-cloud domain. This motivates us to question whether the phenomenon of neural collapse generalization [19] (see Section 2.4) is also evident in our case. Following [19], we explore the relationships between neural collapse and image-to-point transfer by calculating the class-distance normalized variance (CDNV). Informally, the CDNV measures the ratio between the within-class variances of the embeddings and the squared distance of their means (see Appendix B.6 for details). We measure the CDNV of the fine-tuned model on both train and test data of the point-cloud domain. Since neural collapse is essentially a clustering property of features learned by neural networks, we further examine the neural collapse using tSNE visualizations. The results are summarized in Figure 4.

We observe that with finetuning much fewer (345.5x fewer) parameters in ResNet50 pretrained on ImageNet1K, both class-distance-normalized-variance and the clustering of tSNE are worse than training-from scratch, but still show relatively obvious clustering phenomenon. However, when we use the ResNet50 pretrained on ImageNet21K, the top-1 accuracy, and CDNV are significantly improved. More importantly, CDNV of ImageNet1K pretrained ResNet50 and ImageNet21K pretrained ResNet50 is lower than 1. This observation indicates although the image domain and point-cloud domain are quite different, the phenomenon of neural collapse generalization [19] still exists in their transfer. More results and analysis are illustrated in Appendix B.6.

Moreover, the interesting discovery pushes us to think about the reason of cross-modal transfer having neural collapse. Inspired by [19], we briefly explain below. More detailed theoretical proof is presented in Appendix C.

*Theoretical idea.* In this work, we focused on the problem of transferring knowledge between two tasks (source and target) consisting of two different modalities with different classes. Therefore, in the theoretical analysis, we have two separate modes of generalization: *between classes* and *between modalities*. In order to model this problem, we assume that the target and source tasks are decomposed of i.i.d. classes that are samples of two different distributions  $\mathcal{D}_1$  and  $\mathcal{D}_2$  (each stands for a different domain/modality). Each class is defined by a distribution over samples (e.g., samples of dog images). Given a target task (consisting of a set of randomly selected classes  $P_1, \dots, P_k \sim \mathcal{D}_1$ ), the pretrained model is evaluated after training an adaptor and a linear classifier on top of it. Its overall performance is measured in expectation over the selection of target tasks.

To capture the similarity between the two domains, we assume there exists an invertible mapping  $F$  between the classes that preserves the density of the

two distributions, namely,  $\hat{P}_c = F(P_c) \sim \mathcal{D}_2$  for  $P_c \sim \mathcal{D}_1$ . To characterize the similarity between the classes coming from  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , we further assume that the classes  $P_c$  and  $\hat{P}_c$  share a ‘mutual representation space’ from which the class label can be recovered. The shared space is given by two simple functions  $g^*$  and  $\tilde{g}^*$  for which the distance between  $g^* \circ P_c$  and  $\tilde{g}^* \circ \hat{P}_c$  is small (in expectation over  $P_c \sim \mathcal{D}_1$ ). By utilizing tools from the theory of Unsupervised Domain Adaptation [4,48,49], we translate the performance of a pretrained model on randomly selected target tasks into its expected error on randomly selected tasks with classes from  $\mathcal{D}_2$ . Then, in order to bound this error, we use Proposition 5 in [19] that relates the error and the degree of neural collapse of the pretrained model on randomly selected classes from  $\mathcal{D}_2$ . Finally, according to Propositions 1 and 2 in [19], this quantity can be upper bounded by the degree of neural collapse of the pretrained model on the source train data.

## 6 Conclusions

In this work, we use finetuned-image-pretrained models (FIP) to explore the feasibility of transferring image-pretrained models for point-cloud understanding and the benefits of using image-pretrained models on point-cloud tasks. We surprisingly discover that, with simply transforming a 2D pretrained ConvNet and minimal finetuning — input, output, and batch normalization layer (FIP-IO or FIP-IO+BN), FIP can achieve very competitive performance on 3D point-cloud classification, beating a wide range of point-cloud models that adopt a variety of tricks. Moreover, we find that when finetuning all the parameters of the pretrained models (FIP-ALL), the performance can be significantly improved on point-cloud classification, indoor and outdoor scene segmentation. Fully finetuned models generalize to most of the popular point-cloud methods. We also find that FIP-ALL can improve the data efficiency on few-shot learning and accelerate the training speed by a large margin. Additionally, we explore the relationships between neural collapse and cross modal transferring for our case, and shed light on why it works based on neural collapse. Compared with previous works that seek improvements from designing architectures and pretraining only on point-cloud modality, our work is not limited by the architecture design and the small-scale point-cloud dataset. We believe that image pretraining is one of the solutions to the bottleneck of point-cloud understanding and hope this direction can inspire the research community.

## Acknowledgements

Co-authors from UC Berkeley were sponsored by Berkeley Deep Drive (BDD). Tomer Galanti’s contribution was supported by the Center for Minds, Brains and Machines (CBMM), funded by NSF STC award CCF-1231216.

## References

1. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017)
2. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. arXiv preprint arXiv:1906.00910 (2019)
3. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV) (2019)
4. Ben-david, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in Neural Information Processing Systems 19, pp. 137–144. Curran Associates, Inc. (2006)
5. Boulch, A., Le Saux, B., Audebert, N.: Unstructured point cloud semantic labeling using deep segmentation networks. 3DOR **2**, 7 (2017)
6. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
7. Caron, M., Bojanowski, P., Mairal, J., Joulin, A.: Unsupervised pre-training of image features on non-curated data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2959–2968 (2019)
8. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)
9. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
10. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
11. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029 (2020)
12. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
13. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
14. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2017)
15. Dai, A., Nießner, M.: 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 452–468 (2018)
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

18. Feng, D., Zhou, Y., Xu, C., Tomizuka, M., Zhan, W.: A simple and efficient multi-task network for 3d object detection and road understanding. arXiv preprint arXiv:2103.04056 (2021)
19. Galanti, T., György, A., Hutter, M.: On the role of neural collapse in transfer learning. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=SwIp410B6aQ>
20. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361 (2012)
21. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
22. Goyal, A., Law, H., Liu, B., Newell, A., Deng, J.: Revisiting point cloud shape classification with a simple and effective baseline. arXiv preprint arXiv:2106.05304 (2021)
23. Goyal, P., Caron, M., Lefaudeaux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., et al.: Self-supervised pretraining of visual features in the wild. arXiv preprint arXiv:2103.01988 (2021)
24. Gur, S., Wolf, L.: Single image depth estimation trained via depth from defocus cues. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7683–7692 (2019)
25. Han, X.Y., Pappas, V., Donoho, D.L.: Neural collapse under mse loss: Proximity to and dynamics on the central path (2021)
26. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
28. Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: International Conference on Machine Learning. pp. 4182–4192. PMLR (2020)
29. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018)
30. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3d scene understanding with contrastive scene contexts. arXiv preprint arXiv:2012.09165 (2020)
31. Hou, J., Graham, B., Nießner, M., Xie, S.: Exploring data-efficient 3d scene understanding with contrastive scene contexts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15587–15597 (2021)
32. Hua, B.S., Tran, M.K., Yeung, S.K.: Pointwise convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 984–993 (2018)
33. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
34. Kataoka, H., Okayasu, K., Matsumoto, A., Yamagata, E., Yamada, R., Inoue, N., Nakamura, A., Satoh, Y.: Pre-training without natural images. In: Proceedings of the Asian Conference on Computer Vision (2020)
35. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 863–872 (2017)



36. Komarichev, A., Zhong, Z., Hua, J.: A-cnn: Annularly convolutional neural networks on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7421–7430 (2019)
37. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
38. Lawin, F.J., Danelljan, M., Tosteberg, P., Bhat, G., Khan, F.S., Felsberg, M.: Deep projective 3d semantic segmentation. In: International Conference on Computer Analysis of Images and Patterns. pp. 95–107. Springer (2017)
39. Lee, D., Lee, J., Lee, J., Lee, H., Lee, M., Woo, S., Lee, S.: Regularization strategy for point cloud via rigidly mixed sample. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15900–15909 (2021)
40. Li, J., Chen, B.M., Lee, G.H.: So-net: Self-organizing network for point cloud analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9397–9406 (2018)
41. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on  $\chi$ -transformed points. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. pp. 828–838 (2018)
42. Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., Pan, C.: Densepoint: Learning densely contextual representation for efficient point cloud processing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5239–5248 (2019)
43. Liu, Y.C., Huang, Y.K., Chiang, H.Y., Su, H.T., Liu, Z.Y., Chen, C.T., Tseng, C.Y., Hsu, W.H.: Learning from 2d: Pixel-to-point knowledge transfer for 3d pretraining. arXiv preprint arXiv:2104.04687 (2021)
44. Liu, Z., Hu, H., Cao, Y., Zhang, Z., Tong, X.: A closer look at local aggregation operators in point cloud analysis. In: European Conference on Computer Vision. pp. 326–342. Springer (2020)
45. Liu, Z., Qi, X., Fu, C.W.: 3d-to-2d distillation for indoor scene parsing. arXiv preprint arXiv:2104.02243 (2021)
46. Lu, K., Grover, A., Abbeel, P., Mordatch, I.: Pretrained transformers as universal computation engines. arXiv preprint arXiv:2103.05247 (2021)
47. Lu, Y., Xu, C., Wei, X., Xie, X., Tomizuka, M., Keutzer, K., Zhang, S.: Open-vocabulary 3d detection via image-level class and debiased cross-modal contrastive learning. arXiv preprint arXiv:2207.01987 (2022)
48. Mansour, Y.: Learning and domain adaptation. In: Algorithmic Learning Theory, 20th International Conference, ALT. pp. 4–6 (2009)
49. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation: Learning bounds and algorithms. In: COLT - The 22nd Conference on Learning Theory (2009)
50. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4213–4220. IEEE (2019)
51. Pappayan, V., Han, X.Y., Donoho, D.L.: Prevalence of neural collapse during the terminal phase of deep learning training. Proceedings of the National Academy of Sciences **117**(40), 24652–24663 (2020)
52. Park, J., Xu, C., Zhou, Y., Tomizuka, M., Zhan, W.: Detmatch: Two teachers are better than one for joint 2d and 3d semi-supervised object detection. arXiv preprint arXiv:2203.09510 (2022)
53. Pomerleau, F., Colas, F., Siegwart, R.: A review of point cloud registration algorithms for mobile robotics. Foundations and Trends in Robotics **4**(1), 1–104 (2015)

54. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation (2016), <http://arxiv.org/abs/1612.00593>, cite arxiv:1612.00593
55. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)
56. Qiu, S., Anwar, S., Barnes, N.: Dense-resolution network for point cloud classification and segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3813–3822 (2021)
57. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021)
58. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020)
59. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
60. Shan, H., Zhang, Y., Yang, Q., Kruger, U., Kalra, M.K., Sun, L., Cong, W., Wang, G.: 3-d convolutional encoder-decoder network for low-dose ct via transfer learning from a 2-d trained network. IEEE transactions on medical imaging **37**(6), 1522–1534 (2018)
61. Shi, B., Bai, S., Zhou, Z., Bai, X.: Deeppano: Deep panoramic representation for 3-d shape recognition. IEEE Signal Processing Letters **22**(12), 2339–2343 (2015). <https://doi.org/10.1109/LSP.2015.2480802>
62. Sketchup: 3d modeling online free|3d warehouse models. <https://3dwarehouse.sketchup.com> (2021)
63. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 945–953 (2015)
64. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: European Conference on Computer Vision (2020)
65. Wang, H., Liu, Q., Yue, X., Lasenby, J., Kusner, M.J.: Unsupervised point cloud pre-training via view-point occlusion, completion. arXiv preprint arXiv:2010.01089 (2020)
66. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. ACM Transactions on Graphics (TOG) **36**(4), 1–11 (2017)
67. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: CVPR (2019)
68. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog) **38**(5), 1–12 (2019)
69. Wang, Z., Zhan, W., Tomizuka, M.: Fusing bird’s eye view lidar point cloud and front view camera image for 3d object detection. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 1–6. IEEE (2018)
70. Wu, B., Wan, A., Yue, X., Keutzer, K.: Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: ICRA (2018)

71. Wu, B., Zhou, X., Zhao, S., Yue, X., Keutzer, K.: Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In: ICRA (2019)
72. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
73. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: European Conference on Computer Vision. pp. 574–591. Springer (2020)
74. Xu, C., Li, T., Tang, C., Sun, L., Keutzer, K., Tomizuka, M., Fathi, A., Zhan, W.: Pretram: Self-supervised pre-training via connecting trajectory and map. arXiv preprint arXiv:2204.10435 (2022)
75. Xu, C., Wu, B., Wang, Z., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In: European Conference on Computer Vision. pp. 1–19. Springer (2020)
76. Xu, C., Zhai, B., Wu, B., Li, T., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: You only group once: Efficient point-cloud processing with token representation and relation inference module. arXiv preprint arXiv:2103.09975 (2021)
77. Xu, X., Lee, G.H.: Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13706–13715 (2020)
78. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. *Sensors* **18**(10), 3337 (2018)
79. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7652–7660 (2018)
80. Yin, W., Liu, Y., Shen, C.: Virtual normal: Enforcing geometric constraints for accurate and robust depth prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021)
81. Yue, X., Wu, B., Seshia, S.A., Keutzer, K., Sangiovanni-Vincentelli, A.L.: A lidar point cloud generator: from a virtual world to autonomous driving. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval. pp. 458–464 (2018)
82. Zhang, J., Chen, L., Ouyang, B., Liu, B., Zhu, J., Chen, Y., Meng, Y., Wu, D.: Pointcutmix: Regularization strategy for point cloud classification. arXiv preprint arXiv:2101.01461 (2021)
83. Zhang, Z., Girdhar, R., Joulin, A., Misra, I.: Self-supervised pretraining of 3d features on any point-cloud. arXiv preprint arXiv:2101.02691 (2021)
84. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268 (2021)
85. Zhou, H., Zhu, X., Song, X., Ma, Y., Wang, Z., Li, H., Lin, D.: Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. arXiv preprint arXiv:2008.01550 (2020)