sensors

MDPI

Article

Point Cloud Instance Segmentation with Inaccurate Bounding-Box Annotations

Yinyin Peng [1], Hui Feng [1], Tao Chen [1] and Bo Hu [1,2,*]

[1] Department of Electronic Engineering, Fudan University, Shanghai 200433, China
[2] Yiwu Research Institute, Fudan University, Yiwu 322000, China
* Correspondence: bohu@fudan.edu.cn

Abstract: Most existing point cloud instance segmentation methods require accurate and dense point-level annotations, which are extremely laborious to collect. While incomplete and inexact supervision has been exploited to reduce labeling efforts, inaccurate supervision remains under-explored. This kind of supervision is almost inevitable in practice, especially in complex 3D point clouds, and it severely degrades the generalization performance of deep networks. To this end, we propose the first weakly supervised point cloud instance segmentation framework with inaccurate box-level labels. A novel self-distillation architecture is presented to boost the generalization ability while leveraging the cheap but noisy bounding-box annotations. Specifically, we employ consistency regularization to distill self-knowledge from data perturbation and historical predictions, which prevents the deep network from overfitting the noisy labels. Moreover, we progressively select reliable samples and correct their labels based on the historical consistency. Extensive experiments on the ScanNet-v2 dataset were used to validate the effectiveness and robustness of our method in dealing with inexact and inaccurate annotations.

Keywords: point cloud instance segmentation; learning with noisy labels; weakly supervised learning; self-distillation

Citation: Peng, Y.; Feng, H.; Chen, T.; Hu, B. Point Cloud Instance Segmentation with Inaccurate Bounding-Box Annotations. *Sensors* **2023**, *23*, 2343. https://doi.org/10.3390/s23042343

Academic Editors: Miaohui Wang, Guanghui Yue, Jian Xiong and Sukun Tian

Received: 12 January 2023
Revised: 15 February 2023
Accepted: 17 February 2023
Published: 20 February 2023

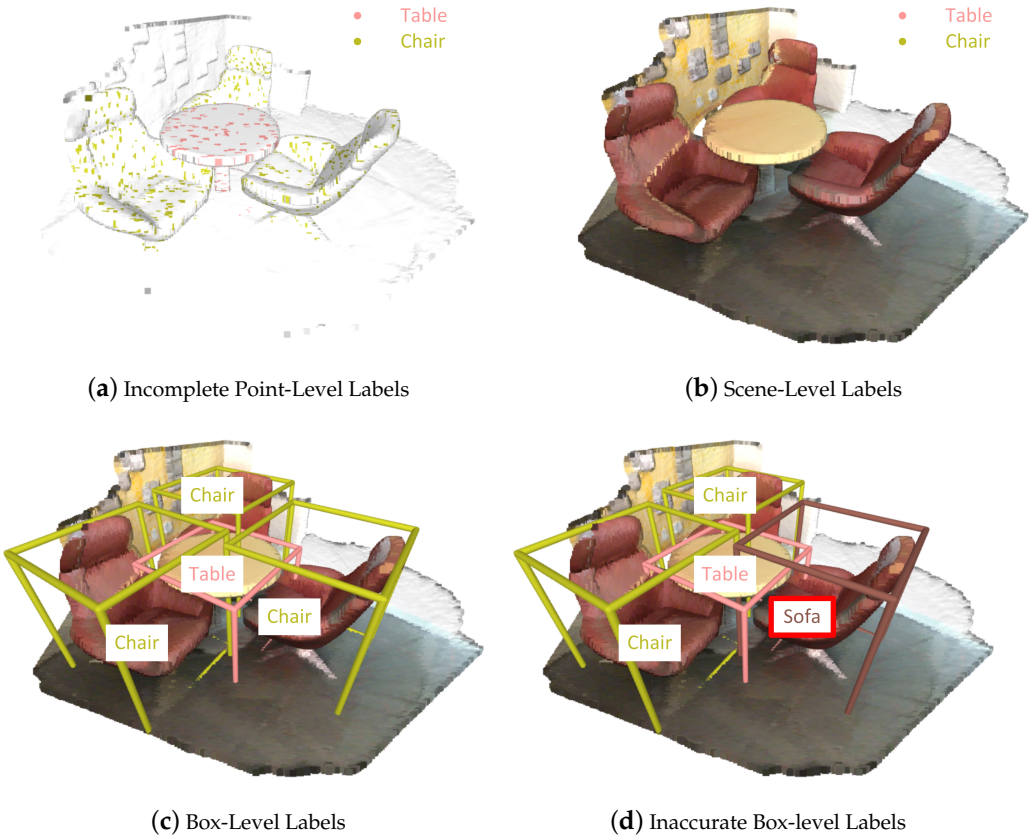
1. Introduction

The rapid development of 3D sensors, such as LiDARs and RGB-D cameras, has brought about an increasing amount of 3D data, thus promoting a wide range of applications, including autonomous driving [1], robotics [2], and medical treatment [3]. With the benefit of rich geometric information and the challenge of intrinsic irregularity, more and more attention has been paid to deep learning on 3D point clouds [4–7].

As one of the fundamental tasks in 3D scene understanding, point cloud instance segmentation aims to predict the semantic label of each point and simultaneously distinguish points within the same class but in different instances. Numerous deep learning methods have been proposed to achieve progressively better performance [8–14]. However, the success of most existing segmentation methods depends heavily on accurately and densely annotated training data, which are time-consuming to collect. For example, it takes about 22.3 min to annotate all of the points of one scene in ScanNet [15]. To alleviate the point-level annotation burden of full supervision, a handful of methods have recently taken weak supervision into consideration; they mainly included incomplete [16–18] and inexact supervision [19–22]. The different kinds of weak supervision are illustrated in Figure 1.

For incomplete supervision, current research works perform semi-supervised learning through self-training, self-supervision, label propagation, etc. However, the way of choosing the small fraction of points to annotate is crucial for the segmentation performance. To represent the location of an instance, SegGroup [18] picks the largest segment, and CSC [23] finds exemplary points through active sampling. In other words, additional labeling efforts are required to implement point cloud instance segmentation with incomplete supervision.

For inexact supervision, there exist two leading types, i.e., scene-level (subcloud-level) and box-level supervision. Since it is difficult to extract object localization information from scene-level or subcloud-level tags [19], we focus on box-level supervision, which is of medium granularity and widely available. The key is to identify the foreground points in each bounding box without point-level instance labels. Box2Seg [21] uses attention map modulation and entropy minimization to generate pseudo-labels. SPIB [22] first conducts object detection with partial bounding-box labels and fulfills instance segmentation with three in-box refinement modules. Both of them need multi-stage training, and the box-level annotations are not fully exploited for instance segmentation. Box2Mask [20] allows each point to predict the box in which it belongs and trains the instance segmentation network from end to end with bounding-box annotations.



(**a**) Incomplete Point-Level Labels (**b**) Scene-Level Labels

(**c**) Box-Level Labels (**d**) Inaccurate Box-level Labels

Figure 1. Illustration of various weak supervision methods for point cloud segmentation. (**a**) Incomplete point-level labels denote the classes to which a small fraction of points belong. (**b**) Scene-level (subcloud-level) labels indicate all of the classes appearing in the scene (subcloud). (**c**) Box-level labels indicate the class and location of each object. (**d**) Inaccurate box-level labels indicate the portion of boxes that are mislabeled. For example, a "chair" is mislabelled as a "sofa".

Nonetheless, the methods presented above implicitly assume that the labels are highly accurate, which may not be guaranteed in practice. Regardless of the granularity at which data are labeled, label noise exists due to the carelessness of annotators and the difficulty of annotating itself. When it comes to box-level label noise, Hu et al. [24] designed a noise-resistant focal loss for 2D object detection. With NLTE [25], it was found that it was essential for domain adaptive objective detection to address noisy box annotations, including miss-annotated boxes and class-corrupted ones. In 3D point cloud instance segmentation, the rough location information of most points is unaffected by slight fluctuations in box coordinates, while mislabeling the box semantics can lead to serious confusion of all of the in-box points. Therefore, we took the semantic label noise of each box into account while leaving the geometric coordinate noise for future work. The inaccurate box-level

annotations are shown in Figure 1d. Since deep neural networks are highly capable of learning any complex function, it is easy to overfit inaccurate labels and reduce the generalization performance [26]. Thus, it is necessary to develop a noise-robust point cloud instance segmentation method. A recent work used PNAL [27] to study point noise in semantic segmentation, but it heavily relied on the early memorization effect, which increased the risk of discarding hard samples or those in the minor class. Furthermore, point cloud instance segmentation with inaccurate bounding-box annotations is even more challenging due to the granularity mismatch of given annotations and the target task. There is an urgent need to combat realistic label noise and sufficiently release the potential of box-level supervision in point cloud instance segmentation.

Extensive research has empirically demonstrated the success of knowledge distillation [28] in boosting the generalization ability, which is in great demand when learning with noisy labels. Traditional knowledge distillation transfers knowledge from a large teacher model, while self-distillation efficiently utilizes knowledge from itself and, thus, attracts more and more attention. As for theoretical analysis, there are various opinions that include label smoothing regularization [29], the multi-view hypothesis [30], and loss landscape flattening [31]. Similarly to our method, PS-KD [32] trained a model with soft targets, which were a weighted summation of the hard targets and the last-epoch predictions, and DLB [33] used predictions from the last iteration as soft targets. However, we considered the entire prediction history and maintained an exponential moving average of the predictions.

In this paper, we present a novel self-distillation framework based on perturbation and history (SDPH) to handle the challenge of point cloud instance segmentation with only inaccurate box annotations. Rather than distilling knowledge from a cumbersome teacher model or an extra clean dataset [34], we perform self-distillation by taking full advantage of self-supervision in the data and the learning process. To be specific, we assume that the predictions over the input point cloud are perturbation-invariant. Both geometric and semantic consistency regularization terms are included to provide additional supervision signals. Furthermore, by investigating the consistency of historical predictions, the model is able to locate and correct refurbishable samples with high precision. Finally, we apply temporal consistency regularization to fully utilize the history information and reduce the unstable prediction fluctuations that may hinder the label refurbishment. In a word, we utilize two kinds of consistency regularization to prevent the network from overfitting inaccurate labels and progressively correct the labels during the training process.

Overall, the main contributions of our paper are summarized as follows:

- To the best of our knowledge, this is the first work to simultaneously explore inexact and inaccurate annotations in the point cloud instance segmentation task.
- We propose a novel self-distillation framework for applying consistency regularization and label refurbishment by using data perturbation and history information.
- Extensive experiments were conducted to demonstrate the effectiveness of our method. The results on ScanNet-v2 show that our SDPH achieved comparable performance to that of densely and accurately supervised methods.

The rest of this paper is organized as follows. First, related research is described in Section 2. Next, we present our self-distillation framework in Section 3. Thereafter, the experimental results and analysis are provided in Section 4. Finally, Section 5 concludes the paper and points out future work.

2. Related Works

2.1. Point Cloud Instance Segmentation

Point cloud instance segmentation methods can be roughly divided into two categories: proposal-based methods and proposal-free methods.

2.1.1. Proposal-Based Methods

Proposal-based methods first conduct object detection to generate region proposals and then perform binary classification to separate all of the foreground points in each proposal. GSPN [8] used an analysis-by-synthesis strategy to enforce geometric understanding in generating proposals with high objectness. These object proposals were further processed by Region-Based PointNet (R-PointNet) to obtain the final segmentation results. The method of 3D-SIS [35] first extracted 2D features from multi-view high-resolution RGB images and then projected them back to the associated 3D voxel grids. The geometry and color features were concatenated and fed into a fully convolutional 3D architecture. The method of 3D-BoNet [36] is a single-stage, anchor-free, and end-to-end trainable network. This method directly predicts a fixed number of bounding boxes and fuses the global information into a point mask prediction branch. The method of 3D-MPA [9] generates proposals through center voting, refines them by using a graph convolutional network, and obtains the final instances through proposal aggregation instead of non-maximum suppression.

2.1.2. Proposal-Free Methods

Proposal-free methods focus on discriminative point feature learning and distinguish instances with the same semantic meaning through clustering. SGPN [11] first embeds all of the input points into feature space and then groups the points into instances based on the pairwise feature similarity, which is not scalable. JSIS3D [12] utilizes a multi-value conditional random field model to jointly optimize semantic labels and instance embeddings predicted by a multi-task point-wise network. ASIS [37] utilizes discriminative loss to pull embeddings of the same instance to its center and push those of different instances apart. Moreover, the association of instance segmentation and semantic segmentation further benefits each. PointGroup [38] predicts point offsets towards their respective instance centers and considers both the original point coordinates and the offset-shifted ones in the clustering stage. OccuSeg [13] introduces the occupancy signal to take part in multi-task learning and guide graph-based clustering. PE [14] encodes each point as a tri-variate normal distribution in the probabilistic embedding space, and a novel loss function that benefits both semantic segmentation and subsequent clustering was proposed. HAIS [10] performs point aggregation and set aggregation to progressively generate instance proposals. SoftGroup [39] groups points based on soft semantic scores to avoid error propagation and suppresses false positive instances by learning to categorize them as the background.

We follow the proposal-free approach because of its superior performance and flexible architecture. Nevertheless, we utilize inaccurate box-level supervision to learn point-level instance segmentation, which greatly alleviates the labeling cost.

2.2. Weakly Supervised Point Cloud Segmentation

Generally speaking, there are three typical types of weak supervision in machine learning: incomplete supervision, inexact supervision, and inaccurate supervision [40].

Most point cloud segmentation methods are concerned with incomplete supervision, where only a small subset of training data are given with labels [16,17,41–43]. This setting is also known as semi-supervised learning. Xu et al. [16] combined multi-instance learning, self-supervision, and smoothness constraints to achieve semantic segmentation with only 10 times fewer labels. Zhang et al. [41] constructed a self-supervised pre-training task through point cloud colorization and proposed an efficient sparse label propagation mechanism to improve the effectiveness of the weakly supervised semantic segmentation task. PSD [42] enforced the prediction consistency between the perturbed branch and the

original branch, and a context-aware module for regularizing the affinity correlation of labeled points was presented. Liu et al. [17] adopted a self-training approach with a super-voxel graph propagation module. Similarly, SSPC-Net [43] built super-point graphs for dynamic label propagation and the coupled attention mechanism to extract discriminative contextual features.

Inexact supervision means that the training data are given with only coarse-grained labels, such as scene-level tags [19,44] and box-level annotations [20–22] in the segmentation context. MPRM [19] applied various attention mechanisms to acquire point class activation maps (PCAMs). After generating pseudo-point-level labels from PCAMs, a segmentation network could be trained in a fully supervised manner. WyPR [44] jointly performed semantic segmentation and object detection through a series of self- and cross-task consistency losses with multi-instance learning objectives. SPIB [22] first leveraged partially labeled bounding boxes to train a proposal generation network with perturbation consistency regularization and then predicted the instance mask inside each target box with three smoothness regularization and refinement modules. Box2Seg [21] learned pseudo-labels from bounding-box-level foreground annotations and subcloud-level background tags, and it achieved semantic segmentation through fully supervised retraining. Box2Mask [20] directly voted for bounding boxes and obtained instance masks via non-maximum clustering.

Inaccurate supervision means that the given labels are not always the ground truth. Although learning from noisy labels with deep neural networks has been explored very much, especially in image classification [45], few researchers have investigated noisy labels with increasing amounts of point cloud data. As the pioneering work in noise-robust point cloud semantic segmentation, PNAL [27] selected reliable points based on their consistency among historical predictions, and it corrected locally similar points with the most likely label, which was voted on in each cluster.

Most of the above weakly supervised methods focused merely on one type of weak supervision. However, the circumstances are usually more complicated in reality, where the label noise in particular is almost inevitable but often ignored. Thus, we consider both inexact and inaccurate supervision and develop a robust point cloud instance segmentation framework with inaccurate box annotations.

3. Our Method

3.1. Overview

The pipeline of our SDPH is depicted in Figure 2. Given a point cloud $\mathcal{P}$ with inaccurate bounding-box annotations, we first assign point-level pseudo-labels based on the spatial inclusion relations between points and boxes. This simple association process allows for a fully supervised training manner. After the label preparation, the backbone network takes a voxelized point cloud as input and produces embeddings for each voxel. To lessen the computational cost, we perform over-segmentation to group voxels into super-voxels. This basic training process will be introduced in Section 3.3. The final instances are obtained through super-voxel-level non-maximum clustering and backward projection.

Apart from the whole forward inference procedure, our self-distillation training framework consists of two main parts that leverage data perturbation and historical information. First, we construct a perturbed branch and keep the prediction consistency between the original branch and the perturbed one. Furthermore, the past predictions are fully exploited to select refurbishable samples and provide soft targets.

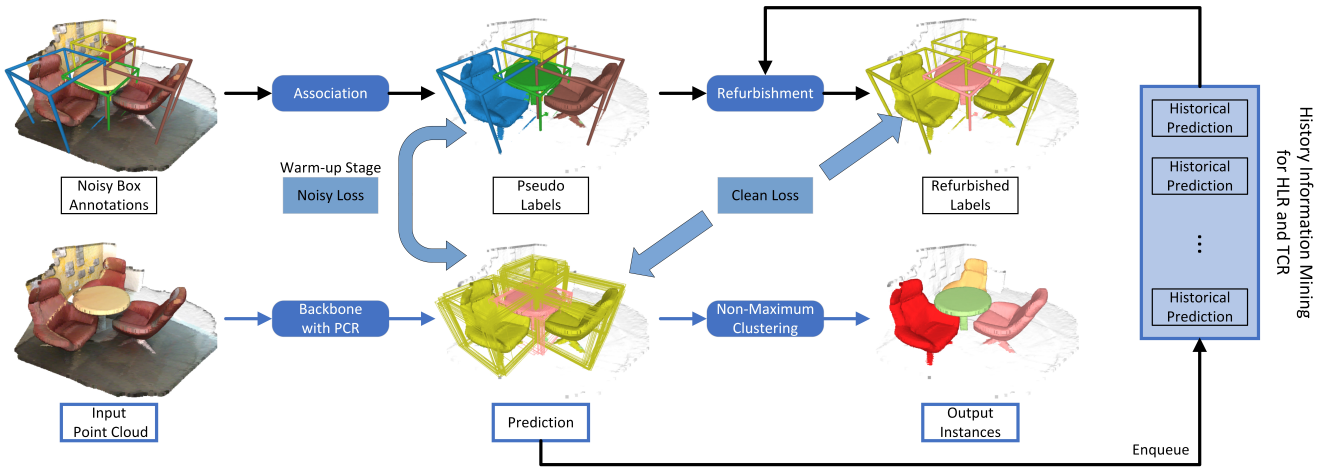


Figure 2. The training framework of self-distillation based on perturbation and history. We first generate pseudo-labels according to the point–box association (c.f. Section 3.2) and train a 3D sparse convolutional network with two types of consistency regularization, namely, PCR (c.f. Section 3.4.1) and TCR (c.f. Section 3.4.3). With the help of regularization, the model is able to perform label refurbishment (HLR, c.f. Section 3.4.2) with higher precision. Note that the noisy loss is used only in the warm-up stage, and afterward, it is replaced by the clean loss, since the cleaned (i.e., refurbished) labels are available.

3.2. Pseudo-Label Generation

Since ground-truth point-level labels are not available, directly training a segmentation network with only box-level labels is infeasible. Therefore, we need to establish the box–point association first. Specifically, we categorize points according to the numbers of boxes containing them. If a point is contained in only one box, it is simply labeled as the unique box, which is represented by both the geometric coordinates and the semantic category. If a point is inside more than one box, the smallest one is associated with it. A point is treated as background if it is outside all of the boxes.

Let $\mathcal{B}$ denote a set of box annotations, with each box $\mathbf{b} \in \mathbb{R}^7$ representing its three-dimensional center, three-dimensional size, and one-dimensional semantic label. For clarity, we use $\mathbf{p}_i \in \mathbf{b}_j$ ($\mathbf{p}_i \notin \mathbf{b}_j$) to show that the i-th point is (not) contained by the j-th box. The pseudo-labels are generated through the following mapping function.

$$\phi(\mathbf{p}_i) = \begin{cases} \mathbf{b}_j, & j = \underset{j \in \{k | \mathbf{p}_i \in \mathbf{b}_k\}}{\arg\min}\ \text{sizeof}(\mathbf{b}_j), \\ \text{background}, & \forall j, \mathbf{p}_i \notin \mathbf{b}_j. \end{cases} \tag{1}$$

Although this mapping function seems plausible, the generated point-level pseudo-labels inevitably suffer from inaccurate associations, as do the super-voxel-level pseudo-labels. The label quality will further degrade due to inaccurate box annotations, which motivated us to design a noise-robust self-distillation training framework.

3.3. Point Cloud Instance Segmentation Network

Before self-distillation, we introduce the basic point cloud instance segmentation network, where the labels are regarded noise-free. As a common choice, we adopted a UNet-like sparse convolutional network as the backbone [10,46,47]. The input point cloud is converted into volumetric grids and then fed into the backbone to extract voxel features, which are pooled into super-voxel features by using the over-segmentation results. Next, multiple output heads are applied to predict the semantic label, the associated box coordinates (offset and size), and the intersection-over-union (IoU) score of the predicted box with the ground-truth box. The basic network is trained with the following multi-task loss.

$$\mathcal{L}_{basic} = \mathcal{L}_{sem} + \mathcal{L}_{offset} + \mathcal{L}_{size} + \mathcal{L}_{score}. \tag{2}$$

Here, $\mathcal{L}_{sem}$ is a normal cross-entropy loss for learning the semantics, which are formulated as

$$\mathcal{L}_{sem} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log p_{ic}, \tag{3}$$

where y_{ic} represents the one-hot semantic label of the i-th super-voxel, $p_{ic} = \frac{\exp(z_{ic})}{\sum_{k=1}^{C}\exp(z_{ik})}$ denotes the probability of being predicted as the c-th category, N is the number of super-voxels, and C represents the number of semantic categories. Note that the background is also included in the categories concerned with $\mathcal{L}_{sem}$.

As for the box regression, we use the L_1 loss.

$$\begin{aligned}\mathcal{L}_{offset} &= \frac{1}{M}\sum_{i=1}^{M}\|\mathbf{d}_i - \hat{\mathbf{d}}_i\|_1,\\ \mathcal{L}_{size} &= \frac{1}{M}\sum_{i=1}^{M}\|\mathbf{s}_i - \hat{\mathbf{s}}_i\|_1,\end{aligned} \tag{4}$$

where M is the number of foreground super-voxels. $\mathbf{d}_i$ and $\hat{\mathbf{d}}_i$ represent the ground-truth and predicted offsets of the i-th super-voxel with respect to the associated box center, respectively. $\mathbf{s}_i$ and $\hat{\mathbf{s}}_i$ represent the corresponding box sizes.

To assist in the later non-maximum clustering and average precision calculation, the IoU score loss is defined as

$$\mathcal{L}_{score} = -\frac{1}{M}\sum_{i=1}^{M}[u_i\log v_i + (1-u_i)\log(1-v_i)], \tag{5}$$

where u_i and v_i represent the true and predicted IoUs between the predicted box and the associated ground-truth box, respectively.

At the inference stage, we follow Box2Mask [20] in performing non-maximum clustering (NMC), which follows exactly the same procedure of non-maximum suppression (NMS) in object detection. Instead of dropping redundant boxes, in NMC, they are collected to form clusters with the corresponding representative boxes. The semantic category of each cluster is assigned through a majority vote. Finally, the clustering structure of super-voxels is projected back to points, which completes the instance segmentation.

3.4. Self-Distillation Based on Perturbation and History

3.4.1. Perturbation-Based Consistency Regularization

Since the original supervision method is inaccurate and untrustworthy, we turn to self-supervision, which has shown great power in deep learning. To provide additional supervision, we construct a perturbed branch and constrain the predictions of the perturbed and original branches to be consistent.

We adopt three kinds of perturbation strategies: scaling, flipping, and rotation. For scaling, we sample a scaling factor ξ from a uniform distribution $\mathcal{U}(0.8, 1.2)$. The origin-centered scaling process is represented as $\tilde{P} = \xi \cdot P$, where $P \in \mathbb{R}^{N_p \times 3}$ is the coordinate matrix of the input point cloud and $\tilde{P}$ is the transformed one. For flipping, we randomly sample the flipping indicators f_x, f_y from $\{-1, 1\}$, where -1 means flipping over the corresponding axis. Thus, the flipping can be expressed as $\tilde{P} = P \cdot \mathrm{diag}(f_x, f_y, 1)$. For rotation, the rotation angle θ around z-axis is denoted as θ_z and sampled from the uniform distribution $\mathcal{U}(0, 2\pi)$. Rotating the point cloud means multiplying its coordinates with a rotation matrix as follows:

$$\tilde{P} = P \cdot R(\theta_z) = P \cdot \begin{bmatrix}\cos\theta_z & \sin\theta_z & 0\\ -\sin\theta_z & \cos\theta_z & 0\\ 0 & 0 & 1\end{bmatrix}. \tag{6}$$

Obviously, both semantic and geometric predictions should be consistent between the two branches, i.e., the perturbation-based consistency regularization loss ("PCR loss" in Figure 3) is defined as

$$\mathcal{L}_{pcr} = \mathcal{L}_{pcr}^{sem} + \mathcal{L}_{pcr}^{geo}. \tag{7}$$

The KL-divergence and MSE losses are used as consistency regularization terms. To be specific, we formulate the semantic consistency loss as

$$\begin{aligned} \mathcal{L}_{pcr}^{sem} &= \frac{1}{N}\sum_{i=1}^{N} D_{KL}(\mathbf{p}_i \| \tilde{\mathbf{p}}_i) \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} p_{ic} \log \frac{p_{ic}}{\tilde{p}_{ic}}. \end{aligned} \tag{8}$$

The geometric consistency loss is defined as

$$\mathcal{L}_{pcr}^{geo} = \frac{1}{N}\sum_{i=1}^{N}\left[\|\tilde{\hat{\mathbf{o}}}_i - \hat{\mathbf{o}}_i\|_2^2 + \|\tilde{\hat{\mathbf{s}}}_i - \hat{\mathbf{s}}_i\|_2^2\right], \tag{9}$$

where $\mathbf{o}_i$ represents the center of the i-th super-voxel's associated box. In addition, $\hat{\cdot}$ indicates the predicted value, and $\tilde{\cdot}$ means perturbation. To ensure valid consistency regularization, the same perturbation should be applied to the geometric predictions of the original branch.

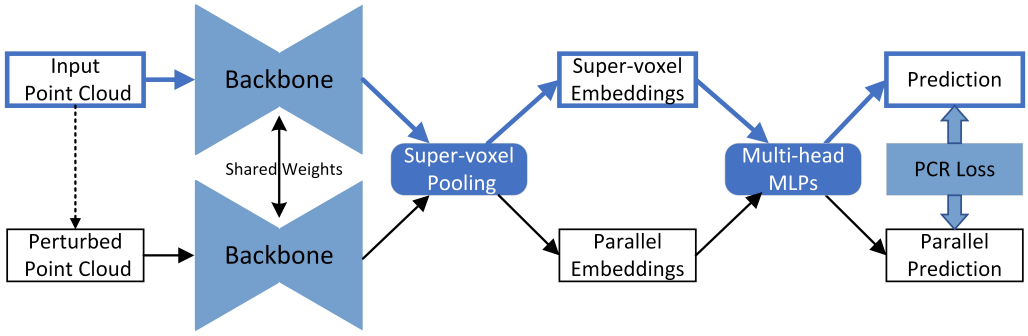


Figure 3. Illustration of the perturbation-based consistency regularization (PCR) module. We construct a parallel branch through data perturbation and force the output predictions of the two branches to be consistent. Note that the predictions include both semantics and geometry.

3.4.2. History-Guided Label Refurbishment

In light of the memorization effect, in which deep networks first learn simple patterns in clean data before memorizing noise by brute force [48], the model is able to identify and correct inaccurate labels by itself during training. Specifically, the consistency of predictions is widely used as a confidence criterion [27,49–51]. Along this line, we consider samples with consistent historical predictions as refurbishable. The refurbishment process is illustrated in Figure 4.

Let $\Psi(q) = \{\hat{y}_{t_1}, \hat{y}_{t_2}, \cdots, \hat{y}_{t_q}\}$ denote the label prediction history of a super-voxel sample, where q is the length of the historical queue. The frequency of the super-voxel being predicted as the c-th category is calculated as

$$F(c|q) = \sum_{i=1}^{q} \frac{[\hat{y}_{t_i} = c]}{q}, \tag{10}$$

where $[\cdot]$ is the Iverson bracket. With the frequency–probability approximation, we apply the following normalized information entropy as the consistency metric:

$$H(q) = \frac{1}{Z}\sum_{c=1}^{C} -F(c|q)\log F(c|q), \tag{11}$$

where $Z = \sum_{c=1}^{C} -\frac{1}{C}\log(\frac{1}{C}) = \log(C)$ is the normalization term representing the maximum entropy. A smaller entropy indicates more consistent predictions. To be concrete, we treat the super-voxel that satisfies $H(q) \leq \epsilon$ $(0 \leq \epsilon \leq 1)$ as the refurbishable sample. The refurbished label is defined as

$$y^* = \underset{1\leq c\leq C}{\operatorname{argmax}} F(c|q). \tag{12}$$

Apparently, the refurbishment will be applied after an appropriate number of warm-up epochs, which is longer than the historical queue. The refurbishable samples are relocated at each new epoch to avoid the accumulation of correction errors. Instead of dropping the remaining samples, we leave them unaffected to enable full exploration of the dataset. In addition, it is noteworthy that we do not impose any restrictions on the label noise, which makes our refurbishment robust to different noise types and different noise rates.

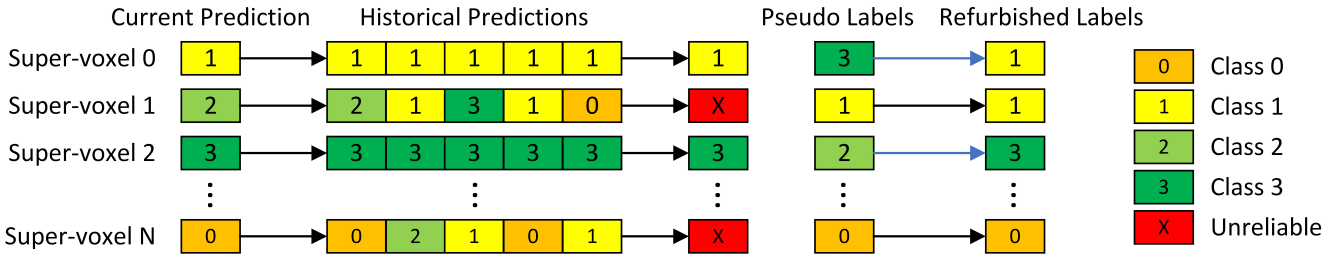


Figure 4. Illustration of the history-guided label refurbishment (HLR) module. We use a historical queue to store the past predictions and correct the previously generated pseudo-labels with consistently predicted classes while keeping the unreliable samples unchanged instead of directly dropping them. Compared with other methods, we take a more conservative strategy, as regularization decreases the overfitting risk.

3.4.3. Temporal Consistency Regularization

The label refurbishment in Section 3.4.2 only utilizes discrete hard labels, overlooking the rich information in the continuous soft distributions. Here, we record the exponential moving average (EMA) of historical logits to impose temporal consistency regularization [32,33].

Let $\mathbf{z}_e$ be the model's output logits at epoch e. After the first trivial epoch, the moving-average logits can be normally updated as

$$\bar{\mathbf{z}}_e = (1-\alpha)\bar{\mathbf{z}}_{e-1} + \alpha\mathbf{z}_e, \tag{13}$$

where α is the weight of the current epoch. In accordance with the conventional practice, we add the temperature τ to further soften the distribution:

$$p_{ic}^{\tau} = \frac{\exp(\mathbf{z}_{ic}/\tau)}{\sum_{k=1}^{C}\exp(\mathbf{z}_{ik}/\tau)}. \tag{14}$$

The temporal consistency regularization term ("TCR loss" in Figure 5) is then defined as

$$\mathcal{L}_{tcr} = \frac{1}{N}\sum_{i=1}^{N}\tau^2 D_{KL}(\bar{\mathbf{p}}_i^{\tau}\|\mathbf{p}_i^{\tau}), \tag{15}$$

where $\bar{\cdot}$ denotes the corresponding EMA version.

With the temporal consistency regularization, the network tries to learn from itself and make comparatively stable predictions, which is important for correcting mislabeled hard samples and promoting the generalization performance.

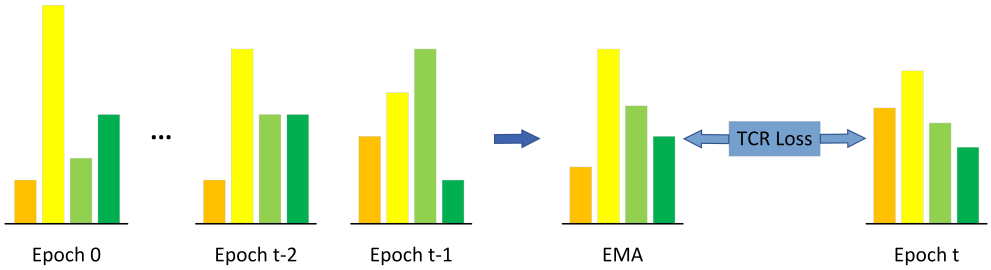


Figure 5. Illustration of the temporal consistency regularization (TCR) module. We record the exponential moving average of the past predicted distributions (logits), which serve as the soft targets for the current prediction.

3.5. Total Loss

Our SDPH can be trained in an end-to-end manner with the total loss $\mathcal{L}$, which contains three parts: the basic loss $\mathcal{L}_{basic}$, the perturbation-based consistency regularization loss $\mathcal{L}_{pcr}$, and the temporal consistency regularization loss $\mathcal{L}_{tcr}$.

$$\mathcal{L} = \mathcal{L}_{basic} + \mathcal{L}_{pcr} + \mathcal{L}_{tcr}, \tag{16}$$

where $\mathcal{L}_{basic}$ is given in Equation (2), $\mathcal{L}_{pcr}$ is given in Equation (7), and $\mathcal{L}_{tcr}$ is given in Equation (15). As we mentioned before, the label refurbishment needs a warm-up stage in which the noisy labels are unchanged in $\mathcal{L}_{basic}$. That is why we call it the "noisy loss" in Figure 2. After the warm-up stage, $\mathcal{L}_{basic}$ is referred to as the "clean loss", since the labels have been cleaned.

4. Experiments

4.1. Experimental Settings

4.1.1. Dataset

We conducted experiments on the widely used ScanNet-v2 [15] dataset. This challenging large-scale indoor point cloud dataset consists of 1201 training scenes, 312 validation scenes, and 100 hidden testing scenes. Each scene of the training and validation sets is richly annotated with point-level semantic-instance labels that are used in the densely supervised methods. However, we created axis-aligned bounding boxes from the point-level annotations to validate our weakly supervised learning framework. To simulate inaccurate annotations, we artificially injected symmetric noise into the training set. Specifically, the semantic labels of the corrupted instance boxes were changed to other labels with equal probability. We used the noise rate, i.e., the probability of each box being mislabeled, to represent the severity of inaccurate supervision. The effects of different noise rates are visualized in Figure 6.

4.1.2. Evaluation Metrics

As with existing methods, we used the mean average precision over 18 foreground object categories as our evaluation metric. To be specific, AP_{25} and AP_{50} denote the scores with IoU thresholds set to 0.25 and 0.5, respectively. In addition, we also report the AP, which averages scores with thresholds varying from 0.5 to 0.95, with a step size of 0.05.

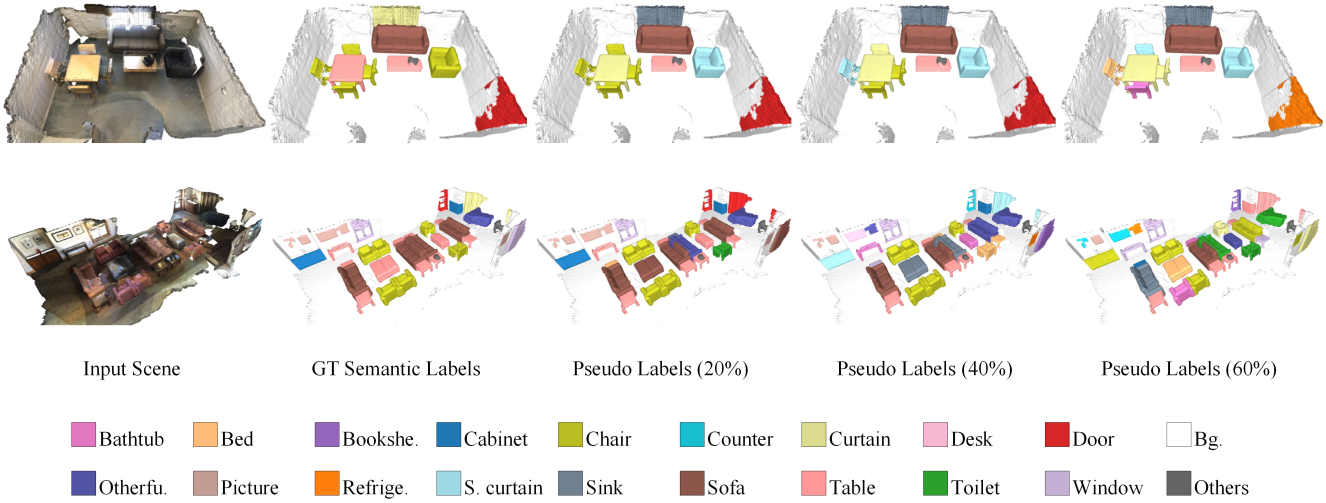


Figure 6. Visualization of different noise rates affecting the semantic labels. From left to right are the input scene, the ground-truth semantics, and the pseudo-labels of noise rates of 20%, 40%, and 60%. The higher the noise rate, the more chaotic the semantics.

4.1.3. Implementation Details

All experiments were performed on a PC with two NVIDIA GeForce RTX 3090 Ti GPUs and an Intel Core i7-12700K CPU. We used two GPUs for distributed training and one for inference. The main software configuration included Python 3.8.13, Pytorch 1.10.2, CUDA 11.3, and MinkowskiEngine 0.5.4. Following the pioneering work of Box2Mask [20], we adopted a six-layer UNet-like sparse convolutional network as our backbone, and the multi-head MLPs were implemented with three layers and 96 hidden units. We set the voxel size to 0.02 m. For history-guided label refurbishment, we set the number of warm-up epochs, the length of the historical queue, and the threshold ϵ to 40, 10, and 0.001, respectively. For temporal consistency regularization, the temperature τ and the EMA coefficient α were empirically set to 3 and 0.9. We trained our network from scratch with a batch size of 4 for 200 epochs in total while using the Adam optimizer with an initial learning rate of 0.001. A cosine annealing scheduler was applied after 100 epochs.

4.2. Instance Segmentation Results

First of all, we conducted comparative experiments with different noise rates to demonstrate the effectiveness of our noise-tolerant learning framework, SDPH. As listed in Table 1, our SDPH achieved consistently better performance than that of Box2Mask (the baseline) under all of the noise rate settings with respect to all of the evaluation metrics. From the overall trend, we observed that higher noise rates were related to larger improvements. When the noise rate is set to 40%, our SDPH still outperformed noise-free Box2Mask in terms of AP. The performance was comparable or even better in terms of AP_{25} and AP_{50} when the noise rate was 20%. These results demonstrate our method's robustness to label noise. Qualitative comparisons of instance and semantic segmentation are shown in Figures 7 and 8, respectively. When training with a noise rate of 40%, our SDPH predicted the semantics more accurately than Box2Mask did, which usually led to better instance segmentation performance.

Table 1. Quantitative comparison of different noise rates on ScanNet-v2.

Method	Metric	0%	10%	20%	30%	40%	50%	60%
Box2Mask [20]	AP	39.1	37.5	36.3	36.3	35.2	33.6	32.0
	AP_{50}	59.7	57.5	55.8	55.4	53.3	50.4	46.7
	AP_{25}	71.8	69.8	68.8	67.3	65.8	62.6	58.2
SDPH	AP	40.1	41.2	40.8	40.0	40.4	37.6	36.5
	AP_{50}	60.4	60.4	60.3	58.7	58.6	55.1	52.5
	AP_{25}	73.0	72.1	71.7	70.7	69.0	65.4	61.9
Improvements	AP	1.0	3.7	4.5	3.7	5.2	4.0	4.5
	AP_{50}	0.7	2.9	4.5	3.3	5.3	4.7	5.8
	AP_{25}	1.2	2.3	2.9	3.4	3.2	2.8	3.7

Even though our method was designed especially for learning with label noise, we acquired a little performance gain in the "noise-free" setting. The possible reasons are two-fold. Firstly, Box2Mask trained the network with associated super-voxel-level pseudo-labels that were not inaccurate. Hence, the label refurbishment worked even without additional noise injection. Secondly, our SDPH benefited from the regularization terms that distilled knowledge from the data and the model itself.

In Table 2, we provide a detailed comparison with state-of-the-art methods that do not explicitly consider label noise. It can be seen that our method performed well in the noise-free setting, which demonstrated the effectiveness of our SDPH. However, instead of attaining consistent performance boosts over different categories, there were some significant declines and increases, especially between SDPH and 3D-MPA [9]. This was probably because 3D-MPA and SDPH adopted different supervision types and instance segmentation routines. The former is a proposal-based method with point-level supervision, while our SDPH is proposal-free and uses the more challenging box-level supervision. As proposal-free methods, PointGroup [38], Box2Mask [20], and SDPH exhibited similar trends when compared with 3D-MPA. For example, their performance greatly declined for refrigerators and shower curtains, and it increases for chairs, desks, sinks, sofas, and other furniture. As shown in Figure 9, the refrigerators had various shapes and sizes and were sometimes surrounded by cabinets. Moreover, curtains were usually beside windows. Even in the case of full point-level supervision—let alone weak box-level supervision—it was difficult to segment them clearly. Furthermore, compared with chairs and desks, there were fewer instances of these categories, which could lead to SDPH's false refurbishment and lower performance.

Table 2. Quantitative comparison with state-of-the-art methods on ScanNet-v2. The highest performance in each column is marked in bold.

Setting	Method	AP_{25}	Bathtub	Bed	Bookshe.	Cabinet	Chair	Counter	Curtain	Desk	Door	Otherfu.	Picture	Refrige.	S. Curtain	Sink	Sofa	Table	Toilet	Window
Full	SegCluster [35]	13.4	16.4	13.5	11.7	11.8	18.9	13.7	12.4	12.2	11.1	12.0	0.0	11.2	18.0	18.9	14.6	13.8	19.5	11.5
	SGPN [11]	22.2	0.0	31.5	13.6	20.7	31.6	17.4	22.2	14.1	16.6	18.6	0.0	0.0	0.0	52.4	40.6	31.9	72.9	15.3
	3D-SIS [35]	35.7	57.6	66.3	16.9	32.0	65.3	22.1	22.6	35.1	26.7	21.1	0.0	28.6	37.2	39.6	56.4	29.4	74.9	10.1
	MTML [52]	55.4	79.4	80.6	45.3	34.6	87.7	9.7	54.2	49.9	45.8	33.5	19.8	44.1	74.9	44.5	80.3	67.4	98.0	47.2
	PointGroup [38]	71.3	86.5	79.5	74.4	67.3	92.5	**64.8**	61.6	74.1	54.8	65.4	**48.2**	38.3	71.1	82.8	85.1	74.2	**100**	**63.6**
	3D-MPA [9]	72.4	**90.3**	83.4	**78.3**	**69.9**	87.6	62.5	**66.0**	69.2	56.6	48.6	48.0	**61.4**	**93.1**	75.2	76.1	74.8	99.2	62.2
Weak	SPIB [22]	61.4	87.4	**86.8**	48.8	45.4	89.0	49.6	47.8	52.3	49.2	45.5	9.9	48.3	82.6	63.2	**88.1**	66.2	95.9	41.9
	Box2Mask [20]	71.8	87.1	83.8	68.2	59.5	94.5	58.5	65.1	78.6	59.8	**67.1**	45.6	46.9	77.4	79.5	87.0	75.5	96.9	61.4
	SDPH	**73.0**	87.1	82.6	73.6	62.1	**95.2**	63.0	61.5	**85.5**	**61.1**	63.1	43.5	46.7	82.0	**85.4**	86.3	**78.2**	98.3	59.3

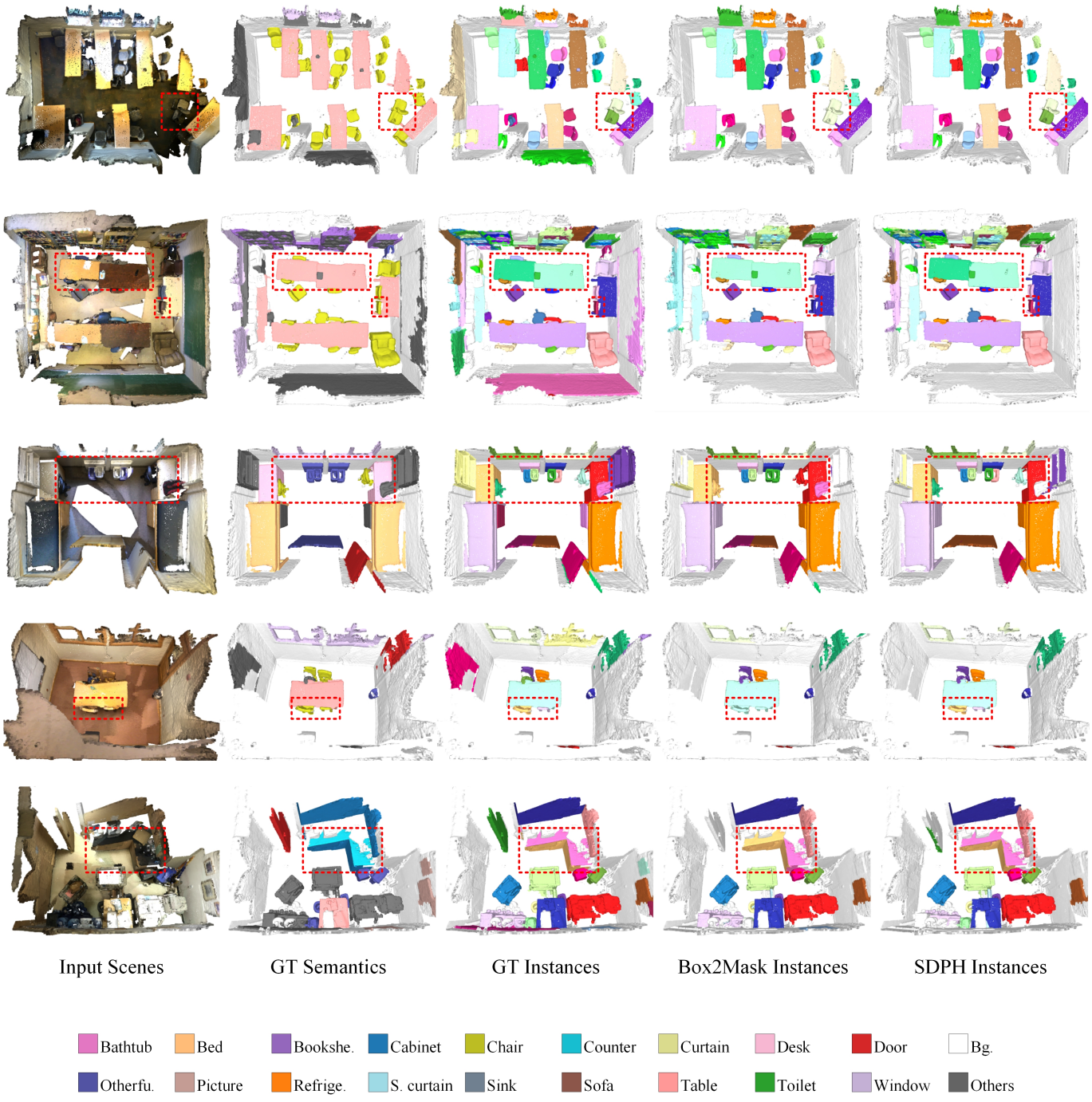


Figure 7. Qualitative comparison at a noise rate of 40% on ScanNet-v2. The legend is employed to distinguish among different semantic meanings, while the individual instances are randomly colored. The key differences are marked out with red dashed rectangles.

Figure 8. Qualitative comparison at a noise rate of 40% on ScanNet-v2. The legend is employed to distinguish among different semantic meanings, and the key differences are marked out with red dashed rectangles.

4.3. Ablation Study

We analyzed the contribution of each component in our learning framework, including perturbation-based consistency regularization (PCR), history-based label refurbishment (HLR), and temporal consistency regularization (TCR). It should be noted that the models in the ablation study were all trained with a noise rate of 40%. The complete ablation results are shown in Table 3. We found that every single component was able to improve the performance by itself. In particular, TCR alone obtained 2.6, 3.4, and 2.0 percent improvements in terms of AP, AP_{50}, and AP_{25}, respectively. The performance could be further boosted through their combination, and the largest increases in AP, AP_{50}, and AP_{25}

reached 5.2, 5.3, and 3.2 by combining all three components. This thorough ablation study demonstrated that each module plays an important role in our framework.

Table 3. Ablation study on ScanNet-v2. The highest performance in each column is marked in bold.

PCR	HLR	TCR	*AP*	AP_{50}	AP_{25}
			35.2	53.3	65.8
✓			37.1	53.7	65.1
	✓		37.6	55.4	66.6
		✓	37.8	56.7	67.8
✓	✓		39.5	58.1	67.9
✓		✓	37.1	54.8	65.6
	✓	✓	39.5	57.4	68.8
✓	✓	✓	**40.4**	**58.6**	**69.0**

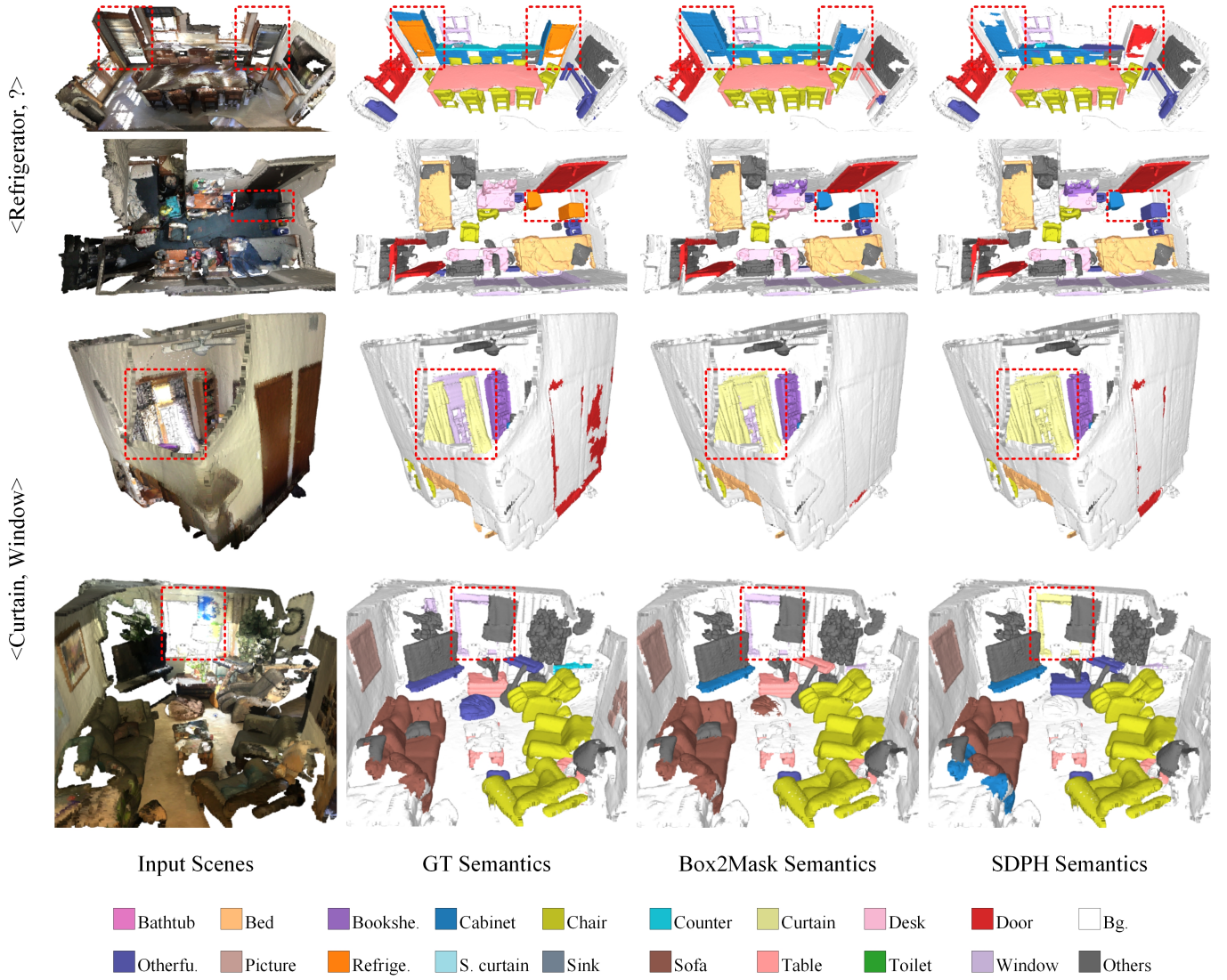


Figure 9. Bad cases on ScanNet-v2 in the noise-free setting. The first two rows show that refrigerators could be misclassified as cabinets, doors, and other furniture. We use "?" to represent this complicated situation. The last two rows show that windows could be misclassified as curtains, which lowered both categories' performance. The legend is employed to distinguish among different semantic meanings, and the key differences are marked out with red dashed rectangles.

4.4. Analysis of Label Refurbishment

To demonstrate the process of label refurbishment, we further recorded two related statistics, as shown in Figure 10. The first was the ratio of refurbishable super-voxel samples, which was defined as

$$\eta = \frac{number\ of\ refurbishable\ samples}{number\ of\ total\ samples}. \tag{17}$$

The second was the correction error, which could be computed as

$$\delta = \frac{number\ of\ mistakenly\ corrected\ samples}{number\ of\ refurbishable\ samples}. \tag{18}$$

Note that both statistics took the entire training set into account. We set the noise rate to 40%.

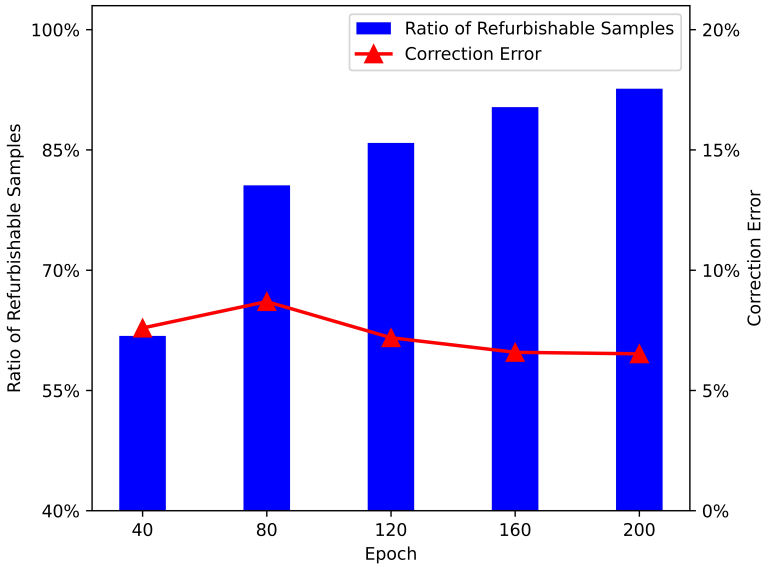


Figure 10. Trend of statistics in history-guided label refurbishment.

The ratio of refurbishable super-voxel samples gradually increased from 61.8% to 92.6%, finally covering the majority of the whole training set. Moreover, the correction error stayed relatively low throughout the training process because we adopted a conservative refurbishment strategy. On the one hand, the refurbishable threshold was quite strict to reduce false correction. On the other hand, we kept the unrefurbishable samples instead of dropping them, which lowered the risk of error accumulation. Therefore, the label quality was steadily improved as the training proceeded, as shown in Figure 11. However, we observed that it was easier to correct the labels of isolated objects with clear boundaries, such as chairs, sofas, and tables. On the contrary, flat objects that were often attached to walls, such as pictures and curtains, were harder to distinguish from the background.

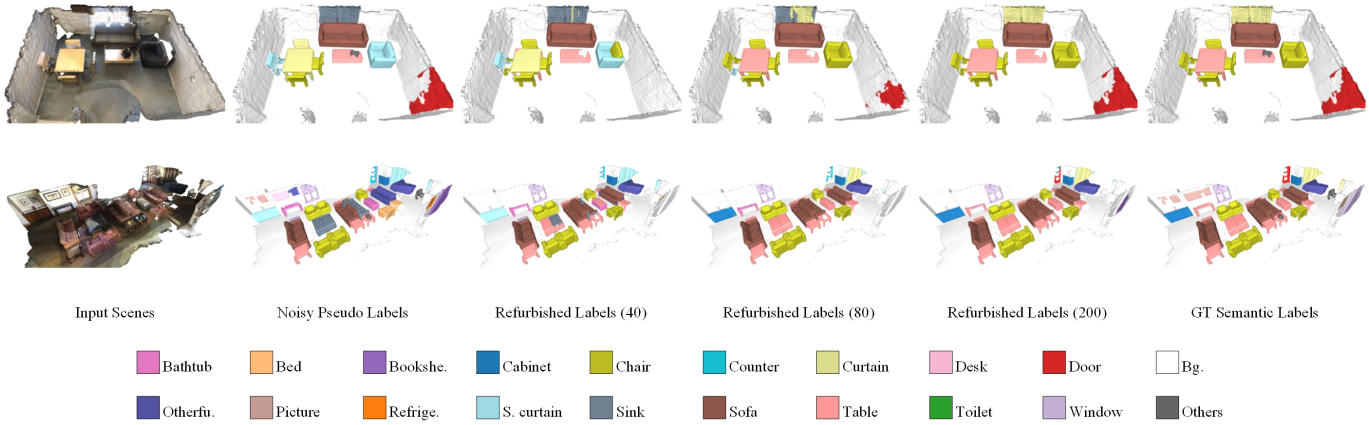


Figure 11. Qualitative demonstration of history-guided label refurbishment. From left to right are the input point clouds, the corresponding noisy pseudo-labels, the refurbished labels in epochs 40, 80, and 200, and the ground-truth semantic labels.

4.5. Complexity Analysis

Apart from the mean average precision, we also compared the time costs to give a full picture of the performance. As shown in Table 4, the inference time of our SDPH was comparable to that of the state-of-the-art weakly supervised method Box2Mask [20], though SDPH required a longer time for training. In fact, our approach mainly focused on the design of loss functions that only affected the training cost. Without extra network parameters, the majority of the additional cost came from perturbation-based consistency regularization (PCR), since it constructed a perturbed network branch. PCR did not affect the inference time, as only the main branch was used in inference.

Table 4. Comparison of the average computation time in milliseconds per scan on ScanNet-v2. The running time was measured in the same environment. Note that a post-processing step was implemented to cluster points into instances in inference.

Method	Training Time (ms)	Inference Time (ms)
Box2Mask [20]	444	1044
SDPH	722	1026

5. Conclusions

In this work, we proposed a novel self-distillation architecture for weakly supervised point cloud instance segmentation with inaccurate bounding boxes as annotations. We employed consistency regularization based on data perturbation and historical records to prevent the network from overfitting noisy labels. Moreover, the noisy labels were refurbished according to the predictions' temporal consistency without knowing the noise rate. An extensive ablation study and analysis verified the importance of each module in SDPH. Our method achieved comparable performance to that of fully supervised methods, and it outperformed recent weakly supervised methods by at least 1.2 percentage points in terms of AP_{25}, which demonstrated the effectiveness and robustness of our framework.

In the future, we plan to extend the noise types to asymmetric semantic noise and geometric coordinate noise, which may require a new confidence criterion. In addition, inspired by the mutual promotion between semantic segmentation and instance segmentation, semantic classification and geometric regression could be associated through smoothness regularization to reduce discontinuity and messy "over-segmentation".

Author Contributions: Conceptualization, Y.P. and H.F.; methodology, Y.P.; software, Y.P.; validation, Y.P.; formal analysis, Y.P.; investigation, Y.P.; resources, H.F.; writing—original draft preparation, Y.P.; writing—review and editing, H.F. and T.C.; visualization, Y.P.; supervision, B.H.; project administration, B.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data was accessed at http://www.scan-net.org/ on 27 August 2022.

Acknowledgments: We are particularly grateful to those who provided useful suggestions and kind help with programming during the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. El Madawi, K.; Rashed, H.; El Sallab, A.; Nasr, O.; Kamel, H.; Yogamani, S. RGB and LiDAR fusion based 3D Semantic Segmentation for Autonomous Driving. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 7–12. [CrossRef]
2. Yan, Z.; Duckett, T.; Bellotto, N. Online learning for 3D LiDAR-based human detection: Experimental analysis of point cloud clustering and classification methods. *Auton. Robot.* **2020**, *44*, 147–164. [CrossRef]
3. Zhao, Y.; Zhang, L.; Liu, Y.; Meng, D.; Cui, Z.; Gao, C.; Gao, X.; Lian, C.; Shen, D. Two-Stream Graph Convolutional Network for Intra-Oral Scanner Image Segmentation. *IEEE Trans. Med. Imaging* **2022**, *41*, 826–835. [CrossRef] [PubMed]
4. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [CrossRef] [PubMed]
5. Lu, H.; Shi, H. Deep Learning for 3D Point Cloud Understanding: A Survey. *arXiv* **2020**, arXiv:2009.08920.
6. Bello, S.A.; Yu, S.; Wang, C.; Adam, J.M.; Li, J. Review: Deep Learning on 3D Point Clouds. *Remote Sens.* **2020**, *12*, 1729. [CrossRef]
7. Liu, W.; Sun, J.; Li, W.; Hu, T.; Wang, P. Deep Learning on Point Clouds and Its Application: A Survey. *Sensors* **2019**, *19*, 4188. [CrossRef] [PubMed]
8. Yi, L.; Zhao, W.; Wang, H.; Sung, M.; Guibas, L.J. GSPN: Generative Shape Proposal Network for 3D Instance Segmentation in Point Cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
9. Engelmann, F.; Bokeloh, M.; Fathi, A.; Leibe, B.; Niessner, M. 3D-MPA: Multi-Proposal Aggregation for 3D Semantic Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 13–19 June 2020.
10. Chen, S.; Fang, J.; Zhang, Q.; Liu, W.; Wang, X. Hierarchical aggregation for 3d instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 10–17 October 2021; pp. 15467–15476.
11. Wang, W.; Yu, R.; Huang, Q.; Neumann, U. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
12. Pham, Q.H.; Nguyen, T.; Hua, B.S.; Roig, G.; Yeung, S.K. JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds with Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
13. Han, L.; Zheng, T.; Xu, L.; Fang, L. OccuSeg: Occupancy-Aware 3D Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 13–19 June 2020.
14. Zhang, B.; Wonka, P. Point Cloud Instance Segmentation Using Probabilistic Embeddings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 8883–8892.
15. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Niessner, M. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
16. Xu, X.; Lee, G.H. Weakly Supervised Semantic Point Cloud Segmentation: Towards 10x Fewer Labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 13–19 June 2020.
17. Liu, Z.; Qi, X.; Fu, C.W. One Thing One Click: A Self-Training Approach for Weakly Supervised 3D Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 1726–1736.
18. Tao, A.; Duan, Y.; Wei, Y.; Lu, J.; Zhou, J. SegGroup: Seg-Level Supervision for 3D Instance and Semantic Segmentation. *IEEE Trans. Image Process.* **2022**, *31*, 4952–4965. [CrossRef] [PubMed]
19. Wei, J.; Lin, G.; Yap, K.H.; Hung, T.Y.; Xie, L. Multi-Path Region Mining for Weakly Supervised 3D Semantic Segmentation on Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 13–19 June 2020.

20. Chibane, J.; Engelmann, F.; Anh Tran, T.; Pons-Moll, G. Box2Mask: Weakly Supervised 3D Semantic Instance Segmentation using Bounding Boxes. In Proceedings of the Computer Vision—ECCV 2022, Tel Aviv, Israel, 23–27 October 2022; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Springer Nature: Cham, Switzerland, 2022; pp. 681–699.
21. Liu, Y.; Hu, Q.; Lei, Y.; Xu, K.; Li, J.; Guo, Y. Box2Seg: Learning Semantics of 3D Point Clouds with Box-Level Supervision. *arXiv* **2022**, arXiv:2201.02963.
22. Liao, Y.; Zhu, H.; Zhang, Y.; Ye, C.; Chen, T.; Fan, J. Point Cloud Instance Segmentation with Semi-Supervised Bounding-Box Mining. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 10159–10170. [CrossRef] [PubMed]
23. Hou, J.; Graham, B.; Niessner, M.; Xie, S. Exploring Data-Efficient 3D Scene Understanding with Contrastive Scene Contexts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 15587–15597.
24. Hu, Z.; Gao, K.; Zhang, X.; Dou, Z. Noise resistant focal loss for object detection. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Nanjing, China, 16–18 October 2020; Springer: Cham, Switzerland, 2020; pp. 114–125.
25. Liu, X.; Li, W.; Yang, Q.; Li, B.; Yuan, Y. Towards Robust Adaptive Object Detection Under Noisy Annotations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022; pp. 14207–14216.
26. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding Deep Learning (Still) Requires Rethinking Generalization. *Commun. ACM* **2021**, *64*, 107–115. [CrossRef]
27. Ye, S.; Chen, D.; Han, S.; Liao, J. Learning with Noisy Labels for Robust Point Cloud Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 6443–6452.
28. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
29. Yuan, L.; Tay, F.E.; Li, G.; Wang, T.; Feng, J. Revisiting Knowledge Distillation via Label Smoothing Regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 13–19 June 2020.
30. Allen-Zhu, Z.; Li, Y. Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning. *arXiv* **2020**, arXiv:2012.09816.
31. Pham, M.; Cho, M.; Joshi, A.; Hegde, C. Revisiting Self-Distillation. *arXiv* **2022**, arXiv:2206.08491.
32. Kim, K.; Ji, B.; Yoon, D.; Hwang, S. Self-Knowledge Distillation with Progressive Refinement of Targets. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 10–17 October 2021; pp. 6567–6576.
33. Shen, Y.; Xu, L.; Yang, Y.; Li, Y.; Guo, Y. Self-Distillation From the Last Mini-Batch for Consistency Regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022; pp. 11943–11952.
34. Li, Y.; Yang, J.; Song, Y.; Cao, L.; Luo, J.; Li, L.J. Learning From Noisy Labels with Distillation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
35. Hou, J.; Dai, A.; Niessner, M. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
36. Yang, B.; Wang, J.; Clark, R.; Hu, Q.; Wang, S.; Markham, A.; Trigoni, N. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In Proceedings of the Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; NIPS: La Jolla, CA, USA, 2019; Volume 32.
37. Wang, X.; Liu, S.; Shen, X.; Shen, C.; Jia, J. Associatively Segmenting Instances and Semantics in Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
38. Jiang, L.; Zhao, H.; Shi, S.; Liu, S.; Fu, C.W.; Jia, J. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 13–19 June 2020.
39. Vu, T.; Kim, K.; Luu, T.M.; Nguyen, T.; Yoo, C.D. SoftGroup for 3D Instance Segmentation on Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022; pp. 2708–2717.
40. Zhou, Z.H. A brief introduction to weakly supervised learning. *Natl. Sci. Rev.* **2018**, *5*, 44–53. [CrossRef]
41. Zhang, Y.; Li, Z.; Xie, Y.; Qu, Y.; Li, C.; Mei, T. Weakly Supervised Semantic Segmentation for Large-Scale Point Cloud. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 3421–3429. [CrossRef]
42. Zhang, Y.; Qu, Y.; Xie, Y.; Li, Z.; Zheng, S.; Li, C. Perturbed Self-Distillation: Weakly Supervised Large-Scale Point Cloud Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 10–17 October 2021; pp. 15520–15528.
43. Cheng, M.; Hui, L.; Xie, J.; Yang, J. SSPC-Net: Semi-supervised Semantic 3D Point Cloud Segmentation Network. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 1140–1147. [CrossRef]
44. Ren, Z.; Misra, I.; Schwing, A.G.; Girdhar, R. 3D Spatial Recognition Without Spatially Labeled 3D. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 13204–13213.
45. Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.G. Learning From Noisy Labels with Deep Neural Networks: A Survey. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: Piscataway, NJ, USA, 2022; pp. 1–19. [CrossRef]

46. Graham, B.; Engelcke, M.; van der Maaten, L. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
47. Choy, C.; Gwak, J.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021.
48. Arpit, D.; Jastrzębski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M.S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. A Closer Look at Memorization in Deep Networks. In *International Conference on Machine Learning, Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017*; Precup, D., Teh, Y.W., Eds.; PMLR: San Diego, CA, USA, 2017; Volume 70, pp. 233–242.
49. Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; Rabinovich, A. Training Deep Neural Networks on Noisy Labels with Bootstrapping. *arXiv* **2014**, arXiv:1412.6596.
50. Song, H.; Kim, M.; Lee, J.G. SELFIE: Refurbishing Unclean Samples for Robust Deep Learning. In *Machine Learning Research, Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019*; Chaudhuri, K., Salakhutdinov, R., Eds.; PMLR: San Diego, CA, USA, 2019; Volume 97, pp. 5907–5915.
51. Nguyen, T.; Mummadi, C.K.; Ngo, T.P.N.; Nguyen, T.H.P.; Beggel, L.; Brox, T. SELF: Learning to filter noisy labels with self-ensembling. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 26–30 April 2020.
52. Lahoud, J.; Ghanem, B.; Pollefeys, M.; Oswald, M.R. 3D Instance Segmentation via Multi-Task Metric Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.