Spatio-temporal Self-Supervised Representation Learning for 3D Point Clouds

Siyuan Huang[1,*], Yichen Xie[2,*], Song-Chun Zhu[3,4,5], Yixin Zhu[3,4]

[1] University of California, Los Angeles [2] Shanghai Jiao Tong University
[3] Beijing Institute for General Artificial Intelligence [4] Peking University [5] Tsinghua University

https://siyuanhuang.com/STRL

Figure 1: **Overview of our method.** By learning the spatio-temporal data invariance from a point cloud sequence, our method self-supevisedly learns an effective representation.

Abstract

To date, various 3D scene understanding tasks still lack practical and generalizable pre-trained models, primarily due to the intricate nature of 3D scene understanding tasks and their immense variations introduced by camera views, lighting, occlusions, etc. *In this paper, we tackle this challenge by introducing a spatio-temporal representation learning (STRL) framework, capable of learning from unlabeled 3D point clouds in a self-supervised fashion. Inspired by how infants learn from visual data in the wild, we explore the rich spatio-temporal cues derived from the 3D data. Specifically, STRL takes two temporally-correlated frames from a 3D point cloud sequence as the input, transforms it with the spatial data augmentation, and learns the invariant representation self-supervisedly. To corroborate the efficacy of STRL, we conduct extensive experiments on three types (synthetic, indoor, and outdoor) of datasets. Experimental results demonstrate that, compared with supervised learning methods, the learned self-supervised representation facilitates various models to attain comparable or even better performances while capable of generalizing pre-trained models to downstream tasks, including 3D shape classification, 3D object detection, and 3D semantic segmentation. Moreover, the spatio-temporal contextual cues embedded in 3D point clouds significantly improve the learned representations.*

1. Introduction

Point cloud is a quintessential 3D representation for visual analysis and scene understanding. It differs from alternative 3D representations (*e.g.*, voxel, mesh) as it is ubiquitous: Entry-level depth sensors (even on cellphones) directly produce point clouds before triangulating into meshes or converting to voxels, making it mostly applicable to 3D scene understanding tasks such as 3D shape analysis [5], 3D object detection and segmentation [58, 10]. Despite its omnipresence in 3D representation, however, annotating 3D point cloud data is proven to be much more difficult com-
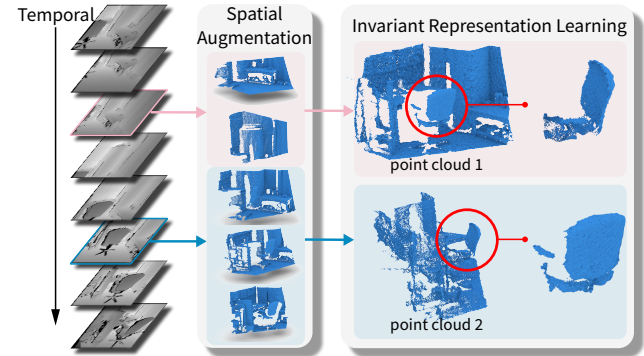
* indicates equal contribution.

pared with labeling conventional 2D image data; this obstacle precludes its potentials in 3D visual tasks. As such, properly leveraging the colossal amount of *unlabeled* 3D point cloud data is a *sine qua non* for the success of *large-scale* 3D visual analysis and scene understanding.

Meanwhile, self-supervised learning from unlabeled images [11, 45, 24, 22, 6, 19, 7] and videos [54, 80, 34, 51] becomes a nascent direction in representation learning with great potential in downstream tasks.

In this paper, we fill in the absence by exploiting self-supervised representation learning for 3D point clouds to address a long-standing problem in our community—the supervised training struggles at producing practical and generalizable pre-trained models due to the supervision-starved nature of the 3D data. Specifically, we consider the following three principles in model design and learning:

Simplicity Although self-supervised learning approaches for 3D point clouds exist, they rely exclusively on spatial analysis by reconstructing the 3D point clouds [1, 75, 53, 20]. This static perspective of self-supervised learning is designed explicitly with complex operations, architectures, or losses, making it difficult to train and generalize to diversified downstream tasks. We believe such intricate designs are artificially introduced and unnecessary, and could be diminished or eliminated by complementing the missing *temporal* contextual cues, akin to how infants may understand this world [18, 57].

Invariance   Learning data invariance via data augmentation and contrasting has shown promising results on images and videos [22, 6, 19]. A natural question arises: How could we introduce and leverage the invariance in 3D point clouds for self-supervised learning?

Generalizability   Prior literature [1, 75, 53, 20] has only verified the self-supervisedly learned representations in shape classification on synthetic datasets [5], which possesses dramatically different characteristics compared with the 3D data of natural indoor [58, 10] or outdoor [16] environments, thus failed to demonstrate sufficient generalizability to higher-level tasks (*e.g.*, 3D object detection).

To adhere to the above principles and tackle the challenges introduced thereby, we devise a spatio-temporal representation learning (STRL) framework to learn from *unlabeled* 3D point clouds. Of note, STRL is remarkably **simple** by learning only from the positive pairs, inspired by the BYOL [19]. Specifically, STRL uses two neural networks, referred to as online and target networks, that interact and learn from each other. By augmenting one input, we train the online network to predict the target network representation of another *temporally* correlated input, obtained by a separate augmentation process.

To learn the **invariant** representation [12, 68], we explore the inextricably spatio-temporal contextual cues embedded in 3D point clouds. In our approach, the online network's and target network's inputs are *temporally* correlated, sampled from a point cloud sequence. Specifically, for natural images/videos, we sample two frames with a natural viewpoint change in depth sequences as the input pair. For synthetic data like 3D shape, we augment the original input by rotation, translation, and scaling to emulate the viewpoint change. The temporal difference between the inputs avails models of capturing the randomness and invariance across different viewpoints. Additional *spatial* augmentations further facilitate the model to learn 3D spatial structures of point clouds; see examples in Fig. 1 and Sect. 3.

To **generalize** the learned representation, we adopt several practical networks as backbone models. By pre-training on large datasets, we verify that the learned representations can be readily adapted to downstream tasks directly or with additional feature fine-tuning. We also demonstrate that the learned representation can be generalized to distant domains, different from the pre-trained domains; *e.g.*, the representation learned from ScanNet [10] can be generalized to shape classification tasks on ShapeNet [5] and 3D object detection task on SUN RGB-D [58].

We conduct *extensive* experiments on various domains and test the performance by applying the pre-trained representation to downstream tasks, including 3D shape classification, 3D object detection, and 3D semantic segmentation. Next, we summarize our main findings.

Our method outperforms prior arts.   By pre-training with STRL and applying the learned models to downstream tasks, it (i) outperforms the state-of-the-art unsupervised methods on ModelNet40 [71] and reaches 90.9% 3D shape classification accuracy with linear evaluation, (ii) shows significant improvements in semi-supervised learning with limited data, and (iii) boosts the downstream tasks by transferring the pre-trained models, *e.g.*, it improves 3D object detection on SUN RGB-D [58] and KITTI dataset [16], and 3D semantic segmentation on S3DIS [2] via fine-tuning.

Simple learning strategy leads to the satisfying performance of learned 3D representation.   Through the ablative study in Tables 7 and 8, we observe that STRL can learn the self-supervised representations with simple augmentations; it robustly achieves a satisfying accuracy (about 85%) on ModelNet40 linear classification, which echoes recent findings [46] that simply predicting the 3D orientation helps learn good representation for 3D point clouds.

The spatio-temporal cues boost the performance of learned representation.   Relying on spatial or temporal augmentation alone only yield relatively low performance as shown in Tables 7 and 8. In contrast, we achieve an improvement of 3% accuracy by learning the invariant representations combining both spatial and temporal cues.

Pre-training on synthetic 3D shapes is indeed helpful for real-world applications.   Recent study [73] shows the representation learned from ShapeNet is not well-generalized to the downstream tasks. Instead, we report an opposite observation in Table 6, showing the representation pre-trained on ShapeNet can achieve comparable and even better performance while applying to downstream tasks that tackle complex data obtained in the physical world.

2. Related Work

Representation Learning on Point Clouds   Unlike conventional representations of structured data (*e.g.*, images), point clouds are unordered sets of vectors. This unique nature poses extra challenges to the learning of representations. Although deep learning methods on unordered sets [66, 77, 42] could be applied to point clouds [52, 77], these approaches do not leverage spatial structures.

Taking spatial structures into consideration, modern approaches like PointNet [48] directly feed raw point clouds into neural networks; these networks ought to be permutation invariant as point clouds are unordered sets. PointNet achieves this goal by using the max-pooling operation to form a single feature vector representing the global context from a set of points. Since then, researches have proposed alternative representation learning methods with hierarchy [49, 33, 13], convolution-based structure [25, 74, 39, 59, 78, 70, 61], or graph-based information aggregation [13, 64, 55, 67]. Operating directly on raw point clouds, these neural networks naturally provide per-point embedding, particularly effective for point-based tasks. Since the proposed STRL is flexible and compatible with various neural models serving as the backbone, our design of STRL leverages the efficacy introduced by per-point embedding.
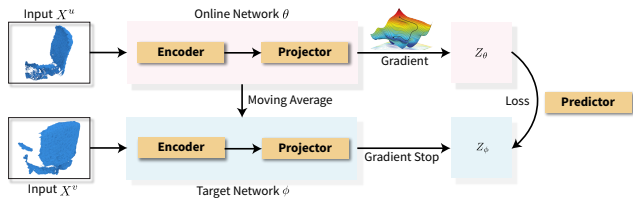
Figure 2: **Illustration of our self-supervised learning framework.** Given two spatio-temporal correlated 3D point clouds, the online network predicts the target network's representation via a predictor. Parameters of the target network are updated by the online network's moving average.

Unsupervised Representation Learning Unsupervised representation learning could be roughly categorized as either generative or discriminative approaches. Generative approaches typically attempt to reconstruct the input data in terms of pixel or point by modeling the distributions of data or the latent embedding. This process could be realized by energy-based modeling [36, 44, 72, 14, 35], auto-encoding [65, 32, 4], or adversarial learning [17]. However, this unsupervised mechanism is computationally expensive, and the learning of generalizable representation unnecessarily relies on recovering such high-level details.

Discriminative approaches, including self-supervised learning, unsupervisedly generate discriminative labels to facilitate representation learning, recently achieved by various contrastive mechanisms [22, 45, 24, 23, 3, 62, 63]. Different from generative approaches that maximize the data likelihood, recent contrastive approaches maximally preserve the mutual information between the input data and its encoded representation. Following BYOL [19], we exclude negative pairs in contrastive learning and devise STRL to construct a stable and invariant representation through a moving average target network.

Self-supervised Learning of Point Clouds Although various approaches [69, 1, 13, 75, 37, 79, 60, 46] have been proposed for unsupervised learning and generation of point clouds, these approaches have merely demonstrated efficacy in shape classification tasks on synthetic datasets while ignoring higher-level tasks of pre-trained models on natural 3D scenes. More recent work starts to demonstrate the potentials for high-level tasks such as 3D object detection and 3D semantic segmentation. For instance, Sauder *et al.* [53] train the neural network to reconstruct point clouds with self-supervised labels generated by randomly arranging object parts, and Xie *et al.* [73] learn from dense correspondences between different views with a contrastive loss. In comparison, the proposed STRL is much simpler without computing the dense correspondences or reconstruction loss; it relies solely on spatio-temporal contexts and structures of point clouds, yielding more robust and improved performances on various high-level downstream tasks.

3. Spatio-temporal Representation Learning

We devise the proposed spatio-temporal representation learning (STRL) based on BYOL [19] and extend its simplicity to the learning of 3D point cloud representation. Fig. 2 illustrates the proposed method.

3.1. Building Temporal Sequence of Point Clouds

To learn a simple, invariant, and generalizable representation for 3D point clouds, we formulate the representation learning as training with sequences of potentially partial and cluttered 3D point clouds of objects or scenes. Given a sequence of potentially non-uniformly sampled time steps, we denote the corresponding point cloud sequence as $\mathcal{P} = \{p_t\}_{t=1}^T$. We devise two approaches to generating the training point cloud sequences to handle various data sources.

Natural Sequence Natural sequences refer to the data sequences captured by RGB-D sensors, wherein each depth image I_t is a projected view of the scene. Given the camera pose (extrinsic parameters) at each time step c_t^{ex}, we back-project depth images with intrinsic parameters c^{in} and obtain a sequence of point clouds $\{p_t\}$ in world coordinate:

$$p_t = Backproj(I_t, c_t^{ex}, c^{in}), \quad t = 1, \cdots, T. \quad (1)$$

Synthetic Sequence Static point clouds are intrinsically spatial, missing the crucial temporal dimension compared to natural sequences. Given a point cloud p_0, we solve this problem by generating a synthetic sequence. Specifically, we consecutively rotate, translate, and scale the original point cloud to construct a sequence of point clouds $\{p_t\}$:

$$p_t = R_t(p_{t-1}), \quad t = 1, \cdots, T, \quad (2)$$

where t is the index of transformations, and R_t the sampled transformation, emulating temporal view changes.

3.2. Representation Learning

We design STRL to unsupervisedly learn the representations through the interactions of two networks: the **online** network and **target** network. Here, the essence of self-supervised learning is to train the online network to accurately predict the target network's representation.

Specifically, the online network parameterized by θ consists of two components: a backbone encoder e_θ and a feature projector f_θ. Similarly, the target network parameterized by ϕ has a backbone encoder e_ϕ and feature projector f_ϕ. In addition, a predictor r with parameters regresses the target presentation: The target network serves as regression targets to train the online network, and its parameters ϕ are an exponential moving average of the online parameters θ,

$$\phi \leftarrow \tau\phi + (1 - \tau)\theta, \quad (3)$$

where $\tau \in [0, 1]$ is the decay rate of the moving average.

Given a sequence of point clouds $\mathcal{P}$, we sample two frames of point clouds $p^u, p^v \in \mathcal{P}$ by a temporal sampler $\mathcal{T}$. With a set of spatial augmentations $\mathcal{A}$ (see details in Sect. 4), STRL generates two inputs $x^u = a^u(p^u)$ and $x^v = a^v(p^v)$, where $a^u, a^v \in \mathcal{A}$. For each input, the online network and target network generate $z_\theta = f_\theta(e_\theta(x^u))$, $z_\phi = f_\phi(e_\phi(x^v))$, respectively. With the additional predictor r, the goal of STRL is to minimize the mean squared error between the normalized predictions and target projections:

$$\mathcal{L}_{u \to v} = \left\| \frac{r(z_\theta)}{\|r(z_\theta)\|_2} - \frac{z_\phi}{\|z_\phi\|_2} \right\|_2^2 = 2 - 2 \cdot \frac{\langle r(z_\theta), z_\phi \rangle}{\|r(z_\theta)\|_2 \cdot \|z_\phi\|_2} \quad (4)$$

Finally, we symmetrize the loss in Eq. (4) to compute $\mathcal{L}_{v \to u}$ by separately feeding x^v to the online network and x^u to the target network. The total loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{u \to v} + \mathcal{L}_{v \to u}. \quad (5)$$

Within each training step, only the parameters of the online network and predictor are updated. The target network's parameters are updated after each training step by Eq. (3). Similar to [22, 19], we only keep the backbone encoder of the online network e_θ at the end of the training as the learned model. Algorithm 1 details the proposed STRL.

4. Implementation Details

Synthetic Sequence Generation We sample the combination of following transformations to construct the function $R()$ in Eq. (2); see an illustration in Fig. 3b:

- Random rotation. For each axis, we draw random angles within $15°$ and rotate around it.
- Random translation. We translate the point cloud globally within 10% of the point cloud dimension.
- Random scaling. We scale the point cloud with a factor $s \in [0.8, 1.25]$.

To further increase the randomness, each transformation is sampled and applied with a probability of 0.5.

Spatial Augmentation The spatial augmentation transforms the input by changing the point cloud's local geometry, which helps STRL to learn a better spatial structure representation of point clouds. Specifically, we apply the following transformations, similar to the image data augmentation; see an illustration in Fig. 3a.

- Random cropping. A random 3D cuboid patch is cropped with a volume uniformly sampled between 60% and 100% of the original point cloud. The aspect ratio is controlled within $[0.75, 1.33]$.
- Random cutout. A random 3D cuboid is cut out. Each dimension of the 3D cuboid is within $[0.1, 0.4]$ of the original dimension.
- Random jittering. Each point's 3D locations are shifted by a uniformly random offset within $[0, 0.05]$.
- Random drop-out. We randomly drop out 3D points by a drop-out ratio within $[0, 0.7]$.

Algorithm 1: STRL of 3D point clouds

Input:
1 $\{\mathcal{P}\}$: a set of 3D point cloud sequences;
2 $\mathcal{T}, \mathcal{A}$: temporal sampler and spatial augmentations;
3 e_θ, f_θ: online encoder and projector with parameter θ;
4 e_ϕ, f_ϕ: target encoder and projector with parameter ϕ;
5 r: predictor;
6 K: number of optimization steps;
7 N: batch size.
Output: online encoder e_θ.
8 **for** $k = 1$ **to** K **do**
```
      /* sample batches of temporal-correlated
         point clouds                        */
```
9    $\mathcal{B} \leftarrow \{p_i^u, p_i^v \in \mathcal{T}(\{\mathcal{P}\})\}_{i=1}^N$
10   **for** $i = 1$ **to** N **do**
```
         /* sample spatial augmentations    */
```
11       $a^u, a^v \in \mathcal{A}$
```
         /* generate inputs                 */
```
12       $x^u = a^u(p^u), x^t = a^v(p^v)$
```
         /* project                         */
```
13       $z_\theta = f_\theta(g_\theta(x^u)), z_\phi = f_\phi(g_\phi(x^v))$
```
         /* compute loss                    */
```
14       $\mathcal{L}_{u \to v} = -2 \cdot \frac{\langle r(z_\theta), z_\phi \rangle}{\|r(z_\theta)\|_2 \cdot \|z_\phi\|_2}$
```
         /* compute total & symmetric loss  */
```
15       $\mathcal{L}_{total} = \mathcal{L}_{u \to v} + \mathcal{L}_{v \to u}$
16   **end**
```
      /* update online network & predictor  */
```
17    $\theta, r = optimize(\theta, r, \mathcal{L}_{total})$
```
      /* update target network              */
```
18    $\phi \leftarrow \tau\phi + (1 - \tau)\theta$
19 **end**

- Down-sampling. We down-sample point clouds based on the encoder's input dimension by randomly picking the necessary amount of 3D points.
- Normalization. We normalize the point cloud to fit a unit sphere while training on synthetic data [5].

Among these augmentations, cropping and cutout introduce more evident changes to the point clouds' spatial structures. As such, we apply them with a probability of 0.5.

Training We use the LARS optimizer [76] with a cosine decay learning rate schedule [40], with a warm-up period of 10 epochs but without restarts. For the target network, the exponential moving average parameter starts with $\tau_{start} = 0.996$ and is gradually increased to 1 during the training. Specifically, we set $\tau = 1 - (1 - \tau_{start}) \cdot (cos(\pi k / K) + 1)/2$ with k being the current training step and K the maximum number of training steps.

STRL is favorable and generalizable to different backbone encoders; see details about the encoder structure for each specific experiment in Sect. 5. The projector and predictor are implemented as multi-layer perceptions (MLPs) with activation [43] and batch normalization [29]; see the *supplementary materials* for detailed network structures. We use a batch size ranging from 64 to 256 split over 8 TITAN RTX GPUs for most of the pre-trained models.
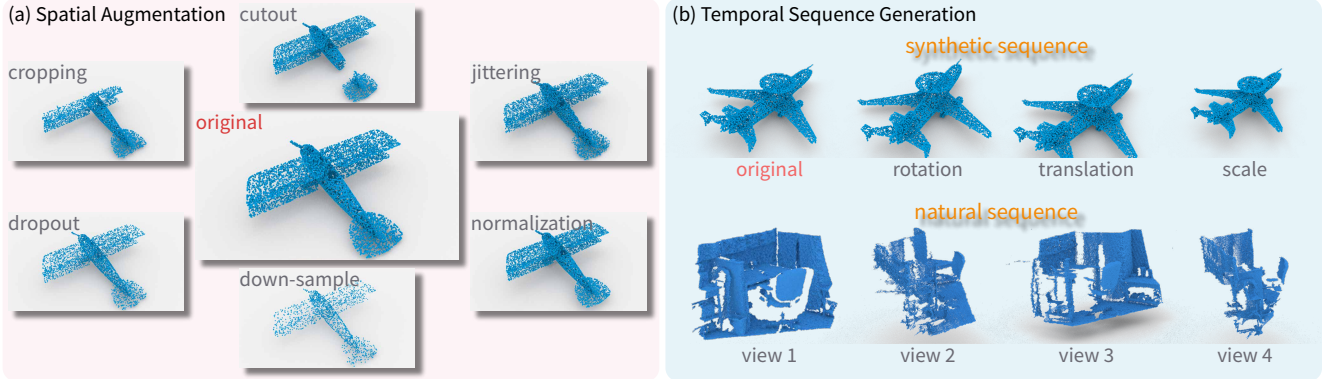
Figure 3: **Spatial data augmentation and temporal sequence generation.** Except for the natural sequence generation, each type of augmentation transforms the input point cloud data stochastically with certain internal parameters.

5. Experiment

We start by introducing how to pre-train STRL on various data sources in Sect. 5.1. Next, we evaluate these pre-trained models on various downstream tasks in Sect. 5.2. At length, in Sect. 5.3, we analyze the effects of different modules and parameters in our model, with additional analytic experiments and discussions of open problems.

5.1. Pre-training

To recap, as detailed in Sect. 3.1, we build the sequences of point clouds and perform the pre-training of STRL to learn the spatio-temporal invariance of point cloud data. For synthetic shapes and natural indoor/outdoor scenes, we generate temporal sequences of point clouds and sample input pairs using different strategies detailed below.

5.1.1 Synthetic Shapes

ShapeNet We learn the self-supervised representation model from the ShapeNet [5] dataset. It consists of 57,448 synthetic objects from 55 categories. We pre-process the point clouds following Yang *et al*. [75]. By augmenting each point cloud into two different views with temporal transformations defined in Eq. (2), we generate two temporal-corrected point clouds. The spatial augmentations are further applied to produce the pair of point clouds as the input.

5.1.2 Natural Indoor and Outdoor Scenes

We also learn the self-supervised representation models from natural indoor and outdoor scenes, in which sequences of point clouds are readily available. Using RGB-D sensors, sequences of depth images can be captured by scanning over different camera poses. Since most scenes are captured smoothly, we learn the temporal invariance from the temporal correlations between the adjacent frames.

ScanNet For indoor scenes, we pre-train on the ScanNet dataset [10]. It consists of 1,513 reconstructed meshes for 707 unique scenes. In experiments, we find that increasing the frame-sampling frequency only makes a limited con-

tribution to the performance. Hence, we sub-sample the raw depth sequences every 100 frames as the keyframes for each scene, resulting in 1,513 sequences and roughly 25 thousand frames in total. During pre-training, we generate fixed-length sliding windows based on the keyframes of each sequence and sample two random frames within each window. By back projecting the two frames with Eq. (1), we generate point clouds in the world coordinate. We use the camera position to translate the two point clouds into the same world coordinate; the camera center of the first frame is the origin.

KITTI For outdoor scenes, we pre-train on the KITTI dataset [15]. It includes 100+ sequences divided into 6 categories. For each scene, images and point clouds are recorded at roughly 10 FPS. We only use the sequences of point clouds captured by the Velodyne lidar sensor. On average, each frame has about 120,000 points. Similar to ScanNet, we sub-sample the keyframes and sample frame pairs within sliding windows as training pairs.

For pre-training on natural scenes, we further enhance the data diversity by applying synthetic temporal transformations in Eq. (2) to the two point clouds. At length, the spatial data augmentation is applied to both point clouds.

5.2. Downstream Tasks

For each downstream task below, we present the model structures, experimental settings, and results. Please refer to the *supplementary materials* for training details.

5.2.1 Shape Understanding

We adopt the protocols presented in prior work [1, 53, 69, 75] to evaluate the shape understanding capability of our pre-trained model using the ModelNet40 [71] benchmark. It contains 12,331 objects (9,843 for training and 2,468 for testing) from 40 categories. We pre-process the data following Qi *et al*. [48], such that each shape is sampled to 10,000 points in unit space.

As detailed in Sect. 5.1, we pre-train the backbone models on ShapeNet dataset. We measure the learned representations using the following evaluation metrics.

Linear Evaluation for Shape Classification   To classify 3D shapes, We append a linear Support Vector Machine (SVM) on top of the encoded global feature vectors. Following Sauder *et al*. [53], these global features are constructed by extracting the activation after the last pooling layer. Our STRL is flexible to work with various backbones; we select two practical ones—PointNet [48] and DGCNN [67]. The SVM is trained with the extracted global features from the training sets of ModelNet40 datasets. We randomly sample 2,048 points from each shape during both pre-training and SVM training.

Table 1 tabulates the classification results on test sets. The proposed STRL outperforms all the state-of-the-art unsupervised and self-supervised methods on ModelNet40.

Table 1: **Comparisons of the linear evaluation for shape classification on ModelNet40**. A linear classifier is trained on the representations learned by different self-supervised approaches on the ShapeNet dataset.

Method	ModelNet40
3D-GAN [69]	83.3%
Latent-GAN [1]	85.7%
SO-Net [38]	87.3%
FoldingNet [75]	88.4%
MRTNet [21]	86.4%
3D-PointCapsNet [75]	88.9%
MAP-VAE [75]	88.4%
Sauder *et al*. + PointNet [53]	87.3%
Sauder *et al*. + DGCNN [53]	90.6%
Poursaeed *et al*. + PointNet [46]	88.6%
Poursaeed *et al*. + DGCNN [46]	90.7%
STRL + PointNet (ours)	88.3%
STRL + DGCNN (ours)	**90.9%**

Supervised Fine-tuning for Shape Classification   We also evaluate the learned representation models by supervised fine-tuning. The pre-trained model serves as the point cloud encoder's initial weight, and we fine-tune the DGCNN network given the labels on the ModelNet40 dataset. Our STRL leads to a marked performance improvement of up to 0.9% on the final classification accuracy; see Table 2a. This improvement is more significant than previous methods; it even attains a comparable performance set by the state-of-the-art supervised learning method [78].

Furthermore, we show that our pre-trained models can significantly boost the classification performance in semisupervised learning where limited labeled training data is provided. Specifically, we randomly sample the training data with different proportions and ensure at least one sample for each category is selected. Next, we fine-tune the pre-trained model on these limited samples with supervision and evaluate its performance on full test sets. Table 2b summarizes the results measured by accuracy. It shows that the proposed model obtains 2.1% and 1.6% performance gain when 1% and 20% of the training samples are available; our self-supervised models would better facilitate downstream tasks when fewer training samples are available.

Table 2: **Shape classification fine-tuned on ModelNet40**. The self-supervised pre-trained model serves as the initial weight for supervised learning methods.

(a) Fine-tuned on Full Training Set

Category	Method	Accuracy
Supervised	PointNet [48]	89.2%
	PointNet++ [49]	90.7%
	PointCNN [39]	92.2%
	DGCNN [67]	92.2%
	ShellNet [78]	93.1%
Self-supervised	Sauder *et al*. + DGCNN [53]	92.4%
	STRL + DGCNN (ours)	**93.1%**

(b) Fine-tuned on Few Training Samples

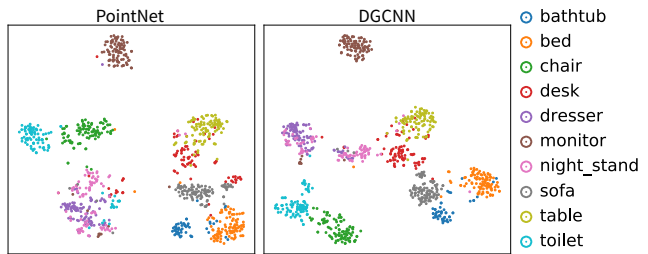| Method	1%	5%	10%	20%
DGCNN	58.4%	80.7%	85.2%	88.1%
STRL + DGCNN	**60.5%**	**82.7%**	**86.5%**	**89.7%**

Figure 4: **Visualization of learned features**. We visualize the extracted features for each sample in ModelNet10 test set using t-SNE. Both models are pre-trained on ShapeNet.

Embedding Visualization   We visualize the learned features of PointNet and DGCNN model with our self-supervised method in Fig. 4; it displays the embedding for samples of different categories in the ModelNet10 test set. t-SNE [41] is adopted for dimension reduction. We observe that both pre-trained models well separate most samples based on categories, except *dressers* and *night stands*; they usually look similar and are difficult to distinguish.

5.2.2 Indoor Scene Understanding

Our proposed STRL learns representations based on view transformation, suitable for both synthetic shapes and natural scenes. Consequently, unlike prior work that primarily performs transfer learning to shape understanding, our method can also boost the indoor/outdoor scene understanding tasks. We start with the indoor scene understanding in this section. We first pre-train our STRL self-supervisedly on the ScanNet dataset as described in Sect. 5.1. Next, we evaluate the performance of 3D object detection and semantic segmentation through fine-tuning with labels.

3D Object Detection   3D object detection requires the model to predict the 3D bounding boxes with their object categories based on the input 3D point cloud. After pretraining, we fine-tune and evaluate the model on the SUN

RGB-D [58] dataset. It contains 10,335 single-view RGB-D images, split into 5,285 training samples and 5,050 validation samples. Objects are annotated with 3D bounding boxes and category labels. We conduct this experiment with VoteNet [47], which is a widely-used model with 3D point clouds as input. During pre-training, we slightly modify its PointNet++ [49] backbone by adding a max-pooling layer at the end to obtain the global features. Table 3 summarizes the results. The pre-training improves the detection performance by 1.2 mAP compared against training VoteNet from scratch, demonstrating that the representation learned from a large dataset, *i.e.*, ScanNet, can be successfully transferred to a different dataset and improve the performances of high-level tasks via fine-tuning. It also outperforms the state-of-the-art self-supervised learning method [73] by 0.7 mAP.[1]

Table 3: **3D object detection fine-tuned on SUN RGB-D**

Model	Method	Input	mAP@0.25 IoU
VoteNet	*from scratch*	Geo+Height	57.7
		Geo	57.0
SR-UNet [9]	PointContrast [73]	Geo	57.5
VoteNet	STRL (ours)	Geo	**58.2**

3D Semantic Segmentation We transfer the pre-trained model to the 3D semantic segmentation task on the Stanford Large-Scale 3D Indoor Spaces (S3DIS) [2] dataset. This dataset contains 3D point clouds scanned from 272 rooms in 6 indoor areas, with each point annotated into 13 categories. We follow the setting in Qi *et al*. [48] and Wang *et al*. [67] and split each room into $1m \times 1m$ blocks. Different from them, we use 4,096 points with only geometric features (XYZ coordinates) as the model input. In this experiment, the DGCNN network is firstly pre-trained on ScanNet with STRL. Here, we focus on semi-supervised learning with only limited labeled data. As such, we fine-tune the pre-trained model on one area in Area 1-5 each time and test the model on Area 6. As shown in Table 4, the pre-trained models consistently outperform the models trained from scratch, especially with a small training set.

5.2.3 Outdoor Scene Understanding

Compared with indoor scenes, point clouds captured in outdoor environments are much sparser due to the long-range nature of Lidar sensors, posing additional challenges. In this section, we evaluate the performance of the proposed STRL by transferring the learned visual representations to the 3D object detection task for outdoor scenes.

As described in Sect. 5.1, we pre-train the model on the KITTI dataset with PV-RCNN [56]—the state-of-the-art model for 3D object detection. Similar to VoteNet, we modify the backbone network of PV-RCNN for pre-training by adding a max-pooling layer to obtain the global features.

[1] The model pre-trained on ShapeNet achieves better results as 59.2 mAP, which is analyzed and explained in Sect. 5.3

Table 4: **3D semantic segmentation fine-tuned on S3DIS**. We train the pre-trained or initialized models in a semi-supervised manner on one of the Areas 1-5. Performances below are evaluated on Area 6 of the S3DIS dataset.

Fine-tuning Area	Method	Acc.	mIoU
Area 1 (3687 *samples*)	*from scratch*	84.57%	57.85
	STRL	**85.28%**	**59.15**
Area 2 (4440 *samples*)	*from scratch*	70.56%	38.86
	STRL	**72.37%**	**39.21**
Area 3 (1650 *samples*)	*from scratch*	77.68%	49.49
	STRL	**79.12%**	**51.88**
Area 4 (3662 *samples*)	*from scratch*	73.55%	38.50
	STRL	**73.81%**	**39.28**
Area 5 (6852 *samples*)	*from scratch*	76.85%	48.63
	STRL	**77.28%**	**49.53**

We fine-tune the pre-trained model on KITTI 3D object detection benchmark [16], a subset of the KITTI raw data. In this benchmark, each point cloud is annotated with 3D object bounding boxes. The subset includes 3,712 training samples, 3,769 validation samples, and 7,518 test samples. Table 5 tabulates results. On all three categories, models pre-trained with STRL outperform the model trained from scratch. In particular, for the cyclist category where the least training samples are available, the proposed STRL generates a marked performance elevation. We further freeze the backbone model while fine-tuning; the results reveal that models with the pre-trained backbone reach a comparable performance compared with training models from scratch.

Table 5: **3D object detection fine-tuned on KITTI**. We report 3D detection performance with moderate difficulty on the val set of KITTI dataset. Performances below are evaluated by mAP with 40 recall positions.

Method	Car (IoU=0.7)		Pedestrian		Cyclist	
	3D	BEV	3D	BEV	3D	BEV
PV-RCNN (*from scratch*)	84.50	90.53	57.06	59.84	70.14	75.04
STRL + PV-RCNN (*frozen backbone*)	81.63	87.84	39.62	42.41	69.65	74.20
STRL + PV-RCNN	**84.70**	**90.75**	**57.80**	**60.83**	**71.88**	**76.65**

5.3. Analytic Experiments and Discussions

Generalizability: ScanNet vs ShapeNet Pre-training What kind of data would endow the learned model with better generalizability to other data domains remains an open problem in 3D computer vision. To elucidate this problem, we pre-train the model on the existing largest natural dataset ScanNet and synthetic data ShapeNet, and test their generalizability to different domains. Table 6 tabulates our cross-domain experimental settings and results, demonstrating the successful transfer from models pre-trained on natural scenes to synthetic shape domain, achieving comparable shape classification performance under linear evaluation.

Table 6: **Ablation study: cross-domain generalizability**

(a) Linear evaluation for shape classification on ModelNet40.

Method	Pre-train Dataset	Accuracy
STRL + DGCNN (linear)	ScanNet	90.4%
	ShapeNet	**90.9%**
STRL + DGCNN (fine-tune)	ScanNet	92.9%
	ShapeNet	**93.1%**

(b) Fine-tuned 3D object detection on SUN RGB-D.

Method	Pre-train Dataset	mAP@0.25 IoU
STRL + VoteNet	ScanNet	58.2
	ShapeNet	**59.2**

Additionally, we report an opposite observation in contrast to a recent study [73]. Specifically, the VoteNet model pre-trained on the ShapeNet dataset achieves better performance than ScanNet pre-training in SUN RGB-D object detection, demonstrating better generalizability of ShapeNet data. We believe three potential reasons lead to such conflicting results: (i) The encoder adapted to learn the point cloud features in Xie *et al.* [73] is too simple such that it fails to capture sufficient information from the pre-trained ShapeNet dataset. (ii) The ShapeNet dataset provides point clouds with clean spatial structures and fewer noises, which benefits the pre-trained model to learn effective representations. (iii) Although the amount of sequence data in ScanNet is large, the modality might still be limited as it only has 707 scenes. This last hypothesis is further backed by our experiments in *Data Efficiency* below.

Temporal Transformation As described in Sect. 3.1 and 5.1, we learn from synthetic view transformation on object shapes and natural view transformation on physical scenes. To study their effects, we disentangle the combinations by removing certain transformations to generate training data of synthetic shapes when pre-training on the ShapeNet dataset; Table 7a summarizes results. For physical scenes, we pre-train PV-RCNN on the KITTI dataset and compare the models trained with and without sampling input data from natural sequences; Table 7b summarizes the results. Temporal transformation introduces substantial performance gains in both cases.

Spatial Data Augmentation We investigate the effects of spatial data augmentations by turning off certain types of augmentation; see Table 8. By augmenting the point clouds into different shapes and dimensions, random crop boosts the performance, whereas random cutout hurts the performance as it breaks the point cloud's structural continuity, crucial for point-wise feature aggregation from neighbors.

Data Efficiency To further analyze how the size of training data affects our model, we pre-train the DGCNN model with a subset of ScanNet dataset by sampling 25,000 frames depth images from the entire 1,513 sequences. Evaluated on ModelNet40, the model's performance only drops about 0.5% for both linear evaluation and fine-tuning com-

Table 7: **Ablation study: temporal transformation**

(a) **Synthetic Shapes.** We evaluate the pre-trained PointNet model on ModelNet40 by linear evaluation under different temporal transformations.

Synthetic View Transformations	Accuracy
Full	**88.3%**
Remove rotation	87.8%
Remove scaling	87.9%
Remove translation	87.2%
Remove rot. + sca. + trans.	85.5%

(b) **Physical Scenes.** We freeze the PV-RCNN backbone and fine-tune the 3D object detector on KITTI. It shows the mAP results (under 40 recall positions) of car detection w./w.o. sampling input data from natural sequence.

Natural Sequence	Car		
	Easy	Moderate	Hard
✓	**91.08**	**81.63**	**79.39**
✗	90.17	81.21	79.05

Table 8: **Ablation study: spatial data augmentation.** We pre-train the PointNet model on ShapeNet with different spatial transformations. Performances below reflect the linear evaluation results on ModelNet40.

Spatial Transformation	Accuracy
Full	**88.3**%
Remove Cutout	88.1%
Remove Crop	87.5%
Remove Crop and Cutout	87.4%
Down-sample only	86.1%

pared with training on the whole set with 0.4 million frames; such results are similar to 2D image pre-training [22]. We hypothesize that increasing the data diversity, but not sampling density, would improve performances in self-supervised 3D representation learning.

Robustness We observe that the proposed STRL can learn the self-supervised representations by simple augmentations; it robustly achieves a satisfying accuracy (about 85%) on ModelNet40 linear classification. Nevertheless, it differs from the results shown in 2D image pre-training [6, 19], where data augmentations affect the ImageNet linear evaluation by up to 10%. We hypothesize that this difference might be ascribed to the general down-sampling process performed on the point clouds, which introduces structural noises and helps the invariant feature learning.

6. Conclusion

In this paper, we devise a spatio-temporal self-supervised learning framework for learning 3D point cloud representations. Our method has a simple structure and demonstrates promising results on transferring the learned representations to various downstream 3D scene understanding tasks. In the future, we hope to explore how to extend current methods to holistic 3D scene understanding [28, 27, 26, 8, 30, 50, 31] and how to bridge the domain gap by joint training of unlabeled data from various domains.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Representation learning and adversarial generation of 3d point clouds. In *Proceedings of International Conference on Machine Learning (ICML)*, 2018. 1, 2, 3, 5, 6

[2] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 7

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 3

[4] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013. 3

[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. 1, 2, 4, 5

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 8

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[8] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 8

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[10] 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 (CVPR)*, 2017. 1, 2, 5

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 1

[12] Peter Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991. 2

[13] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 2, 3

[14] Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning generative convnets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2013. 5

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2, 7

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 3

[18] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: What early learning tells us about the mind*. William Morrow Paperbacks, 2000. 1

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 1, 2, 3, 4, 8

[20] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker. View inter-prediction gan: Unsupervised representation learning for 3d shapes by learning global shape memories to support local view predictions. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 1, 2

[21] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker. Multi-angle point cloud-vae: Unsupervised feature learning for 3d point clouds from multiple angles by joint self-reconstruction and half-to-half prediction. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 6

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 4, 8

[23] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 3

[24] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 1, 3

[25] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[26] Siyuan Huang, Yixin Chen, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Perspectivenet: 3d object detection from a single rgb image via perspective points. *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 8

[27] Siyuan Huang, Siyuan Qi, Yinxue Xiao, Yixin Zhu, Ying Nian Wu, and Song-Chun Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout and camera pose estimation. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 8

[28] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 8

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 4

[30] baoxiong Jia, Yixin Chen, Siyuan Huang, Yixin Zhu, and Song-Chun Zhu. Lemma: A multi-view dataset for learning multi-agent multi-task activities. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 8

[31] Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars. *International Journal of Computer Vision (IJCV)*, 126(9):920–941, 2018. 8

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[33] Roman Klokov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[34] Joshua Knights, Anthony Vanderkop, Daniel Ward, Olivia Mackenzie-Ross, and Peyman Moghadam. Temporally coherent embeddings for self-supervised video representation learning. *arXiv preprint arXiv:2004.02753*, 2020. 1

[35] Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019. 3

[36] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 3

[37] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018. 3

[38] 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 (CVPR)*, 2018. 6

[39] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 2, 6

[40] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 4

[41] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 6

[42] Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements. *arXiv preprint arXiv:2002.08599*, 2020. 2

[43] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of International Conference on Machine Learning (ICML)*, 2010. 4

[44] Jiquan Ngiam, Zhenghao Chen, Pang W Koh, and Andrew Y Ng. Learning deep energy models. In *Proceedings of International Conference on Machine Learning (ICML)*, 2011. 3

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 3

[46] Omid Poursaeed, Tianxing Jiang, Quintessa Qiao, Nayun Xu, and Vladimir G Kim. Self-supervised learning of point clouds via orientation estimation. *Proceedings of International Conference on 3D Vision (3DV)*, 2020. 2, 3, 6

[47] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 7

[48] 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 (CVPR)*, 2017. 2, 5, 6, 7

[49] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2, 6, 7

[50] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 8

[51] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020. 1

[52] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Deep learning with sets and point clouds. *arXiv preprint arXiv:1611.04500*, 2016. 2

[53] Jonathan Sauder and Bjarne Sievers. Self-supervised deep learning on point clouds by reconstructing space. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 3, 5, 6

[54] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *Proceedings of International Conference on Robotics and Automation (ICRA)*, 2018. 1

[55] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[56] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[57] Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11(1-2):13–29, 2005. 1

[58] 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 (CVPR)*, 2015. 1, 2, 7

[59] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[60] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, 2020. 3

[61] 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 International Conference on Computer Vision (ICCV)*, 2019. 2

[62] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 3

[63] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 3

[64] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Feastnet: Feature-steered graph convolutions for 3d shape analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[65] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of International Conference on Machine Learning (ICML)*, 2008. 3

[66] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*, 2015. 2

[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. 2, 6, 7

[68] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002. 2

[69] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 3, 5, 6

[70] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[71] 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 (CVPR)*, 2015. 2, 5

[72] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *Proceedings of International Conference on Machine Learning (ICML)*, 2016. 3

[73] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. *Proceedings of European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 7, 8

[74] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2018. 2

[75] 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 (CVPR)*, 2018. 1, 2, 3, 5, 6

[76] Yang You, Igor Gitman, and Boris Ginsburg. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 2017. 4

[77] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 2

[78] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2019. 2, 6

[79] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[80] Chengxu Zhuang, Tianwei She, Alex Andonian, Max Sobol Mark, and Daniel Yamins. Unsupervised learning from video with deep neural embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1