# TWIST: Two-Way Inter-label Self-Training for Semi-supervised 3D Instance Segmentation

Ruihang Chu[1] Xiaoqing Ye[2] Zhengzhe Liu[1] Xiao Tan[2]
Xiaojuan Qi[3*] Chi-Wing Fu[1,4] Jiaya Jia[1,5]
[1]CUHK [2]Baidu Inc. [3]HKU [4]SHIAE [5]SmartMore

## Abstract

*We explore the way to alleviate the label-hungry problem in a semi-supervised setting for 3D instance segmentation. To leverage the unlabeled data to boost model performance, we present a novel Two-Way Inter-label Self-Training framework named TWIST. It exploits inherent correlations between semantic understanding and instance information of a scene. Specifically, we consider two kinds of pseudo labels for semantic- and instance-level supervision. Our key design is to provide object-level information for denoising pseudo labels and make use of their correlation for two-way mutual enhancement, thereby iteratively promoting the pseudo-label qualities. TWIST attains leading performance on both ScanNet and S3DIS, compared to recent 3D pre-training approaches, and can cooperate with them to further enhance performance,* e.g.*, +4.4% $AP_{50}$ on 1%-label ScanNet data-efficient benchmark. Code is available at https://github.com/dvlab-research/TWIST.*

## 1. Introduction

Deep learning methods have achieved great success on 3D point cloud learning. They demand large-scale annotated data. Compared to work on scanning, methods for annotations consume substantially more manual effort. For ScanNet [9], 20 people were hired for collecting the RGB-D scans. However, it took 500 crowd-based workers, each using around 22.3 minutes, to label one scan on average.

To alleviate this label-hungry problem, a direction is to exploit semi-supervised learning (SSL). This setting needs ground-truth labels only for a small fraction of the training set. The target is to leverage a large volume of completely unlabeled data to boost model performance. Contrary to intensive research on image understanding [1, 14, 26, 27, 38, 39], less [20, 49] was carried out on this setting for 3D instance segmentation, which is an important task for 3D perception. Methods of [20, 49] utilize unlabeled data for model pre-training through a contrastive loss. They, however, only explore SSL by means of consistency regularization.

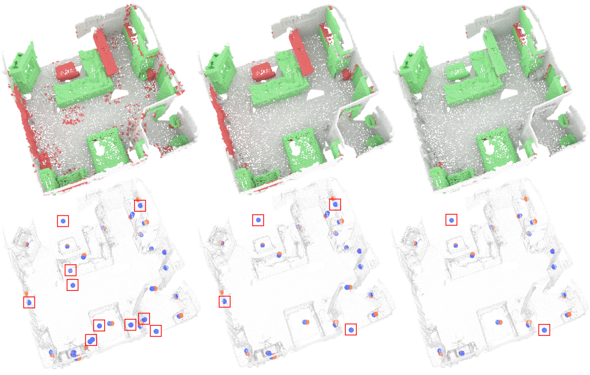

Figure 1. Top: Pseudo semantic labels (green means correct and red means incorrect results) produced on unlabeled point cloud by (a) confidence thresholding, (b) our method without the re-correction module, and (c) our full method. Bottom: Pseudo centroids (blue dots) found by various methods. Orange dots mark the ground truth (GT) and red boxes mark the blue dots far from GTs. Our TWIST framework (c) effectively promotes the quality of both semantic and offset-to-centroid pseudo labels.

In this paper, we address semi-supervised 3D instance segmentation by designing a new self-training framework, which is the first of this kind. We aim to generate high-quality pseudo labels from unlabeled data to improve model training. This goal is challenging to achieve. First, the task requires both semantic- and instance-level understanding of a 3D scene. These two goals may conflict with each other. For example, we may want different instances of the same class to have different instance IDs but the same semantic ID. Therefore, it is non-trivial to generate high-quality (joint) pseudo labels for supporting both prediction tasks.

Second, the way to promote consistency among pseudo labels in the instance-level task has much room to explore. For instance, pseudo semantic labels within the same instance should be consistent. Otherwise, it could confuse how points are separated into object instances. Third, an effective pseudo-label evaluation and selection mechanism is in high

[*] Corresponding Author

demand. It cannot be achieved easily by common strategies, such as simple confidence thresholding [3, 48].

To address these issues, we design the **T**wo-**W**ay **I**nter-label **S**elf-**T**raining (TWIST) framework that collectively considers two kinds of pseudo labels, *i.e.*, *pseudo semantic labels* for semantic-level supervision and *pseudo offset vectors* for instance-level supervision. Importantly, TWIST iteratively updates the two pseudo-label sets, while promoting their consistency and quality. The key designs include a novel proposal re-correction module to leverage object-level predictions to denoise the pseudo labels and strategies to enable inter-label mutual enhancement.

Specifically, TWIST does not generate pseudo labels in the point level like point-wise confidence thresholding due to its vulnerability to noise, as observed in Fig. 1(a). Instead, we utilize the model to predict instance proposals and leverage this prior to update the pseudo labels in proposal level, naturally preserving the intra-proposal consistency. To further improve the pseudo-label quality, we develop the proposal re-correction module to provide object-wise evaluation along with pseudo-label denoising. This module can be trained in a learnable fashion and takes input of diverse proposal-level samples to mitigate the label-hungry issue.

Another notable characteristic of TWIST is to explicitly encourage mutual enhancement between two pseudo-label sets. Here, we design two-way bidirectional inter-label interactions, implemented by the semantic-guided instance proposal generation module and the proposal-based pseudo-label update module. Also, we design the proposal re-correction module between them as a safeguard to assess proposal quality and correct the labels of low-quality ones. It encourages better model convergence. By these means, we jointly enhance the two pseudo-label sets significantly, as shown in Fig. 1(c).

We evaluate TWIST on two large-scale 3D datasets of ScanNet v2 [9] and S3DIS [2]. TWIST outperforms both the supervised-only baseline and state-of-the-art unsupervised pre-training approaches [20, 49] by a large margin. Also, it can cooperate with other 3D pre-training approaches [20, 49] for further performance gain of 0.8 to 4.4 points. This property indicates that TWIST has complementary strength for semi-supervised 3D instance segmentation. Our overall contributions are as follows.

- We demonstrate the effectiveness of self-training for semi-supervised 3D instance segmentation, using two kinds of pseudo labels for effective model training.
- We present TWIST that enables generation of more accurate pseudo labels with object-level denoise and two-way inter-label enhancement.
- A new SOTA semi-supervised learning framework is proposed for 3D instance segmentation. It is verified on two large-scale datasets and shows complementary strength with existing 3D pre-training approaches.

## 2. Related Work

**3D Instance Segmentation.** Given a point cloud, this task aims to predict object instances, each with a semantic class. 3D instance segmentation approaches [4, 5, 11, 15, 18, 19, 22, 24, 25, 28, 29, 34, 44, 45, 53, 54] can be classified into top-down and bottom-up methods. Top-down methods [4, 19, 53] often adopt a detect-and-segment pipeline that first leverages the geometry and/or color features to produce 3D proposals and then refines the proposals by the mask predictions.

On the other hand, bottom-up methods [5, 8, 11, 15, 18, 22, 24, 25, 28, 44, 45, 54] form object instances by clustering the input points based on their embedded similarity. SPGN [44] and ASIS [45] promote intra- and inter-instance similarity by a discriminative loss. Later, methods of [11, 24, 25] consider semantic prediction and geometric distribution in the clustering. A natural prior is to only group points of consistent semantic category into the same instance. Such pipelines are further improved by various techniques, such as multi-task learning [15], hierarchical aggregation [5], dynamic kernel [18], and superpoint traversal [28]. In this paper, we focus on label-efficient settings and develop our baseline framework, following the bottom-up paradigm.

**Label-efficient Learning in 3D.** Labeling point cloud is laborious and error-prone. Several recent approaches explore label-efficient learning for point clouds. Instead of requiring labels for every point in the training set, incomplete/indirect labels, *e.g.*, 2D image labels [40, 43, 47], sparse 3D point labels [21, 30–32, 50, 55], region/scene tags [36, 37, 41, 46], and labels from partial training set [6, 10, 23, 40, 42, 56] were used. Though the supervision becomes weak, the model is designed to exploit accessible information for optimizing performance. These techniques have been verified in diverse 3D tasks, *e.g.*, single CAD-model classification and part segmentation [12, 16, 33, 57], large-scale semantic segmentation [6, 10, 21, 23, 30, 41, 46, 50, 55], and object detection [31, 32, 36, 37, 40, 42, 47, 56].

We focus on 3D instance segmentation, which requires label-efficient learning. Here, we have ground-truth labels only for a small portion of the training set and aim to adopt large unlabeled data to boost model performance.

Beyond the SOTA semi-supervised methods for 3D semantic segmentation [6, 10, 23], our model further produces instance labels by clustering the points. It needs to efficiently extract object localization knowledge from the unlabeled data (in large quantity) and assimilate such knowledge with that from the labeled data (in small quantity). Compared to works on semi-supervised 3D object detection [40, 42, 56], we focus on crowded indoor scenes that need point-level instance separation. So far, there are only a few unsupervised pre-training methods [20, 49] that can help on this task. This is the first work with a novel self-training model.

**Self-Training.** Self-training is a popular technique for semi-supervised learning and has been successfully applied to many 2D image understanding tasks [17, 48, 51, 58, 60, 61]. Pursuing this direction is desirable for 3D data to reduce high annotation cost. Until now, only a few methods benefit self-training on 3D tasks, *e.g.*, 3D shape classification [59], semantic segmentation [30], and object detection [37, 52]. In this paper, we demonstrate the effectiveness of self-training on 3D instance segmentation. The key to the success of our method is the object-level pseudo-label denoising and inter-label mutual enhancement to promote the pseudo-label quality in self-training.

## 3. Preliminaries on 3D Instance Segmentation

Given a point cloud $P = \{(p_i, c_i)\}_{i=1}^N$, where each $p_i = (x_i, y_i, z_i)$ is a 3D coordinate and each $c_i$ is a RGB color, the model produces a set of 3D object instance proposals $\hat{G} = \{\hat{g}_j\}_{j=1}^M$, where each $\hat{g}_j$ is a subset of points in $P$ of the same inferred semantic class. We use $i$ as the point index and $j$ as the instance proposal index.

**Revisiting Supervised Baseline Framework.** For labeled supervision, we directly use a bottom-up framework [20], which adopts a neural model $\Phi$ with a shared Sparse U-Net [13] and two separate MLP-based branches. One branch is for predicting a set of per-point semantic classes $\hat{S} = \{\hat{s}_i \in \{1, ..., K\}\}_{i=1}^N$ and the other is for predicting a set of per-point offset vectors in 3D $\hat{O} = \{\hat{o}_i \in \mathbb{R}^3\}_{i=1}^N$ for shifting points on object surface towards the respective instance centroids, where $K$ is the number of semantic classes. $\hat{S}$ is supervised by ground-truth semantic labels $S$ under the standard cross entropy loss and $\hat{O}$ is explicitly supervised by the ground-truth point-to-centroid vector $O$ via a regression loss. We train the network by jointly optimizing $\hat{S}$ and $\hat{O}$.

During test, both the semantic predictions and compact point locality $(p_i + \hat{o}_i)$ are incorporated for point-level clustering. Also, we use breadth-first search to explore neighboring points of the same semantic category within an $x$-cm sphere and cluster these points into an object instance.

**Discussions.** We choose such a supervised pipeline as our baseline since it is simple and decent on evaluation benchmarks [25]. A pivotal success factor is that accurate semantic prediction provides a strong prior for filtering out noisy inter-class points in the vicinity, thus purifying the instance clustering results. We leverage this principle for semi-supervised learning.

## 4. Our Method

In the semi-supervised setting, only a small fraction of point cloud scenes have labels, while the remaining large set is unlabeled. We use superscripts $l$ and $u$ to indicate *labeled* and *unlabeled* quantities, respectively. $P^l$ denotes a labeled point cloud and $P^u$ denotes an unlabeled one. Also, we denote ($\hat{S}^l$,$\hat{O}^l$,$\hat{G}^l$) and ($\hat{S}^u$,$\hat{O}^u$,$\hat{G}^u$) as the per-point semantic classes, per-point offsets, and instance proposals predicted on $P^l$ and $P^u$, respectively, and denote $S^l$ and $O^l$ as the per-point ground-truth semantic labels and offset vectors of $P^l$, respectively.

Our goal is to exploit the unlabeled data in training to boost 3D instance segmentation performance. To learn knowledge effectively from the unlabeled data, we train the network through the following self-training pipeline.

**Step (i)** is the *initialization stage*, in which we train model $\Phi(., \theta_0^r)$, using the model described in Sec. 3 and with $\theta_0^r$ model weights at self-training round 0, on all labeled point clouds. The objective on each $P^l$ is

$$\mathcal{L}^l = \mathcal{L}_s(S^l, \hat{S}^l) + \mathcal{L}_o(O^l, \hat{O}^l), \tag{1}$$

where $\mathcal{L}_s$ is the cross entropy loss on semantic predictions $\hat{S}^l$, and $\mathcal{L}_o$ is the regression loss for supervising both the $L1$ distance and direction of the predicted offset vectors $\hat{O}^l$. For a point cloud $P^l$ with $N$ points, $\mathcal{L}_o$ is formulated as

$$\mathcal{L}_o(O^l, \hat{O}^l) = \frac{1}{N}\sum_i^N (||o_i^l - \hat{o}_i^l|| - \frac{o_i^l}{||o_i^l||_2} \cdot \frac{\hat{o}_i^l}{||\hat{o}_i^l||_2}). \tag{2}$$

**Step (ii)** is the *pseudo-label generation stage*. At round $t$ of the self-training, we first use the learned model $\Phi(., \theta_{t-1}^r)$ to predict semantic classes $\hat{S}^u$ and offset vectors $\hat{O}^u$ for each unlabeled point cloud $P^u$, and then refine them to produce pseudo semantic labels $\tilde{S}^u$ and pseudo offset vectors $\tilde{O}^u$.

**Step (iii)** is the *training stage* for updating the network model. Also, at round $t$ of the self-training, we use pseudo labels $\tilde{S}^u$ and $\tilde{O}^u$ to refine model $\Phi(., \theta_{t-1}^r)$ into $\Phi(., \theta_t^r)$. For each point cloud pair ($P^l$, $P^u$), the training objective is

$$\mathcal{L}^{\Phi} = \mathcal{L}^l + \mathcal{L}^u, \tag{3}$$

$$\text{where } \mathcal{L}^u = \mathcal{L}_s(\tilde{S}^u, \hat{S}^u) + \mathcal{L}_o(\tilde{O}^u, \hat{O}^u). \tag{4}$$

The self-training iterates between steps **(ii)** and **(iii)** until the performance converges. Importantly, pseudo semantic labels $\tilde{S}^u$ provide class-level supervision and pseudo offset vectors $\tilde{O}^u$ provide instance-level supervision, enabling the use of unlabeled data for updating the network model.

The key to the success of self-training is to produce accurate pseudo labels in step **(ii)**. This cannot be easily achieved by simple point-wise confidence thresholding. Also, for 3D instance segmentation, we consider pseudo-labels consistency and explore their mutual correlation to promote their quality; see the three components in our TWIST in Fig. 2.

First, the semantic-guided proposal generation module (Sec. 4.1) clusters points of same semantic predictions into candidate instance proposals $\hat{G}^u$ in each unlabeled point cloud $P^u$. Since these proposals may not be accurate, we design the proposal re-correction module (Sec. 4.2), which is a learnable model, to locate more reliable instance proposals with object-level assessment/refinement. After that, the

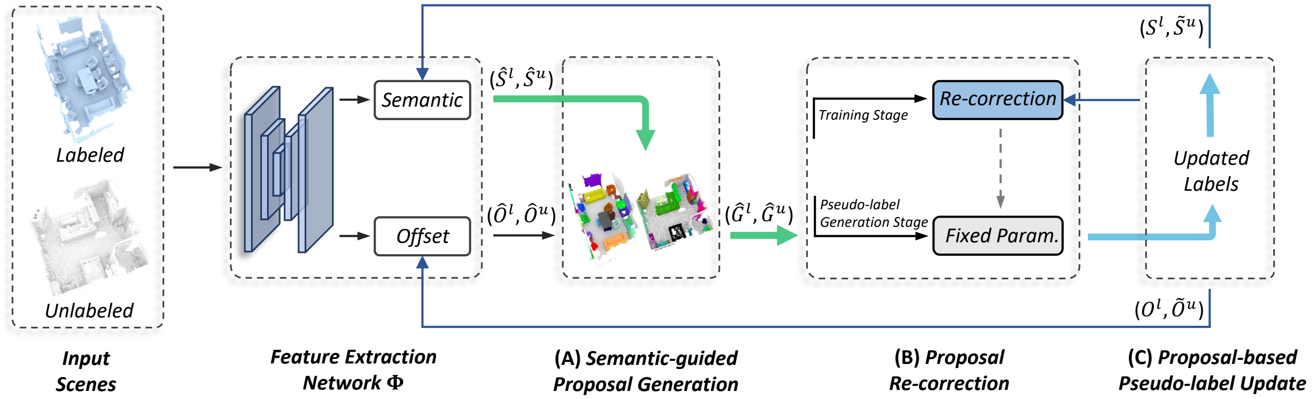

Figure 2. Overview of our TWIST framework. Given a pair of labeled and unlabeled point clouds ($P^l$, $P^u$), the feature extraction network $\Phi$ first predicts per-point semantic classes ($\hat{S}^l$, $\hat{S}^u$) and per-point offsets ($\hat{O}^l$, $\hat{O}^u$). At the *training stage* of each self-training round, $\hat{S}^l$ and $\hat{O}^l$ are forwarded by module **A** to generate instance proposals $\hat{G}^l$, which is then fed into module **B** for evaluation/rectification. Here, ($\hat{S}^l$,$\hat{O}^l$) and ($\hat{S}^u$,$\hat{O}^u$) are supervised by ground-truth labels ($S^l$, $O^l$) and pseudo labels ($\tilde{S}^u$, $\tilde{O}^u$), respectively. Module **B** can also be trained (see Sec. 4.2). At the *pseudo-label generation stage*, only the unlabeled point cloud is processed. We pass $\hat{S}^u$ and $\hat{O}^u$ through modules **A**, **B**, and **C** for object-level denoising and finally update pseudo labels ($\tilde{S}^u$, $\tilde{O}^u$). The thick arrows in green and blue represent the two-way mutual enhancement between pseudo labels $\tilde{S}^u$ and $\tilde{O}^u$, as discussed in Sec. 4.3.

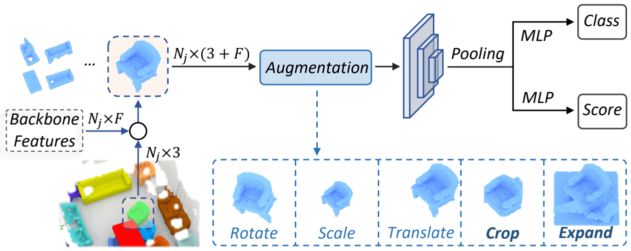

Figure 3. The re-correction module. Its input includes multiple object-level features, each formed by a concatenation of 3D coordinates of an instance proposal and associated backbone features. The module returns a semantic class along with an instance certainty score. Both can be supervised by ground-truth labels. We design five augmentation strategies to diversify the input features.

proposal-based pseudo-label update module (Sec. 4.3) generates pseudo labels $\tilde{S}^u$ and $\tilde{O}^u$ from the reliable instance proposals and helps enforce the intra-proposal consistency.

These modules work together to explore mutual correlation between the pseudo labels. They enhance pseudo-label consistency and quality (analysis is in Sec. 4.3).

### 4.1. Semantic-guided Instance Proposal Generation

The semantic-guided instance proposal generation module (module **A** in Fig. 2) produces instance proposals in input point cloud $P^l$ or $P^u$ by employing the clustering algorithm described in Sec. 3 with semantic predictions as the guidance. Note that we involve this step in both the end-to-end network training (output $\hat{G}^l$ in $P^l$) and pseudo-label generation (output $\hat{G}^u$ in $P^u$). The output proposals are fed to the re-correction module for proposal-level evaluation.

### 4.2. Proposal Re-correction

The predicted semantic class of each instance proposal is not accurate enough, since it is just a combination of the per-point semantic predictions. To address the issue, we design the proposal re-correction module, called $\Psi$, to assess how well each point set in a proposal forms a single instance. We then re-classify the proposal to re-correct the misclassified labels. At the *training stage* of each round, this proposal re-correction module receives $\hat{G}^l$ for its training, and at the *pseudo-label generation stage*, it assesses a set of $\hat{G}^u$ for updating the pseudo labels.

**Input Data.** Instead of taking the entire point cloud scene as the input sample, we forward object-level features as input to $\Psi$. So, we can considerably enlarge the set of trainable samples, better predicting individual objects and making it easier for object recognition from a global point of view.

For an instance proposal $\hat{g}_j$ of $N_j$ points, the re-correction module concatenates the 3D point coordinates ($\mathbb{R}^{N_j \times 3}$) and the associated backbone features ($\mathbb{R}^{N_j \times F}$), *i.e.*, the output of Sparse U-Net in model $\Phi$, to form the input sample $\hat{k}_j \in \mathbb{R}^{N_j \times (3+F)}$ to module $\Psi$. We select backbone features instead of the original RGB features, since they effectively capture the contextual information, which could be critical for locating 3D objects, as ablated in Sec. 5.3.

**Module Training.** Fig. 3 shows the module training workflow. At the *training stage*, the re-correction module first diversifies feature $\hat{k}^l_j$ by various object-wise data augmentation. We consider two classes of augmentations. Geometric transformation includes rotation, translation, and point-wise scaling. Geometric mutation consists of cropping and expanding, which introduces stronger disturbance on changing the instance certainty score of $\hat{k}_j^l$, *i.e.*, $iou_j^l$ in Eq. (5). Specifically, they first narrow down or enlarge the bounding box of $\hat{k}_j^l$ by a random ratio and then select all points within the updated box as new $\hat{k}_j^l$. These operations remove part of $\hat{k}_j^l$ or bring points from other objects to $\hat{k}_j^l$.

Then, the re-correction module employs a small SparseConv encoder [13] and two MLP heads to process $\hat{k}_j^l$. They predict a semantic class $\hat{e}_j^l$ and an instance certainty score $\hat{v}_j^l$ for each instance proposal. $\hat{v}_j^l$ is scaled to (0,1) by Sigmoid operation. Inspired by [5, 24], we calculate point-wise IoUs between $\hat{k}_j^l$ and use the ground-truth instance, which matches the best to supervise the instance certainty score as

$$\mathcal{L}_{sc}^{\Psi} = -\frac{1}{M}\sum_{j=1}^{M}[iou_j^l \cdot \log(\hat{v}_j^l) + (1 - iou_j^l) \cdot \log(1 - \hat{v}_j^l)], \tag{5}$$

where $M$ is the number of instance proposals. For semantic supervision, we use the cross-entropy loss on $\hat{e}_j^l$ only if its $iou_j^l$ is larger than 0.5. The ground-truth value should be the semantic class of the actual instance that $\hat{k}_j^l$ mostly matches. The overall objective for training the re-correction module is $\mathcal{L}^{\Psi} = \mathcal{L}_{sem}^{\Psi} + \mathcal{L}_{sc}^{\Psi}$. It is optimized along with $\mathcal{L}^{\Phi}$ (see Eq. (3)) at the *training stage* of each round.

### 4.3. Proposal-based Pseudo-label Update

At the *pseudo-label generation stage* of each self-training round, for an unlabeled point cloud $P^u$, the re-correction module produces a re-predicted semantic class $\hat{e}_j^u$ and an instance certainty score $\hat{v}_j^u$ for each instance proposal $\hat{g}_j^u$ predicted from $P^u$. Instance proposals with a score higher than 0.5 are employed for proposal-level pseudo-label update.

For all points in $\hat{g}_j^u$, *i.e.*, $p_i^u \in \hat{g}_j^u$, we update their pseudo semantic labels using the re-predicted semantic class $\hat{e}_j^u$ as

$$\tilde{S}^u\{p_i^u \in \hat{g}_j^u\} = \hat{e}_j^u. \tag{6}$$

To update the pseudo offset vectors, we first produce the pseudo instance center of each proposal $\hat{g}_j^u$. Instead of directly selecting the centroid of $\hat{g}_j^u$, we adopt the mean-shift result by considering all points in $\hat{g}_j^u$ to make use of the predicted offset vectors as

$$\tilde{a}_j^u = \frac{1}{N_j}\sum_{p_i^u \in \, \hat{g}_j^u}(p_i^u + \hat{o}_i^u), \tag{7}$$

where $N_j$ is the number of points in $\hat{g}_j^u$. Then, we can update the set of pseudo offset vectors $\tilde{O}^u$ by

$$\tilde{O}_{\{p_i^u \in \hat{g}_j^u\}}^u = \tilde{a}_j^u - p_i^u. \tag{8}$$

Through this mechanism, we update the two types of pseudo labels in the instance-proposal level, thereby naturally preserving the intra-proposal pseudo-label consistency. Hence, even if $\hat{e}_j^u$ is wrong, points of $\hat{g}_j^u$ are still likely to be clustered in a group under the shifting guidance from $\tilde{a}_j^u$.

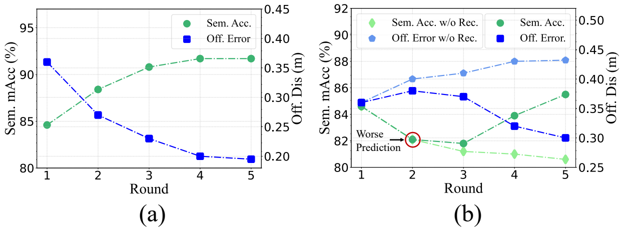

Figure 4. Pseudo-label quality plots. We evaluate the pseudo semantic labels by point-wise mean accuracy (denoted as Sem. Acc.) and pseudo offset vector by vector difference (denoted as Off. Error). (a) The pseudo-label accuracy increases consistently with more self-training rounds. (b) Using less accurate semantic predictions harms pseudo-label quality. Thus the accuracy of both pseudo-label sets continues to decrease without the re-correction module (green diamonds and blue pentagons). Our re-correction module steers the situation back (green dots and blue squares). Experiments are conducted on ScanNet v2 with 10% labels.

**Mutual Enhancement Analysis.** An intriguing merit of TWIST is that the quality of the two pseudo-label sets can be reciprocally improved through their two-way interaction.

First, a better $\tilde{S}^u$ encourages better semantic predictions, and hence guides the network to acquire better instance proposals (denoted by the thick green arrows in Fig. 2). Within one proposal, points are more likely from the same object and their mean-shift result (Eq. (7)) can be more reliable, thus promoting the quality of $\tilde{O}^u$. In turn, with a better $\tilde{O}^u$ to train more accurate offset predictions, more and better point clusters can lead to production of valid instance proposals for updating the pseudo semantic labels (denoted by the thick blue arrows in Fig. 2), where more $\tilde{S}^u$ can be produced and assigned to high-quality point sets. Hence, pseudo-label quality is jointly improved in the self-training process, as shown in Fig. 4(a).

Further, the designed re-correction module can effectively encourage their mutual effect to converge towards a positive direction. As shown in Fig. 4(b), we reduce the quality of pseudo labels on purpose by first corrupting 10% GT labels' semantic class for model training, and then using the updated model's prediction to generate pseudo labels. When removing the re-correction module, both $\tilde{S}^u$ and $\tilde{O}^u$ suffer from accuracy reduction. Fortunately, they can still recover and even converge to a better condition eventually, when the re-correction module is enabled again. The re-correction mechanism safeguards the model's fault-tolerant capability, facilitating the mutual promotion effect for both the semantic and offset pseudo labels.

## 5. Experiments

### 5.1. Experimental Setup

**Datasets.** We conduct extensive experiments on two large-scale indoor datasets of ScanNet v2 [9] and S3DIS [2]. Scan-

| Dataset | Method | 1% | | | 5% | | | 10% | | | 20% | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | mAP | $AP_{50}$ | $AP_{25}$ | mAP | $AP_{50}$ | $AP_{25}$ | mAP | $AP_{50}$ | $AP_{25}$ | mAP | $AP_{50}$ | $AP_{25}$ |
| ScanNet v2 | Sup-only | 5.1 | 9.8 | 17.6 | 18.2 | 32.0 | 47.0 | 26.7 | 42.8 | 58.9 | 29.3 | 47.9 | 63.0 |
| | PointContrast [49] | 7.2 | 12.5 | 20.3 | 19.4 | 35.4 | 48.5 | 27.0 | 43.9 | 59.5 | 30.2 | 49.5 | 63.6 |
| | CSC [20] | 7.1 | 13.0 | 21.2 | 20.9 | 36.7 | 50.6 | 27.3 | 45.0 | 60.2 | 30.6 | 50.3 | 64.1 |
| | TWIST | **9.6** (+4.5) | **17.1** (+7.3) | **26.2** (+8.6) | **27.0** (+8.8) | **44.1** (+12.1) | **56.2** (+9.2) | **30.6** (+3.9) | **49.7** (+6.9) | **63.0** (+4.1) | **32.8** (+3.5) | **52.9** (+5.0) | **66.8** (+3.8) |
| | TWIST + CSC [20] | 11.5 (+6.4) | 20.0 (+10.2) | 31.1 (+13.5) | 28.6 (+10.4) | 45.9 (+13.9) | 58.2 (+11.2) | 32.8 (+6.1) | 51.5 (+8.7) | 65.1 (+6.2) | 34.1 (+4.8) | 53.7 (+5.8) | 67.8 (+4.8) |
| S3DIS | Sup-only | 9.0 | 12.7 | 20.7 | 21.5 | 30.4 | 42.8 | 25.2 | 36.8 | 48.3 | 29.9 | 41.2 | 54.5 |
| | PointContrast [49] | 13.4 | 15.9 | 23.1 | 22.9 | 33.6 | 44.5 | 27.1 | 38.7 | 50.2 | 31.2 | 43.1 | 56.6 |
| | CSC [20] | 14.6 | 16.7 | 23.2 | 24.9 | 34.2 | 44.9 | 29.7 | 41.0 | 52.1 | 33.5 | 44.7 | 57.8 |
| | TWIST | **17.9** (+8.9) | **22.5** (+9.8) | **27.1** (+6.4) | **27.1** (+5.6) | **37.1** (+6.7) | **48.6** (+5.8) | **33.6** (+8.4) | **45.6** (+8.8) | **55.8** (+7.5) | **36.7** (+6.8) | **48.4** (+7.2) | **59.7** (+5.2) |
| | TWIST + CSC [20] | 18.9 (+9.9) | 24.8 (+12.1) | 28.9 (+8.2) | 29.3 (+7.8) | 39.6 (+9.2) | 49.9 (+7.1) | 35.0 (+9.8) | 46.9 (+10.1) | 57.8 (+9.5) | 37.9 (+8.0) | 49.5 (+8.3) | 61.6 (+7.1) |

Table 1. Results on ScanNet v2 validation set and S3DIS Area-5 set with various ratios of labeled data. 'Sup-only' is the baseline model trained with only labeled data. TWIST consistently attains the best result and can cooperate with CSC for even better performance.

| Method | 1% | 5% | 10% | 20% |
|---|---|---|---|---|
| Sup-only | 10.1 | 27.3 | 41.3 | 47.3 |
| PointContrast [49] | 11.7 | 29.8 | 43.2 | 48.8 |
| CSC [20] | 11.9 | 32.5 | 44.0 | 52.9 |
| TWIST | **14.2** (+4.1) | **40.1** (+12.8) | **46.6** (+5.3) | **53.5** (+6.2) |
| TWIST + CSC [20] | 18.6 (+8.5) | 42.1 (+14.8) | 48.1 (+6.8) | 55.0 (+7.5) |

Table 2. Results on the test set of ScanNet v2 data efficient benchmark-*limited reconstructions*. $AP_{50}$ = evaluation metric.

Net v2 consists of 1,613 real-world 3D scenes with point-level semantic and instance annotations. The whole dataset is split into training, validation, and testing sets, each with 1201, 312, and 100 scans, respectively. S3DIS has 272 scanned 3D layouts across 6 large areas. We follow the common split in previous work [20, 24] to adopt Area 5 as the validation set and the other five areas as the training set.

**SSL Training Set Partition.** On ScanNet v2, we directly adopt the setting of ScanNet data-efficient benchmark [20] and split the training set into the labeled and unlabeled sets, with {1%, 5%, 10%, 20%} labeled data, respectively. Further, we follow these four labeling ratios to split S3DIS training set by randomly sampling the 3D scenes.

**Implementation Details.** On both datasets, we train the 3D feature extraction network $\Phi$ and re-correction module $\Psi$ by the SGD optimizer, with learning rate set to 0.1 and 0.005, respectively. The learning rate is scheduled with the polynomial decay of power 0.9. In each round (*i.e.*, step **(iii)** and step **(ii)** in Sec. 4), our model is trained on 4 NVIDIA 2080Ti GPUs for 10k steps with batch size 8 with 4 labeled scenes and 4 unlabeled scenes.

We adopt standard point cloud data augmentation strategies in [35] to process 3D scenes before feeding them to $\Phi$. The bounding box scaling ratios for geometric mutation augmentations when training $\Psi$ are within [0.7, 1.3]. For all experiments, we adopt the same Sparse U-Net [20, 49] implemented by MinkowskiEngine [7] as the backbone of $\Phi$ and a much smaller decoder with fewer layers and stages, as the backbone of $\Psi$, with a voxelization size of 2.0 cm. The self-training converges in 3 or 4 rounds. More training details on self-training initialization are illustrated in the supplementary material.

| L.R. | Method | mAP | $AP_{50}$ | $AP_{25}$ |
|---|---|---|---|---|
| 20% | TWIST | 32.8 | 52.9 | 66.8 |
| | TWIST+CSC [20] | 34.1 | 53.7 | 67.8 |
| 100% | Our Baseline | 36.8 | 57.2 | 71.8 |
| | GSPN [53] | 19.3 | 37.8 | 53.4 |
| | MTML [25] | 20.3 | 40.2 | 55.4 |
| | PointGroup [24] | 34.8 | 56.9 | 71.3 |
| | 3D-MPA [11] | 35.3 | 59.1 | 72.4 |
| | DyCo3D [18] | 35.4 | 57.6 | - |

Table 3. Comparisons with approaches that are supervised by 100% of labels on ScanNet validation set. 'L.R.' means the label ratio.

## 5.2. Main Results

We evaluate our TWIST and other recent methods on ScanNet v2 and S3DIS. Table 1 depicts quantitative results on their validation sets, with 1%, 5%, 10% and 20% labeled data for supervision. Table 2 presents online comparison results on the test set of ScanNet v2 data-efficient benchmark[1]. The compared methods include (i) Sup-only baseline use the baseline method in Sec. 3 on the labeled portion of data; (ii) PointContrast [49]; (iii) CSC [20]; (iv) TWIST; and (v) TWIST with CSC-pretrained model as initialization.

As shown in Table 1, TWIST significantly improves the performance over the baselines by leveraging unlabeled data for model training. In contrast to the two SOTA approaches (ii)-(iii) that leverage the unsupervised pre-training, TWIST generates high-quality pseudo labels as explicit supervisions and surpasses them consistently on all metrics.

A noteworthy finding is that, TWIST with the CSC-pretrained weights as initialization in the first round (v) further boosts the performance as shown in the gray rows of Tables 1 and 2, revealing their complementary strength to this task. Specifically, their cooperative performance even doubles the mAP on ScanNet validation set with 1% labels and achieves 34.1% mAP, given only 20% labels. The latter is only 2.7% lower than the fully-supervised baseline and can be compa-

[1] http://kaldir.vc.in.tum.de/scannet_benchmark/data_efficient/

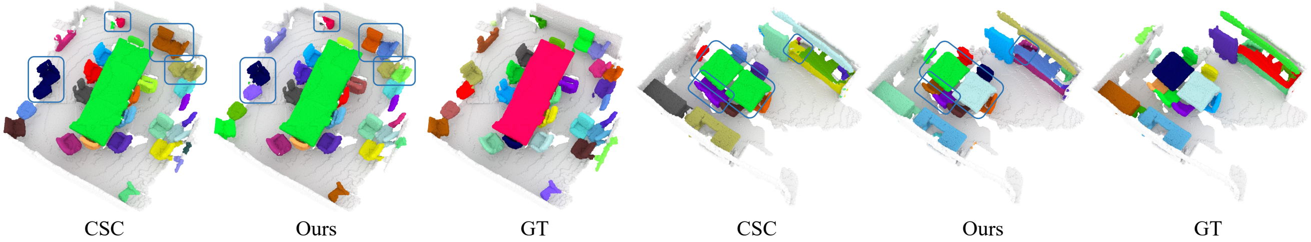

Figure 5. Comparisons of visualization results with CSC (trained with only 5% of labeled scenes). Distinct instances have different colors.

rable to several recent competitive approaches also trained with 100% labels, as shown in Table 3.

Fig. 5 shows the visual comparisons between TWIST and CSC on the ScanNet v2 validation set. With only 5% labeled data for training, TWIST generates clean instance predictions and shows a high capability of separating spatially-close objects, such as similar adjacent chairs.

### 5.3. Ablation Study

We also conduct ablation studies to evaluate the key designs in TWIST. Unless otherwise specified, we evaluate on the ScanNet v2 validation set with 5% training data labeled.

**Effects of different components in TWIST.** To analyze the effect of the core components, we try different combinations and summarize the ablation results in Table 4. We first set the baseline as Group I. Based on it, the improvement is mainly derived from the following three aspects.

- **Pseudo labels.** Groups II and III generate pseudo semantic labels and pseudo offset vectors using the naive thresholding strategy discussed in Sec. 4, improving mAP by +2.7% and +1.5%, respectively. Their joint effect (Group IV) brings further gain (+0.4% mAP).
- **Proposal-based pseudo-label update.** Group V exploits component **C** of TWIST (Fig. 2) to generate proposal-level pseudo labels and preserve pseudo-label consistency inside an instance proposal. Compared to the naive strategy (Group IV) for pseudo-labeling, the strategy enhances the performance by (+3.5% mAP).
- **Re-correction module.** This module contributes to the object-level pseudo-label denoising in two aspects. For instance proposals with instance certainty scores $\hat{v}_j$ lower than the threshold, we filter them out. Otherwise, we rectify their semantic categories by re-prediction. The generated pseudo labels thus become more reliable (+2.2% mAP), as observed in the last two rows.

**Effects of mutual enhancement.** TWIST enables inter-label mutual enhancement by semantic-guided proposal generation and proposal-based pseudo-label update (see Sec. 4.3). So, we disable two-way enhancement by altering either of the above modules and show ablation results in Table 5.

| Group | $\tilde{S}^u$ | $\tilde{O}^u$ | *(C)* | *(B)* | mAP | AP$_{50}$ | AP$_{25}$ |
|---|---|---|---|---|---|---|---|
| I | | | | | 18.2 | 32.0 | 47.0 |
| II | ✓ | | | | 20.9 | 36.4 | 50.1 |
| III | | ✓ | | | 19.7 | 35.2 | 48.6 |
| IV | ✓ | ✓ | | | 21.3 | 36.8 | 50.2 |
| V | ✓ | ✓ | ✓ | | 24.8 | 41.2 | 53.6 |
| VI | ✓ | ✓ | ✓ | ✓ | **27.0** | **44.1** | **56.2** |

Table 4. Effects of different components of TWIST. $\tilde{S}^u$ and $\tilde{O}^u$ denote the pseudo semantic label and pseudo offset vector, respectively. *(C)* denotes the proposal-based pseudo-label update and *(B)* is the re-correction module (refer to modules **B** and **C** in Fig. 2).

| *sem-to-off* | *off-to-sem* | mAP | AP$_{50}$ | AP$_{25}$ | mIoU |
|---|---|---|---|---|---|
| ☑ | ☒ | 22.5 | 39.4 | 51.7 | 49.4 |
| ☒ | ☑ | 21.7 | 38.7 | 50.5 | 52.0 |
| ☑ | ☑ | **27.0** | **44.1** | **56.2** | **54.9** |

Table 5. Effect of mutual enhancement in TWIST. We disable the inter-label mutual enhancement by blocking either *semantic-to-offset* or *offset-to-semantic* enhancement during self-training.

| Group | *Feature* | *Augmentation* | mAP | AP$_{50}$ | AP$_{25}$ |
|---|---|---|---|---|---|
| I | xyz+rgb | - | 22.3 | 38.0 | 51.8 |
| II | xyz+b.feats. | - | 25.6 | 42.3 | 54.5 |
| III | xyz+b.feats. | trans. | 25.9 | 42.7 | 55.0 |
| IV | xyz+b.feats. | mut. | 26.6 | 43.6 | 55.7 |
| V | xyz+b.feats. | trans.+mut. | **27.0** | **44.1** | **56.2** |

Table 6. Comparison with different input features and augmentation strategies of the re-correction module, where ‘b.feats.’ denotes the backbone feature; ‘trans.’ denotes the geometric transformation; and ‘mut.’ denotes the geometric mutation.

To disable the offset-to-semantic enhancement, we simply produce the pseudo semantic labels by point-wise confidence thresholding, instead of adopting the proposal-based pseudo-label update. For fair comparison, the generation strategy of the pseudo offset vector (Eqs. (7) and (8)) remains unchanged. As observed in rows 1 and 3, it yields inferior semantic predictions (-5.5% mIoU) and dramatic drop on instance segmentation accuracy (-4.5% mAP).

On the other hand, to block the semantic-to-offset enhancement, we change the clustering algorithm in the instance proposal generation procedure to a class-agnostic

| *Round* | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1% | 9.8 | 14.1 | 15.6 | 16.5 | 17.0 | 17.1 |
| 5% | 32.0 | 38.7 | 42.3 | 43.7 | 44.1 | 44.1 |
| 10% | 42.8 | 45.8 | 48.3 | 49.5 | 49.7 | 49.7 |
| 20% | 47.9 | 51.2 | 52.0 | 52.8 | 52.9 | 52.9 |

Table 7. Performance at each self-training round on ScanNet v2, given {1%, 5%, 10%, 20%} ratios of labeled data. The round 0 denotes the "Sup-only" baseline method. $AP_{50}$ = evaluation metric.

mechanism. As observed in Table 5 rows 2 and 3, despite the descent performance of semantic segmentation, its effect on guiding instance generation is disabled, thus resulting in 5.3% mAP decrease in instance segmentation.

**Ablations for the re-correction module.** To further explore impact of the re-correction module, we conduct experiments among the different settings shown in Table 6.

Regarding the input network features, we observe obvious performance gain (+3.3% mAP) with the features extracted from the backbone, *i.e.*, the output of Sparse U-Net of model $\Phi$, in comparison with the RGB features. It can be attributed to the captured contextual information in the network feature derived by the feature aggregation operation, which allows the re-correction module to leverage neighboring environment and boost 3D object recognition.

Besides, we compare the effect of different data augmentation strategies. Comparing II and III, we find that simple geometric transformation, such as rotation, contributes minor improvement. By including and removing some adjacent points for geometric mutation, the instance proposals are enhanced with more variants. Group IV validates the effectiveness of our proposed geometric mutation strategy and Group V further enhances the performance by combining it with common augmentation.

Note that we only employ the re-correction module for proposal evaluation/rectification at the *pseudo-label generation stage*. It can also be used at test time to rectify the semantic class of instance proposals onsite. In another comparison, we add the re-correction module to both the baseline ("Sup-only" in Table 1) and TWIST during the inference. It brings +2.5% mAP over the baseline (still 6.3% mAP lower than TWIST) but only +0.2% mAP over TWIST. This result manifests that TWIST has been well optimized by the pseudo labels that are consistent with the re-correction module output. Thus, we do not need this module at test time for computational efficiency.

**Ablations on different self-training rounds.** TWIST typically converges in 3-4 rounds. The quality of the two sets of pseudo labels (Fig. 4(a)) and model performance (Table 7) can both be gradually improved iteratively.

**Separate semantic and instance improvement.** Our method benefits both 3D semantic and instance segmentation in the semi-supervised setting. In Table 8, we show improvement in each task and compare with SOTA pre-training approach CSC [20]. Our approach outperforms CSC with all data split settings (+1.6%∼+5.8% mIoU) on semantic segmentation task, thanks to the pseudo semantic labels.

| Split | Method | On Sem. | | On Ins. | |
|---|---|---|---|---|---|
| | | mIoU | mAcc | mAP | $AP_{50}$ |
| 1% | CSC [20] | 29.3 | 40.0 | 7.1 | 13.0 |
| | TWIST | 31.0 (+1.7) | 41.2 (+1.2) | 9.0 (+1.9) | 16.1 (+3.1) |
| 5% | CSC [20] | 49.1 | 57.2 | 20.9 | 36.7 |
| | TWIST | 54.9 (+5.8) | 63.6 (+6.4) | 24.0 (+3.1) | 40.7 (+4.0) |
| 10% | CSC [20] | 59.5 | 69.3 | 27.3 | 45.0 |
| | TWIST | 61.1 (+1.6) | 70.6 (+1.3) | 29.2 (+1.9) | 47.4 (+2.4) |
| 20% | CSC [20] | 64.1 | 73.1 | 30.6 | 50.3 |
| | TWIST | 66.5 (+2.4) | 74.7 (+1.6) | 31.8 (+1.2) | 51.6 (+1.3) |

Table 8. Performance improvement on both semantic segmentation and instance segmentation tasks compared with CSC.

In the instance segmentation task, as our semantic branch yields higher accuracy over CSC, for fair comparison on the instance clustering quality, we discard our semantic results and directly replace it with the semantic predictions from CSC. As presented in the gray regions, TWIST still achieves better improvement with the help of pseudo offset supervision for instance clustering.

## 6. Conclusion

We have presented TWIST, a new self-training-based framework for semi-supervised 3D instance segmentation. TWIST considers two kinds of pseudo labels for providing semantic- and instance-level supervisions on unlabeled data to effectively enhance the model training. Our three proposed modules in TWIST work hand-in-hand to improve the pseudo-label accuracy, through the object-level denoise and the two-way inter-label mutual enhancement. With the enhanced pseudo labels, TWIST outperforms existing 3D pre-training approaches and demonstrates its complementary strength, since TWIST can cooperate with the existing 3D pre-training approaches to further boost performance.

Our approach does not generate pseudo labels for poor-quality instance proposals. For some "hard" objects, their instance proposals always have low quality, even when predicted by strong fully-supervised networks with 100% labels. So, these objects may not receive pseudo supervisions in self-training. In future work, we will recall the instance proposals to the pseudo-labeling process and further facilitate label-efficient 3D instance segmentation.

**Acknowledgements.** This work is supported in part by the Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), HKU Startup Fund, HKU Seed Fund for Basic Research, SmartMore donation fund, and project MMT-p2-21 of the Shun Hing Institute of Advanced Engineering, CUHK.

## References

[1] Tarvainen Antti and Valpola Harri. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

[2] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016.

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 2019.

[4] Yang Bo, Wang Jianan, Clark Ronald, Hu Qingyong, Wang Sen, Markham Andrew, and Trigoni Niki. Learning object bounding boxes for 3d instance segmentation on point clouds. In *NeurIPS*, pages 6737–6746, 2019.

[5] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *ICCV*, pages 15467–15476, 2021.

[6] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang. Sspc-net: Semi-supervised semantic 3d point cloud segmentation network. *arXiv preprint arXiv:2104.07861*, 2021.

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

[8] Ruihang Chu, Yukang Chen, Tao Kong, Lu Qi, and Lei Li. Icm-3d: Instantiated category modeling for 3d instance segmentation. *IEEE Robotics and Automation Letters*, 7(1):57–64, 2021.

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017.

[10] Shuang Deng, Qiulei Dong, and Bo Liu. Scss-net: Superpoint constrained semi-supervised segmentation network for 3d indoor scenes. *arXiv preprint arXiv:2107.03601*, 2021.

[11] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *CVPR*, pages 9031–9040, 2020.

[12] Matheus Gadelha, Aruni RoyChowdhury, Gopal Sharma, Evangelos Kalogerakis, Liangliang Cao, Erik Learned-Miller, Rui Wang, and Subhransu Maji. Label-efficient learning on point clouds using approximate convex decompositions. In *ECCV*, pages 473–491. Springer, 2020.

[13] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018.

[14] Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

[15] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *CVPR*, pages 2940–2949, 2020.

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

[17] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *ICCV*, pages 6930–6940, 2021.

[18] Tong He, Chunhua Shen, and Anton van den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *CVPR*, pages 354–363, 2021.

[19] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *CVPR*, pages 4421–4430, 2019.

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

[21] Qingyong Hu, Bo Yang, Guangchi Fang, Yulan Guo, Ales Leonardis, Niki Trigoni, and Andrew Markham. Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds with 1000x fewer labels. *arXiv preprint arXiv:2104.04891*, 2021.

[22] Haiyong Jiang, Feilong Yan, Jianfei Cai, Jianmin Zheng, and Jun Xiao. End-to-end 3d point cloud instance segmentation without detection. In *CVPR*, pages 12796–12805, 2020.

[23] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6423–6432, 2021.

[24] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *CVPR*, pages 4867–4876, 2020.

[25] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *ICCV*, pages 9256–9266, 2019.

[26] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1205–1214, 2021.

[27] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, page 896, 2013.

[28] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *ICCV*, pages 2783–2792, 2021.

[29] Jinxian Liu, Minghui Yu, Bingbing Ni, and Ye Chen. Self-prediction for joint instance and semantic segmentation of point clouds. In *ECCV*, pages 187–204, 2020.

[30] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. One thing one click: A self-training approach for weakly supervised 3d semantic segmentation. In *CVPR*, pages 1726–1736, 2021.

[31] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Yunde Jia, and Luc Van Gool. Towards a weakly supervised framework for 3d point cloud object detection and annotation. *TPAMI*, 2021.

[32] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object detection from lidar point cloud. In *ECCV*, pages 515–531, 2020.

[33] Sanjeev Muralikrishnan, Vladimir G Kim, and Siddhartha Chaudhuri. Tags2parts: Discovering semantic regions from shape tags. In *CVPR*, pages 2926–2935, 2018.

[34] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *CVPR*, pages 8827–8836, 2019.

[35] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019.

[36] Zengyi Qin, Jinglu Wang, and Yan Lu. Weakly supervised 3d object detection from point clouds. In *ACM MM*, pages 4144–4152, 2020.

[37] Zhongzheng Ren, Ishan Misra, Alexander G Schwing, and Rohit Girdhar. 3d spatial recognition without spatially labeled 3d. In *CVPR*, pages 13204–13213, 2021.

[38] Laine Samuli and Aila Timo. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[39] Nasim Souly, Concetto Spampinato, and Mubarak Shah. Semi supervised semantic segmentation using generative adversarial network. In *ICCV*, pages 5688–5696, 2017.

[40] Yew Siang Tang and Gim Hee Lee. Transferable semi-supervised 3d object detection from rgb-d data. In *ICCV*, pages 1931–1940, 2019.

[41] An Tao, Yueqi Duan, Yi Wei, Jiwen Lu, and Jie Zhou. Seggroup: Seg-level supervision for 3d instance and semantic segmentation. *arXiv preprint arXiv:2012.10217*, 2020.

[42] He Wang, Yezhen Cong, Or Litany, Yue Gao, and Leonidas J Guibas. 3dioumatch: Leveraging iou prediction for semi-supervised 3d object detection. In *CVPR*, pages 14615–14624, 2021.

[43] Haiyan Wang, Xuejian Rong, Liang Yang, Shuihua Wang, and Yingli Tian. Towards weakly supervised semantic segmentation in 3d graph-structured point clouds of wild scenes. In *BMVC*, page 284, 2019.

[44] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, pages 2569–2578, 2018.

[45] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *CVPR*, pages 4096–4105, 2019.

[46] Jiacheng Wei, Guosheng Lin, Kim-Hui Yap, Tzu-Yi Hung, and Lihua Xie. Multi-path region mining for weakly supervised 3d semantic segmentation on point clouds. In *CVPR*, pages 4384–4393, 2020.

[47] Benjamin Wilson, Zsolt Kira, and James Hays. 3d for free: Crossmodal transfer learning using hd maps. *arXiv preprint arXiv:2008.10592*, 2020.

[48] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10687–10698, 2020.

[49] 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 *ECCV*, 2020.

[50] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *CVPR*, pages 13706–13715, 2020.

[51] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

[52] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *CVPR*, pages 10368–10378, 2021.

[53] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *CVPR*, pages 3947–3956, 2019.

[54] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. In *CVPR*, pages 8883–8892, 2021.

[55] Yachao Zhang, Zonghao Li, Yuan Xie, Yanyun Qu, Cuihua Li, and Tao Mei. Weakly supervised semantic segmentation for large-scale point cloud. In *AAAI*, pages 3421–3429, 2021.

[56] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Sess: Self-ensembling semi-supervised 3d object detection. In *CVPR*, pages 11079–11087, 2020.

[57] Chenyang Zhu, Kai Xu, Siddhartha Chaudhuri, Li Yi, Leonidas J Guibas, and Hao Zhang. Adacoseg: Adaptive shape co-segmentation with group consistency loss. In *CVPR*, pages 8543–8552, 2020.

[58] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. *NeurIPS*, 2020.

[59] Longkun Zou, Hui Tang, Ke Chen, and Kui Jia. Geometry-aware self-training for unsupervised domain adaptation on object point clouds. In *ICCV*, pages 6403–6412, 2021.

[60] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018.

[61] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, pages 5982–5991, 2019.