ContrastMotion: Self-supervised Scene Motion Learning for Large-Scale LiDAR Point Clouds

Xiangze Jia[1] , **Hui Zhou**[2] , **Xinge Zhu**[2] , **Yandong Guo**[3] , **Ji Zhang**[1,4*] , **Yuexin Ma**[5*]
[1]Nanjing University of Aeronautics and Astronautics
[2]The Chinese University of Hong Kong
[3]OPPO Research Institute
[4]Zhejiang Lab
[5]ShanghaiTech University
jxzxiao5@nuaa.edu.cn, ji_zhang_nuaa@gmail.com, mayuexin@shanghaitech.edu.cn

Abstract

In this paper, we propose a novel self-supervised motion estimator for LiDAR-based autonomous driving via BEV representation. Different from usually adopted self-supervised strategies for data-level structure consistency, we predict scene motion via feature-level consistency between pillars in consecutive frames, which can eliminate the effect caused by noise points and view-changing point clouds in dynamic scenes. Specifically, we propose *Soft Discriminative Loss* that provides the network with more pseudo-supervised signals to learn discriminative and robust features in a contrastive learning manner. We also propose *Gated Multi-frame Fusion* block that learns valid compensation between point cloud frames automatically to enhance feature extraction. Finally, *pillar association* is proposed to predict pillar correspondence probabilities based on feature distance, and whereby further predicts scene motion. Extensive experiments show the effectiveness and superiority of our **ContrastMotion** on both scene flow and motion prediction tasks.

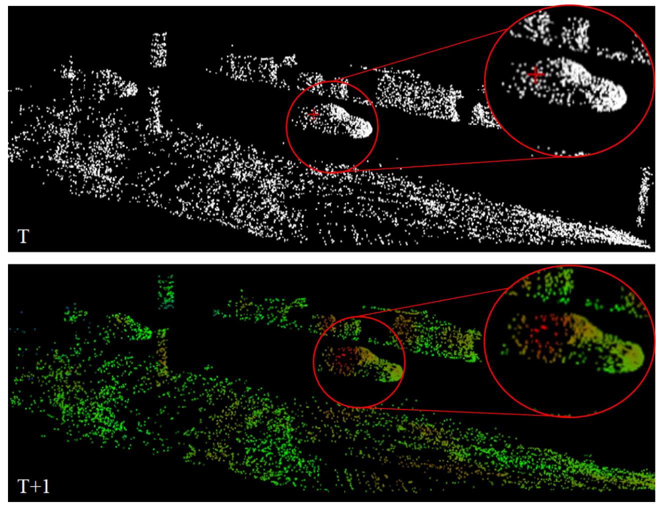

Figure 1: **The corresponding probabilities between pillars in $T+1$ and the reference pillar (red cross) in T inferred from ContrastMotion.** ContrastMotion extracts discriminative features based on geometric context information and predict corresponding probabilities between pillars in $T+1$ and the reference based on feature distance. The transition from red to green indicates that the corresponding probability is decreasing.

1 Introduction

For autonomous vehicles, it is critical to understand the dynamic cues in large-scale scenarios, e.g. distinguishing movable and static objects in the environment, to avoid obstacles and guarantee the safety [Mahjourian *et al.*, 2018; Luo *et al.*, 2018]. Due to the advantage of long-range depth capture, LiDAR has become one popular sensor for scene perception [Cong *et al.*, 2022; Yin *et al.*, 2021; Zhu *et al.*, 2020; Zhu *et al.*, 2021] and boosted the development of autonomous driving. Extracting motion features from LiDAR point cloud has attracted more and more attention in recent years, which is a fundamental technique for many downstream tasks, such as detection, segmentation, trajectory prediction, navigation, etc.

However, 3D labeling on large-scale point clouds requires complex processes and labor-intensive human efforts. Unlike the regular pixel representation of images, LiDAR point cloud is sparsity-varying and unordered in one scene and the discrepancy is more obvious for cross-scenes and cross-LiDAR captured data, leading to the situation that supervised methods are difficult to adapt to other domains. Considering above limitations, we focus on the self-supervised learning strategy to extract the motion pattern existing in consecutive frames of LiDAR point cloud.

Scene motion is a fine-grained representation of the motion pattern, which is defined as a 3D displacement vector between each point in two sequential frames [Wu *et al.*, 2019]. There are already some self-supervised approaches [Gu *et al.*, 2022; Baur *et al.*, 2021; Li *et al.*, 2022b] for scene flow estimation. However, most of them rely on point-wise feature extraction and correlation corresponding, which bring

*Corresponding author
https://github.com/JXZxiaowu/ContrastMotion

heavy costs of memory and computation for large-scale point clouds. Meanwhile, the data-level structure consistency is not applicable for dynamic scenes, because LiDAR is view-dependent and will capture different views of objects in different frames, causing many points having no correspondences in adjacent frames and leading to low accuracy for the motion estimation. These properties limit the deployment of such methods in real scenes, which is highly dynamic and requires high-efficiency point cloud processing.

Instead of dense motion prediction, researchers tend to explore the motion representation on Bird's Eye View (BEV). Due to the rich semantic and geometry information existing in the BEV representation, especially for the traffic scenarios, BEV provides a trade-off choice for accurate and efficient perception in large-scale scenes [Ma *et al.*, 2022]. Moreover, BEV representation can, to some extent, eliminate the effect caused by unordered and noisy points by discretizing the point cloud into BEV grid cells. Such compact representation is friendly for defining the search window for point matching. Thus, based on one popular BEV [Li *et al.*, 2022c] and pillar [Lang *et al.*, 2019] representation, , [Luo *et al.*, 2021; Baur *et al.*, 2021] propose effective motion estimators, which are more applicable for autonomous driving. However, the self-supervised architecture is similar to previous scene flow methods by mainly using the point-level structure consistency and regularization between consecutive frames of point cloud for self-supervision. Such methods, depending on data-level correspondences, still suffers from dynamic changes of scenes.

In this paper, based on the pillar representation, we propose a novel self-supervised scene motion estimator for LiDAR captured large-scale point clouds, which has solved above problems. To reduce the effect of point-level correspondences missing during the view-dependent scanning in dynamic scenes, we extract the high-level feature representation of each pillar and then predict pillar corresponding probabilities between consecutive frames (Fig 1). After obtaining the pillar correspondences, we easily calculate the motion information of the whole scene. Specifically, we utilize contrastive learning to bootstrap the network to extract discriminative features of each pillar, where the positive pillar pairs are two pillars that have a corresponding relationship in consecutive frames. To acquire positive pairs without any labels, we leverage commonly used data augmentation methods to generate two frames of point clouds from the same sample and record the pillar correspondences according to the transformation matrix. Although the hard one-to-one positive pillar assignment increases the discriminability of features, it provides less supervision and makes it difficult for the network to learn strong and robust features. Therefore, we propose *Soft Discriminative Loss(SD–Loss)*, which generates positive pillar pairs with a soft constraint to further provide a margin of fault tolerance in the dynamic scenes. We also propose *Gated Multi-frame Fusion (GMF)* block learning valid compensation to enhance feature consistency of corresponding pillars, where the gate status also provides dynamic and static information about the scene. In *pillar association*, we predict the pillar correspondence probabilities mainly through the feature distance of pillars between consecutive frames. Moreover, in order to reduce the complexity of pillar association from quadratic of the number of pillars to linear, we adopt *Patching* and *Asymmetric Patch* strategy.

We validate our ContrastMotion on the KITTI Scene Flow dataset and nuScenes dataset and achieve state-of-the-art performance for both scene flow and motion prediction tasks. We also conduct ablation studies to illustrate the effectiveness of the novel designs in our method.

The contributions of our work can be summarized as follows.

- We propose a novel pillar-based self-supervised motion estimator for LiDAR point cloud, which predicts pillar corresponding probabilities between point cloud pair by feature distance, and further acquires scene motion.
- We propose Soft Discriminative Loss(SD–Loss) to provide the network with more pseudo-supervised signals to learn discriminative and robust features, and it can also tackle the case of correspondence points missing in real scenes.
- We propose Gated Multi-frame Fusion(GMF) block learning valid compensation between point cloud pairs automatically to enhance feature consistency of corresponding pillars, which also has the capability of perceiving dynamic-static information of the scene.
- We analyze the degradation solution, and our ContrastMotion has achieved state-of-the-art performance on both scene flow estimation and motion prediction tasks.

2 Related Work

2.1 Supervised Motion Learning

This task aims to estimate motions and predict the future locations of various agents via past information [Luo *et al.*, 2021]. The previous works [Ma *et al.*, 2019; Sadeghian *et al.*, 2019] formulate this task as a trajectory problem based on 3D object detection and tracking. Another active research line [Wu *et al.*, 2020; Lee *et al.*, 2020; Wei *et al.*, 2021; Li *et al.*, 2022a; Wang *et al.*, 2022] estimate the motions in a end-to-end framework. However, 3D labeling on large-scale point clouds requires complex processes and labor-intensive human efforts, and there are significant discrepancies between cross-scenes and cross LiDAR captured data.

2.2 Self Supervised Motion Learning

In the field of the point cloud, There are already self-supervised [Wu *et al.*, 2019; Mittal *et al.*, 2020; Kittenplon *et al.*, 2021; Li *et al.*, 2021a; Hur and Roth, 2021; Kittenplon *et al.*, 2021; Wu *et al.*, 2019] and non-learning [Li *et al.*, 2021b] methods for motion estimation. PointPWC [Wu *et al.*, 2019] estimates the motions with a coarse-to-fine paradigm and introduces the losses for scene consistency, smoothness and shape approximation. However, these methods perform significant sampling in the training and inference pipelines due to the heavy costs of memory and computation. Instead of dense motion, researchers tend to explore the motion in pillar representation. PillarMotion [Luo *et al.*,

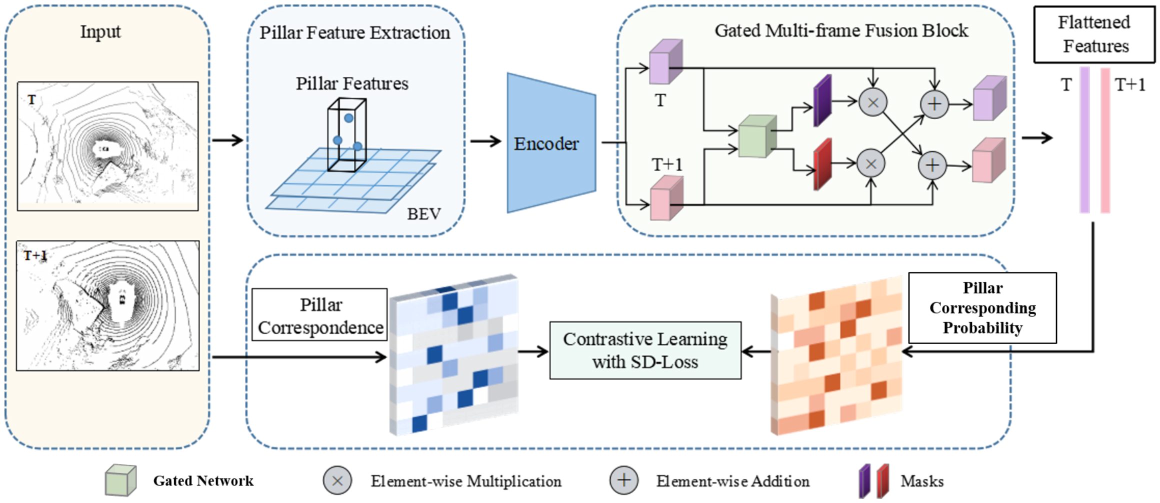

Figure 2: A schematic overview of the proposed self-supervised motion estimator for LiDAR point cloud. We introduce a feature-level consistency augmented with *SD–Loss* to achieve self-supervision in a contrastive learning manner. We also propose *Gated Multi-frame Fusion* block learning valid compensation to enhance feature similarity of corresponding pillars

2021] introduces structural consistency augmented with motion masking and a cross-sensor (LiDAR and camera) regularization to predict the motion of pillars. SLIM [Baur *et al.*, 2021] proposes a self-supervised model for motion estimation and segmentation in a BEV map. However, these methods are similar to previous scene flow methods by mainly using the point-level structure consistency and regularization between consecutive frames, which suffer from the dynamic changes of scenes.

2.3 Contrastive Learning

Contrastive learning [Radford *et al.*, 2021; Bai *et al.*, 2022] has been widely used due to the great potential in pre-training and feature correspondence learning. In the point cloud field, PointContrast [Xie *et al.*, 2020] proposes a pre-trained method for learning representations via matching corresponding points in different views. Specifically, it establishes the connections of corresponding points in different perspectives with the proposed PointInfoNCE to enhance the representation of the model. However, The pretext task only involves point-level correspondence matching which completely ignores the spatial configuration and context in the scene [Hou *et al.*, 2021], leading to a limited performance in motion tasks.

3 Method

3.1 Overview

We represent the point cloud at time t as $\mathcal{PC}_t = \{p_i^t\}_{i=1}^{Nt}$, where p_i^t represents one point and N_t represents the number of points. Considering the efficiency requirement in dealing with large-scale point cloud for autonomous driving, the encoder of our ContrastMotion is based on the popular pillar representation. The point cloud $\mathcal{PC}_t$ are discretized into pillars $\{P_i^t\}_{i=1}^{N_p}$, where N_p is the number of pillars.

As shown in Figure 2, our ContrastMotion is in a contrastive learning mechanism. In training pipeline, We leverage commonly used data augmentation methods to generate $\mathcal{PC}_t$ and $\mathcal{PC}_{t+1}$ from the same point cloud sample, learn the representations of $\mathcal{PC}_t, \mathcal{PC}_{t+1}$ organized in pillars with *PFE* [Qi *et al.*, 2017], and then feed them to our *Encoder* which is similar to backbone in FastFlow3D [Jund *et al.*, 2021], respectively. After that, The Gated Multi-frame Fusion(GMF) block learn valid compensation to enhance feature consistency of corresponding pillars, and its outputs are two flattened feature maps of $N_p \times D$, where N_p is the number of pillars and D denotes feature dimension. We then predict pillar corresponding probabilities based on feature distance between pillars in point cloud pair. The errors between predicted pillar corresponding probabilities and ground truth pillar correspondences are taken as the loss to optimize parameters. Given consecutive frames during inference, ContrastMotion generates pillar correspondences based on pillar corresponding probabilities, and further scene motion.

3.2 Contrastive Learning

Self-Contrast and Cross-Contrast

Different from the methods with point-level correspondences and structural consistency suffering from dynamic changes of scenes, we learn the motion pattern via feature-level consistency. To learn the discriminative features distinguishing each pillar from others and feature consistency between corresponding pillars, we formulate the problem as a one-to-N_p classification task in contrastive learning, and propose *Self-contrast* and *Cross-contrast*

Specifically, Self-contrast and Cross-contrast are used for learning the discriminative features of different pillars in the same frame and feature consistency of corresponding pillars in consecutive frames, respectively. For the former, each pillar forms a positive pair with itself only and forms the negative pairs with all others in the same frame. For the latter, since there are no pillar correspondence labels for consecutive frames in self-supervised training, we leverage commonly used data augmentation methods to generate $\mathcal{PC}_t$ and $\mathcal{PC}_{t+1}$

from the same point cloud training sample, including random rotation, shift, scaling, and jitter. All the data augmentation transformations are applied to the entire point clouds to ensure the completeness of the object surfaces. Then we acquire pillar correspondences based on the known transformation matrices in data augmentation, where each pillar P_i^t forms a positive pair with its corresponding pillar $P_{i'}^{t+1}$ and negative pairs with all other pillars P_j^{t+1} where $j \neq i'$. After that, the feature consistency of corresponding pillars can be learnt by Cross-contrast in a contrastive learning manner. In addition, we apply the random removal to the points in generated $\mathcal{PC}$ to simulate the change in the appearance of the objects caused by ego-motion to avoid over-fitting. Self-contrast and Cross-contrast learn the discriminative and consistent pillar features on which pillar corresponding probabilities depend, which reduces the effect of noise points and point correspondences missing in dynamic scenes.

Soft Discriminative Loss

As mentioned above, The training pipeline of Self- and Cross-contrast is considered a contrastive learning manner. For unsupervised contrastive learning, InfoNCE [Oord *et al.*, 2018] and PointInfoNCE proposed by PointContrast [Xie *et al.*, 2020] are widely used recently. Specifically, they encourage one query to be similar to one positive key and dissimilar to, typically many, negative keys. However, the one-to-one correspondence is not applicable for dynamic scenes, because LiDAR is view-dependent and will capture different views of objects in different frames, causing many points having no correspondences in adjacent frames. Moreover, most points of one rigid object have similar motion and contextual geometric information. Therefore, we generate positive keys with a soft constraint and propose soft discriminative loss (SD-Loss).

$$\mathcal{L} = -\sum_i \sum_{j \in V(P_i^t)} w_{ij} log \frac{exp(\sigma(P_i^t) * \sigma(P_j^{t+1}))}{\sum_k exp(\sigma(P_i^t) * \sigma(P_k^{t+1}))} \quad (1)$$

Here $*$ denotes vector inner product. $V(P_i^t) = \{P_j^{t+1} | d(P_j^{t+1}, P_{i'}^{t+1}) < \varepsilon\}$ represents the positive keys of P_i^t, where $d(\cdot,\cdot)$ is the Euclidean distance function, $P_{i'}^{t+1}$ is determined by the transformation matrix and the corresponding pillar of P_i^t, and ε represents the distance threshold. $\sigma(x)$ is the pillar features extracted by the networks, where $ReLU$ guarantees a positive feature vector, and w_{ij} represents the weight of each positive key, which has a negative correlation with the distance between pillar P_j^{t+1} and $P_{i'}^{t+1}$. For each pillar P_i^t, we generates the weighted positive keys with not only the corresponding pillar $P_{i'}^{t+1}$ defined by transformation matrix but also the neighbouring pillars of $P_{i'}^{t+1}$. Compared to the PointInfoNCE, our SD–Loss retains the feature similarities of local pillars and provides a margin of fault tolerance to avoid the destruction of changing views of point clouds in dynamic scene. Moreover, SD–Loss tackles a widely observed challenge that one-to-one positive key assignment provides less supervision, making it difficult for the network to learn strong and robust features.

3.3 Gated Multi-Frame Fusion Block

Adjacent frames can provide additional information to enhance the features of current sparse point cloud. However, naive fusion of multiple frames makes the moving objects trailing and makes the pillar association which relies heavily on the feature consistency of pillars ambiguous. Therefore, we propose Gated Multi-frame Fusion(GMF) block to learn the valid complement features from adjacent frames, which consists of a gated network. After the *Encoder* extract the features $\mathbf{F}_t$ and $\mathbf{F}_{t+1}$, we align two feature maps according to ego-motion to make the features at the same grid correspond to the same real-world location. Considering a static scene, the fusion feature can be denoted as $\mathbf{F}_t + \mathbf{F}_{t+1}$ due to warping ego-motion between consecutive frames. To avoid wrong fusion caused by moving objects in real scene, we introduce pillar description maps. We feed $\mathbf{F}_t$ and $\mathbf{F}_{t+1}$ to gated network consisting of two convolutional layers, followed activation functions, a linear layer and $Sigmoid$ to get pillar description maps $\mathbf{m}_t, \mathbf{m}_{t+1}$, respectively.

$$\mathbf{m} = \delta(linear(conv(\mathbf{F})) \quad (2)$$

where δ represents $Sigmoid$ to generate pillar description between 0 and 1. Then, the results of multiplying description maps with the original feature maps are added to another original feature maps for feature enhancement.

$$\mathbf{z}_t = \mathbf{F}_t + \mathbf{m}_{t+1} \cdot \mathbf{F}_{t+1} \quad (3)$$

$$\mathbf{z}_{t+1} = \mathbf{F}_{t+1} + \mathbf{m}_t \cdot \mathbf{F}_t \quad (4)$$

where $\cdot$ denotes elemental multiplication, and $\mathbf{z}$ represents the modulated feature map. GMF is straightforward but adaptively distinguishes between static and dynamic pillars and achieves correct feature fusion. As a result, the feature maps $\mathbf{z}$ contain much more geometric features than $\mathbf{F}$, even if the relevant point is missing. GMF facilitates the extraction of discriminative features of the scene due to feature compensation between multiple frames and is beneficial for pillar association due to the enhanced feature consistency between corresponding pillars. Even without additional supervised signals, the pillar description maps actually embodies the dynamic-static information of the pillars and provides more information for the feature extraction, which is introduced in the ablation experiments.

3.4 Pillar Association

After extracting the feature representation of each pillar, we implement pillar association to predict pillar correspondences $\{(P_i^t, P_{i'}^{t+1})\}_{i=1}^{N_p}$ using feature distance. Specifically, the pillar association is in a *query-key* manner where *query* is the feature of P_i^t and *keys* are features of $\{P_j^{t+1}\}_{j=1}^{N_p}$. However, the cost of calculating the feature distance between query and all keys is expensive. Therefore, we adopt *Patching* strategy, where the pseudo-image composed of pillars is partitioned into several $s \times s$ patches, named query patches $\{Q_i^t\}_{i=1}^{\frac{N_p}{s^2}}$. For each query patch Q_i^t, there exists a corresponding key patch K_i^{t+1} whose patch size is $\alpha s \times \alpha s$ where $\alpha > 1$, and the central pillars of the corresponding patches have the same row and column numbers. The pillars in each query patch only calculate the feature distance with the keys in the corresponding key patch $K_{\cdot}^{t+1}$ and then predict pillar correspondence probability. We define pillar corresponding probability between P_i^t and P_j^{t+1} as

$$pb(i,j) = \frac{exp(\mathbf{z}(P_i^t) * \mathbf{z}(P_j^{t+1}))}{\sum_{k \in K_{\cdot}^{t+1}} exp(\mathbf{z}(P_i^t) * \mathbf{z}(P_k^{t+1}))} \tag{5}$$

where $\mathbf{z}(P)$ represents pillar features, $*$ denotes vector inner product. We identify the one with the maximum correspondence probability with P_i^t as its corresponding pillar $P_{i'}^{t+1}$. The adoption of asymmetric patch sizes ensures that queries at the border of query patch still have sufficient keys to implement pillar association. In summary, We adopt *Patching* to achieve greater efficiency and *Asymmetric Patch* to ensure its accuracy. After acquiring the pillar correspondences $\{(P_i^t, P_{i'}^{t+1})\}_{i=1}^{N_p}$, we calculate the pillar flow of P_i^t through $\mathbf{c}(P_{i'}^{t+1}) - \mathbf{c}(P_i^t)$ where $\mathbf{c}(P)$ represents the center of pillars in the real-world coordinate system.

3.5 Task-Specific Post-Processing

Scene Flow

Scene flow is defined as a 3D displacement vector between each surface point rather than each pillar, we directly scatter the flow of pillars to points.

Motion Prediction

Generally, the related works [Wu *et al.*, 2020; Luo *et al.*, 2021] predict future motion with the scene motions from the past by assuming consistent velocities and accelerations. ContrastMotion estimates the motions with past frames and converts them into velocities, then predict future motion.

3.6 Analysis

Compared to point cloud registration which estimates the transformation matrix between point cloud pairs, scene motion estimates fine-grained motion and is more complex. From the perspective of training samples, our ContrastMotion looks like addressing registration rather than scene motion. Because we utilize data augmentation to generate a pair of point clouds as training sample, and the data augmentation transformation is applied to all points in the point cloud in order to ensure the integrity of the object surface. *Does the degradation solution occur during training, where the network infers the transformation matrix and obtains the motion of each pillar directly, making the model much less capable of inferring scene motion?* The reason why ContrastMotion is working relies on the fact that we use a pillar association process instead of the conventional prediction head. In pillar association, we use a query-key manner and for each key we have no additional positional encoding. The distinction between the different pillars(keys) relies on their geometric contextual information rather than on their real position. In pillar association process, the network is agnostic to pillar motion, which avoids the degenerate solution. Thus, even with the training samples generated from the transformation matrix, our model still achieves good performance in real scenes.

Method	EPE3D↓	Acc3DS↑	Acc3DR↑	Outliers3D↓
PointPWC	0.3648	0.0726	0.2974	0.8579
PoseFlow	0.3256	0.1104	0.2058	0.9778
SLIM*	0.0688	**0.7695**	0.9342	0.2488
FlowStep3D	0.1741	0.5821	0.7397	0.3596
RCP†	0.0974	0.6058	0.8021	0.3263
ContrastMotion	**0.0656**	0.7382	**0.9417**	**0.1847**

Table 1: **Evaluation results on FT3D → KITTI Scene Flow.** These self-supervised methods are trained only on FT3D and evaluated on KITTI Scene Flow dataset for all points. PointPWC, FlowStep3D and RCP are the point-based methods, while SLIM and our ContrastMotion are pillar-based. *: models trained on KITTI-RL. †: reproducing experiments. ‡: 8192 points downsampling.

Method	Moving		Stat.
	EPE3D(m)↓	Acc3DR↑	EPE3D(m)↓
Zero	0.6381	0.1632	0.5248
PointPWC	0.3539	0.2543	0.1974
PoseFlow	0.7399	0.0000	0.0570
SLIM	0.1050	**0.7365**	0.0925
ContrastMotion(ours)	**0.0925**	0.7082	**0.0545**

Table 2: **Self-supervised training & evaluation on nuScenes scene flow.** *Zero* predicts that the environment is static.

4 Experiments

4.1 Dateset

KITTI Scene Flow [Menze *et al.*, 2015; Menze *et al.*, 2018] consists of 200 training scenes and 200 test scenes. Point clouds and ground truth scene flows are obtained by lifting the annotated disparity maps and optical flows to 3D. Following previous works [Gu *et al.*, 2019; Wu *et al.*, 2019], FlyingThings3D (FT3D) dataset, a synthetic dataset with 19640 samples, is used for training and the training scenes in KITTI are used for inference.
nuScenes [Caesar *et al.*, 2019] is a large-scale public dataset for autonomous driving collected from 1000 real scenes, whose initial purpose is object detection and segmentation. The point clouds are captured with a frequency of 20Hz, and [Caesar *et al.*, 2019] annotates the entire dataset with accurate 3D bounding boxes for 23 object classes at 2Hz. Following previous works [Wu *et al.*, 2020; Jund *et al.*, 2021; Li *et al.*, 2022b], we assume that each object is in rigid motion and derive the ground truth motion from the origin detection and tracking annotations.

4.2 Experimental Setup

Implementation Details

We implement all the experiments on PyTorch [Paszke *et al.*, 2019]. For FT3D and nuScenes datasets, we train the ContrastMotion for 300 and 100 epochs respectively, and set the initial learning rate to 0.001, weight decay to 0.001. The batch size is set to the maximum available on a 32G GPU, and Adam is used as the optimizer. We crop the point clouds following baseline models [Li *et al.*, 2022b; Luo *et al.*, 2021], and the pillar size is set to $[0.25, 0.25]$. For training $\mathcal{PC}$ generation, the upper limits of shift, rotation angle, scaling and jitter are $3m, 0.17, 1.05$ and $0.1m$. We scale the samples of FT3D to fit the cropping range during training. In KITTI, we predict the motion of points beyond crop-

Method	Sup.	Static		Speed ≤ 5m/s		Speed ≥ 5m/s	
		Mean	Median	Mean	Median	Mean	Median
Zero	-	**0.0000**	**0.0000**	0.6111	0.0971	8.6517	8.1412
FlowNet3D [Liu *et al.*, 2019]	Pre.	2.0514	**0.0000**	2.2058	0.3172	9.1923	8.4923
HPLFlowNet [Gu *et al.*, 2019]	Pre.	2.2165	1.4925	1.5477	1.1269	5.9841	4.8553
PillarMotion [Luo *et al.*, 2021]	Self	0.1620	0.0010	0.6972	0.1758	3.5504	2.0844
ContrastMotion (ours)	Self	0.0829	**0.0000**	**0.4522**	**0.0959**	**3.5266**	**1.3233**

Table 3: **Comparison with the state-of-the-art results on the nuScenes motion prediction**. We report the mean and median errors on the three speed groups. *Self* indicates self-supervised methods trained with nuScenes, and *Pre.* indicates the methods that are not trained with the annotations of nuScenes but are supervised pre-trained on two scene flow datasets. *Zero* predicts that the environment is static.

Method	SD–Loss	PointInfoNCE Loss	Ground Mask	GMF	Static	Speed ≤ 5m/s	Speed > 5m/s
(a)	✓				0.1377	0.7791	3.9497
(b)	✓			✓	**0.0682**	0.4579	3.9872
(c)	✓		✓	✓	0.0829	**0.4522**	**3.5266**
(d)		✓	✓	✓	0.0837	0.6546	3.9601

Table 4: **Ablation study for different modules of ContrastMotion.** We evaluate each model on nuScenes motion prediction and report mean errors. Ground Mask: Removing estimated ground points before feeding to network. Gated Block: *Gated Multi-frame Fusion Block*.

ping range as the average motion of that scene. The distance threshold ε in $V(P_i^t)$ is set to 1.1 times pillar size, and the weights of positive keys w_{ij} are defined as

$$w_{ij} = \begin{cases} 0.6 & if j = i' \\ 0.1 & else \end{cases} \tag{6}$$

where pillar i' is the corresponding pillar of pillar i. The dimension of output feature map $D = 32$. In pillar association, the patch size $s = 32$ and $\alpha = 2$. The point clouds from KITTI are more intensive than real scenes. Therefore we adopt a post-processing to scatter the flows of pillars more accurately to points. We randomly select a reference point from each pillar and take the point with the minimum Chamfer [Wu *et al.*, 2019] loss in corresponding pillar as its corresponding point. The post-processing mentioned above is not applied to the nuScenes dataset or to the motion prediction task. In motion prediction task, we predict the motion for the next 1.0s by two past motions for fair comparisons with the methods. Refer to supplementary materials for more details of the Chamfer Loss and ground points pre-processing.

Evaluation Metrics

We report the evaluation criteria used by Hplflownet [Gu *et al.*, 2019] in scene flow: **EPE3D(m)** average end-point-error over each point; **Acc3DS** point ratio where EPE3D < 0.05 or relative error $< 5\%$; **Acc3DR** point ratio where EPE3D < 0.1 or relative error $< 10\%$; **Outliers3D** point ratio where EPE3D > 0.3 or relative error $> 10\%$;

We report the average **L2** distances and **median** errors on non-empty pillars which are classified as static, slow and fast groups in motion prediction, which used in PillarMotion.

4.3 Comparison with SOTA Methods

Scene Flow

We implement the experiments on FT3D → KITTI-SF, and mainly compare our ContrastMotion with the point-based methods, including PointPWC [Wu *et al.*, 2019], FlowStep3D [Kittenplon *et al.*, 2021], PoseFlow [Tishchenko *et al.*, 2020] and RCP [Gu *et al.*, 2022], and pillar-based method SLIM [Li *et al.*, 2022b]. Our ContrastMotion and SLIM can process a full scene from KITTI. While the baselines can run inference on a full scene, during training it is infeasible to run them with roughly $30K$ points due to the cost of memory and the training time [Li *et al.*, 2022b].

As shown in Table 1, the point-based methods, e.g., PointPWC and RCP, achieve limited results compared with the pillar-based approaches. These methods perform significant sampling in the training pipelines due to the heavy cost of memory and computation for large-scale point clouds, while pillar-based methods can be trained with the raw data. The increased points providing more fine-grained information enhance the stable inference ability on the dense point clouds. In Table 1, we also report the performance of a pillar-based method, SLIM. Although trained with KITTI-RL [Geiger *et al.*, 2013], the performance of SLIM is not satisfying, which is due to the effect of point-level correspondence problems in the dynamic scenes. Our ContrastMotion leverages feature-level consistency to predict pillar flows and achieves SOTA performances in EPE3D and Outliers3D metrics. The results show the effectiveness of pillar-based methods and our ContrastMotion.

In order to get closer to real-world evaluation, we also implement self-supervised experiments on the dataset without point correspondences, nuScenes, and compare our ContrastMotion with SLIM [Li *et al.*, 2022b]. As shown in Table 2, Our ContrastMotion achieves the SOTA performance in moving EPE3D. Where around 95% points in nuScenes are static. Our proposed GMF enhances the feature consistency between the corresponding static pillars and helps our ContrastMotion achieve the best performance compared to the baseline methods in static EPE3D.

Motion Prediction

As same as PillarMotion [Luo *et al.*, 2021], we use the same 500 training scenes, 100 validation scenes and 250 testing scenes with PillarMotion [Luo *et al.*, 2021] .But unlike PillarMotion which uses LiDAR, camera and pose data,

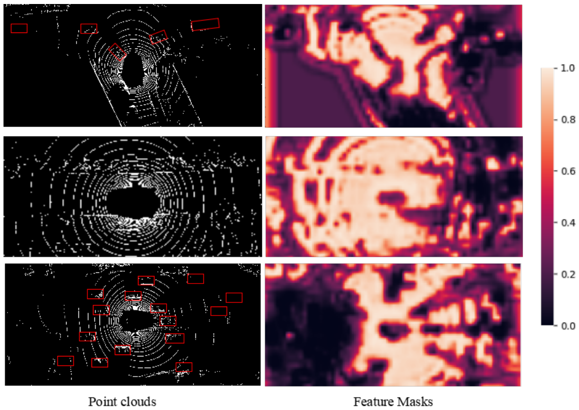

Figure 3: **The pillar description maps m generated by Gated Network in *GMF*.** The red boxes indicate moving objects. Overlaying the description maps on the point clouds, we can see that there are m close to *zero* in positions where movable objects and empty pillars appear, but there are m close to *one* in positions where static objects such as ground, road edges, etc., appear.

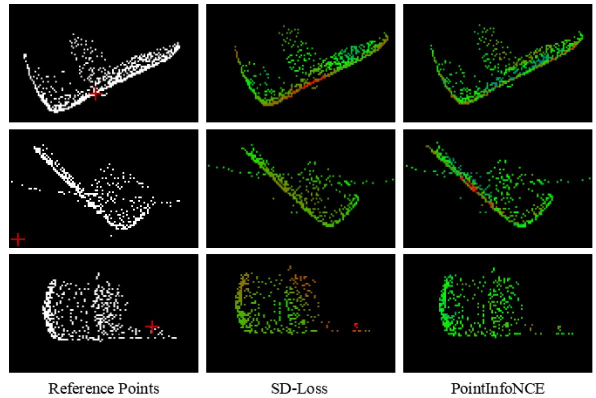

Figure 4: **The corresponding probabilities between points in $T+1$ and the reference points in T inferred from model (c) and model (d), which respectively adopt SD–Loss and PointInfoNCE as loss function.** Top row: a false negative estimation by model(d). Middle row: a false positive estimation by model (d). Bottom row: the case that the corresponding points are not scanned. For all cases, model (c) which adopt proposed SD–Loss provides reasonable results.

only LiDAR and pose data are used in our work. The related methods we compare include self-supervised PillarMotion [Luo *et al.*, 2021], and pre-trained supervised estimators on other datasets, e.g., FlowNet3D [Liu *et al.*, 2019] and HPLFlowNet [Gu *et al.*, 2019].

Table 3 shows the results of different approaches. The performances of FlowNet3D and HPLFlowNet which are trained in FT3D and tested on nuScenes without fine-tuning indicate the existence of a large domain gap. In addition, 3D labeling on real large-scale scenes requires complex processes and labor-intensive human efforts, and the discrepancy is obvious for cross-scenes and cross-LiDAR captured data. PillarMotion achieves a limited performance with consistency and cross-device regularization, which is also due to the effect of point-level correspondence missing in the dynamic scenes. In addition, the optical flow used as a signal to guide motion learning introduces errors, including optical flow prediction and coordinate transformation errors. Our ContrastMotion achieves SOTA performances among the compared methods in the low and high speed groups, and the prediction errors decrease by 35% and 0.02 compared to PillarMotion. In terms of median errors, our ContrastMotion achieves SOTA performances in all the groups. The results show the advantages of self-supervision compared with supervision and the potential of ContrastMotion in motion prediction tasks.

4.4 Ablation Studies

Gated Multi-Frame Fusion Block

We perform experiments to explore the effects of proposed GMF, and the results are shown in Table 4. The baseline model (a) trained only with our SD–Loss does not perform well for static and low speed group. However, model (b) incorporating of proposed GMF performs significantly better in the static and low speed groups, and achieves comparable performances with (a) in high speed group. In addition, we visualize the pillar description maps in GMF, which represents the probability of inserting the features of each pillar to another point cloud feature map. As shown in Figure 3, there are significantly high mask values in positions without motions, such as the ground, but low values where movable objects and empty pillars appear. What is mentioned above shows GMF learns valid movable-static information and compensation, and further enhance the feature consistency between corresponding pillars which facilitates the pillar association to acquire accurate pillar correspondence.

Soft Discriminative Loss

We also conduct experiments to compare proposed SD–Loss with PointInfoNCE proposed by PointContrast [Xie *et al.*, 2020]. Comparing models (c) and (d) in Table 4, we find that model (c) achieves better performances in all the groups. As shown in Figure 4, we believe that SD–Loss considering the feature similarities of the local points can make an approximate estimation in the case that the corresponding points are not scanned and the corresponding probabilities inferred by model (c) show a Gaussian-like distribution due to the soft pillar correspondence adopted in SD–Loss, which provide a margin of fault tolerance and avoid the destruction of noises and changing views of dynamic scenes.

For ablation experiments of ground points, qualitative and runtime analysis, please refer to the supplementary material.

5 Conclusion

In this paper, we propose ContrastMotion, a self-supervised framework in contrastive learning manner for scene motion. We propose Soft Discriminative Loss to bootstrap discriminative feature extraction and alleviate the correspondence missing situations. We also propose Gated Multi-frame Fusion block learning valid compensation . Pillar association predicts pillar correspondence probabilities and further predicts scene motion. We will focus on improving the inference speed of pillar association in the future.

Acknowledgements

This work was supported by NSFC (No.62206173), Shanghai Frontiers Science Center of Human-centered Artificial Intelligence (ShangHAI) and Natural Science Foundation of China (No. 62172372).

References

[Bai *et al.*, 2022] Yutong Bai, Xinlei Chen, Alexander Kirillov, Alan Yuille, and Alexander C Berg. Point-level region contrast for object detection pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16061–16070, 2022.

[Baur *et al.*, 2021] Stefan Andreas Baur, David Josef Emmerichs, Frank Moosmann, Peter Pinggera, Björn Ommer, and Andreas Geiger. Slim: Self-supervised lidar scene flow and motion segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13126–13136, 2021.

[Caesar *et al.*, 2019] 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. *arXiv preprint arXiv:1903.11027*, 2019.

[Cong *et al.*, 2022] Peishan Cong, Xinge Zhu, Feng Qiao, Yiming Ren, Xidong Peng, Yuenan Hou, Lan Xu, Ruigang Yang, Dinesh Manocha, and Yuexin Ma. Stcrowd: A multimodal dataset for pedestrian perception in crowded scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19608–19617, June 2022.

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

[Gu *et al.*, 2019] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3254–3263, 2019.

[Gu *et al.*, 2022] Xiaodong Gu, Chengzhou Tang, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Rcp: Recurrent closest point for point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8216–8226, 2022.

[Hou *et al.*, 2021] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021.

[Hur and Roth, 2021] Junhwa Hur and Stefan Roth. Self-supervised multi-frame monocular scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2684–2694, 2021.

[Jund *et al.*, 2021] Philipp Jund, Chris Sweeney, Nichola Abdo, Zhifeng Chen, and Jonathon Shlens. Scalable scene flow from point clouds in the real world. *IEEE Robotics and Automation Letters*, 7(2):1589–1596, 2021.

[Kittenplon *et al.*, 2021] Yair Kittenplon, Yonina C Eldar, and Dan Raviv. Flowstep3d: Model unrolling for self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4114–4123, 2021.

[Lang *et al.*, 2019] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[Lee *et al.*, 2020] Kuan-Hui Lee, Matthew Kliemann, Adrien Gaidon, Jie Li, Chao Fang, Sudeep Pillai, and Wolfram Burgard. Pillarflow: End-to-end birds-eye-view flow estimation for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2007–2013. IEEE, 2020.

[Li *et al.*, 2021a] Ruibo Li, Guosheng Lin, and Lihua Xie. Self-point-flow: Self-supervised scene flow estimation from point clouds with optimal transport and random walk. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15577–15586, 2021.

[Li *et al.*, 2021b] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Neural scene flow prior. *Advances in Neural Information Processing Systems*, 34:7838–7851, 2021.

[Li *et al.*, 2022a] Bing Li, Cheng Zheng, Silvio Giancola, and Bernard Ghanem. Sctn: Sparse convolution-transformer network for scene flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1254–1262, 2022.

[Li *et al.*, 2022b] Ruibo Li, Chi Zhang, Guosheng Lin, Zhe Wang, and Chunhua Shen. Rigidflow: Self-supervised scene flow learning on point clouds by local rigidity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16959–16968, 2022.

[Li *et al.*, 2022c] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022.

[Liu *et al.*, 2019] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 529–537, 2019.

[Luo *et al.*, 2018] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on*

Computer Vision and Pattern Recognition, pages 3569–3577, 2018.

[Luo *et al.*, 2021] Chenxu Luo, Xiaodong Yang, and Alan Yuille. Self-supervised pillar motion learning for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3183–3192, 2021.

[Ma *et al.*, 2019] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6120–6127, 2019.

[Ma *et al.*, 2022] Yuexin Ma, Tai Wang, Xuyang Bai, Huitong Yang, Yuenan Hou, Yaming Wang, Yu Qiao, Ruigang Yang, Dinesh Manocha, and Xinge Zhu. Vision-centric bev perception: A survey. *arXiv preprint arXiv:2208.02797*, 2022.

[Mahjourian *et al.*, 2018] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5667–5675, 2018.

[Menze *et al.*, 2015] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 2:427, 2015.

[Menze *et al.*, 2018] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018.

[Mittal *et al.*, 2020] Himangi Mittal, Brian Okorn, and David Held. Just go with the flow: Self-supervised scene flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11177–11185, 2020.

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

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[Qi *et al.*, 2017] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[Radford *et al.*, 2021] 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.

[Sadeghian *et al.*, 2019] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1349–1358, 2019.

[Tishchenko *et al.*, 2020] Ivan Tishchenko, Sandro Lombardi, Martin R Oswald, and Marc Pollefeys. Self-supervised learning of non-rigid residual flow and ego-motion. In *2020 international conference on 3D vision (3DV)*, pages 150–159. IEEE, 2020.

[Wang *et al.*, 2022] Yunlong Wang, Hongyu Pan, Jun Zhu, Yu-Huan Wu, Xin Zhan, Kun Jiang, and Diange Yang. Besti: Spatial-temporal integrated network for class-agnostic motion prediction with bidirectional enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17093–17102, 2022.

[Wei *et al.*, 2021] Yi Wei, Ziyi Wang, Yongming Rao, Jiwen Lu, and Jie Zhou. Pv-raft: point-voxel correlation fields for scene flow estimation of point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6954–6963, 2021.

[Wu *et al.*, 2019] Wenxuan Wu, Zhiyuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: A coarse-to-fine network for supervised and self-supervised scene flow estimation on 3d point clouds. *arXiv preprint arXiv:1911.12408*, 2019.

[Wu *et al.*, 2020] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11385–11395, 2020.

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

[Yin *et al.*, 2021] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021.

[Zhu *et al.*, 2020] Xinge Zhu, Yuexin Ma, Tai Wang, Yan Xu, Jianping Shi, and Dahua Lin. Ssn: Shape signature networks for multi-class object detection from point clouds. *ECCV*, 2020.

[Zhu *et al.*, 2021] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Wei Li, Yuexin Ma, Hongsheng Li, Ruigang Yang, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *TPAMI*, 2021.