Implicit Surface Contrastive Clustering for LiDAR Point Clouds

Zaiwei Zhang[*]
Nuro Inc
zaizhang@nuro.ai

Min Bai
AWS AI
baimin@amazon.com

Li Erran Li
AWS AI
lilimam@amazon.com

Abstract

Self-supervised pretraining on large unlabeled datasets has shown tremendous success in improving the task performance of many 2D and small scale 3D computer vision tasks. However, the popular pretraining approaches have not been impactfully applied to outdoor LiDAR point cloud perception due to the latter's scene complexity and wide range. We propose a new self-supervised pretraining method ***ISCC*** *with two novel pretext tasks for LiDAR point clouds. The first task uncovers semantic information by sorting local groups of points in the scene into a globally consistent set of semantically meaningful clusters using contrastive learning, complemented by a second task which reasons about precise surfaces of various parts of the scene through implicit surface reconstruction to learn geometric structures. We demonstrate their effectiveness through transfer learning on 3D object detection and semantic segmentation in real world LiDAR scenes. We further design an unsupervised semantic grouping task to show that our approach learns highly semantically meaningful features.*

1. Introduction

Robust and reliable 3D LiDAR perception is critical for autonomous driving systems. Unlike images, LiDAR provides unambiguous measurements of the vehicle's 3D environment. A plethora of perception models have arisen in recent years to enable a variety of scene understanding tasks using LiDAR input, such as object detection and semantic segmentation. However, training these models generally relies on a large quantity of human annotated data, which is tedious and expensive to produce.

Recently, self-supervised learning has attracted significant research attention [5, 7, 16–18], as it has the potential to increase performance on downstream tasks with limited quantities of annotated data in the image domain. However, self-supervised learning has shown less impact for outdoor 3D point clouds. A core difficulty stems from the relative difficulty in designing appropriate pretext tasks used for self-supervision. In the image domain, the ImageNet dataset [35] provides millions of canonical images of everyday objects, allowing for straightforward manipulations to generate pretext tasks that lead to strong object-centric or semantic group-centric feature learning. While large-scale unlabeled outdoor LiDAR datasets are relatively easy to collect, the data samples exhibit a high level of scene complexity, sparsity of measurements, and heavy dependency on the observer positioning and sensor type. These factors pose great challenges for creating useful pretext tasks.

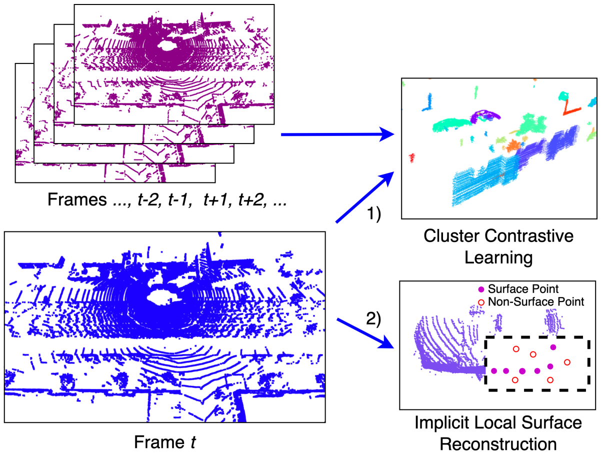
Frames ..., t-2, t-1, t+1, t+2, ...
1)
Cluster Contrastive Learning
Surface Point
Non-Surface Point
2)
Frame t
Implicit Local Surface Reconstruction

Figure 1. With frame t as input, we 1) apply contrastive cluster learning to reason about unsupervised semantic clustering and also enforce feature consistency across different views, and 2) conduct implicit surface reconstruction in randomly cropped out regions.

Recent works have proposed 3D-specific self-supervised learning, starting with scene-level contrastive learning [44, 49], followed by the work of [30] and [47], which use finer, region-level granularity to better encode individual components of large scale LiDAR point clouds. However, these techniques do not explicitly make use of regularities in 3D shapes and surfaces.

In our work, we propose **ISCC** (Implicit Surface Con-

[*]Work done at AWS AI.

trastive Clustering) which consists of two new pretext tasks to automatically learn semantically meaningful feature extraction without annotations for LiDAR point clouds. The first task focuses on learning semantic information by sorting local groups of points in the scene into a globally consistent set of semantically meaningful clusters using the contrastive learning setup [5]. This is augmented with a second task which reasons about precise surfaces of various parts of the scene through implicit surface reconstruction to learn geometric regularities. A high level overview is found in Figure 1. Furthermore, we showcase a novel procedure to generate training signals for implicit surface reasoning in the absence of dense 3D surface meshes which are difficult to obtain for large scale LiDAR point clouds.

Using the large real world KITTI [13] and Waymo [37] datasets, we show that our approach is superior to related state-of-the-art self-supervised learning techniques in downstream finetuning performance in semantic segmentation and object detection. For example, we see a 72% gain in segmentation performance on SemanticKITTI versus the state-of-the-art [48] when fine-tuned with 1% of the annotations, and exceeds the accuracy achieved by using twice the annotations with random initialized weights. As well, we analyze the semantic consistency of the learned features through a new *unsupervised semantic grouping* task, and show that our learned features are able to form semantic groups even in the absence of supervised fine-tuning.

2. Related work

We aim to push the boundaries of self-supervised feature learning for large-scale 3D LiDAR point clouds in the autonomous driving setting. We draw inspiration from numerous existing methods for related tasks.

Self-supervised feature learning Automatically learning to extract features using deep neural networks has attracted significant attention, pioneered by researchers in the natural language processing domain [10, 12, 27], followed by notable achievements using camera images [5, 7, 16–18]. In this setting, the goal is to learn general purpose feature extractors which can be finetuned to tackle a variety of tasks, such as understanding and generation for language, or classification, segmentation, and detection for images. The core component of these approaches is the design of pretext tasks (and its accompanying solution) which can be automatically generated from unlabeled data. The features learned by the network while attempting to solve these tasks have been shown to transfer well to the desired downstream tasks.

While these works have demonstrated significant capabilities in the language and images, the drastically different modality of large-scale 3D LiDAR point clouds have given rise to a related but distinct line of work [8, 19, 44, 49]. These works use differently augmented views of the same scene or temporally different frames of the same sequence to construct pretext tasks and learn features using contrastive objectives. More recently, [47] samples a limited number of points from the scene and considers their spherical neighborhoods as region proposals, and then apply local and global contrastive loss on those region features. Unlike these approaches, we introduce the 3D geometry-inspired pretext task of local surface reasoning to improve our self-supervised feature extraction.

Self-supervised task-specific learning for 3D point clouds There exist cases where a surrogate loss function can be designed to directly train a neural network to solve the end-goal task. For example, [3, 42] use self-supervision for LiDAR scene flow estimation, while [11, 23] extend self-supervised LiDAR motion reasoning to learn suitable features for detecting other traffic participants. Unlike these approaches, we note that our technique is driven by semantic reasoning and is thus capable of improving perception performance over various stationary semantic regions.

Learning surface representations from 3D point clouds Previously, [2, 28, 40, 50] explored numerous models and output parameterization for converting a relatively sparse input point cloud into a denser representation. In our technique, we further leverage 3D geometric learning as a component of its pretext task to learn useful local point cloud feature extraction, instead of as an end goal in itself. Our approach leverages the implicit surface [24, 34] concept where the presence of the surface is represented by the characterization of distinct points near or on the surface. However, unlike the existing work which generally require dense ground truth, our objective is simpler and uses automatically generated targets for its prediction.

Unsupervised semantic clustering Lastly, our work draws inspiration from the existing work on using unlabeled data to automatically discover semantically meaningful groupings within a data sample. Previously, this task has been studied extensively in the image domain by works including [20, 21, 39, 41]. In the 3D domain, a recent work [30] targets self-supervised semantic reasoning over point clouds by applying heuristics-based methods to isolate individual objects (segments) in 3D LiDAR sweeps. They extract segment-specific features and apply contrastive objective over each segment to learn to distinguish from each other. Our method proposes an additional objective over the segments to learn to sort the segments into a predetermined number of semantic clusters.

Algorithm 1 Training Framework for **ISCC**

```
Input  Network encoder $F_\theta$ and momentum encoder $F_m$;
    Point cloud frames $\mathcal{D} = \{X\}_{i=1}^{N}$; Pre-computed point
    group labels $\{V\}_{i=1}^{N}$; Global feature queues $F_g \in \mathrm{R}^{C\times d}$;
Output  Pre-trained weight for the network encoder $F_\theta$
  for $x_i$ in $X$ do
      - Sample a different frame $x_j$ from $D$
      - Generate two augmented versions $\hat{x}_i$ and $\hat{x}_j$
      - For shared point groups $V$ between $\hat{x}_i$ and $\hat{x}_j$, apply
    random cropping and create query points $Q^1$ for $\hat{x}_i$
      - Feature embeddings: $h^1 = F_\theta(\hat{x}_i), h^2 = F_m(\hat{x}_j)$
      - Point group features: $f^1 = \text{avg\_pool}(h^1, V), f^2 =$
    $\text{avg\_pool}(h^2, V)$
      - Global contrastive clustering loss: $L_c(f^1, f^2, F_g)$
      - Local occupancy prediction loss: $L_o(h^1, Q^1)$
      - Update $F_g$ with $f^2$ and update $F_\theta$
  end for
```

3. Method

We introduce a new self-supervised pretraining framework **ISCC** for outdoor LiDAR point clouds. The main approach is outlined in Algorithm 1, and visualized in Figure 1. We propose two pre-text tasks for self-supervised learning: contrasting cluster assignments and occupancy prediction. We first describe the steps for local point group generation and the formulation of the contrastive clustering task in Section 3.1. Then we show how to generate supervisions for the occupancy prediction task and how to perform the prediction in Section 3.2. Finally, we list the implementation details in Section 3.3.

3.1. Contrasting Semantic Cluster Assignments

Outdoor LiDAR point clouds usually contain large and complex scenes with numerous objects within each sample. Unlike most previous self-supervised learning approaches that reason at the sample level, a single global vector encoding is incapable of describing the attributes and locations of the various objects in the scene. Thus, we perform representation learning on local point group features. This pretext task is based on two ideas: 1) point group features should be sorted into consistent semantic clusters across the dataset for semantic reasoning, and 2) the features of different point groups should be descriptive and distinct to capture their unique attributes and discourage feature collapse.

3.1.1 Point Group Generation

We start by grouping points in the scene to produce a tractable number of entities. In a typical scene, foreground objects classes (such as cars, pedestrians and cyclists) are spatially well-separated, while the ground plane is relatively easy to identify. Due to this data characteristic, we remove the ground plane points and apply unsupervised geometric clustering to group the remaining points, similar to [14, 30]. Additionally, with sequential data streams, we make use of temporal pairs of LiDAR frames and form consistent clusters across multiple frames for multi-frame reasoning.

For each frame X_i in a stream of LiDAR point clouds, we gather a set of frames $\{x_j\}$ from its neighbouring frames $\{x_{i-ks}, ..., x_{i-s}, x_{i+s}, ..., x_{i+ks}\}$ and align the point clouds with the provided relative ego-vehicle poses, yielding $2k+1$ sample frames spaced s frames apart. To balance the desire for increased view variability with computational efficiency, we set $k = 3$ and $s = 5$. Next, we apply the plane estimation algorithm [22] to remove the ground plane points and use HDBSCAN [26] to form different point groups. Note that each groups contains points from x_i and $\{x_j\}$, providing consistent groups IDs across the various frames. We denote the generated group IDs as $\{V\}_{i=1}^{N}$. This process is visualized in a) and b) of Figure 2.

Point group augmentation For a sampled point cloud pair (x_i, x_j), we select up to $K = 100$ shared point groups between the two frames. We then apply local cropping [49] on each selected groups in x_i. We use 20% as cropping ratio for our experiments. We keep the original geometry in x_j so that it can propagate dense feature information through the global cluster reasoning. Then, we apply global random rotation, translation and scaling as additional data augmentation to produce $\hat{x}_i$ and $\hat{x}_j$ for use by the pretraining.

Point group feature extraction After the data preparation, $\hat{x}_i$ and $\hat{x}_j$ are mapped to feature embeddings by applying a 3D U-Net. Adapting the method of He *et al.* [18], we use an encoder F_θ to extract features for $\hat{x}_i$ and a momentum encoder F_m to extract features for the augmented view $\hat{x}_j$. The weights of the momentum encoder are updated with an exponential moving average (EMA) from the encoder's weights. The update rule is $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$ with a fixed $\lambda = 0.999$. We then group the per-voxel features by their corresponding point groups and apply average-pooling to the last layer's features to extract the point group features f^1 and also f^2 from the momentum encoder. Finally, we apply a two layer MLP as in [7] and L2 normalization to compute the cluster embeddings .

3.1.2 Contrastive Clustering Loss

Contrasting cluster assignments has been studied in 2D computer vision [6]. However, we focus on conducting cluster assignments on local point groups and reasoning across different frames globally. Traditional methods such as k-means and DBSCAN have proven to be effective but time-consuming to apply on very large datasets. Recently, [1, 5] have shown that pseudo-label assignment

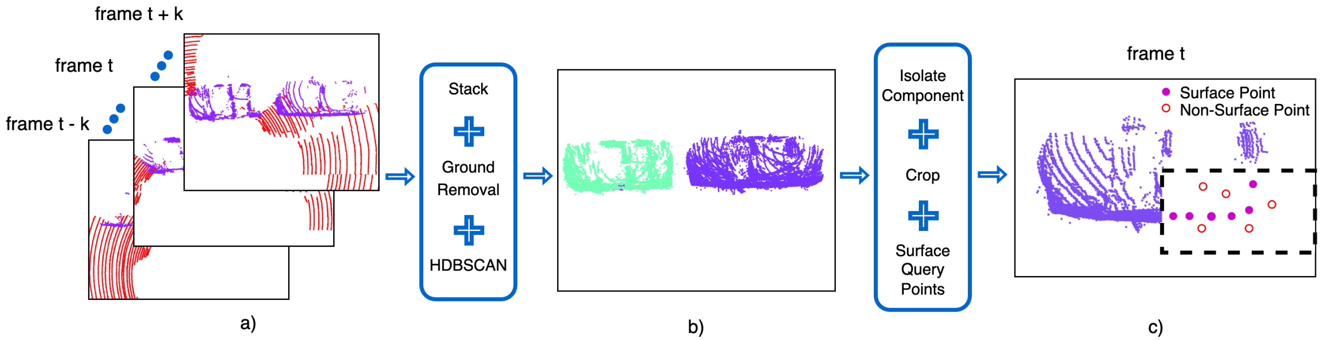

Figure 2. Data preparation process: we (a) stack LiDAR frames in global coordinates for multiple frames, remove ground points [22], and apply geometric clustering to generate linked point group IDs (b) across multiple frames to enable the contrastive semantic clustering task. For each component, we further apply crop augmentation in one frame, and use the cropped out points for the implicit surface reasoning task (c).

can be formulated as an optimal transport problem, where the Sinkhorn-Knopp algorithm [9] can be used for efficient clustering. We follow a similar approach for our unsupervised feature clustering.

Global feature queue We first build a global feature queue $F_g \in \mathrm{R}^{C\times d}$ to store local point group features over multiple data instances. We set queue size C to 1M and feature dimension d to 100 for our experiments. The global feature queue is created with cluster features f_2 from the momentum encoder.

Global feature clustering For building global clusters, we follow the same method proposed by Asano [1]. During training, given query point group features f^2 and global feature queue F_g, we first concatenate them and then apply a classification head $h_g : \mathrm{R}^d \rightarrow \mathrm{R}^C$ to convert the feature vectors to class (cluster ID) scores. C is number of classes and we set it to 100 for our experiments. We then map them to class probabilities via a softmax operator: $P_g = \mathrm{softmax}(\mathrm{concat}(f^2, F_g)\cdot h_g)$. We find the class label assignments Y for the query features f^2 by performing iterative Sinkhorn-Knopp updates [9].

Global contrastive clustering loss After extracting the class label assignments Y, we directly map it to f^1 using the paired correlation between f^1 and f^2. Using the query features to find the cluster assignments introduces additional data augmentation since it provides a different view of the local points from a different timestamp.

Given a pair of point group features f^1, f^2 and their class label assignments Y, we apply the classification head h_g to f^1 and softmax to compute the predicted class probabilities: $P^c = \mathrm{softmax}(f^1 \cdot h_g)$. We use multi-class cross-entropy loss to learn the global constrastive clustering:

$$L_g(P^c, Y) = -\sum_{m=1}^{M} y_m \log(p^c_m) \tag{1}$$

where M is the number of point groups. We set the upper bound of M to 1k for our experiments. To encourage the network learning to form more distinct semantic clusters, we also add a local contrastive loss, which is a modified InfoNCE loss to conduct contrastive learning between different point groups of the same point cloud:

$$L_l = -\sum_{i\in K} \log \frac{\exp(f_i^1 \cdot f_i^2/\tau)}{\sum_{j\in K}\exp(f_i^1 \cdot f_j^2/\tau)} \tag{2}$$

τ is a temperature parameter [43]. Note that the positive examples are two point group features from different frames, while negative examples are from other point groups under different frames. This local contrastive loss enables the network to learn view consistent features from different LiDAR frames. Combining the two losses, we have the global contrastive clustering loss: $L_c = L_g + L_l$.

3.2. Occupancy Prediction

While implicit representation has been widely used for scene and object level reconstruction, we study its potential as a pretext task for pretraining. Although recent works [31, 45] have shown some success of using this pretext task, it has not been studied with large scale LiDAR point clouds. There exists several challenges for occupancy prediction in outdoor LiDAR point clouds. Due to the point cloud's scale and sparsity, it is computationally infeasible to form a dense 3D feature volume for occupancy prediction and during training as uniform query point sampling across the scene is extremely inefficient. Most of the query points will be lying in empty voxels or near background points, such as road. Moreover, ground truth surface information for learning surface reconstruction is difficult to obtain as manual generation of watertight surface meshes is cost prohibitive, while moving objects pose issues for automatic surface reconstruction.

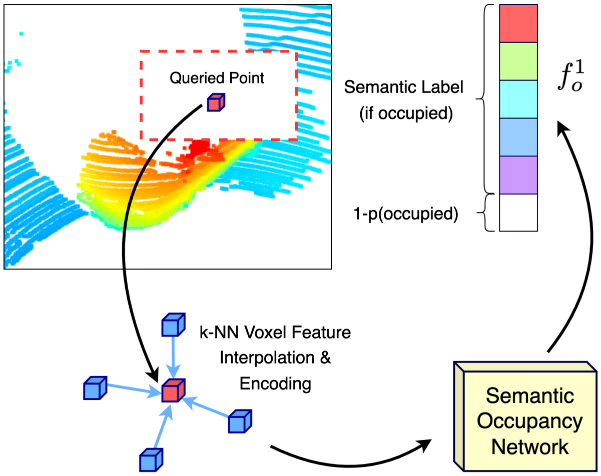

Figure 3. Implicit surface pretext task: we remove a part of a point cloud, and generate features of queried points using neighborhood interpolation to predict the semantic label of the point if it is on the surface, or not a surface point otherwise.

Occupancy Label Generation We generate the query points $Q^1 = \{q_k^1\}_{k=1}^K$ near K sampled local point groups. We apply local cropping to the point groups in x_i, while sampling 100 points from the cropped regions as q_k^1. We add random translation noise with a maximum norm of $1m$ to 50 points with the assumption that the result is highly unlikely to remain valid surface points. As such, we assign them with zero occupancy probability. The remaining 50 points are assigned with full occupancy probability. We visualize this procedure in part c) of Figure 2.

Sparse Feature Aggregation Due to extensive computational complexities, we directly aggregate features for query points from a sparse volume representation h^1. As illustrated from Figure 3, we first apply k-NN to find k neighbouring voxel centers for each query point. Then we interpolate the selected voxel features weighted by the inverse distance norms between the query point and the selected voxel centers. We denote the aggregated features as $\phi(h^1, Q^1)$. In order to encode the position information, we calculate the offsets between the query point and neighbor voxel centers and project them to feature space with one-layer MLP. We also interpolate the position embeddings with the inverse distance norms, and denote them as f_q^1.

Semantic Aware Reconstruction Following the practice from [29, 32], we apply an occupancy network to extract the final embedding: $f_o^1 = \psi(f_q^1, \phi(h^1, Q^1))$, where ψ contains one ResNet fully connected block [32]. Since there are unsupervised cluster assignments for all point groups, we assign the labels Y to the K sampled point groups and add one more class for the points with zero occupancy probabilities to construct label Z. We apply the multi-class cross entropy loss to predict the semantic aware occupancy points:

$$L_o(P^o, Z) = -\sum_{m=1}^{K} z_m \log(p_m^o) \tag{3}$$

where K is the number of sampled point groups, and $P^o = \text{softmax}(f_o^1)$. We visualize this in Figure 3.

3.3. Implementation Details

In our experiments, we use a standard SGD optimizer with momentum 0.9, and we use a cosine learning rate scheduler [25] which decreases from 0.06 to 0.00006 and train the model for 500 epochs with a batch size of 80.

4. Experimental results

A key goal of self-supervised learning is to learn general features using unlabeled datasets and provide better pretrained models for different downstream tasks, especially when the quantity of task-specific annotations is limited. The generalizability of the learned features to different tasks improves the usability of the trained models and reduces the need for task-specific method design.

In this section, we first describe the datasets and network architectures used during pretraining in Section 4.1. Then, we discuss the baseline methods in Section 4.2, followed by present quantitative improvements in the downstream tasks of semantic segmentation and object detection. These results show that the representations learned by our method generalizes to different applications. Finally, we analyze the semantic information captured by the learned features with a new unsupervised semantic grouping task.

4.1. Network pretraining

Pretraining Datasets We use two large scale, well-established public datasets for pretraining: SemanticKITTI (SK) [4] and Waymo Open Dataset (WOD) [37]. SK is based on the original KITTI [13] dataset, which enabled a tremendous amount of foundational progress in the LiDAR perception field. It contains 19k training frames over 10 sequences with approximately 122k points per scene. Waymo Open Dataset [37] is recently released and provides a further leap in the scale and variability of data by providing 160k training frames from 850 driving sequences, with a higher point density of 161k points per scene on average.

Network Architecture We use the popular 3D sparse U-Net as the backbone[15, 51], but note that our technique is not specific to this backbone. The input point cloud is voxelized using a regular grid (at a resolution of $0.1 \times 0.1 \times 0.1m$ for SemanticKITTI and $0.1 \times 0.1 \times 0.15m$ for Waymo). To maximize generalizability, we only take the

	% of SK Used for Fine-Tuning				% of WOD Used for Fine-Tuning			
Self-Supervision Method	1%	2%	5%	10%	1%	2%	5%	10%
No Pre-training	38.9	44.0	51.7	53.4	42.5	45.8	50.4	52.8
PointContrast [44]	41.1(+2.2)	45.0(+1.0)	51.0(-0.7)	52.3(-1.1)	43.8(+1.3)	46.7(+0.9)	49.0(-1.4)	53.4(+0.6)
DepthContrast [49]	39.2(+0.3)	44.7(+0.7)	49.9(-1.8)	52.3(-1.1)	42.7(+0.2)	45.8(+0.0)	50.7(+0.3)	53.0(+0.2)
SegContrast [30]	42.2(+3.3)	45.7(+1.7)	51.0(-0.7)	53.9(+0.5)	43.4(+0.9)	46.2(+0.4)	50.9(+0.5)	53.8(+1.0)
SSPL [48]	42.5(+3.6)	46.4(+2.4)	51.0(-0.7)	53.6(+0.2)	44.8(+2.3)	47.3(+1.5)	51.3(+0.9)	53.5(+0.7)
Ours	**45.1**(+6.2)	**49.0**(+5.0)	**53.0**(+1.3)	**55.2**(+1.8)	**46.0**(+3.5)	**47.9**(+2.1)	**51.7**(+1.3)	**54.1**(+1.3)

Table 1. Semantic segmentation fine-tuning performance on SemanticKITTI Dataset and Waymo Open Dataset (mIoU)

	Car (Moderate)				Pedestrian (Moderate)				Cyclist (Moderate)			
Self-Supervision Method	5%	10%	20%	50%	5%	10%	20%	50%	5%	10%	20%	50%
No Pre-training	60.2	69.1	74.3	77.8	48.2	**58.8**	59.7	59.2	44.9	57.6	63.3	70.5
PointContrast [44]	62.2	70.6	66.9	77.4	48.6	58.2	58.6	59.1	46.8	58.4	64.6	70.9
DepthContrast [49]	65.0	72.5	77.1	77.7	48.5	55.1	57.1	57.7	51.9	59.6	65.3	71.8
SegContrast [30]	65.4	73.0	77.0	77.9	48.0	57.2	57.6	58.1	50.6	59.3	65.8	72.0
SSPL [48]	63.3	71.1	76.8	76.8	48.1	55.3	57.0	58.2	48.0	58.8	64.2	71.3
Ours	**68.9**	**74.3**	**77.3**	**78.4**	**48.9**	56.5	**59.9**	**59.8**	**53.2**	**60.7**	**69.5**	**73.8**

Table 2. 3D object detection fine-tuning performance on sub-sampled KITTI Dataset (mAP_R11)

point coordinates as input data, and ignore other attributes including intensity. For the semantic segmentation task, we use the same backbone as [46]. The sparse input voxel grid is first processed by an input convolution. Next, the features are encoded by a sparse U-Net[51] architecture with 6 layers. Each layer uses a convolution and non-linearity blocks to first transform the incoming features, perform 2x downsampling, and decode the result by merging features from the next higher layer. The final output of the network is reprojected to yield the 64-dimensional per-point features which are used during pretraining. For the object detection task, we use a similar U-Net model as in OpenPCDet [38], which consists of four sparse downsampling blocks for encoding and four sparse upsampling blocks for decoding.

4.2. Baseline methods

We select 4 relevant baseline methods for LiDAR-only self-supervised learning. The first baseline **PointContrast** [44] generates point pairs between different views and enforces feature consistency between point pairs with contrastive learning. We reproduce their approach with 2048 sampled point pairs for each frame during training. Next, we compare against **DepthContrast** [49] uses cross representation feature consistency for self-superwised learning. However, they only perform feature learning on a globally-pooled feature embedding. Similar to our approach, the baseline of **SegContrast** [30] also applies ground removal and DBSCAN clustering to extract point groups, but they only perform contrastive learning on the point group features. The most recent baseline **SSPL** first crops the entire scene to few hundreds of occupied volumes and then applies both local and global contrastive learning on the extracted volume features. Finally, we compare against random initialization to quantify the absolute impact of pretraining.

To ensure fairness, we use the same model architectures, pretraining datasets, and procedures for our method as well as to reproduce all baseline methods. All training is conducted on systems with 8 NVIDIA V100 GPUs. More details can be found in the supplementary materials.

4.3. Downstream Task 1: Semantic Segmentation

Our algorithm implicitly reasons about the semantic identities of groups of points, and is most directly related to the task of assigning meaningful labels to every 3D point in the environment. Therefore, we first evaluate our method on semantic segmentation as the downstream task.

We conduct the pretraining on both SemanticKITTI and WOD, and finetune with reduced quantities of labeled data. To avoid the domain adaptation issues due to different sensor configuration, we pretrain and finetune on the same dataset. In self-supervised learning settings, the dataset used for pretraining is usually orders of magnitude larger than the labeled data for downstream tasks. Therefore we present semantic segmentation results on several fractional subsampled sets of the training data provided by SemanticKITTI and WOD. The sets are selected such that each smaller set is fully contained within all larger sets. The results are shown in Table 1. Similar to [48], we add a small

	Car	Cycle*	Truck	Ped	Cyclist	Build	Fence	Vege	Trunk	Pole	Sign	Mean**
GT	89.6	46.9	76.3	64.2	49.6	80.4	42.0	69.2	59.1	48.6	62.6	62.6
No Pretraining	20.8	2.7	4.1	2.1	0.0	42.1	17.4	35.7	7.8	4.1	2.7	12.7
PointContrast [44]	40.0	3.1	7.4	**6.5**	2.7	39.3	5.6	29.8	15.1	9.9	3.9	14.8
DepthContrast [49]	57.3	2.2	0.4	0.1	0.0	45.1	10.2	47.0	15.9	12.1	0.1	17.3
SegContrast [30]	51.6	2.5	4.8	3.5	**4.8**	52.7	12.0	48.6	20.7	**21.8**	**5.8**	20.8
SSPL [48]	34.6	1.1	12.4	2.1	1.9	51.2	17.2	26.6	5.7	2.6	1.6	14.3
Ours	**67.7**	**4.1**	**14.7**	4.8	3.9	**71.1**	**27.5**	**51.4**	**23.6**	18.0	5.5	**26.6**

Table 3. Feature Evaluation: unsupervised semantic grouping on prominent classes in SemanticKITTI Dataset (IoU). * cycle class is the average of bicycle and motorcycle. ** mean is computed over selected classes. Further details are in the supplementary materials.

decoder with two pointwise layers to transform the features produced by the self-supervised backbone, and finetune with annotations until convergence. To reduce noise in the performance estimation, we report averaged results over the last 15% of the checkpoints.

As shown in Table 1, our approach significantly improves downstream semantic segmentation when the annotation is limited. Most impressively, we observe that finetuning on only 1% of labeled data with our pretrained model results in better performance than training from scratch with 2% of labeled data. As well, we observe that competitor approaches can often produce worse results after fine-tuning when larger quantities of annotated data (5% and 10%) are available. This indicates that their learned features can be helpful in scenarios with very little annotations, but introduce harmful biases in the network initialization that prevent the model from optimally learning from larger quantities of labels. In contrast, our approach consistently sees performance improvements across all settings.

4.4. Downstream Task 2: 3D Object Detection

The 3D object detection task primarily focuses on identifying foreground objects of interest and proposing a bounding cuboid that encompasses the complete extent of the object. However, there are many situations where the object is heavily obscured by itself or other objects, or otherwise has only limited perceived points due to its distance to the LiDAR sensor. As such, this task requires a precise understanding of the complete shapes of foreground objects. Our approach directly reasons about the missing information via the implicit surface reconstruction, which allows it to significantly outperform the baseline methods.

Similar to the semantic segmentation task, we pretrain and finetune on the same dataset. We present the finetuning results on uniformly subsampled sets of the training data provided by KITTI. We use the detection framework from Part-A^2 [36] and pretrain the 3D U-Net backbone used by their method. We follow the same training settings as in OpenPCDet [38] for finetuning. Likewise, we average the performance over the last five checkpoints to reduce noise.

As shown in Table 2, our approach significantly improves downstream detection performance, especially when the annotation is limited. For example, we observe 8.7% and 8.3% improvements in mAP on car and cyclist, respectively, when annotations for 1% subsets of the full datasets are provided. These improvements are much more significant than those gained from the baseline methods, which shows the effectiveness of our approach.

4.5. Feature evaluation: unsupervised semantic grouping

To directly examine the features learnt, we propose a feature evaluation method on LiDAR point clouds: unsupervised semantic grouping. After obtaining the pretrained networks, we extract the point group features for each scene and aggregate them globally to perform *k*-means clustering. We then assign the generated *k*-means labels to each cluster. With the *k*-means labels and ground truth semantic labels, we formulate a classic assignment problem where we optimally match each *k*-means label to one ground truth label to maximizing the resulting mIoU.

For our experiments, we use 1000 clusters for *k*-means and randomly sample 1% (191 frames) from SemanticKITTI for this evaluation for computation tractability considerations. We use the MIP solver from the Google Optimization Tools [33] to solve the assignment problem. With the optimal assignment, we map the cluster IDs to 19 ground truth class IDs defined in SemanticKITTI, and compute the segmentation accuracy.

In Table 3, we show the per-class IoU evaluation on selected classes for baseline methods and our approach. Since a point group can contain multiple semantic classes, we also show the upper-bound of per-class IoU by using the ground truth labels, representing the limitations of HDBSCAN. Overall, our approach outperforms baseline methods in mIoU over most categories, especially for frequent classes such as building, vegetation and cars, where the performance of our method is close to the upper-bound. It shows that our approach is able to identify common semantic features across the dataset and cluster them well during pretraining. However, features corresponding to rarely seen objects such as pedestrians and cyclists may be considered

	1%	2%	5%	10%
None	38.9	44.0	51.7	53.4
occ-only(8)	40.5(+1.6)	45.7(+1.7)	51.4(-0.3)	52.0(-1.4)
occ-only(16)	43.0(+4.1)	**46.1**(+2.1)	**51.9**(+0.2)	54.0(+0.6)
occ-only(32)	**43.6**(+4.7)	46.0(+2.0)	51.7(+0.0)	**54.1**(+0.7)

Table 4. Occupancy Prediction Only: We show the relative performance gain with only performing the occupancy prediction pretext task and we show different results for using 8, 16 and 32 neighbouring voxel features respectively.

	1%	2%	5%	10%
None	38.9	44.0	51.7	53.4
Single-view	43.7(+4.8)	48.1(+4.1)	51.7(+0.0)	54.4(+1.0)
Multi-view(2)	44.0(+5.1)	48.4(+4.4)	52.8(+1.1)	54.9(+1.5)
Multi-view(5)	**45.1**(+6.2)	**49.0**(+5.0)	**53.0**(+1.3)	**55.2**(+1.8)

Table 5. Single-view vs Multi-view: We show the relative performance gain with single-view alone, multi-view sequences with 2 frames separated and 5 frames separated.

as noisy points during clustering and result in incorrect clusters. Therefore, for those categories, our approach performs worse than the baseline methods which use less global reasoning, such as PointContrast and SegContrast. We believe that this issue can be solved by applying weights based on appearance frequency to the clustering loss. Please see the supplemental materials for evaluation on other classes.

5. Ablation studies

In this section, we analyze the impact of each component in our pretraining framework. We use fine-tuning for semantic segmentation on SemanticKITTI for this analysis.

5.1. Importance of occupancy prediction

We examine the impact of occupancy prediction by removing the objective. As seen in Table 6 where only the global clustering objective is used, the performance is significantly lower than in Table 1, especially when more annotations are available during finetuning. It shows that our approach learns additional features with this local geometry reconstruction task. Furthermore, Table 4 also shows that our pretraining method benefits from an increase in neighbouring voxel features used for aggregation, gradually saturating with 16 neighboring features.

5.2. Single-view vs multi-view

Although our method does not directly require multi-view data as input to the network, it benefits from using multi-view information as a form of natural data augmentation for the global contrastive clustering task. In Table 5, we observe that our approach performs best with multi-view data. As the ego-vehicle moves, a larger frame separation allows elements in the scene to be observed from widely different angles. This increases the difficulty of the contrastive task, as the object has very different perceived appearance in the paired frames. However, it is interesting to note that our approach can already outperform baseline methods in Table 1 even with single view data.

	1%	2%	5%	10%
None	38.9	44.0	51.7	53.4
local-only	41.3(+2.4)	45.9(+1.9)	51.7(+0.0)	53.0(-0.4)
global-only	41.7(+2.8)	45.5(+1.5)	51.8(+0.1)	52.5(-0.9)
full(100)	**43.6**(+4.7)	**47.0**(+3.0)	51.8(+0.1)	53.1(-0.3)
full(200)	43.5(+4.6)	46.1(+2.1)	**51.9**(+0.2)	**53.6**(+0.2)

Table 6. Global Clustering Only: We show the relative performance gain with only performing the global clustering pretext task under different settings: only using the local contrastive loss (local-only), only using the global culstering loss (global-only), and full contrastive clustering with 100 and 200 clusters.

5.3. Importance of global clustering

Next, we showcase the impact of the global clustering task. In Table 4, we see that the performance degrades relative to Table 1 if only the occupancy prediction task is used. Moreover, if we only apply the local contrastive loss, the performance improvements are minimal based on Table 6. If we only apply the global clustering loss, the number of distinct clusters after Sinkhorn clustering will reduce, resulting in minimal improvements. Therefore, our approach reaches the best performance only when applying the full contrastive loss. We observed that setting the maximum number of clusters to 100 and 200 have similar results, showing that the method is robust to the maximum number of clusters for Sinkhorn clustering.

6. Conclusion and future work

In conclusion, we propose **ISCC**, a new self-supervised pretraining approach with two pretext tasks: global clustering and occpancy prediction. We show that our approach extracts meaningful features for semantic segmentation and object detection over different datasets. Due to the special characteristics of LiDAR pointclouds, our approach contains several components that can only be applied to similar data, such as ground plane removal. Additionally, we only study the in-domain finetunig performance instead of producing a general purpose foundation model as seen in images and language understanding. We believe that these are possible areas for future research. We hope **ISCC** will inspire further advances in 3D self-supervised learning.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *International Conference on Learning Representations (ICLR)*, 2020. 3, 4

[2] Ma Baorui, Liu Yu-Shen, Zwicker Matthias, and Han Zhizhong. Surface reconstruction from point clouds by learning predictive context priors. In *CVPR*, 2022. 2

[3] Stefan Baur, David Emmerichs, Frank Moosmann, Peter Pinggera, Bjorn Ommer, and Andreas Geiger. Slim: Self-supervised lidar scene flow and motion segmentation. In *International Conference on Computer Vision (ICCV)*, 2021. 2

[4] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 5

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 1, 2, 3

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 3

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 3

[8] Yujin Chen, Matthias Nießner, and Angela Dai. 4dcontrast: Contrastive learning with dynamic correspondences for 3d scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[9] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 4

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[11] Emeç Erçelik, Ekim Yurtsever, Mingyu Liu, Zhijie Yang, Hanzhen Zhang, Pınar Topçam, Maximilian Listl, Yılmaz Kaan Çaylı, and Alois Knoll. 3d object detection with a self-supervised lidar scene flow backbone. *ECCV*, 2022. 2

[12] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020. 2

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2, 5

[14] Zan Gojcic, Or Litany, Andreas Wieser, Leonidas J Guibas, and Tolga Birdal. Weakly supervised learning of rigid 3d scene flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5692–5703, 2021. 3

[15] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *CVPR*, 2018. 5

[16] 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. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1, 2

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

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 1, 2, 3

[19] Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6535–6545, 2021. 2

[20] Jyh-jing Hwang, Stella X. Yu, Jianbo Shi, Maxwell D Collins, Tien-ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: segmentation by discriminative sorting of segments. In *ICCV*, 2019. 2

[21] Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X. Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformer. In *CVPR*, 2022. 2

[22] Seungjae Lee, Hyungtae Lim, and Hyun Myung. Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3D point cloud. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022. Submitted. 3, 4

[23] 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. *ICCV*, 2021. 2

[24] Shi-Lin Liu, Hao-Xiang Guo, Hao Pan, Peng-Shuai Wang, Xin Tong, and Liu Yang. Deep implicit moving least-squares functions for 3d reconstruction. In *CVPR*, 2021. 2

[25] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[26] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 3

[27] Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jefferey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (ICLR)*, 2013. 2

[28] Himangi Mittal, Brian Okorn, Arpit Jangid, and David Held. Self-supervised point cloud completion via inpainting. In *BMVC*, 2021. 2

[29] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 5

[30] Lucas Nunes, Rodrigo Marcuzzi, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Segcontrast: 3d point cloud feature representation learning through self-supervised segment discrimination. *IEEE Robotics and Automation Letters*, 7(2):2116–2123, 2022. 1, 2, 3, 6, 7

[31] 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. 4

[32] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 5

[33] Laurent Perron and Vincent Furnon. Or-tools. 7

[34] Anjana Deva Prasad, Anushrut Jignasu, Zaki Jubery, Soumik Sarkar, Baskar Ganapathysubramanian, Aditya Balu, and Adarsh Krishnamurthy. Deep implicit surface reconstruction of 3d plant geometry from point cloud. In *AAAI*, 2022. 2

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115, 2015. 1

[36] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *arXiv preprint arXiv:1907.03670*, 2019. 7

[37] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. This publication was made using the Waymo Open Dataset, provided by Waymo LLC under license terms available at waymo.com/open. 2, 5

[38] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 6, 7

[39] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgeoulis, and Luc V Gool. Revisiting contrastive methods for unsupervised learning of visual representations. In *NeurIPS*, 2021. 2

[40] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *CVPR*, 2020. 2

[41] Xinlong Wang, rufeng Zhang, Chunhua Shen, Tao Kong, and Li Lei. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 2

[42] Wenxuan Wu, Zhi Yuan Wang, Zhuwen Li, Wei Liu, and Li Fuxin. Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation. In *European Conference on Computer Vision*, pages 88–107. Springer, 2020. 2

[43] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[44] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 6, 7

[45] Siming Yan, Zhenpei Yang, Haoxiang Li, Li Guan, Hao Kang, Gang Hua, and Qixing Huang. Implicit autoencoder for point cloud self-supervised representation learning. *arXiv preprint arXiv:2201.00785*, 2022. 4

[46] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3101–3109, 2021. 6

[47] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposal-contrast: Unsupervised pre-training for lidar-based 3d object detection. In *ECCV*, 2022. 1, 2

[48] Zaiwei Zhang, Min Bai, and Erran Li. Self-supervised pre-training for large-scale point clouds. In *NeurIPS 2022*, 2022. 2, 6, 7

[49] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10252–10263, 2021. 1, 2, 3, 6, 7

[50] Huang Zitian, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *CVPR*, 2020. 2

[51] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *MICCAI*, 2016. 5, 6