Unsupervised 3D Point Cloud Representation Learning by Triangle Constrained Contrast for Autonomous Driving

Bo Pang* Hongchi Xia* Cewu Lu†
Shanghai Jiao Tong University
{pangbo, xiahongchi, lucewu}@sjtu.edu.cn

Abstract

Due to the difficulty of annotating the 3D LiDAR data of autonomous driving, an efficient unsupervised 3D representation learning method is important. In this paper, we design the Triangle Constrained Contrast (TriCC) framework tailored for autonomous driving scenes which learns 3D unsupervised representations through both the multimodal information and dynamic of temporal sequences. We treat one camera image and two LiDAR point clouds with different timestamps as a triplet. And our key design is the consistent constraint that automatically finds matching relationships among the triplet through "self-cycle" and learns representations from it. With the matching relations across the temporal dimension and modalities, we can further conduct a triplet contrast to improve learning efficiency. To the best of our knowledge, TriCC is the first framework that unifies both the temporal and multimodal semantics, which means it utilizes almost all the information in autonomous driving scenes. And compared with previous contrastive methods, it can automatically dig out contrasting pairs with higher difficulty, instead of relying on handcrafted ones. Extensive experiments are conducted with Minkowski-UNet and VoxelNet on several semantic segmentation and 3D detection datasets. Results show that TriCC learns effective representations with much fewer training iterations and improves the SOTA results greatly on all the downstream tasks. Code and models can be found at https://bopang1996.github.io/.

1. Introduction

For the perception of autonomous driving, semantic segmentation and 3D object detection on point cloud are two fundamental tasks [18, 89]. At present, deep learning algorithms based on supervised learning have pushed their performances to the applicable level [37, 42, 83] with large scale datasets [7, 47, 66, 86]. Nevertheless, in the supervised process, obtaining annotations for these dense localization tasks is expensive and time-consuming. Thus, a mass of easy-collected perception data lacks efficient utilization [7, 86] and it needs much repetitive work to transfer supervised trained models to different scenes which makes the extension of applications difficult. Based on this, in this paper, we build an unsupervised dense 3D point cloud representation learning framework tailored for autonomous driving scenes, aiming at achieving better performance on dense tasks with less annotated data.

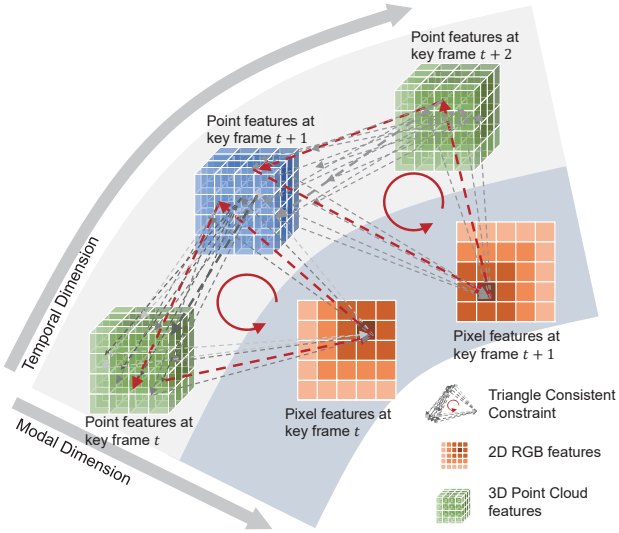

Figure 1. Brief illustration of the Triangle Constrained Contrast (TriCC). In order to learn dense 3D point cloud representations from temporal information and multimodal semantics in one unified algorithm, TriCC is designed to adopt the triangle consistent constraint to automatically find the pixel (point)-level matching relationships among each triangle (the red cycle) and learn effective representations spanning temporal and modality dimensions, instead of relying on hand-crafted pre-defined dense positive pairs that traditional contrastive learning methods adopt. Thus, it can unify all the semantics in one concise and compact algorithm.

Inspired by contrastive learning, a mainstream unsuper-

*Equal contribution. † Cewu Lu is the corresponding author, the member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China, and Shanghai Qi Zhi Institute.

vised framework, we choose to adopt the same discriminative route to build our algorithm. Compared with currently popular generative models like MAE [17,28,90], discriminative methods are more propitious to temporal and multimodal information, and have better applicability for structures of point cloud models. Some previous works try to utilize contrastive learning in point cloud tasks. PointContrast [78] and PPKT [44] adopt affine transformation and multimodal information respectively to get positive discrimination pairs. However, this kind of hand-crafted manner leads to many false negatives. STRL [34] learns semantics from video temporal frames. Though the positive pairs have more diversity, it learns global semantics which is not optimal for dense tasks. More importantly, these methods cannot learn from the complete multimodal & temporal information since they model these two sources of semantics in two different ways which cannot extend to each other. This makes them incapable of efficient unsupervised learning in autonomous driving scenarios where multimodal and temporal information is commonly available.

In this paper, we design a new unsupervised representation learning framework: Triangle Constrained Contrast (TriCC) for the autonomous driving scene. It aims at getting rid of discriminative unsupervised algorithms' demand on hand-crafted positive temporal dense pairs and replacing it with an elegant self-driven scheme so that TriCC can learn both the dense temporal semantics and multimodal semantics in one unified end-to-end model. Inspired by [74], the core idea of TriCC is to treat a camera image, a LiDAR point cloud with timestamp t_0, and a point cloud with nearby timestamp t_1 as a triple-pair. Then it learns representations and gets dense matching relationships among the triplet through the designed triangle consistent constraint which forces the pixels or points to match back with themselves after transition through the triplet-cycle as Fig. 1 shows. The implicitly learned dense relationships among the temporal dimension and multiple modalities are harder and more effective positive pairs for discrimination which are important for representation learning. Moreover, we design the cycle shortcut technique and triplet contrast to enhance the learning efficiency, allowing TriCC to learn better representations with half iterations of previous methods.

The proposed TriCC framework is effective and simple to implement. We adopt nuScenes dataset [7] to learn the point cloud representation and evaluate it on point cloud semantic segmentation and 3D object detection downstream tasks with several datasets and backbones. TriCC pushes all the performances to the new state-of-the-art even if it only adopts half of the pre-training iterations than baselines. In particular, on nuScenes semantic segmentation, TriCC relatively improves Res16UNet's fine-tuning performance with 1% annotations by 7.9% and on KITTI [20] 3D object detection with 5% annotations, it produces a 4% relative improvements. We hope this new framework will provide the community with new insights.

2. Related Work

2D Image Unsupervised Representation Learning Up to now, the contrastive learning method [9, 13, 29] is the mainstream discriminative framework for image representation learning and has received lots of success on many downstream tasks [4, 51, 54, 75, 77]. To learn good representations, its core idea is to make the features of positive samples closer and repulse the negative ones with InfoNCE loss [50] or its variants [9, 10, 13, 52]. In practice, contrastive learning methods benefit from large amounts of negative samples in a batch [13] or memory bank [29]. Also, there are methods [8–10, 52] that adopt clustering-based methods to reduce the dependence on negatives. Further on, BYOL [21] learns representations without negatives. Moreover, generative frameworks [17,28,90] achieve great progress in unsupervised representation learning, but they are more suitable for Transformer-based backbones instead of others like convolution. In this work, we follow the discrimination way to build our TriCC.

3D Point Cloud Unsupervised Representation Learning For point cloud, unsupervised representation learning methods can also be classified into generative methods and discriminative methods. For the former, GAN [1, 26, 69], self-reconstruction [19, 63, 82], up-sampling [39, 56], point cloud completion [60, 71, 76] are proposed as pretext tasks. And recently, masked auto-encoders [31, 53] are adopted inspired from the success of them on 2D image representation learning. For discriminative frameworks, contrastive method [11, 16, 27, 32, 38, 58, 72, 87] is also popular recently and for outdoor scenes that we focus on, many methods adopt it [34, 41, 78, 88]. STRL [34] learns representations from instance invariance in temporal dimension. GCC-3D [41] generates pseudo instances for clustering. CO^3 [12] utilizes vehicle-side and infrastructure-side LiDAR points to learn. ProposalContrast [84] learns robust 3D representations for detection by extending the contrast operation from point level to region proposals.

Multimodal Unsupervised Representation Learning Previously, much effort has been made into multimodal representation learning. Inspired from the success in paired image and text [55], paired audio and images [24], multimodal approach also extends to 3D area. [3,23] transfers semantics between RGB and depth images. [33,40] propose a distillation method that train 2D image backbones with 3D point cloud geometric information. While [2, 43, 45, 59], on the contrary, learn 3D representation with the help of pre-trained 2D image backbones. For autonomous driving, camera sensors are commonly available. Thus, we also take multimodal information as one of TriCC's source of semantics for helping 3D point cloud representation learning.

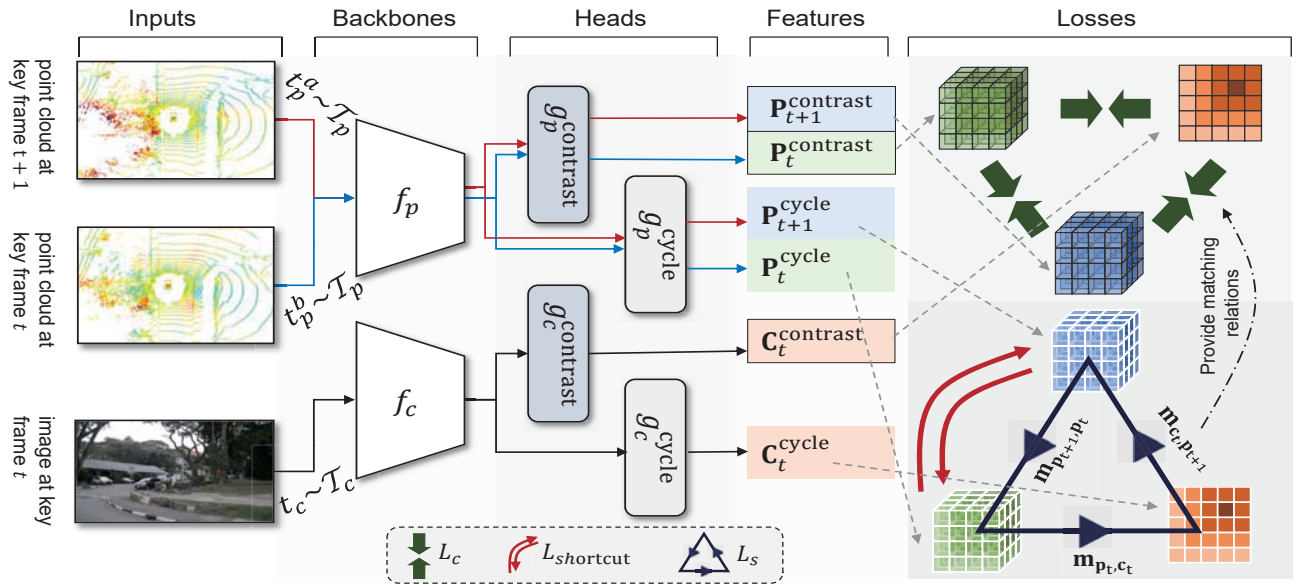

Figure 2. The pipeline of our TriCC. The triplet input composed of camera images, point cloud in the current key frame, and point cloud in the next key frame are augmented by a group of transformations ($\mathcal{T}_p$ for point clouds and $\mathcal{T}_c$ for images). After forward propagation in the backbones and projection heads, we get their feature maps. The feature maps from the "cycle" heads participate in the calculation of consistent constraint loss (the blue triangle and red double sided arrows in the figure) to learn representations and get matching relationships. And the relationships guide the triplet contrast loss (the green arrows) conducted on feature maps from the "contrast" heads.

Unsupervised Dense Temporal Correspondence TriCC also learns representations from temporal semantics. Therefore, unsupervised dense temporal correspondence learning is related to this paper. Optical flow [22, 46, 64, 67] is a well known technique for dense temporal correspondence. Dense tracking [65, 79] is a task to predict mask in latter frames of a given current mask. To free it from the costly annotation, many unsupervised methods have been developed [25, 49, 68, 70, 73, 92]. [36, 74] use cycle-consistency as the pretext task which inspires us to design the TriCC.

3. Triangle Constrained Contrast

In autonomous driving scene, abundant multimodal information with temporal dimension is commonly available. Due to the expensive annotation cost of dense tasks, in this paper we design an unsupervised 3D representation learning algorithm named TriCC tailored for autonomous driving. Our core objective is to provide a unified semantic learning method that can simultaneously utilize multimodal and temporal information so that it can boost the performance of outdoor point cloud semantic segmentation and 3D object detection to the hilt.

Our TriCC is composed of two main designs: 1) the Triangle Consistent Constraint (Sec. 3.1) and 2) the Triplet Contrast (Sec. 3.2). The former solves the methodological problem of uniformly utilizing the dense temporal and dense multimodal information under the discriminative framework. While the latter extends the traditional contrastive learning to the triplet contrast, a unified unsupervised framework for autonomous driving, as Fig. 2 shows.

3.1. Triangle Consistent Constraint

For dense 3D point cloud semantic segmentation and object detection task, current mainstream discrimination-based unsupervised representation framework contrastive learning relies on handcrafted positive pairs to learn effective representations. Its algorithm can be summarized by the following equation:

$$L = -\log\left(\frac{\exp(\mathrm{sim}(\mathbf{x}_a, \mathbf{x}_b)/\tau)}{\sum_j \exp(\mathrm{sim}(\mathbf{x}_a, \mathbf{x}_j)/\tau)}\right) \tag{1}$$

where $\mathbf{x}_a$ and $\mathbf{x}_b$ are the features of the positive pairs and $\{\mathbf{x}_j\}$ is the set containing all the negative samples and $\mathbf{x}_b$.

Motivation In previous works, dense matching relationships of different views from the same point cloud [78, 88], or relationships of calibrated camera image and LiDAR points [44, 59] are adopted as the positive pairs. However, dense relations in temporal dimension are hard to obtain and previous methods [34] can only contrast in frame level. This makes it difficult to unify the temporal semantics into the discriminative representation learning frameworks of other semantics. Thus, we design the Triangle Consistent Constraint to automatically find the temporal matching relations and learn representations from all the semantics uniformly.

Basic Framework The core design of the consistent constraint is to find matching relations and learn representations from a "self-cycle" formed among a group of dense feature maps. Specifically, we first get this feature map group

$\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^{n_i \times c}, i = 1, ..., k\}$ by backbone networks, where the features can be in any modality and with different timestamps. There is one restriction for the features in the group that they need to be under the same scenes with similar semantics so that their similarity can be calculated.

On this group of feature maps, the built "self-cycle" can be formulated in the following steps:

- Define the transition matrix between two feature maps:

$$\mathbf{M} = \{\mathbf{m}_{i,j} = \mathrm{sm}(\langle \mathbf{x}_i, \mathbf{x}_j \rangle / \tau) \in \mathbb{R}^{n_i \times n_j}\} \quad (2)$$

  where $\langle\rangle$ is the similarity function and sm is the softmax function in the second dimension which transforms the similarity matrix to the transition matrix of the pixel (point)-level features. τ is the temperature for softmax for better optimization.

- To build our "self-cycle", we only need the transition matrices of adjacent features to form a cycle:

$$\mathbf{S} \in \mathbb{R}^{n_1 \times n_1} = (\prod_{i=1}^{k-1} \mathbf{m}_{i,i+1})\mathbf{m}_{k,1} = P(\mathbf{x_1}|\mathbf{x}_1) \quad (3)$$

  where $\prod$ denotes the accumulated matrix multiplication. $\mathbf{S}$ is the consistent matrix of our "self-cycle", which links all the transition matrices of adjacent features and connects the tail and head to form a cycle. Each row in $\mathbf{S}$ represents the probability of one pixel (point) transferring back to every pixel (point) in its own feature map.

- A good representation should make every feature vector in the feature map return to itself with a higher probability after the cycle transition, since this kind of feature maps has better discrimination. We call this cycle-back to self as the consistent constraint and it can be optimized by:

$$L_s = \mathrm{CrossEntropy}(\log(\mathbf{S}), \mathbf{I}) \quad (4)$$

  where $\mathbf{I}$ is the identity matrix. L_s forces $\mathbf{M}$ to contain meaningful transition matrices through learning effective discrimination representations.

Practice Details In our experiments, we adopt the triangle consistent constraint. That is the feature map group is $\mathbf{X} = \{\mathbf{C}_t, \mathbf{P}_{t+1}, \mathbf{P}_t\}$, where $\mathbf{P}_t$, $\mathbf{C}_t$ are the point cloud feature and camera image feature of the tth key frame and $\mathbf{P}_{t+1}$ is the point cloud feature of the next key frame. Thus, we call it the triangle consistent constraint as Fig. 1 shows. We adopt cosine similarity as our similarity function.

Derived from the basic framework, we adopt the calibrated relationship $\hat{\mathbf{m}}_{\mathbf{P}_t,\mathbf{C}_t}$ between $\mathbf{P}_t$ and $\mathbf{C}_t$ to replace the original $\mathbf{m}_{\mathbf{P}_t,\mathbf{C}_t}$ gotten from feature similarity. Each row of $\hat{\mathbf{m}}_{\mathbf{P}_t,\mathbf{C}_t}$ is a one-hot vector representing the one (many) -to-one projected mapping from $\mathbf{P}_t$ to $\mathbf{C}_t$. This calibrated relationship is easy to obtain by the poses of the LiDAR and camera sensors in autonomous driving scene and will provide the optimization process with extra guidances.

This relatively short-term cycle method has a higher optimization efficiency. And we can also get the long-term temporal feature matching relationships through the learned representations (*e.g.* $\mathrm{sim}(\mathbf{P}_t, \mathbf{P}_{t+\Delta t}) = \langle \mathbf{P}_t, \mathbf{P}_{t+\Delta t} \rangle$) for further utilization as discussed in ablation study (Sec 4.4).

Advantages Consistent constraint forms an effective chain of feature similarity matching. Thus, it does not need any handcrafted temporal pixel (point)-level positive pairs to learn unsupervised representations simultaneously from both the multimodal and temporal information, instead, automatically finding matching relationships $\mathbf{M}$ across the modality and temporal dimensions. Therefore, it is tailored for the autonomous driving perception scenes with temporal multimodal information.

3.1.1 Cycle Shortcut

To improve the optimization efficiency, we add a mini cycle between $\mathbf{P}_t$ and $\mathbf{P}_{t+1}$ as a shortcut, which can be represented as:

$$\begin{aligned} \mathbf{S}_d &= \mathrm{sm}(\langle \mathbf{P}_t, \mathbf{P}_{t+1} \rangle / \tau) \times \mathrm{sm}(\langle \mathbf{P}_{t+1}, \mathbf{P}_t \rangle / \tau) \\ L_{\text{shortcut}} &= \mathrm{CrossEntropy}(\log(\mathbf{S}_d), \mathbf{I}) \end{aligned} \quad (5)$$

This is a small trick that learns a simpler consistent constraint to speed up the main constraint. In terms of symmetry, we should add three shortcuts on all the edges of the triangle cycle respectively, but from experiments, we find that one shortcut between $\mathbf{P}_t$ and $\mathbf{P}_{t+1}$ is enough.

3.2. Triplet Contrast

After introducing our core consistent constraint, we have an effective method to get the matching relationships among multimodal and temporal information. To enhance the representation learning efficiency, we further design a triplet contrastive loss based on these matching relationships. That is to contrast all the feature maps within the cycle chain to further strengthen the discrimination of representations on the feature hypersphere. Taking $\mathbf{C}_t$ and $\mathbf{P}_{t+1}$ as an example, the contrastive loss can be written as:

$$L_c^{\mathbf{C}_t,\mathbf{P}_{t+1}} = \frac{1}{n_{\mathbf{C}_t}} \sum_q -\log \frac{\exp(\mathrm{sim}(\mathbf{C}_t^q, \mathbf{P}_{t+1}^{\sigma(\mathbf{m}^q_{\mathbf{C}_t,\mathbf{P}_{t+1}})})/\tau)}{\sum_k \exp(\mathrm{sim}(\mathbf{C}_t^q, \mathbf{P}_{t+1}^k)/\tau)} \quad (6)$$

where $\mathbf{C}^q, \mathbf{P}^q \in \mathbb{R}^c$ denote the qth feature vector in the feature map and $\mathbf{m}^q_{\mathbf{C_t},\mathbf{P_{t+1}}}$ is the qth row of the transition matrix between $\mathbf{C}_t$ and $\mathbf{P}_{t+1}$. σ is the argmax function.

In our triplet contrast, this kind of contrast is also conducted between $(\mathbf{P}_t, \mathbf{P}_{t+1})$ and $(\mathbf{C}_t, \mathbf{P}_t)$. Thus, the integrated triplet contrast loss is:

$$L_c = L_c^{\mathbf{C}_t,\mathbf{P}_{t+1}} + L_c^{\mathbf{P}_t,\mathbf{P}_{t+1}} + L_c^{\mathbf{C}_t,\mathbf{P}_t} \quad (7)$$

Following SLidR [59] we also aggregate the image pixels into super-pixels and conduct triplet contrast in the unit of super-pixel (see supplementary for details).

3.3. Pipeline & Analysis

With the basic learning framework of our TriCC, we follow the current mainstream discriminative unsupervised learning structure to build our networks. The learning network is composed of a 2D backbone f_c, a 3D point cloud backbone f_p, and their projection heads for consistent constraint and contrast: g_c^{cycle}, g_c^{contrast}, g_p^{cycle}, and g_p^{contrast}, which means that the consistent constraint loss and contrast loss share the same 2D and 3D backbones but have their own heads. In previous sections, we do not distinguish these two kinds of features in our constraint and contrast losses for concision since this is only the network design, not the core algorithm. The projection heads are simple MLPs to assist the representation learning as the mainstream methods do [9, 10, 13, 29] and they are removed from the backbones in downstream transferring. The overall pipeline of TriCC is shown in Fig. 2. All the inputs are augmented by a group of transformation $\mathcal{T}$ and the outputs of the projection heads attached on the backbones are the feature maps we adopt in losses. The total training loss of TriCC is just:

$$L = L_s + L_{\text{shortcut}} + L_c \tag{8}$$

One limitation of our TriCC is that it needs the 2D and 3D backbones to keep the number of pixels and points. Thus, a UNet [57] like backbone is preferred. When applying it on backbones with large strides like VoxelNet [91], straightforward methods to satisfy this requirement can be adding a small decoder or adopting an interpolation layer.

Compared with previous discriminative unsupervised methods, TriCC is the first method that gets rid of dependences on pre-defined temporal positive pairs for 3D point cloud representation learning. The consistent constraint and the contrast loss divide the work of positive relationship digging and representation learning in a concise and compact algorithm. TriCC is the first method that can learn 3D representations from both multimodal and temporal semantics.

4. Experiments

To evaluate the proposed TriCC method, we transfer our pre-trained backbones (Res16UNet [15] and VoxelNet [91]) to several semantic segmentation (nuScenes [7] and SemanticKITTI [5]) and 3D object detection datasets (nuScenes and KITTI [20]) to compare with our baselines (PPKT [44] and SLidR [59]) and other SOTA unsupervised algorithms. Ablation studies are provided to show the effectiveness of each designed technique.

4.1. Pre-Training Details

Backbone & Training Structure We pre-train two 3D point cloud backbones with our TriCC: Res16UNet implemented by Minkowski Engine [15] and VoxelNet [91]. We apply kernels with size of $3 \times 3 \times 3$ for all sparse convolutions in Res16UNet as what is done in [59]. Different from Res16UNet which generates features for each input point, VoxelNet downsamples the feature maps with a $16 \times 8 \times 8$ stride and we get the features of each point by bilinear interpolation. Res16UNet takes in input with cylindrical coordinates and the voxel size is 0.1m for $z - axis$ and radius and 1° for azimuth angle. While VoxelNet takes input with Cartesian coordinates and the voxel size is (0.1m, 0.1m, 0.2m) for (x, y, z)-axis. As what is commonly accepted in the mainstream contrastive unsupervised learning methods [9, 10, 13, 21, 29], we adopt MLP projection heads (g_p^{cycle}, $g_p^{contrast}$) composed of two $1 \times 1 \times 1$ convolution layers and the first one is followed by a batchnorm [35] and a ReLU [48] layer. The selected 2D backbone is ResNet-50 [30] and uses the pre-trained weights obtained by unsupervised learning through MoCov2 [14]. The last 3 strided convolution layers are replaced by dilated convolution to get a total 4 stride for the whole backbone. We fix the parameters of ResNet and only train its projection heads g_c^{cycle}, $g_c^{contrast}$: a 1×1 convolution layer followed by a 4 times bilinear upsampling layer.

Pre-training Details We pre-train all models on nuScenes [7] dataset, which is a public large-scale dataset for autonomous driving containing 1000 20s scenes, 1,400,000 camera images as well as 90,000 LiDAR sweeps covering 16 semantic classes, designed for point cloud semantic segmentation and 3D object detection tasks. Models are trained on the official training set with 700 training scenes without any annotations. All the baselines are pre-trained for 50 epochs. For Res16UNet backbone, we train our TriCC on 8 GPU for 20 or 50 epochs using SGD with a initial learning rate of 2.0, momentum of 0.9, weight decay of 10^{-4}. The batch size is 16 and a cosine annealing scheduler that decreases the learning rate from its initial value to 0 is adopted. For the VoxelNet pre-training, the differences lie in that the initial learning rate is set to 0.4, the mini-batch size is 24. For TriCC, the temperature τ is 0.1 for consistent constraint and 0.07 for contrast.

Data Augmentation Basically we follow the data augmentation adopted in SLidR [59]. For 3D point cloud, a random rotation around the z-axis, a random flip with 50% probability over the direction of the x and y-axis, and a random cuboid dropping are applied to the points. On the other hand, images are transformed under a random horizontal flip and a random crop-resizing.

4.2. Transfer to Semantic Segmentation

Firstly, we evaluate TriCC's learned representation on semantic segmentation task and compare it with the baselines. We choose nuScenes [7] with 16 categories and SemanticKITTI [5] covering 19 categories as the transferring

Table 1. Comparisons of different pre-training methods and different backbones under the linear probing and few-shots fine-tuning evaluation protocols. On nuScenes, we use 100% annotated scans for linear probing and 1%, 5%, 10%, 25%, 100% annotation for fine-tuning. The results we report is the mIoU on the validation set of nuScenes. "P" and "C" denote point cloud and camera image modality.

Pre-training Method	Pretrain Modality	Linear	Fine-tuning				
		100%	1%	5%	10%	25%	100%
Res16UNet as the backbone							
fine-tune from scratch	-	8.1	30.3	47.7	56.6	64.8	74.2
PointContrast [78]	P	21.9	32.5 (+2.2)	-	57.1 (+0.5)	-	74.3 (+0.1)
DepthContrast [88]	P	22.1	31.7 (+1.4)	-	57.3 (+0.7)	-	74.1 (-0.1)
PPKT [44]	P, C	36.4	37.8 (+7.5)	51.7 (+4.0)	59.2 (+2.6)	66.8 (+2.0)	73.8 (-0.4)
SLidR [59]	P, C	38.0	38.2 (+7.9)	52.2 (+4.5)	58.8 (+2.2)	66.2 (+1.4)	74.6 (+0.4)
TriCC(ours), 20 epoch	P, C	37.8	40.8 (+10.5)	**54.1** (+6.4)	60.2 (+3.6)	**67.6** (+2.8)	75.3 (+1.1)
TriCC(ours), 50 epoch	P, C	38.0	**41.2** (+10.9)	**54.1** (+6.4)	**60.4** (+3.8)	**67.6** (+2.8)	**75.6** (+1.4)
VoxelNet as the backbone							
fine-tune from scratch	-	2.6	24.5	35.7	43.1	48.1	53.9
SLidR [59]	P, C	33.5	32.1 (+7.6)	40.3 (+4.6)	45.4 (+2.3)	50.3 (+2.2)	54.3 (+0.4)
TriCC (ours), 20 epoch	P, C	33.6	**34.0** (+9.5)	**42.0** (+6.3)	**46.7** (+3.6)	**51.6** (+3.5)	**56.0** (+2.1)

Table 2. Comparisons of different pre-training methods and different backbones for few-shots fine-tuning on SemanticKITTI. We use 1%, 5%, and 10% annotated scans. The results we report is the mIoU on the validation set of SemanticKITTI.

Pre-training Method	Fine-tuning		
	1%	5%	10%
Res16UNet as the backbone			
fine-tune from scratch	39.5	52.1	55.6
PPKT [44]	43.9 (+4.4)	53.1 (+1.0)	57.3 (+1.7)
SLidR [59]	44.6 (+5.1)	52.6 (+0.5)	56.0 (+0.4)
TriCC (ours), 20 epoch	45.8 (+6.3)	55.7 (+3.6)	58.4 (+2.8)
TriCC (ours), 50 epoch	**45.9** (+6.4)	**55.9** (+3.8)	**59.0** (+3.4)
VoxelNet as the backbone			
fine-tune from scratch	28.8	40.8	46.4
SLidR [59]	35.2 (+6.4)	45.5 (+4.7)	48.6 (+2.2)
TriCC (ours), 20 epoch	**36.5** (+7.7)	**46.8** (+6.0)	**49.8** (+3.4)

datasets. Following the common setting [5,59,80], we evaluate the results using the validation set of nuScenes and the sequence 08 of SemanticKITTI. The fine-tuning and linear probing training details can be found in supplementary.

On nuScenes The semantic segmentation models are built by adding a point-wise linear classification head on pre-trained backbones. Two evaluation protocols are utilized to evaluate the pre-trained models: 1) linear probing and 2) fine-tuning. For the former, we initialize the parameters of the backbones with the pre-trained weights, fix them, and only train the linear segmentation head. For fine-tuning, we train the whole segmentation models with different proportions of available annotated training data to compare the annotation efficiency. The training objective is the combination of the cross-entropy and Lovász loss [6]. PPKT [44] and SLidR [59] are also tested as baselines serving as comparisons to our models. Results are reported in Tab. 1.

Table 3. Comparisons of different pre-training methods for few-shots fine-tuning on KITTI. We use 5%, 10%, and 20% annotated scans. The results we report are the mAP_R40 in easy, moderate, and hard level on the validation set of KITTI dataset.

Pretrain	Fine-tuning								
	5% label			10% label			20% label		
	E	M	H	E	M	H	E	M	H
Res16UNet + PointRCNN									
random	73.7	56.6	50.7	74.6	58.8	53.9	77.9	63.7	59.2
PPKT [44]	75.7	59.6	54.4	78.3	63.7	58.4	78.9	64.8	59.9
SLidR [59]	74.5	58.8	52.9	78.1	63.5	58.3	77.6	63.8	59.2
Ours, 20ep	**77.9**	**61.3**	**56.2**	**79.6**	**64.6**	**59.3**	**80.0**	**65.9**	**60.7**
VoxelNet + PV-RCNN									
random	79.4	65.4	61.6	78.8	67.1	63.2	81.9	70.1	66.9
SLidR [59]	80.5	67.9	63.9	80.8	68.2	64.5	81.6	70.5	67.1
Ours, 20ep	**81.4**	**68.6**	**64.7**	**82.3**	**69.4**	**65.8**	**83.3**	**72.1**	**68.5**
VoxelNet + SECOND									
random	68.0	54.7	51.7	71.9	60.2	56.8	73.2	61.8	58.5
SLidR [59]	69.7	57.8	54.5	73.0	**62.3**	**59.0**	73.9	63.0	59.6
Ours, 20ep	**70.7**	**59.6**	**56.4**	**73.5**	62.0	58.8	**74.6**	**63.8**	**60.7**

From the results, we can see that all the unsupervised pre-trained methods achieve better results than the random initialization. Compared with pure point-cloud unsupervised methods, methods that learn representations from multimodal semantics achieve better performances, revealing the importance of multimodal information.

Compared with all the baselines, our TriCC on Res16UNet achieves much better performances and pushes the SOTA fine-tuning results by 1.0 mIoU. And its learning efficiency is much higher too since TriCC pre-trained for 20 epochs surpasses all the baselines pre-trained for 50 epochs

Table 4. Comparisons of different pre-training methods for few-shots fine-tuning on nuScenes. We use 5%, 10%, and 20% annotated scans. We report the mAP and NDS metrics.

Pretrain	Fine-tuning					
	5% label		10% label		20% label	
	mAP	NDS	mAP	NDS	mAP	NDS
VoxelNet + CenterPoint						
random	38.0	44.3	46.9	55.5	50.2	59.7
Point Con. [78]	39.8	45.1	47.7	56.0	-	-
GCC-3D [41]	41.1	46.8	48.4	56.7	-	-
SLidR [59]	43.3	52.4	47.5	56.8	50.4	59.9
TriCC, 20epoch	**44.6**	**54.4**	**48.9**	**58.1**	**50.9**	**60.3**
VoxelNet + SECOND						
random	35.8	45.9	39.0	51.2	43.1	55.7
SLidR [59]	36.6	48.1	39.8	52.1	44.2	56.3
TriCC, 20epoch	**37.8**	**50.0**	**41.4**	**53.5**	**45.5**	**57.7**

a large gap. For few-shot fine-tuning, TriCC on Res16UNet improves 10.9 mIoU than the random initialization setting, 3.0 mIoU better than the previous SOTA results, under the 1% fine-tuning setting. And for 25% fine-tuning, TriCC also improves the SOTA by 1.4 mIoU. For the VoxelNet backbone, our TriCC achieves similar improvements: 1.9%, 1.3%, 1.7% mIoU boost for 1%, 25%, 100% fine-tuning. All the above results prove the effectiveness of the new proposed unified method TriCC that integrates all the semantics in the autonomous driving scene. As for linear probing results, regrettably, TriCC does not learn obviously better linearly-separable representations. We believe the reason is that non-linear features are important here as [28] shows.

On SemanticKITTI We also conduct experiments on SemanticKITTI [5] dataset with 1%, 5%, and 10% few-shot fine-tuning setting. We adopt PPKT and SLidR as our baselines and the results are reported in Tab. 2.

From the results, we can see that TriCC achieves much better performances than the baselines on both Res16UNet and VoxelNet backbones, and pushes the results by $\sim$2.5 mIoU for Res16UNet and $\sim$1.3 mIoU for VoxelNet.

4.3. Transfer to 3D Object Detection

Here, we further evaluate TriCC on 3D object detection task with the nuScenes [7] and KITTI [20] datasets. Baselines are pre-trained for 50 epochs (except GCC-3D, also 20 epochs) while TriCC is only pre-trained for 20 epochs. Detailed fine-tuning settings can be found in supplementary.

Results The few-shot fine-tuning results on KITTI [20] are reported in Tab. 3 and the results on nuScenes [7] are reported in Tab. 4. It's obviously seen that TriCC gives a higher performance on almost all the detection models including PointRCNN [62], PV-RCNN [61], SECOND [81], and CenterPoint [85]. The non-ideal results on 10% KITTI with SECOND model are strange and may be due to the coincidence of uniform sampling of training data.

Table 5. Comparisons with SOTA unsupervised 3D representation learning methods on KITTI fine-tuning with 100% annotations. We report mAP@R11 for SECOND models and mAP@R40 for PV-RCNN models.

Pretrain	Det.	Fine-tuning		
		Easy	Moderate	Hard
mAP@R11 w/o road planes				
random	Sec.	73.3	63.2	60.3
SwAV [9]	Sec.	73.2 (-0.1)	64.0 (+0.8)	60.9 (+0.6)
DeepCluster [8]	Sec.	73.2 (-0.1)	63.4 (+0.2)	60.1 (-0.2)
BYOL [21]	Sec.	71.1 (-2.2)	60.4 (-2.8)	57.0 (-3.3)
Point Con. [78]	Sec.	72.7 (-0.6)	62.7 (-0.5)	59.2 (-1.1)
GCC-3D [41]	Sec.	73.9 (+0.6)	63.5 (+0.3)	59.8 (-0.5)
STRL [34]	Sec.	74.0 (+0.7)	63.9 (+0.7)	60.9 (+0.6)
SLidR [59]	Sec.	73.6 (+0.3)	64.6 (+1.4)	61.5 (+1.2)
CO^3 [12]	Sec.	74.4 (+1.1)	64.4 (+1.2)	60.9 (+0.6)
TriCC (ours)	Sec.	**75.0** (+1.7)	**65.7** (+2.5)	**62.2** (+1.9)
mAP@R40 with road planes				
random	PV	81.3	70.6	66.1
Point Con. [78]	PV	82.8 (+1.5)	71.6 (+1.0)	67.5 (+1.4)
GCC-3D [41]	PV	-	71.3 (+0.7)	-
STRL [34]	PV	-	71.5 (+0.9)	-
SLidR [59]	PV	82.9 (+1.6)	71.9 (+1.3)	68.0 (+1.9)
Pro. Con. [84]	PV	**84.5** (+3.2)	72.9 (+2.3)	69.0 (+2.9)
TriCC (ours)	PV	84.1 (+2.8)	**73.3** (+2.7)	**69.4** (+3.3)

Comparison with SOTA In Tab. 5, we report the KITTI detection fine-tuning results with 100% annotation and comparisons with previous SOTA unsupervised representation learning methods. From the results, we can see that TriCC provides a 2.5 mAP and 2.7 mAP performance boost over the random initialization for SECOND and PV-RCNN models. And TriCC achieves new SOTA results even compared with ProposalContrast [84], a pre-training method tailored for 3D object detection, and CO^3 that adopts extra LiDAR point clouds from the infrastructure side.

4.4. Ablation Study

We conduct ablation studies on core techniques of TriCC and the results are shown in Tab. 6. All the ablation studies are conduct on 1% nuScenes segmentation fine-tuning. From Tab. 6a, we can see that the three designed losses are important for learning better representations. Besides, the consistent constraint alone can learn effective representations by itself, and results are competitive to previous methods. From Tab. 6b, it is shown that with the contrast pairs of $(\mathbf{P}_t, \mathbf{C}_t)$ and $(\mathbf{P}_{t+1}, \mathbf{C}_t)$, our TriCC can work well. $(\mathbf{P}_{t+1}, \mathbf{P}_t)$ boosts the performance a bit. Since it does not

Table 6. Ablations on nuScenes semantic segmentation. We pre-train the backbones for 20 epochs and fine-tune them with 1% annotations.

(a) The three sub-losses are necessary for the TriCC framework.

backbone	losses	mIoU
Res16UNet	L_s	38.1
Res16UNet	$L_s + L_c$	39.7
Res16UNet	$L_s + L_c + L_{shortcut}$	40.8
VoxelNet	L_s	32.2
VoxelNet	$L_s + L_c$	33.1
VoxelNet	$L_s + L_c + L_{shortcut}$	34.0

(b) Three components in L_c are enough. Long-term contrast leads to no further improvement.

contrast items	mIoU
$(\mathbf{P}_t, \mathbf{C}_t)$	39.6
$(\mathbf{P}_t, \mathbf{C}_t), (\mathbf{P}_{t+1}, \mathbf{C}_t)$	40.6
$(\mathbf{P}_t, \mathbf{C}_t), (\mathbf{P}_{t+1}, \mathbf{C}_t), (\mathbf{P}_{t+1}, \mathbf{P}_t)$	40.8
Triplet Contrast + $(\mathbf{P}_t, \mathbf{P}_{t+2})$	40.2
Triplet Contrast + $(\mathbf{C}_t, \mathbf{P}_{t+2})$	39.9

(c) Standalone heads for L_s, L_c are necessary for avoiding mutual interference.

backbone	heads setting	mIoU
Res16UNet	Only L_s	38.1
Res16UNet	L_s L_c share head	39.5
Res16UNet	Two heads for L_s, L_c	40.8
VoxelNet	Only L_s	32.2
VoxelNet	L_s L_c share head	32.8
VoxelNet	Two heads for L_s, L_c	34.0

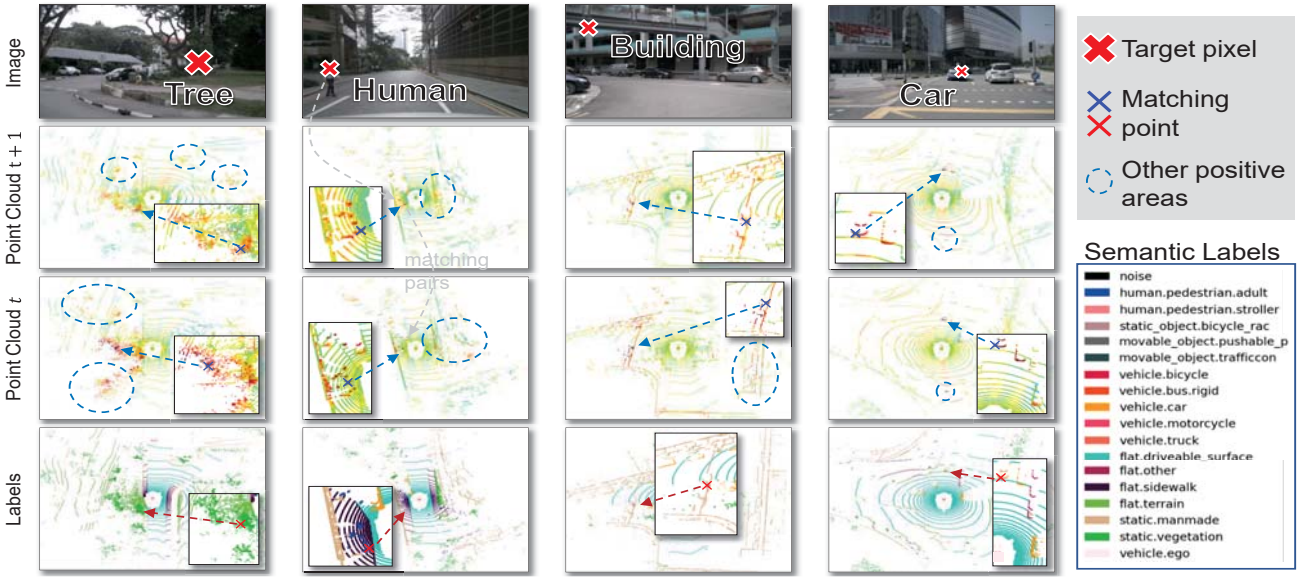

Figure 3. Visualization of the dense matching quality. We pick a pixel from the input image and show the similarity heat maps of all the point cloud points in the current and next key frame according to the features from the "cycle" head. It is seen that all the target pixels are successfully matched to points with the right semantics and this matching is not a one-to-one pre-defined matching since many positive areas are dug out as the blue dashed cycles show. This reveals our consistent constraint can efficiently find positive pairs for discrimination.

lead to much overhead, we keep it in our algorithm. When further adding long-term contrast pairs, the performances get worse. We believe this is because the long-term matching relationships contain too much noises to guide the representation learning. In Tab. 6c, we can see that two standalone heads for contrast and consistent constraint lead to better results than sharing one head.

4.5. Visualization & Analysis

We visualize the matching quality through the similarity heat map in Fig. 3. We can find that the proposed consistent constraint can successfully match the right semantics across the multimodal and temporal information. Due to the soft matching algorithm in the constraint, TriCC can dig out all the similar pairs as the blue dashed cycle shows, which is another key advantage over the handcrafted one-to-one positive pairs, besides the automatically pairing process.

5. Conclusion

We propose the Triangle Constrained Contrast (TriCC) model for learning unsupervised dense 3D point cloud representations from both multimodal and temporal information. TriCC follows the mainstream discriminative framework but does not need the pre-defined handcrafted dense temporal positive pairs. Triangle consistent constraint and triplet contrast are its two main components which are designed to find dense temporal matching relations automatically and learn representations. Thus, TriCC can unify multimodal and temporal semantics which are commonly available in autonomous driving scenes in one concise algorithm. Experiments show its superiorities in downstream semantic segmentation and 3D object detection. We hope it provides new insights for the community of representation learning.

Acknowledgement This work was supported by the National Key Research and Development Project of China (No. 2021ZD0110704), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Qi Zhi Institute, and Shanghai Science and Technology Commission (21511101200). Bo Pang would like to thank the support of Baidu Scholarship.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *Int. Conf. Mach. Learn.*, pages 40–49. PMLR, 2018. 2

[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 *CVPR*, pages 9902–9912, 2022. 2

[3] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. *arXiv preprint arXiv:2204.01678*, 2022. 2

[4] Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Contrastive learning for unsupervised video highlight detection. In *CVPR*, pages 14042–14052, 2022. 2

[5] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, pages 9297–9307, 2019. 5, 6, 7

[6] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, pages 4413–4421, 2018. 6

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11621–11631, 2020. 1, 2, 5, 7

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 2, 7

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33:9912–9924, 2020. 2, 5, 7

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 2, 5

[11] Haolan Chen, Shitong Luo, Xiang Gao, and Wei Hu. Unsupervised learning of geometric sampling invariant representations for 3d point clouds. In *ICCV*, pages 893–903, 2021. 2

[12] Runjian Chen, Yao Mu, Runsen Xu, Wenqi Shao, Chenhan Jiang, Hang Xu, Zhenguo Li, and Ping Luo. Coˆ 3: Cooperative unsupervised 3d representation learning for autonomous driving. *arXiv preprint arXiv:2206.04028*, 2022. 2, 7

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conf. Mach. Learn.*, pages 1597–1607. PMLR, 2020. 2, 5

[14] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 5

[15] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 5

[16] Bi'an Du, Xiang Gao, Wei Hu, and Xin Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In *ACM MM*, pages 3133–3142, 2021. 2

[17] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *NeurIPS*, 2022. 2

[18] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020. 1

[19] Xiang Gao, Wei Hu, and Guo-Jun Qi. Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations. In *CVPR*, pages 7163–7172, 2020. 2

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 5, 7

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020. 2, 5, 7

[22] Xiuye Gu, Yijie Wang, Chongruo Wu, Yong Jae Lee, and Panqu Wang. Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds. In *CVPR*, pages 3254–3263, 2019. 3

[23] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *CVPR*, pages 2827–2836, 2016. 2

[24] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP*, pages 976–980. IEEE, 2022. 2

[25] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *NeurIPS*, 2020. 3

[26] 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 *AAAI*, volume 33, pages 8376–8384, 2019. 2

[27] Kaveh Hassani and Mike Haley. Unsupervised multi-task feature learning on point clouds. In *ICCV*, pages 8160–8171, 2019. 2

[28] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 2, 7

[29] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 2, 5

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[31] Georg Hess, Johan Jaxing, Elias Svensson, David Hagerman, Christoffer Petersson, and Lennart Svensson. Masked autoencoders for self-supervised learning on automotive point clouds. *arXiv preprint arXiv:2207.00531*, 2022. 2

[32] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *CVPR*, pages 15587–15597, 2021. 2

[33] Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. Pri3d: Can 3d priors help 2d representation learning? In *ICCV*, pages 5693–5702, 2021. 2

[34] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *ICCV*, pages 6535–6545, 2021. 2, 3, 7

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Int. Conf. Mach. Learn.*, pages 448–456. PMLR, 2015. 5

[36] Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020. 3

[37] Chiyu Max Jiang, Mahyar Najibi, Charles R Qi, Yin Zhou, and Dragomir Anguelov. Improving the intra-class long-tail in 3d detection via rare example mining. *ECCV*, 2022. 1

[38] Shamit Lal, Mihir Prabhudesai, Ishita Mediratta, Adam W Harley, and Katerina Fragkiadaki. Coconets: Continuous contrastive 3d scene representations. In *CVPR*, pages 12487–12496, 2021. 2

[39] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: a point cloud upsampling adversarial network. In *ICCV*, pages 7203–7212, 2019. 2

[40] Zhenyu Li, Zehui Chen, Ang Li, Liangji Fang, Qinhong Jiang, Xianming Liu, Junjun Jiang, Bolei Zhou, and Hang Zhao. Simipu: Simple 2d image and 3d point cloud unsupervised pre-training for spatial-aware visual representations. In *AAAI*, volume 36, pages 1500–1508, 2022. 2

[41] Hanxue Liang, Chenhan Jiang, Dapeng Feng, Xin Chen, Hang Xu, Xiaodan Liang, Wei Zhang, Zhenguo Li, and Luc Van Gool. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In *ICCV*, pages 3293–3302, 2021. 2, 7

[42] Minghua Liu, Yin Zhou, Charles R Qi, Boqing Gong, Hao Su, and Dragomir Anguelov. Less: Label-efficient semantic segmentation for lidar point clouds. *ECCV*, 2022. 1

[43] Yunze Liu, Li Yi, Shanghang Zhang, Qingnan Fan, Thomas Funkhouser, and Hao Dong. P4contrast: Contrastive learning with pairs of point-pixel pairs for rgb-d scene understanding. *arXiv preprint arXiv:2012.13089*, 2020. 2

[44] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.04687*, 2021. 2, 3, 5, 6

[45] Yueh-Cheng Liu, Yu-Kai Huang, Hung-Yueh Chiang, Hung-Ting Su, Zhe-Yu Liu, Chin-Tang Chen, Ching-Yu Tseng, and Winston H Hsu. Learning from 2d: Contrastive pixel-to-point knowledge transfer for 3d pretraining. *arXiv preprint arXiv:2104.04687*, 2021. 2

[46] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*. Vancouver, British Columbia, 1981. 3

[47] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, et al. One million scenes for autonomous driving: Once dataset. *arXiv preprint arXiv:2106.11037*, 2021. 1

[48] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Int. Conf. Mach. Learn.*, 2010. 5

[49] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 3

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

[51] Bo Pang, Yizhuo Li, Yifan Zhang, Gao Peng, Jiajun Tang, Kaiwen Zha, Jiefeng Li, and Cewu Lu. Unsupervised representation for semantic segmentation by implicit cycle-attention contrastive learning. In *AAAI*. AAAI, 2022. 2

[52] Bo Pang, Yifan Zhang, Yaoyi Li, Jia Cai, and Cewu Lu. Unsupervised visual representation learning by synchronous momentum grouping. *ECCV*, 2022. 2

[53] 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. 2

[54] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, pages 6964–6974, 2021. 2

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[56] Edoardo Remelli, Pierre Baque, and Pascal Fua. Neuralsampler: Euclidean point cloud auto-encoder and sampler. *arXiv preprint arXiv:1901.09394*, 2019. 2

[57] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[58] Aditya Sanghi. Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning. In *ECCV*, pages 626–642. Springer, 2020. 2

[59] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *CVPR*, pages 9891–9901, 2022. 2, 3, 5, 6, 7

[60] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *ECCV*, pages 236–250. Springer, 2016. 2

[61] 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 *CVPR*, pages 10529–10538, 2020. 7

[62] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 7

[63] Yi Shi, Mengchen Xu, Shuaihang Yuan, and Yi Fang. Unsupervised deep shape descriptor with point distribution learning. In *CVPR*, pages 9353–9362, 2020. 2

[64] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 3

[65] Jianhua Sun, Zehao Wang, Jiefeng Li, and Cewu Lu. Unified and fast human trajectory prediction via conditionally parameterized normalizing flow. *IEEE Robotics and Automation Letters*, 7(2):842–849, 2021. 3

[66] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2446–2454, 2020. 1

[67] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 3

[68] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning video object segmentation with visual memory. In *ICCV*, pages 4481–4490, 2017. 3

[69] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3d point clouds via graph convolution. In *ICLR*, 2018. 2

[70] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, pages 9481–9490, 2019. 3

[71] Hanchen Wang, Qi Liu, Xiangyu Yue, Joan Lasenby, and Matt J Kusner. Unsupervised point cloud pre-training via occlusion completion. In *ICCV*, pages 9782–9792, 2021. 2

[72] 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. In *AAAI*, volume 35, pages 2773–2781, 2021. 2

[73] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, pages 1328–1338, 2019. 3

[74] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, pages 2566–2576, 2019. 2, 3

[75] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pages 3024–3033, 2021. 2

[76] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *CVPR*, pages 1939–1948, 2020. 2

[77] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pages 8392–8401, 2021. 2

[78] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*, pages 574–591. Springer, 2020. 2, 3, 6, 7

[79] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018. 3

[80] Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *ECCV*, 2022. 6

[81] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 7

[82] Juyoung Yang, Pyunghwan Ahn, Doyeon Kim, Haeil Lee, and Junmo Kim. Progressive seed generation auto-encoder for unsupervised point cloud learning. In *ICCV*, pages 6413–6422, 2021. 2

[83] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *CVPR*, pages 11495–11504, 2020. 1

[84] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposalcontrast: Unsupervised pre-training for lidar-based 3d object detection. *ECCV*, 2022. 2, 7

[85] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, pages 11784–11793, 2021. 7

[86] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018. 1

[87] Ling Zhang and Zhigang Zhu. Unsupervised feature learning for point cloud understanding by contrasting and clustering using graph convolutional neural networks. In *Int. Conf. on 3D Vis.*, pages 395–404. IEEE, 2019. 2

[88] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *ICCV*, pages 10252–10263, 2021. 2, 3, 6

[89] Dingfu Zhou, Jin Fang, Xibin Song, Liu Liu, Junbo Yin, Yuchao Dai, Hongdong Li, and Ruigang Yang. Joint 3d instance segmentation and object detection for autonomous driving. In *CVPR*, pages 1839–1849, 2020. 1

[90] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2

[91] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 5

[92] Fangrui Zhu, Li Zhang, Yanwei Fu, Guodong Guo, and Weidi Xie. Self-supervised video object segmentation. *arXiv preprint arXiv:2006.12480*, 2020. 3