Implicit Autoencoder for Point-Cloud Self-Supervised Representation Learning

Siming Yan[1]   Zhenpei Yang[1]   Haoxiang Li[2]   Chen Song[1]   Li Guan[2]   Hao Kang[2]

Gang Hua[2]   Qixing Huang[1]

[1]The University of Texas at Austin   [2]Wormpex AI Research

Abstract

This paper advocates the use of implicit surface representation in autoencoder-based self-supervised 3D representation learning. The most popular and accessible 3D representation, i.e., point clouds, involves discrete samples of the underlying continuous 3D surface. This discretization process introduces sampling variations on the 3D shape, making it challenging to develop transferable knowledge of the true 3D geometry. In the standard autoencoding paradigm, the encoder is compelled to encode not only the 3D geometry but also information on the specific discrete sampling of the 3D shape into the latent code. This is because the point cloud reconstructed by the decoder is considered unacceptable unless there is a perfect mapping between the original and the reconstructed point clouds. This paper introduces the Implicit AutoEncoder (IAE), a simple yet effective method that addresses the sampling variation issue by replacing the commonly-used point-cloud decoder with an implicit decoder. The implicit decoder reconstructs a continuous representation of the 3D shape, independent of the imperfections in the discrete samples. Extensive experiments demonstrate that the proposed IAE achieves state-of-the-art performance across various self-supervised learning benchmarks. Our code is available at https://github.com/SimingYan/IAE.

1. Introduction

The rapid advancement and growing accessibility of commodity scanning devices have made it easier to capture vast amounts of 3D data represented by point clouds. The success of point-based deep networks [46, 47] additionally enables many 3D vision applications to exploit this natural and flexible representation. Exciting results have emerged in recent years, ranging from object-level understanding, including shape classification [5] and part segmentation [55, 75, 77], to scene-level understanding, such as 3D object detection [11, 17, 56] and 3D semantic segmentation [3].

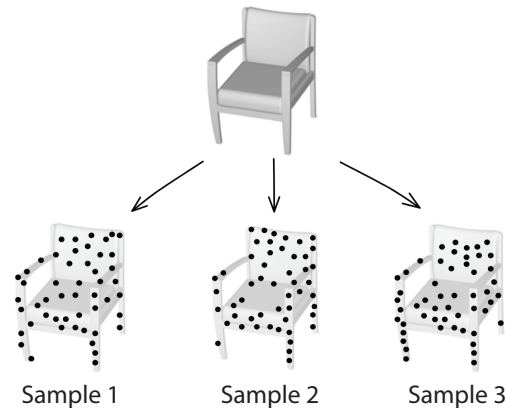Annotating point-cloud datasets is a highly labor-

Figure 1: **The Sampling Variation Problem**. Given a continuous 3D shape, there are infinitely many ways to sample a point cloud. The proposed Implicit AutoEncoder (IAE) learns a latent representation of the true 3D geometry independent of the specific discrete sampling process. By alleviating the sampling variation problem, IAE improves existing point-cloud self-supervised representation learning methods in various downstream tasks.

intensive task due to the challenges involved in designing 3D interfaces and visualizing point clouds. As a result, researchers are motivated to explore self-supervised representation learning paradigms. The fundamental concept is to pre-train deep neural networks on large unlabeled datasets and fine-tune them on smaller datasets annotated based on specific task requirements. A carefully designed self-supervised representation learning paradigm effectively initializes the network weights, enabling fine-tuning on downstream tasks (*e.g.*, classification and segmentation) to avoid weak local minima and achieve improved stability [15].

Research in self-supervised representation learning predates the history of 3D computer vision. Notably, significant effort has been dedicated to developing self-supervised learning methods for 2D images [6, 12, 31, 41, 69, 82, 83], with autoencoders being one of the most popular tools [4, 21, 41, 59, 63]. An autoencoder-based self-supervised representation learning pipeline comprises an encoder that transforms the input into a latent code and a decoder that expands the

latent code to reconstruct the input. Since the latent code has a much lower dimension than the input, the encoder is encouraged to summarize the input by condensed latent features with the help of the reconstruction loss.

There is a growing interest in developing self-supervised representation learning methods for point clouds by drawing inspiration from image-based autoencoders. For example, Yang et al. [76] propose a novel folding-based decoder, and Wang et al. [66] develop a denoising autoencoder based on the standard point-cloud completion model. However, to the best of our knowledge, prior works in point-cloud self-supervised representation learning have relied on the same design principles as image-based methods, where **both the encoder and decoder represent the 3D geometry in the same format** (*i.e.*, point clouds).

Point clouds are noisy, discretized, and unstructured representations of 3D geometry. As shown in Figure 1, a 3D shape can be represented by many different point clouds, all of which are valid representations of the same object. Different point cloud samples are subject to different noises, which are induced from various sources such as the intrinsic noises from the sensors and the interference from the environment. The unordered and noisy nature distinguishes point clouds from conventional structured data, such as pixel images defined on rectangular grids. When training a point-cloud autoencoder, the encoder is forced to capture *sampling variations*, limiting the model's ability to extract valuable information about the true 3D geometry.

This paper, for the first time, formalizes the concept of sampling variations and proposes to combine the implicit surface representation with point-cloud self-supervised representation learning. Specifically, we introduce an asymmetric point-cloud autoencoder scheme, where the encoder takes a point cloud as input, and the decoder uses the implicit function as the output 3D representation. The rest of the paper refers to this design as the Implicit AutoEncoder (IAE). IAE enjoys many advantages over existing methods. First, reconstruction under the implicit representation discourages latent space learning from being distracted by the imperfections brought by sampling variations and encourages the encoder to capture generalizable features from the true 3D geometry. Second, the reconstruction loss defined on implicit functions bypasses the expensive and unstable data association operation commonly used in point-based reconstruction losses such as the Earth Mover Distance [52] and the Chamfer Distance [16]. The added efficiency allows IAE to process up to 40k input points with a single Tesla V100 GPU, making it possible to capture very fine geometric details when existing methods can only work with sparse data with approximately 1k points.

To demonstrate the effectiveness of IAE, we conduct experiments to verify that the learned representation from our pre-trained model can adapt to various downstream tasks,

both at the object level and the scene level, including shape classification, linear evaluation, object detection, and indoor semantic segmentation. IAE consistently outperforms state-of-the-art methods in all settings. Specifically, under the best setting, IAE achieves 88.2% / 94.3% classification accuracy on ScanObjectNN [60] / ModelNet40 [68] respectively. IAE is also the first to support the autoencoding paradigm in scene-level pre-training for the application to various scene-level downstream tasks. For example, compared to training from scratch, IAE achieves +0.7% and +1.1% absolute improvements in object detection quality evaluated by mAP@0.5 on ScanNet [11] and SUN RGB-D [56], respectively.

We summarize the contributions of our paper as follows:

- We propose an asymmetric point-cloud autoencoder called Implicit AutoEncoder (IAE). The IAE takes a point cloud as input and uses an implicit function as the output 3D representation. We combine the implicit surface representation with point-cloud self-supervised representation learning for the first time.

- We formalize the concept of sampling variations in point clouds and demonstrate that IAE is more effective at capturing generalizable features from true 3D geometry than standard point-cloud autoencoders.

- We conduct experiments to demonstrate the effectiveness of IAE on various downstream tasks, including shape classification, object detection, and indoor semantic segmentation. IAE consistently outperforms state-of-the-art methods in all settings.

2. Related Work

Point-Cloud Self-Supervised Representation Learning. The unordered nature of 3D point clouds poses a unique challenge to representation learning, which is known to be effective on conventional structured data representations like images. Several point-cloud self-supervised methods have been developed to learn representations from individual 3D objects [1, 7, 20, 25, 44, 54, 66, 74, 76, 78], with the synthetic ShpapeNet [5] being the most popular dataset used in pre-training. However, existing methods exhibit limited generalizability to real scene-level datasets due to a significant domain gap [72, 73].

The remarkable success achieved in 2D has sparked a growing interest in extending the contrastive learning paradigm to self-supervised representation learning on point clouds [51, 72, 80]. These approaches rely on the contrastive loss to separate positive and negative instance pairs in the latent space, necessitating meticulous design efforts to define similar and dissimilar examples. They also consume a substantial amount of computational resources, particularly when working with large datasets. While 3D contrastive

learning is a popular field of research, this paper addresses the sampling variation problem under the simple autoencoder paradigm discussed below.

Point-Cloud Autoencoders. Several existing works exploit the autoencoder paradigm for point-cloud self-supervised representation learning. They simultaneously train an encoder and a decoder, with the encoder mapping the input point cloud to a latent code, and the decode reconstructing another point cloud from the latent code [38, 66, 76, 79]. While these approaches have demonstrated promising performance improvements, the sampling variation problem remains a significant challenge when the decoder output is represented as point clouds. This is because the network must encode not only the geometry but also the imperfections of the specific point cloud sampling into the latent code. Additionally, training a decoder under a point-based network architecture is not as straightforward as training a regular convolutional or fully-connected model, which limits the quality of the output. This paper proposes the use of implicit representation as the decoder output in the autoencoder-based self-supervised representation learning paradigm. As we will see, the implicit decoder alleviates the sampling variation issue and allows for a more efficient pipeline than existing approaches. Closely related to our design, ParAE [13] learns the 3D data distribution through discrete generative models. While ParAE builds the supervision upon a point-wise partitioning matrix, the proposed IAE utilizes implicit representations that are conceptually more relevant to the 3D geometry and easier to train.

Implicit Representations. A 3D shape can be represented implicitly by a deep network that maps 3D coordinates to signed distances [33, 39] or occupancy grids [8, 32, 42]. Unlike explicit representations such as point clouds, voxels, or triangle meshes, implicit representations offer a continuous shape parameterization and do not suffer from discretization errors. Implicit representations have been successfully applied in various 3D tasks, including 3D reconstruction from images [27, 28, 36], primitive-based 3D reconstruction [18, 19, 40], 4D reconstruction [35], and continuous texture representation [37]. However, to the best of our knowledge, no existing research has investigated the integration of implicit representations into the self-supervised representation learning paradigm, making IAE the pioneer in the field.

3. Method

3.1. Problem Setup

Assume we have access to a large dataset of 3D point clouds $\mathbf{D} = \{\mathcal{P}|\mathcal{P} \in \mathbb{R}^{n \times 3}\}$ without any annotations. The goal of autoencoder-based point-cloud self-supervised learning is to train an encoder network f_Θ and a decoder network g_Φ, where the encoder maps the point cloud to an

m-dimensional latent code:

$$f_\Theta : \mathbb{R}^{n \times 3} \to \mathbb{R}^m, m \ll n \qquad (1)$$

and the decoder maps the latent back to the 3D geometry:

$$g_\Phi : \mathbb{R}^m \to \mathbb{S} \qquad (2)$$

Here, $\mathbb{S}$ refers to the space of 3D shapes under the representation of choice. In practice, a popular choice is to use a sub-sampled version of the point cloud as the output shape representation: $\mathbb{S} = \mathbb{R}^{n' \times 3}$, where $n' \leq n$. The parameters Θ and Φ are jointly trained by minimizing the distance metric d between the input point cloud and the reconstructed shape:

$$\Theta^*, \Phi^* = \underset{\Theta, \Phi}{\arg \min} \, d((g_\Phi \circ f_\Theta)(\mathcal{P}), \mathcal{P}) \qquad (3)$$

After training the autoencoder, the encoder f_{Θ^*} is fine-tuned on a small dataset with task-specific annotations (*e.g.*, bounding boxes and segmentation labels).

3.2. Challenges

Sampling Variations. As shown in Figure 1, there are infinitely many valid point clouds representing the same 3D shape. This is because point clouds are discrete samples of the continuous 3D geometry. Due to the finite sampling size and noisy conditions, each point cloud contains a unique set of imperfections from the sampling process. Ideally, an autoencoder is expected to encode the information just about the true 3D geometry into the latent code. In practice, the encoder is compelled to capture the imperfections in the latent code since the decoder output is represented as another point cloud with a distance metric in place enforcing the two discrete samples to be identical. Sampling variation causes distractions to the autoencoder learning and leads to suboptimal performance in downstream tasks.

Efficient Distance Calculation. The Earth Mover Distance (EMD) [52] in Eq. 4 and the Chamfer Distance (CD) [16] in Eq. 5 are commonly used metrics to calculate the distance between the input and the reconstructed point clouds. For each point $\mathbf{x}$ in point cloud $\mathcal{P}$, EMD and CD find the nearest point in the other cloud $\hat{\mathcal{P}}$ for distance calculation. The correspondence discovery operation makes both EMD and CD expensive metrics to use, especially when the point clouds are dense.

$$d_{\text{EMD}}(\mathcal{P}, \hat{\mathcal{P}}) = \min_{\phi:\mathcal{P} \to \hat{\mathcal{P}}} \sum_{\mathbf{x} \in \mathcal{P}} ||\mathbf{x} - \phi(\mathbf{x})||_2 \qquad (4)$$

$$d_{\text{CD}}(\mathcal{P}, \hat{\mathcal{P}}) = \sum_{\hat{\mathbf{x}} \in \hat{\mathcal{P}}} \min_{\mathbf{x} \in \mathcal{P}} ||\hat{\mathbf{x}} - \mathbf{x}||_2 + \sum_{\mathbf{x} \in \mathcal{P}} \min_{\hat{\mathbf{x}} \in \hat{\mathcal{P}}} ||\mathbf{x} - \hat{\mathbf{x}}||_2 \quad (5)$$

Data Density. Many large-scale open 3D datasets provide very dense point clouds. For example, the popular ShapeNet dataset [5] contains at least 50k points per shape. Due to the inefficiency discussed above, existing explicit autoencoders
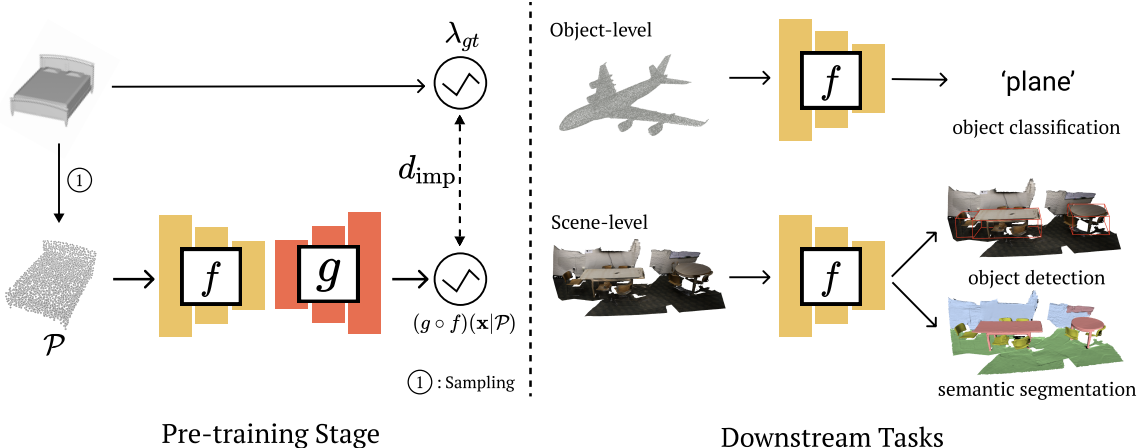
Figure 2: **The Implicit AutoEncoder.** In the **pre-training stage**, we jointly train an encoder f_Θ and a decoder g_Φ to reconstruct the underlying 3D shape from the input[1]. While the encoder takes the point cloud as input, the output of the decoder is represented by an implicit function. The implicit function maps a 3D query point $\mathbf{x} = (x, y, z)$ to the signed [33, 39], unsigned [10], or occupancy [8, 32, 42] representation of the shape. After pre-training, we fine-tune the encoder f_Θ on different **downstream tasks**. At the object level, we evaluate IAE on object classification. At the scene level, we evaluate IAE on object detection and semantic segmentation.

lack the ability to exploit such density and are limited to a sparely sub-sampled version of the original dense point cloud. Sub-sampling introduces information loss and additional sampling variations. If it *could* use dense point clouds during training, an explicit autoencoder *would* potentially be as good as the proposed IAE. The problem, however, is that it is non-trivial to use dense data in explicit autoencoders. A naive adoption of the EMD [52] and CD [16] losses leads to an unaffordable computation cost. We show more analysis in Section 4.5.

3.3. The Implicit AutoEncoder (IAE)

Design. As illustrated by Figure 2, we propose the Implicit AutoEncoder (IAE), which elegantly addresses the challenges discussed above. In IAE, while the encoder still takes a point cloud as input, the decoder outputs an implicit function: $\mathbb{S} = \{\lambda | \lambda(\mathbf{x}) \in \mathbb{R}, \forall \mathbf{x} \in \mathbb{R}^3\}$. Let λ_{gt} be the "ground-truth" implicit function derived analytically from the input shape, the training objective of IAE is to minimize the distance between two implicit surfaces through the distance metric d_{imp}:

$$\Theta^*, \Phi^* = \arg\min_{\Theta, \Phi} d_{\text{imp}}((g_\Phi \circ f_\Theta)(\mathbf{x}|\mathcal{P}), \lambda_{\text{gt}}(\mathbf{x})) \quad (6)$$

Our experiments use three popular variants of the implicit function, including the signed distance function [33, 39], the unsigned distance function [10], and the occupancy grid [8, 32, 42]. When the output shape is represented by either the signed (SDF) or the unsigned (UDF) distance function, d_{imp} is implemented as the mean $\mathcal{L}_1$ distance between the ground-truth and reconstructed distance functions on N uniformly

sampled points inside the 3D bounding box:

$$d_{\text{sdf}} = \frac{1}{N} \sum_{\mathbf{x} \in X} |(g_\Phi \circ f_\Theta)(\mathbf{x}|\mathcal{P}) - s(\mathbf{x})| \quad (7)$$

$$d_{\text{udf}} = \frac{1}{N} \sum_{\mathbf{x} \in X} ||(g_\Phi \circ f_\Theta)(\mathbf{x}|\mathcal{P})| - u(\mathbf{x})| \quad (8)$$

where s denotes SDF value of sampled point $\mathbf{x}$, and u denotes UDF value of $\mathbf{x}$.

On the other hand, when the output shape is represented by the occupancy grid, d_{imp} becomes the cross-entropy loss:

$$d_{\text{occ}} = \frac{1}{N} \sum_{\mathbf{x} \in X} CE((g_\Phi \circ f_\Theta)(\mathbf{x}|\mathcal{P}), o(\mathbf{x})) \quad (9)$$

where CE denotes cross entropy, o denotes occupancy.

4. Experiments

4.1. Pre-Training Datasets

ShapeNet [5] is a synthetic dataset commonly used in 3D vision, consisting of 57,748 shapes from 55 object categories. In addition to point clouds, ShapeNet offers high-quality water-tight triangular meshes as a more accurate description of 3D geometry. Following [32, 39], we generate ground-truth signed distance functions and occupancy grids from the provided mesh data. While ShapeNet is able to provide such high-quality 3D geometric information thanks to its synthetic nature, water-tight meshes are rarely available in real-world settings. To assess the effectiveness of IAE under more realistic conditions, we further use the nearest neighbor strategy [43] to obtain unsigned distance functions from the point clouds in ShapeNet (50k points).

ScanNet [11] is a real-world dataset with over 1,500 point-cloud scans. Unlike ShapeNet [5], which focuses on

---

[1]For simplification, we denote f as f_Θ, g as g_Φ, and do not show the query point $\mathbf{x}$ in Figure 2.

Table 1: **Classification results on ScanObjectNN and ModelNet40 datasets.** The model parameters number (#Params), GFLOPS, and overall accuracy (%) are reported. The FULL section presents the results of fine-tuning the pre-trained models. The LINEAR section reports the results of training a linear SVM. To establish a fair comparison, we present the performance of IAE using two distinct architectures: DGCNN and Transformer-like Point-M2AE. These architectures are commonly adopted in prior methods. The methods within the same block share the same architecture. 'w/o mesh' indicates that pre-training does not involve mesh data.

| Method | #Params(M) | GFLOPS | ScanObjectNN | | | ModelNet40 | |
			OBJ_BG	OBJ_ONLY	PB_T50_RS	1k P	8k P
Supervised Learning Only							
PointNet [47]	3.5	0.5	73.3	79.2	68.0	89.2	90.8
PointNet++ [46]	1.5	1.7	82.3	84.3	77.9	90.7	91.9
PointCNN [26]	0.6	-	86.1	85.5	78.5	92.2	-
KPConv [58]	-	-	86.4	84.1	80.2	92.9	-
GBNet [49]	8.8	-	-	-	81.0	93.8	-
PointMLP [30]	12.6	31.4	-	-	85.7	94.1	-
PointNeXt [48]	1.4	3.6	-	-	88.1	93.2	-
with Self-Supervised Representation Learning (FULL)							
DGCNN [67]	1.8	2.4	82.8	86.2	78.1	92.9	93.1
JigSaw [54]	1.8	2.4	-	-	83.5	92.6	-
STRL [23]	1.8	2.4	-	-	-	93.1	-
OcCo [65]	1.8	2.4	88.2	87.5	84.3	93.0	-
IAE (DGCNN)	1.8	2.4	**90.2**	**89.0**	**85.6**	**94.2**	**94.2**
Transformer [78]	22.1	4.8	83.0	84.1	79.1	91.4	91.8
Point-BERT [78]	22.1	4.8	89.3	88.1	84.3	93.2	93.8
Point-MAE [38]	22.1	4.8	90.0	88.3	85.2	93.8	94.0
Point-M2AE [79]	15.3	3.6	91.2	88.8	86.4	94.0	-
IAE (M2AE) w/o mesh	15.3	3.6	92.3	91.2	88.0	94.1	94.2
IAE (M2AE)	15.3	3.6	**92.5**	**91.6**	**88.2**	**94.2**	**94.3**
with Self-Supervised Representation Learning (LINEAR)							
JigSaw [54]	1.8	2.4	-	-	59.5	84.1	-
STRL [23]	1.8	2.4	-	-	77.9	90.9	-
OcCo [65]	1.8	2.4	-	-	78.3	89.7	-
CrossPoint [2]	1.8	2.4	-	-	81.7	91.2	-
IAE (DGCNN)	1.8	2.4	-	-	**83.0**	**92.1**	-
Point-M2AE [79]	15.3	3.6	-	-	83.1	92.9	-
IAE (M2AE) w/o mesh	15.3	3.6	-	-	84.3	93.1	-
IAE (M2AE)	15.3	3.6	-	-	**84.4**	**93.2**	-

individual objects, ScanNet collects indoor scene data with a significantly larger perceptive range. We apply the sliding window strategy and crop each instance into $d \times d \times d$ cubes, where $d = 3.0$ m. Following Qi et al. [45], we divide the dataset into training (8k point clouds) and validation (2.6k point clouds) splits. We randomly sub-sample 10,000 points from each point cloud as the input to the encoder. Due to the lack of high-quality water-tight meshes, we use the nearest neighbor strategy [43] to calculate ground-truth unsigned distance functions. We refer interested readers to the supplementary material for implementation details. Importantly, we point out that IAE is the first method that supports the autoencoding paradigm in real-world scene-level training. As observed by Xie et al. [72], pre-training on synthetic datasets exhibits limited generalizability to

real-world tasks. Real-world scene-level pre-training is extremely challenging due to noises, incomplete scans, and the complication of indoor scene contents. As a result, existing self-supervised methods fail to achieve any noticeable performance improvement [44, 54, 66].

4.2. Pre-Training Setting

As outlined in Section 3, we propose a general framework for Implicit Autoencoder. The IAE consists of an encoder part that maps a given point cloud into a high-dimensional latent code, and a decoder part that fits an implicit function to the latent code and query point to output the implicit function value. Our framework can easily incorporate any backbone architecture for point cloud understanding into the encoder part. In order to ensure a fair comparison, we adopt the same

Method	ScanNet		SUN RGB-D	
	AP_{25}	AP_{50}	AP_{25}	AP_{50}
VoteNet [45]	58.6	33.5	57.7	32.9
3D-MPA [14]	64.2	49.2	-	-
MLCVNet [71]	64.5	41.4	59.8	-
BRNet [9]	66.1	50.9	61.1	43.7
3DETR [34]	65.0	47.0	59.1	32.7
GroupFree [29]	69.1	52.8	63.0	45.2
FCAF3D [53]	71.5	57.3	64.2	48.9
CAGroup3D [64]	75.1	61.3	66.8	50.2
STRL [23]	59.5	38.4	58.2	35.0
RandomRooms [51]	61.3	36.2	59.2	35.4
PointContrast [72]	59.2	38.0	57.5	34.8
DepthContrast[2] [80]	62.1	39.1	60.4	35.4
Point-M2AE [79]	66.3	48.3	-	-
IAE (VoteNet)	61.5	39.8	60.4	36.0
IAE (CAGroup3D)	**76.1**	**62.0**	**67.6**	**51.3**

Table 2: **3D Object Detection Results.** We fine-tune our pre-trained model on ScanNetV2 [11] and SUN-RGBD [56] validation sets using VoteNet [45] and CAGroup3D [64]. We show mean Average Precision (mAP) across all semantic classes with 3D IoU thresholds of 0.25 and 0.5. Methods in the second section denote self-supervised methods.

Method	S3DIS 6-Fold	
	OA	mIoU
PointNet [46]	78.5	47.6
PointCNN [26]	88.1	65.4
DGCNN [67]	84.1	56.1
DeepGCN [24]	85.9	60.0
KPConv [58]	-	70.6
RandLA-Net [22]	88.0	70.0
BAAF-Net [50]	88.0	70.0
PointTransformer [81]	90.2	73.5
CBL [57]	89.6	73.1
PointNeXt [48]	90.3	74.9
Jigsaw [54]	84.4	56.6
OcCo [66]	85.1	58.5
STRL [23]	84.2	57.1
IAE (DGCNN)	85.9	60.7
IAE (PointNeXt)	**90.8**	**75.3**

Table 3: **Semantic Segmentation Results on S3DIS [3] 6-Fold.** We show Overall Accuracy (OA) and mean Intersection over Union (mIoU) across six folds.

backbone architecture as previous approaches.

The decoder part of the IAE fits an implicit function that takes both the latent code obtained from the encoder and the query point as input, and outputs the implicit function value at that point. We explore two different designs for the decoder part based on Occupancy Network-style [32] and Convolutional Occupancy Network-style [42]. The pretrained models used for downstream tasks in our experiments are based on the Convolutional Occupancy Network-style decoder. Further implementation details are provided in the supplementary material.

4.3. Downstream Tasks

After pre-training IAE on ShapeNet [5] and ScanNet [11], we discard the decoder in pre-training and append different network heads onto the pre-trained encoder for different downstream tasks, at both the object level and the scene level.

4.3.1 Shape Classification

We assess the quality of object-level feature learning on ModelNet40 [68] and ScanObjectNN [61] datasets. ModelNet40 consists of 9,832 training objects and 2,468 testing objects across 40 different categories. We additionally pre-process the dataset following the procedure developed by Qi et al [46]. ScanObjectNN contains approximately 15,000 objects from 15 categories, which are scanned from real-world indoor scenes with cluttered backgrounds. We conduct

experiments on three variants: OBJ-BG, OBJ-ONLY, and PBT50-RS. Details are provided in supplementary materials.

Linear SVM Evaluation. To evaluate the 3D representation learning ability of IAE, we first conduct linear classification experiments on ModelNet40 [68] and ScanObjectNN [61] after pre-training on ShapeNet. We use our frozen pre-trained encoder to extract features from 1,024 points sampled from each 3D shape and train a linear SVM classifier on top of them. The classification accuracies are reported in Table 1 LINEAR section. To establish a fair comparison with previous approaches, we construct two different encoders, DGCNN [67] and M2AE [79]. Under M2AE backbone, IAE achieves 93.2%/84.4% classification accuracy on the ModelNet40/ScanObjectNN testing split, representing a 0.3%/1.3% absolute improvement from the state-of-the-art method. Furthermore, we highlight that IAE is capable of surpassing state-of-the-art performance even without high-quality water-tight mesh labels. In the second-to-last row of Table 1 LINEAR section, we present the results when the ground-truth implicit representation is directly generated from the dense point cloud on ShapeNet using the nearest neighbor strategy. The 1.2% absolute improvement on ScanObjectNN indicates that even though point clouds are discrete samples of the true 3D geometry, converting point clouds to the implicit format is an effective way of alleviating sampling variations.

Supervised Fine-Tuning. In addition to fitting an SVM in the latent space, we fine-tune the encoder pre-trained on ShapeNet using ModelNet40 [68] and ScanObjectNN [61] labels. We again adopt both DGCNN [67] and M2AE [79]

---

[2]DepthContrast used a slightly larger model than Votenet. For a fair comparison, we reproduce DepthContrast with the original Votenet model.

Task	Pre-train	Acc/AP$_{25}$
Object detection	ScanNet	**67.6**
	ShapeNet	67.1
MN40 Linear	ScanNet	91.3%
	ShapeNet	**92.1%**

Table 4: **Cross-Domain Generalizability** between ShapeNet [5] and ScanNet [11]. For the 3D object detection task, we report mAP at IoU=0.25 on the SUN RGB-D dataset [56]. For ModelNet40 [68] linear evaluation, we report the classification accuracy.

architecture to construct the encoder network for a fair comparison, whose outputs are utilized as class predictions in this downstream task. As shown in Table 1 FULL section, IAE shows consistent improvements and achieves state-of-the-art performance with 94.3%/88.2% classification accuracy on ModelNet40/ScanObjectNN respectively. IAE also surpasses previous state-of-the-art with 94.2%/88.0% classification accuracy when the ground-truth implicit function is computed without meshes.

4.3.2 Indoor 3D Object Detection

By leveraging the convolutional occupancy network [42], IAE demonstrates a robust ability to handle complex 3D scenes. We build the encoder with the VoteNet [45] architecture and fine-tune the weights using ScanNet [11] detection labels after self-supervised pre-training on the same dataset. As shown in Table 2, IAE outperforms random weight initialization with 18.8% and 4.9% relative improvements in mAP@0.5 and mAP@0.25, respectively. In addition to ScanNet, we further evaluate IAE on the more challenging SUN RGB-D dataset [56] and observe similar performance improvements.

Recently, Wang et al. [64] propose the CAGroup3D model, which outperforms the commonly used VoteNet [45] by a significant margin. Table 2 demonstrates that incorporating CAGroup3D as the IAE encoder further boosts the performance over the primitive CAGroup3D model.

4.3.3 Indoor 3D Semantic Segmentation

On the Stanford Large-Scale 3D Indoor Spaces (S3DIS) dataset [3], we assess how well IAE can transfer its knowledge in self-supervised pre-training to scene-level semantic segmentation. S3DIS consists of 3D point clouds collected from 6 different large-area indoor environments, annotated with per-point categorical labels. Following Qi et al. [46], we divide each room instance into 1m × 1m blocks. We randomly sub-sample 4,096 points from the original point cloud as the encoder input and use 6-fold cross-validation during fine-tuning. As shown in Table 3, IAE significantly improves baseline approaches when the encoder is constructed from the same DGCNN [67] architecture, with an 0.8% absolute improvement in overall accuracy
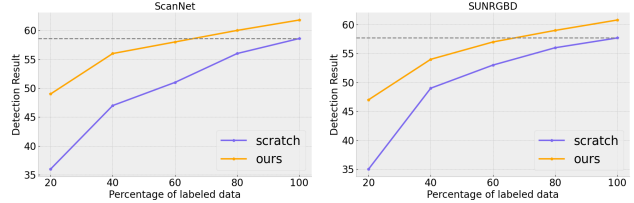

Figure 3: **Label efficiency training.** We pre-train our model on ScanNet and then fine-tune on ScanNet and SUN RGB-D separately. During fine-tuning, different percentages of labeled data are used. Our pre-training model outperforms training from scratch and achieves nearly the same result with only 60% labeled data.

and 2.2% improvement in mIoU. When integrated with PointNeXt [48], the latest model design in the field, IAE is still able to outperform the from-scratch training.

4.3.4 Cross-Domain Generalizability

Up to this point, our discussion on scene-level fine-tuning assumes the pre-training process uses scene-level data as well. Whether synthetic object-level pre-training helps real-world scene-level fine-tuning remains an open debate in 3D vision. Xie et al. [72] suggest pre-training on ShapeNet [5] may have a detrimental effect on ScanNet [11] fine-tuning. In contrast, Huang et al. [23] demonstrate that with a sufficiently powerful encoder, object-level pre-training is actually beneficial regardless of whether the downstream task is at the object level or the scene level. As shown in Table 4, with ShapeNet pre-training, IAE achieves a mAP@0.25 of 67.1 on the SUN RGB-D dataset [56] for object detection. Albeit not as good as scene-level pre-training, object-level pre-training demonstrates a noticeable improvement over training-from-scratch (66.8 mAP@0.25). On ModelNet40 [68], we use linear evaluation to determine the effect on scene-level classification. While it is meaningless to fit an SVM using an encoder with randomly initialized weights, Table 4 demonstrates that pre-training the IAE encoder on both ScanNet and ShapeNet leads to state-of-the-art classification accuracy under DGCNN [67] backbone (*c.f.*, Table 1). Empirically, we present additional evidence to support Huang et al.'s claim in the open debate.

4.3.5 Label Efficiency Training

Pre-training helps models to be fine-tuned with a small amount of labeled data. We study the labeling efficiency of our model on 3D object detection by varying the portion of supervised training data. Results can be found in Figure 3. We take the VoteNet [45] as the training network and use 20%, 40%, 60%, and 80% of the training data from ScanNet and SUN RGB-D datasets. We observe that our pre-training method gives larger gains when the labeled data is less. With

	Method	MN40
EAE	FoldingNet	90.1%
	OcCo	89.7%
	Snowflake	89.9%
IAE	OccNet	91.5%
	ConvONet	**92.1%**

(a) Different decoders

	Method	SUN RGB-D
EAE	FoldingNet	58.2
	OcCo	58.4
	Snowflake	58.1
IAE	ConvOccNet	**60.4**

(b) Different decoders on real data task

	Function	MN40
EAE	PC	90.1%
IAE	Occ	91.3%
	UDF	91.7%
	SDF	**92.1%**

(c) Different implicit functions

Table 5: (a) **Ablation Study on Different Decoder Models** on shape-level ModelNet40 [68] linear evaluation. Our implicit autoencoders (IAE) consistently outperform their explicit counterparts (EAE). (b) **Ablation Study on Different Decoder Models** on real data task, mAP at IoU=0.25 on SUN RGB-D [56] detection. Our approach using the convolutional occupancy network shows consistent improvements. (c) **Ablation Study on Implicit Functions.** For explicit representation, we use FoldingNet [76] as the decoder and point cloud (PC) at the output. Three implicit representations show consistent improvements over the explicit representation.
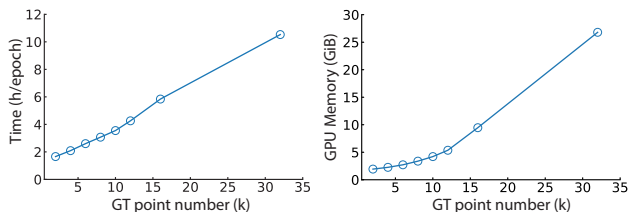

Figure 4: **Computation time and GPU memory utilization of explicit autoencoder.** We use the standard explicit autoencoder framework, OcCo [65], and take DGCNN [67] as the encoder backbone. We use Chamfer Distance as the loss function.

Figure 5: **Experiment analysis on Explicit and Implicit Autoencoders. (a)** T-SNE visualization for encoder latent codes of IAE (blue) and EAE (orange). The IAE clusters are noticeably smaller. **(b)** The average cluster radius for IAE and EAE in (a).

only 60% training data on ScanNet/SUN RGB-D, our model can get similar performance compared with using all training data from scratch. This suggests our pre-training can help the downstream task to obtain better results with less data.

4.4. Ablation Study

We proceed with an ablation study to understand the different design choices we made in our experiments.

Explicit vs. Implicit Decoders. Under otherwise identical settings, we conduct two sets of parallel experiments to compare conventional explicit decoders with our proposed implicit decoders. Specifically, we choose three state-of-the-art explicit models, including FoldingNet [76], OcCo [66], and SnowflakeNet [70], and compare them with two different implicit models, including Occupancy Network [32], and Convolutional Occupancy Network [42]. We pre-train all models on ShapeNet [5] and report the linear SVM classification accuracies on ModelNet40 [68] under DGCNN [67] backbone. As shown in Table 5 (a), implicit decoders consistently improve all explicit methods. Furthermore, we performed a comparison on real data by pre-training the models on ScanNet and reporting the SUN RGB-D detection result. As shown in Table 5 (b), the implicit decoder outperforms the explicit methods.

Choice of Implicit Representations. Among the signed distance function [33, 39], unsigned distance function [10], and occupancy grid [8, 32, 42], what is the implicit representation of choice in self-supervised representation learning? As
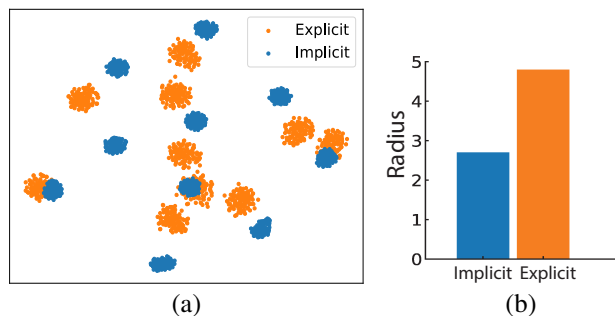
shown in Table 5 (c), while they all outperform the explicit representation, the signed distance function demonstrates the highest linear evaluation accuracy on ModelNet40 [68] classification with ShapeNet [5] pre-training. When ground-truth signed distance functions are unavailable due to the lack of high-quality meshes, the unsigned distance function becomes the second-best choice.

4.5. Analysis

In this section, we show more experiment analysis between explicit autoencoders and implicit autoencoders.

Computational Cost for Loss Function. As mentioned in Section 3.2, existing explicit autoencoders lack the ability to utilize dense point clouds from open 3D datasets, but sub-sampling inevitably introduces information loss and additional sampling variation. To verify this issue, we implemented the OcCo [65] framework and tested its computation time and GPU memory utilization as the ground truth point cloud density increased. In Figure 4, we observe that by applying Chamfer loss and training the model on a single Tesla V100 GPU with a batch size of 1, the training time increases to over 10.53 hours per epoch, and the GPU memory usage increases to 26.8 GiB as the number of points for the ground-truth point cloud increased to 32k. Since pre-training usually takes at least 200 epochs, this cost is

unacceptable. In comparison, IAE only takes 0.3 hours per epoch with 6 GiB GPU memory.

Sampling Variation Problem. We conduct additional experiments to gain insights into the internal mechanism of IAE. Specifically, we pre-train both IAE and an explicit autoencoder, as discussed in Section 4.3.1, on the ShapeNet dataset [5]. Next, we randomly sample 100 different point clouds from 10 characteristic shapes and use the pre-trained encoders to predict latent codes. We use T-SNE [62] to obtain embeddings of the latent codes in $\mathbb{R}^2$. As visualized in Figure 5 (a), both the implicit and explicit models are successful in mapping the shapes into 10 distinctive clusters. However, the implicit clusters are noticeably smaller in size, with an average radius of only 56.3% of the explicit clusters (Figure 5 (b)). This result supports our claim that IAE is less sensitive to sampling variations and is able to learn more generalizable features about the true 3D geometry.

5. Analysis of IAE under a Linear AE model

We proceed to gain more insights of IAE under a linear AE model.

Specifically, suppose we have $N > n$ points $\boldsymbol{x}_i \in R^n, 1 \le i \le N$. Without losing generality, we assume $\boldsymbol{x}_i$ lies on a low-dimensional linear space $\{L\}$ of dimension $m < n$. These data points are used to model the underlying 3D models. Now let us perturb each point $\boldsymbol{x}'_i = \boldsymbol{x}_i + \epsilon_i$, where ϵ_i is used to model sampling variations. We assume $\epsilon_i \in \{L\}^\perp$, meaning they encode variations that are orthogonal to variations among the underlying 3D models. Denote $X := (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$ and $X' := (\boldsymbol{x}'_1, \cdots, \boldsymbol{x}'_N)$.

We consider two linear autoencoding models. The first one, which is analogous to IAE, takes $\boldsymbol{x}'_i$ as input and seeks to reconstruct $\boldsymbol{x}_i$:

$$A^\star, Q^\star = \underset{A', Q' \in \mathbb{R}^{n \times m}}{\operatorname{argmin}} \sum_{k=1}^{N} \|A'Q'^T \boldsymbol{x}'_k - \boldsymbol{x}_k\|^2$$
$$s.t.\ Q'^T Q' = I_m,\ Q' \in \{C\}(X')$$

Here Q' is the encoder, and A' is the decoder. $\{C\}(X')$ denotes the column space of matrix X'. The constraints $Q'^T Q' = I_m$ and $Q' \in \{C\}(X')$ ensure that the encoder-decoder pair is unique up to a unitary transformation in $O(m)$. Below we show that $Q^\star$ is independent of ϵ_k.

Proposition 1. *Let $Q \in \mathbb{R}^{n \times m}$ collect the top-m eigenvectors of the convariance matrix $C = \sum\limits_{k=1}^{N} \boldsymbol{x}_k \boldsymbol{x}'_k$. Then under the assumption that $\epsilon_k \in \{L\}^\perp, 1 \le k \le N$, $Q^\star = Q$.*

Prop. 1 indicates that $Q^\star$ does not encode sampling variations.

Now consider the second model where we force the encoder-decoder pair to reconstruct the original inputs,

which is analogous to standard autoencoding:

$$\hat{A}^\star, \hat{Q}^\star = \underset{A', Q' \in \mathbb{R}^{n \times m}}{\operatorname{argmin}} \sum_{k=1}^{N} \|A'Q'^T \boldsymbol{x}'_k - \boldsymbol{x}'_k\|_{\mathcal{F}}^2$$
$$s.t.\ Q'^T Q' = I_m,\ Q' \in \{C\}(X')$$

In this case, $\hat{A}^\star = \hat{Q}^\star$, and both of them are given by the top m eigenvectors of $C' = \sum \boldsymbol{x}'_k \boldsymbol{x}'^T_k$.

To quantitatively compare encoders $\hat{Q}^\star$ and Q, we need the following definition.

Definition 1. *Consider two unitary matrices $Q_1, Q_2 \in \mathbb{R}^{n \times m}$ where $Q_i^T Q_i = I_m$. We define the deviation between them as*

$$\mathcal{D}(Q_1, Q_2) := Q_1 - Q_2 R^\star,$$
$$R^\star = \underset{R \in O(m)}{\operatorname{argmin}} \|Q_1 - Q_2 R\|_{\mathcal{F}}^2$$

The following proposition specifies the derivatives between $\hat{Q}^\star$ and ϵ_k.

Proposition 2. *Under the assumption that $\epsilon_k \in \{L\}^\perp, 1 \le k \le N$, we have*

$$\frac{\partial \mathcal{D}(\hat{Q}^\star, Q)}{\partial \epsilon_{ki}} = (I_n - QQ^T)(\boldsymbol{e}_i \boldsymbol{x}_k^T) Q \Lambda^+ \qquad (10)$$

where $\Lambda = \operatorname{diag}(\lambda_1, \cdots, \lambda_m)$ is a diagonal matrix that collects the top eigenvalues of C that correspond to Q. $\boldsymbol{e}_k$ is the k-th standard basis vector.

In other word, $\hat{Q}^\star$ is sensitive to ϵ_k. Therefore, it encodes sampling variations. This theoretical analysis under a simplified setting suggests that IAE may potentially learn more robust representations.

6. Conclusion

We present Implicit AutoEncoder (IAE), a simple yet effective model for point-cloud self-supervised representation learning. Unlike conventional point-cloud autoencoders, IAE exploits the implicit representation as the output of the decoder. IAE prevents latent space learning from being distracted by sampling variations and encourages the encoder to capture generalizable features from the true 3D geometry. Extensive experiments demonstrate that IAE achieves considerable improvements over a wide range of downstream tasks, including shape classification, linear evaluation, object detection, and indoor semantic segmentation. In the future, we plan to extend IAE to support not only hand-crafted but also trainable implicit representations that can be jointly learned with the autoencoder pipeline.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[2] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9902–9912, 2022.

[3] 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.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[5] 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.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] Ye Chen, Jinxian Liu, Bingbing Ni, Hang Wang, Jiancheng Yang, Ning Liu, Teng Li, and Qi Tian. Shape self-correction for unsupervised point cloud understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8382–8391, 2021.

[8] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[9] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8963–8972, 2021.

[10] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020.

[11] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[13] Benjamin Eckart, Wentao Yuan, Chao Liu, and Jan Kautz. Self-supervised learning on 3d point clouds by learning discrete generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8257, 2021.

[14] 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.

[15] Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208. JMLR Workshop and Conference Proceedings, 2010.

[16] 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.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[18] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020.

[19] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7154–7164, 2019.

[20] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8160–8171, 2019.

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[22] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117, 2020.

[23] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021.

[24] Guohao Li, Matthias Müller, Guocheng Qian, Itzel Carolina Delgadillo Perez, Abdulellah Abualshour, Ali Kassem Thabet, and Bernard Ghanem. Deepgcns: Making gcns go as deep as cnns. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[25] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018.

[26] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018.

[27] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. *arXiv preprint arXiv:1911.00767*, 2019.

[28] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2019–2028, 2020.

[29] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021.

[30] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022.

[31] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, pages 52–59. Springer, 2011.

[32] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[33] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4743–4752, 2019.

[34] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021.

[35] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5379–5389, 2019.

[36] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[37] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4531–4540, 2019.

[38] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *arXiv preprint arXiv:2203.06604*, 2022.

[39] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[40] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1060–1070, 2020.

[41] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[42] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020.

[43] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[44] Omid Poursaeed, Tianxing Jiang, Han Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028. IEEE, 2020.

[45] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019.

[46] 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.

[47] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[48] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv preprint arXiv:2206.04670*, 2022.

[49] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24:1943–1955, 2021.

[50] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1757–1767, 2021.

[51] Yongming Rao, Benlin Liu, Yi Wei, Jiwen Lu, Cho-Jui Hsieh, and Jie Zhou. Randomrooms: Unsupervised pre-training from synthetic shapes and randomized layouts for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3283–3292, 2021.

[52] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[53] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: Fully convolutional anchor-free 3d object detection. *arXiv preprint arXiv:2112.00322*, 2021.

[54] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. *arXiv preprint arXiv:1901.08396*, 2019.

[55] Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Měch. Parsenet: A parametric surface fitting network for 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 261–276. Springer, 2020.

[56] 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.

[57] Liyao Tang, Yibing Zhan, Zhe Chen, Baosheng Yu, and Dacheng Tao. Contrastive boundary learning for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8489–8499, 2022.

[58] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.

[59] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.

[60] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *International Conference on Computer Vision (ICCV)*, 2019.

[61] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019.

[62] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[63] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[64] Haiyang Wang, Lihe Ding, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, and Liwei Wang. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. *arXiv preprint arXiv:2210.04264*, 2022.

[65] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *Proceedings of the IEEE/CVF*

[66] Peng-Shuai Wang, Yu-Qi Yang, Qian-Fang Zou, Zhirong Wu, Yang Liu, and Xin Tong. Unsupervised 3d learning for shape analysis via multiresolution instance discrimination. *ACM Trans. Graphic*, 2020.

[67] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[68] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[69] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[70] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5499–5509, 2021.

[71] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10447–10456, 2020.

[72] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[73] Siming Yan, Chen Song, Youkang Kong, and Qixing Huang. Learning from multi-view representation for point-cloud pre-training. *arXiv preprint arXiv:2306.02558*, 2023.

[74] Siming Yan, Yuqi Yang, Yuxiao Guo, Hao Pan, Peng-shuai Wang, Xin Tong, Yang Liu, and Qixing Huang. 3d feature prediction for masked-autoencoder-based point cloud pretraining. *arXiv preprint arXiv:2304.06911*, 2023.

[75] Siming Yan, Zhenpei Yang, Chongyang Ma, Haibin Huang, Etienne Vouga, and Qixing Huang. Hpnet: Deep primitive segmentation using hybrid representations. *arXiv preprint arXiv:2105.10620*, 2021.

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

[77] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

[78] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud

transformers with masked point modeling. *arXiv preprint arXiv:2111.14819*, 2021.

[79] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *arXiv preprint arXiv:2205.14401*, 2022.

[80] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. *arXiv preprint arXiv:2101.02691*, 2021.

[81] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021.

[82] Chengxu Zhuang, Siming Yan, Aran Nayebi, Martin Schrimpf, Michael C Frank, James J DiCarlo, and Daniel LK Yamins. Unsupervised neural network models of the ventral visual stream. *Proceedings of the National Academy of Sciences*, 118(3), 2021.

[83] Chengxu Zhuang, Siming Yan, Aran Nayebi, and Daniel Yamins. Self-supervised neural network models of higher visual cortex development. In *2019 Conference on Cognitive Computational Neuroscience*, pages 566–569. CCN, 2019.