ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

ClickSeg: 3D Instance Segmentation with Click-Level Weak Annotations

Anonymous ICCV submission

Paper ID 8783

Abstract

3D instance segmentation methods often require fully-annotated dense labels for training, which are costly to obtain. In this paper, we present ClickSeg, a novel click-level weakly supervised 3D instance segmentation method that requires one point per instance annotation merely. Such a problem is very challenging due to the extremely limited labels, which has rarely been solved before. We first develop a baseline weakly-supervised training method, which generates pseudo labels for unlabeled data by the model itself. To utilize the property of click-level annotation setting, we further propose a new training framework. Instead of directly using the model inference way, i.e., mean-shift clustering, to generate the pseudo labels, we propose to use k-means with fixed initial seeds: the annotated points. New similarity metrics are further designed for clustering. Experiments on ScanNetV2 and S3DIS datasets show that the proposed ClickSeg surpasses the previous best weakly supervised instance segmentation result by a large margin (e.g., +9.4% mAP on ScanNetV2). Using 0.02% supervision signals merely, ClickSeg achieves ∼90% of the accuracy of the fully-supervised counterpart. Meanwhile, it also achieves state-of-the-art semantic segmentation results among weakly supervised methods that use the same annotation settings.

1. Introduction

While 3D point cloud semantic and instance segmentation have achieved remarkable results in recent years, they often require dense supervision, which brings a great human burden. On average, a scene in ScanNet [8] contains 150k points and takes 22.3 minutes to annotate, while there are a total number of 1513 scenes. The situation is more severe in some datasets [2, 16] that contain millions of points in a single scene. This largely hampers the application of fully-supervised al-
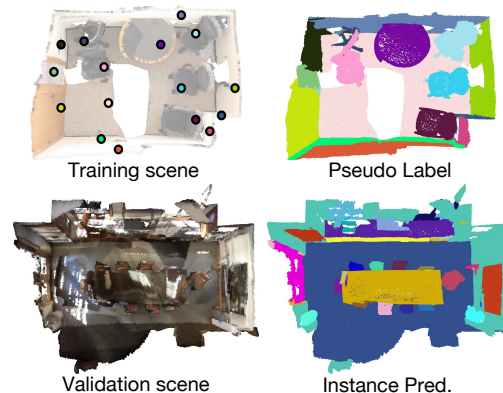

Figure 1. ClickSeg only requires the annotators to annotate one point per instance and can generate fine quality instance segmentation results.

gorithms.

Some recent methods [35, 25, 33] use weak labels to train the 3D semantic segmentation model and generate pseudo labels, but they mainly focus on semantic segmentation. Instance segmentation, however, is more complicated and hard to achieve using limited labels since it also requires the separation of all instances. Recently, Tao [28] propose a method for this task. They first recover the complete training set by gradually merging supervoxels to their neighboring labeled ones, then perform fully-supervised methods. However, the training set is fixed once recovered, which contains many errors, thus their accuracy largely falls behind fully-supervised methods. Besides, they require the annotators to annotate the most central supervoxel of each instance, which is hard for humans to realize.

In this paper, we present ClickSeg, a weakly supervised 3D instance segmentation method that only requires the annotators to annotate one arbitrary point per instance. We first develop a weakly-supervised training method that performs on 3D instance segmentation. It initially trains the model on weak labels by metric learning, then generates pseudo labels for unla-

ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

beled data by the inference way of the model. To fully utilize the property of click-level annotation setting, we further propose a new training framework. Since each instance is annotated for one point, every unlabeled point must belong to one of the annotated points. Thus, instead of directly using the model inference way, i.e., mean-shift clustering, to generate pseudo labels, we propose to use k-means clustering with fixed initial seeds: the annotated points. Furthermore, we define new similarity metrics used in clustering to better adapt to weak supervision setting, which helps pseudo label generation. The pseudo labels are generated online during training, which are increasingly accurate. This novel framework significantly improves the final accuracy.

Experimental results show that ClickSeg surpasses the previous best weakly supervised 3D instance segmentation method by a large margin, and achieves ~90% of the accuracy of the fully supervised counterpart. Meanwhile, it also achieves state-of-the-art semantic segmentation results among other weakly supervised methods that use the same annotation settings. To summarize, our contributions include:

- An effective and efficient 3D instance segmentation training framework which only requires click-level annotations. It generates pseudo labels online during training and the training time is almost the same as fully-supervised methods.

- A novel pseudo instance label generation method specifically designed for click-level annotation setting, which utilize k-means with fixed initial seeds: the annotated points.

- Extensive experiments demonstrate our superior performance over other weakly-supervised methods in both instance and semantic segmentation.

2. Related Work

2.1. Point Cloud Segmentation

3D point cloud segmentation is a fundamental computer vision task that has been widely studied. Recent methods could be mainly divided into two categories: point-based and voxel-based, according to the convolution type. The point-based methods [27, 34, 30, 26] use point convolution to extract deep per point features. Voxel-based methods like [7, 14] first voxelize the point cloud, then perform 3D CNN on the grids. SparseConv [12] is later proposed to reduce the computation and memory consumption of these methods.

Compared with semantic segmentation methods, instance segmentation methods face a more challenging situation, as they are also required to separate different instances. Current fully-supervised 3D instance segmentation methods can be categorized into proposal-based and grouping-based paradigms. The proposal-based methods use a top-down strategy to first generate the region proposals, then perform segmentation inside the region [14, 36]. The grouping-based methods use a bottom-up scheme that group points to form the instances. We follow this kind of scheme and mainly introduce these methods below. MTML [20] outputs an embedding and a directional vector for each point. They use the embedding to cluster points and the directional vector to score the proposals. Occuseg [13] proposes an occupancy signal to guide the clustering process. PointGroup [18] uses semantics to group original and offset-shifted points, and then uses ScoreNet to score the proposals. HAIS [4] follows this method and further absorbs small fragments into instances. SST-Net [22] constructs a superpoint tree to form instances via merge and split. These methods achieve remarkable results but require fully-annotated labels, which are difficult to obtain.

2.2. Weakly Supervised Segmentation

Weakly supervised image semantic segmentation have been extensively studied, including image-level [1], click-level [3], scribble-level [23], and box-level [9] supervisions. The image-level supervision only indicates the contained classes of an image. Most of these methods [32, 17] use class activation map (CAM) to generate pixel-level labels. What's the point [3] is the first to introduce click-level supervision that annotates one point per object. It uses objectness prior, image-level and click-level supervision to form the loss function. ClickSeg uses the same annotation strategy. The scribble-level supervision is proposed in [23] to draw a curve for each object. The box-level supervision uses bounding boxes to train a segmentation model.

For weakly supervised 3D segmentation, the recently emerged methods mainly focus on semantic segmentation. MPRM [33] generates point-level labels from sub-cloud-level supervision using CAM. Xu *et al.* [35] propose a method that only requires 10% of labels to reach comparable results to their fully-supervised counterpart. SQN [15] shows that very few labels are enough for semantic segmentation and propose a method that requires only 1% labels for training. One-Thing-One-Click [25] uses click-level labels and proposed a self-training semantic segmentation method that utilizes CRF for label propagation. Weakly supervised 3D instance segmentation, however, is an under-explored area. Recently, SegGroup [28] is proposed to tackle this problem. They require the annotator to annotate the most central supervoxel of each instance. They
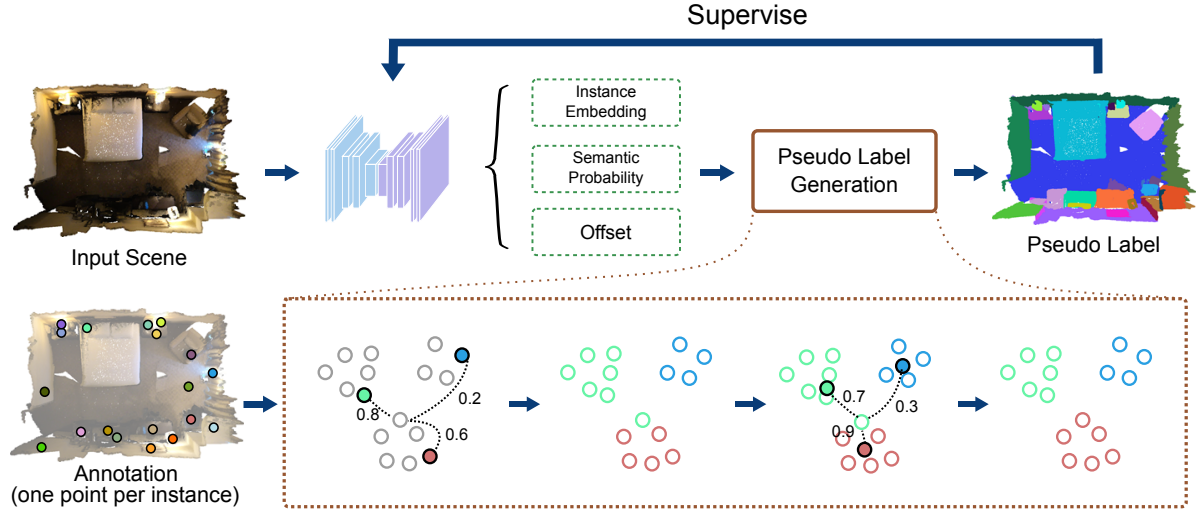
Figure 2. The pipeline of ClickSeg. The network outputs several terms for each point of the input scene. We generate pseudo labels by using these terms as similarity metrics to perform k-means clustering with initial seeds being the annotated points. The network is in turn trained using these pseudo labels. Please refer to Sec. 3 for more details.

first recover the training set by merging neighboring supervoxels according to the features from graph convolution, and then perform fully-supervised training.

3. Method

We first introduce the fully supervised baseline method, which could be seen as the upper bound. We adopt a common fully supervised method [13, 37] that uses a network backbone to output for each point i an embedding vector e_i, a semantic probability s_i, and an offset vector o_i pointing to the center of its belonging instance. The backbone is the widely used 3D UNet implemented by Submanifold Sparse Convoliton [12], which is shown to be effective with efficiency.

The semantic probabilities and the offset vectors are trained using cross-entropy loss L_{CE} and L2 regression loss $L_{regress}$, respectively. The embeddings are trained by metric learning [19]. Specifically, discriminative loss [10] is used, which reduces the distances of embeddings from the same instance by using the loss function of:

$$L_{var} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} [||\mu_c - e_i|| - \delta_v]_+^2, \qquad (1)$$

and push away embeddings from different instances by using:

$$L_{dist} = \frac{1}{C(C-1)} \sum_{c_A=1}^{C} \sum_{c_B=1}^{C} [2\delta_d - ||\mu_{c_A} - \mu_{c_B}||]_+^2. \quad (2)$$

Here C is the number of instances, N_c is the number of points in instance c, μ_c is the mean embedding of

instance c, and threshold δ_v is set to be smaller than δ_d. $[x]_+$ is x if $x > 0$ or 0 otherwise.

At the inference stage, we use the embeddings and offsets as similarity metrics to perform mean-shift clustering to group points into instances. The similarity between point i and j is defined as:

$$S_{ij} = \exp(-(\frac{||e_i - e_j||}{\sigma_e})^2 - (\frac{||(p_i + o_i) - (p_j + o_j)||}{\sigma_p})^2), \tag{3}$$

where p_i denotes the position vector of point i, σ_e and σ_p are learned to automatically weight these two terms as in [13].

3.1. Weakly-Supervised Approach

The metric learning is not directly applicable in our weak annotation setting. This is because since only one point is annotated per instance, it is infeasible to reduce the distances of embeddings from the same instance in Eqn. 1. Thus, we need to expand our labels first.

We perform supervoxel partition using low-level features to segment the whole scene into small supervoxels. Points in the same supervoxel can be considered to belong to the same instance. This enables us to expand the weak point labels to supervoxel-level.

The embeddings can then be trained using Eqn. 1 and 2 on the labeled points. μ_c in the current case denotes the mean embedding of labeled points in instance c. The offset vectors, however, can not be learned using weak labels. Therefore, at the inference stage, only the embeddings are used to group points to instances.

3

ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

The similarity is defined as:

$$S_{ij} = \exp(-(||e_i - e_j||)^2). \qquad (4)$$

Using only weak labels for training produces poor performance as shown in the Tab. 3. To fully utilize the unlabeled points, we propose a training framework that generates pseudo labels during training.

Baseline version. We first train the embeddings by metric learning on the weak labels for several epochs as described above. Then we generate pseudo instance labels by using the model to perform inference on the training set, that is, grouping points to form instances by the predicted embeddings. After that, we continue to train our model using the pseudo labels with high confidence. Specifically, Eqn. 1 is only performed on point i that satisfies $||\mu_c - e_i|| < \alpha$. Eqn. 2 is only performed on two instances c_A and c_B if they satisfy $||\mu_{c_A} - \mu_{c_B}|| > \beta$. The training and pseudo label generation iterate several times. We denote this method as the baseline version.

The network trained in this way could utilize the unlabeled data, which boosts the model performance as shown in Tab. 3 and Appendix. However, it still largely falls behind the fully-supervised baseline.

Click annotation version. The baseline version generates pseudo labels by directly using the inference way of the model, that is, performing mean-shift to group points by the network output embeddings. However, it fails to utilize the original annotation information. We annotate one point per instance, which means every point in the scene belongs to one of the annotated points. Thus the generation of pseudo instance labels in each scene can be treated as a per-point classification problem that assigns each unlabeled point to one of the annotated points, instead of directly clustering. This can be achieved by nearest neighbor search using the similarity metrics.

However, we take a step further by using k-means clustering with initial seeds being the annotated points. Our intuition lies in that, some annotated points may locate far from the center of their belonging instance in feature space, making the unlabeled points difficult to match them correctly. With the moving of seeds in k-means clustering, this problem can be alleviated. Each cluster contains only one annotated point, otherwise the k-means will stop. The process is visualized in Fig. 2.

We generate pseudo instance labels for all unlabeled points. This enables us to learn the offset vector described in the fully-supervised counterpart. The network is trained on all pseudo instance labels without thresholding. We empirically find that the pseudo label generation accuracy increases with the training goes

(see Fig. 3), thus we regenerate the pseudo labels for each input scene during training.

We also generate pseudo semantic labels. For an unlabeled point i, we use the semantic label of its matched annotated point j as the pseudo label. The network is trained only on point i that satisfies $S_{ij} >= \theta$. This is because for semantic segmentation, only a small number of labels is needed for training to achieve a high accuracy [15], and we find that more false pseudo labels will have negative effects on the accuracy. The whole training process is depicted in Alg. 1. We denote this method as the click annotation version.

Algorithm 1: Training process of the click annotation version. PSEUDO_GEN denotes the pseudo label generation process. s, e, o are semantic probabilities, embeddings, and offsets, respectively. t is a hyper-parameter.

Input: A training sample (x, y_1, y_2), denoting the input scene, the click-level instance and semantic labels.
Output: Training loss L.
$s, e, o \leftarrow \text{NETWORK}(x)$
$\overline{y_1} \leftarrow$ expand y_1 by supervoxel
$\overline{y_2} \leftarrow$ expand y_2 by supervoxel
$L \leftarrow L_{var}(e, \overline{y_1}) + L_{dist}(e, \overline{y_1}) + L_{CE}(s, \overline{y_2})$
if *current training epoch* $\geq t$ **then**
    $\hat{y_1}, \hat{y_2} \leftarrow \text{PSEUDO\_GEN}(s, e, o, x, y_1)$
    Filter out unreliable labels in $\hat{y_2}$
    $L \leftarrow L + L_{var}(e, \hat{y_1}) + L_{dist}(e, \hat{y_1}) +$
    $L_{CE}(s, \hat{y_2}) + L_{regress}(o, \hat{y_1})$
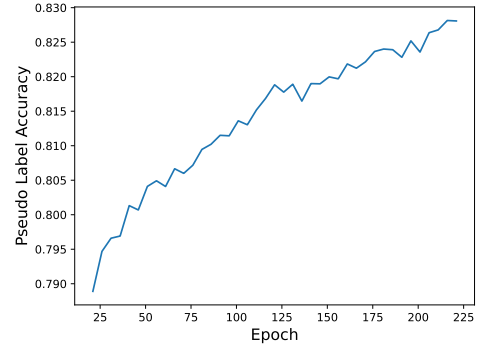**end**

Figure 3. Pseudo label generation accuracy in different training epochs.

Similarity. We modify the similarity metrics to better adapt to weak supervision setting, thus helping to generate more accurate pseudo labels. The new similarity formula between point i and j is defined as:

$$S_{ij} = Q_{ij} * \exp(-(\frac{D_{ij}^e}{\sigma_e})^2 - (\frac{D_{ij}^p}{\sigma_p})^2), \qquad (5)$$

ICCV
#8783

ICCV
#8783

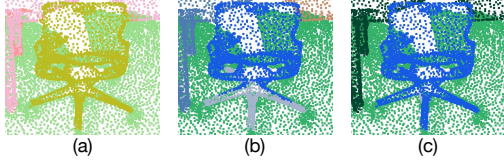ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 4. A chair in the training set. (a) the predicted semantics, (b)(c) the pseudo instance label generated without and with using semantics.

which contains three terms: the embedding distance D_{ij}^e, the spatial distance D_{ij}^p, and the semantic similarity Q_{ij}. The embedding distance D_{ij}^e is simply $||e_i - e_j||$ which is the same as Eqn. 4.

The semantic similarity score Q_{ij} between two points is calculated by: $Q_{ij} = \frac{s_i \cdot s_j}{||s_i|| ||s_j||}$. We take this term into account for the following reason: the embeddings trained on weak supervoxel-level labels can only capture local similarities by low-level features like colors, normals, etc., depending on the supervoxel partition method. If an instance is composed of several parts, with each part having large differences in their "looks", simply using the embeddings may not be able to match them correctly. In contrast, the prediction of semantics is relatively robust for different parts of an object. For example, the chair leg can be correctly classified as *chair* in Fig. 4, while it can not be recognized as part of the chair in the instance embedding space. This is because although the number of semantic labels of *chair* is limited in one scene, there are still many labels of *chair* in the whole dataset. The learning on these labels helps the network to recognize the different parts of the chair. Points belonging to the same instance should share the same semantic labels, so we add this term to help better match them.

The spatial distance D_{ij}^p is defined as $||p_i - p_j||$, where p_i denotes the position vector of point i. This is based on the observation that points belonging to the same instance are often spatially adjacent. Adding this term avoids some false cases that assign unlabeled points to the annotated points that have the same semantic labels and similar local structures but belong to different instances. After some training epochs on the pseudo labels, the offset vectors are learned, we can then calculate D_{ij}^p by $||(p_i + o_i) - (p_j + o_j)||$, same as the fully supervised baseline method.

One-thing-multi-click. The current method can also be extended to the annotation setting of one-thing-multi-click, which means annotating multiple points per instance. The pseudo label generation way is a bit different than that of one-thing-one-click. In the multi-click case, the k-means is not applicable. Thus we directly assign unlabeled points to instances by nearest neighbor search. We calculate the likelihood of an un-

labeled point belonging to an instance by the mean of similarities between this point and all the annotated points of that instance. Then we simply choose the most likely instance the point belongs to as its pseudo label.

3.2. Inference

At inference stage, for instance segmentation, we perform mean-shift clustering using the similarity metrics to get the predicted instances. For semantic segmentation, we perform a fusion step for each point to fuse the semantic probabilities of other points according to the similarities. The insight is that for similar points, they should share the same semantic labels. The calculation formula is:

$$\hat{s}_i = \frac{\sum_{j=1}^{N} \mathbb{I}_{ij} * S_{ij} * s_j}{\sum_{j=1}^{N} \mathbb{I}_{ij} * S_{ij}}, \tag{6}$$

where $\hat{s}_i$ denotes the fused semantic probability of point i, N is the number of points in the scene, $\mathbb{I}_{ij}$ is 1 if $S_{ij} > \gamma$ and 0 otherwise. The final prediction for point i is $argmax\, \hat{s}_i$. This helps to smooth the predictions of similar points, resulting in a more consistent semantic map.

4. Experiments

4.1. Experimental Setup

Datasets and metrics. We evaluate ClickSeg on two popular datasets, ScanNetV2 [8] and S3DIS [2]. The ScanNetV2 dataset contains 1513 scenes that have publicly available labels and 100 scenes for hidden testing. These 1513 scenes are further split into 1201 and 312 scenes officially for training and validation. We follow this split and evaluate on both validation and testing sets. The S3DIS dataset includes 6 large-scale areas. Each area contains dozens of indoor scenes. We follow other methods [25, 18] to choose Area-5 as the testing set, and other areas as the training set. We evaluate both semantic and instance results. The evaluation metric for semantic segmentation is mean Intersection-over-Union (mIoU), and for instance segmentation is mean Average Precision (mAP).

Annotation details. We annotate one single arbitrary point for every object that appears in the scene to indicate its semantic class, which is the same as [25]. The instance id is naturally contained. There are on average 31.93 points per scene in ScanNetV2 training set and 36.7 points per scene in S3DIS training set are annotated under this setting. The average number of points per scene in ScanNetV2 training set is 145,170.8, and 955,975.5 in S3DIS. The annotation ratio is only about 0.02% in ScanNetV2 and 0.004% in S3DIS.
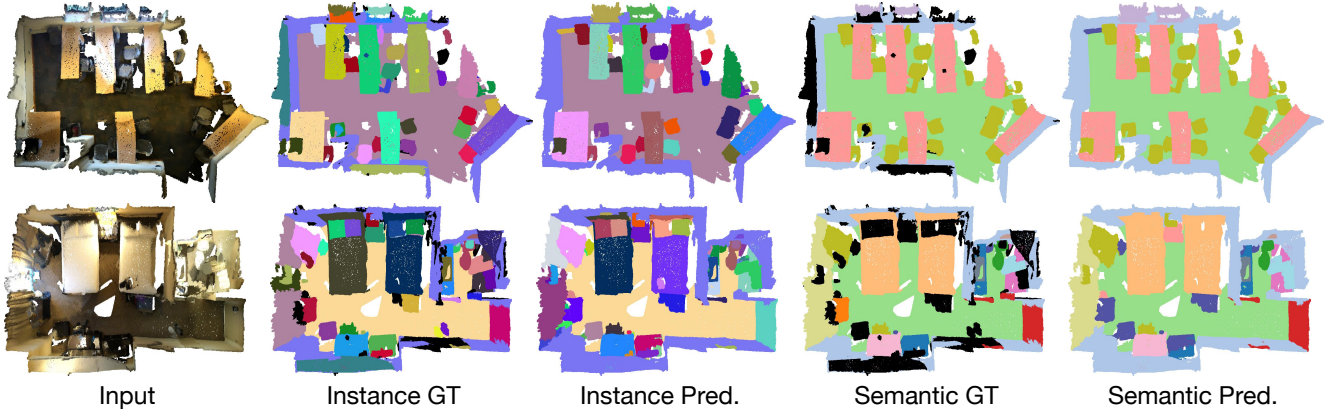
ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Input   Instance GT   Instance Pred.   Semantic GT   Semantic Pred.

Figure 5. Visualization of the semantic and instance results of our click annotation version on the ScanNetV2 validation set.

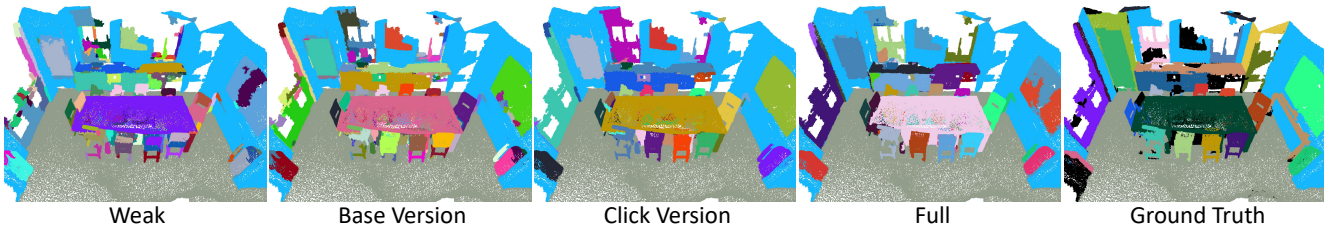Weak   Base Version   Click Version   Full   Ground Truth

Figure 6. Qualitative results of different versions of our method on the ScanNetV2 validation set. "Weak" denotes training only on weak labels. "Base Version" denotes the baseline version of our method. "Click Version" denotes the click annotation version. "Full" denotes the fully-supervised counterpart method.

Implementation details. We use the Adam optimizer with a learning rate of 5e-3. The batch size is 8 for ScanNetV2 and 4 for S3DIS. A single Tesla V100 (32GB) GPU is used for training. We generate the supervoxels by using the graph-based method [11] for ScanNetV2, and SSP [21] for S3DIS. Following other methods [18, 22], we pre-process scenes in S3DIS by randomly sub-sample points at a rate of 1/4 since the point density is too high. The voxel size is set to 0.02m. We first train the network using expanded supervoxel-level weak labels for 250 epochs. Then we generate pseudo labels and train using both the pseudo labels and the weak labels for another 250 epochs. We set the hyper-parameters $\delta_v = 0.2$, $\delta_d = 1.5$, $\alpha = 0.6$, $\beta = 1.5$, $\theta = 0.9$, $\gamma = 0.3$.

4.2. Results

ScanNetV2. We compare ClickSeg with fully-supervised methods and other weakly or semi-supervised methods. The instance and the semantic segmentation results are shown in Tab. 1. It achieves about 95% mIoU and 90% mAP@50 of the fully-supervised baseline using only 0.02% labels for training. Meanwhile, ClickSeg also surpass all other weakly supervised methods in both semantic and instance results. Notice that SegGroup [28] requires the annotator to annotate the most central supervoxel for each

instance, while our method only requires to annotate one arbitrary point for each instance. The qualitative results are shown in Fig .5.

We additionally report the results of ClickSeg under the annotation setting of one-thing-two/three-click in Tab. 1. With each instance annotating up to three points, it even achieves about 99% accuracy of our fully-supervised baseline in terms of both semantic and instance segmentation.

S3DIS. We report the semantic and instance segmentation results of ClickSeg on S3DIS Area-5. Since no other weakly or semi-supervised 3D instance segmentation method have tested on S3DIS, we only compare with fully-supervised methods and other weakly or semi-supervised semantic segmentation methods. The results are shown in Tab. 2, the semantic result of ClickSeg outperforms other weakly supervised methods. It surpasses [25] by 7% of mIOU using the same annotations. It also achieves 85.5% instance mAP@50 compared with the fully-supervised baseline by using only 0.004% labels for training.

4.3. Ablation Study

We test the effectiveness of different components in ClickSeg. The ablation studies are conducted on ScanNetV2 validation set, under the one-thing-one-click annotation setting.

ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

(a) Instance Segmentation on ScanNetv2			
Methods	Supervision	mAP@50 (%) (val split)	mAP@50 (%) (test split)
SGPN [31]	100%	-	14.3
3D-SIS [14]	100%	22.5	38.2
3D-BoNet [36]	100%	-	48.8
MASC [24]	100%	-	44.7
MTML [20]	100%	40.2	54.9
PointGroup [18]	100%	56.9	63.6
Occuseg [13]	100%	60.7	67.2
Fully-sup baseline	100%	57.7	-
Weakly-sup baseline	1t1c (0.02%)	25.5	-
SegGroup [28]	Supervoxels	43.4	44.5
ClickSeg	1t1c (0.02%)	51.9	53.9
ClickSeg	1t2c (0.04%)	55.2	-
ClickSeg	1t3c (0.06%)	57.0	-
(b) Semantic Segmentation on ScanNetv2			
Methods	Supervision	mIoU (%) (val split)	mIoU (%) (test split)
PointNet [27]	100%	53.5	55.7
PointConv [34]	100%	61.0	66.6
KPConv [30]	100%	69.2	68.4
SparseConvNet [12]	100%	69.3	72.5
MinkowskiNet [5]	100%	72.2	73.6
Occuseg [13]	100%	-	76.4
Fully-sup baseline	100%	72.0	-
Fully-sup baseline*	100%	74.6	-
Weakly-sup baseline	1t1c (0.02%)	61.9	-
MPRM [33]	Subcloud	46.6	41.1
SegGroup [28]	Supervoxels	62.4	61.1
SQN [15]	0.1%	-	51.6
OTOC [25]	1t1c (0.02%)	70.5	69.1
ClickSeg	1t1c (0.02%)	71.4	70.3
ClickSeg	1t2c (0.04%)	72.6	-
ClickSeg	1t3c (0.06%)	74.1	-

Table 1. Instance and semantic segmentation results on ScanNetV2 validation and test set. "1t1c" means annotating one arbitrary point per instance. Seggroup [28] annotates the most central supervoxel of each instance. "Fully-sup baseline*" means the fully-supervised baseline method with the semantic fusion strategy described in Sec. 3.2. "Weakly-sup baseline" means training only on weak supervoxel-level labels as described in Sec. 3.1. Due to the ScanNetV2 submission policy, we only report the test set result of ClickSeg under the "1t1c" setting.

4.3.1   Ablation on Instance Segmentation

Different versions. We show the instance mAP@50 results of different versions of ClickSeg in Tab. 3 and the qualitative results in Fig. 6. The network trained directly on weak supervoxel-level labels only achieves 25.5% mAP@50. This is because the embeddings can only capture local similarity, thus resulting in over-segmentation. The baseline version utilizes the un-labeled data for training, which improves the results by 9.0%. However, the pseudo labels are not accurate enough, and the high threshold filters out most of the

pseudo labels. Thus the accuracy still largely falls be-hind the fully-supervised method. The click annotation version use a different way of pseudo label generation, and proposes some modifications to the similarity met-rics, which greatly increase the pseudo label accuracy. The final results outperform the baseline version by a significant increase of 17.4% mAP@50, which validates the effectiveness of our method.

Effect of k-means. In the click annotation ver-sion, instead of using nearest neighbor search to as-sign pseudo labels, the k-means helps to alleviate the

ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

(a) Instance Segmentation on S3DIS

Methods	Supervision	mAP@50 (%)
PointGroup [18]	100%	57.8
SSTNet [22]	100%	59.3
Fully-sup baseline	100%	53.4
Weakly-sup baseline	1t1c (0.004%)	21.0
ClickSeg	1t1c (0.004%)	45.7

(b) Semantic Segmentation on S3DIS

Methods	Supervision	mIoU (%)
PointNet [27]	100%	41.1
TangentConv [29]	100%	52.8
MinkowskiNet[6]	100%	65.4
KPConv [30]	100%	67.1
Fully-sup baseline	100%	65.5
Weakly-sup baseline	1t1c (0.004%)	51.0
SQN [15]	0.01%	45.3
OTOC [25]	1t1c (0.004%)	50.2
ClickSeg	1t1c (0.004%)	57.2

Table 2. Instance and semantic results on S3DIS Area-5.

	Weak	Base	Click	Full
mAP@50	25.5	34.5	51.9	57.7

Table 3. Instance segmentation results of different versions of our method. "Weak" denotes training only on weak labels. "Base" denotes the baseline version. "Click" denotes the click annotation version. "Full" denotes the fully-supervised counterpart method.

	Nearest neighbor	K-means
mAP@50	50.5	51.9

Table 4. Results of different way of generating pseudo labels in the click annotation version.

problem of misclassification. Because some annotated points may locate at the border of their belonging instances in the feature space, unlabeled points can be misclassified. The k-means gradually moves the seeds to help to correct these false cases. Tab. 4 shows using k-means increases the accuracy by 1.4% mAP@50.

Effect of different similarity metrics. We test the effect of different designs of similarity metrics. There are three terms in the similarity calculation formula: the embedding distance, the spatial distance, and the semantic similarity. These terms help generate more accurate pseudo labels. As shown in Tab. 5, we remove some of these terms to test their effects. The results show that they all improve the instance accuracy in terms of mAP@50, thus proving the effectiveness of our designs.

Embedding	Spatial	Semantic	mAP@50
✓			48.7
✓	✓		49.2
✓		✓	50.7
✓	✓	✓	51.9

Table 5. Effects of different terms in the similarity formula.

Pseudo Inst.	Pseudo Sem.	Fusion	mIoU
			61.9
✓			66.1
✓	✓		67.9
✓	✓	✓	71.4

Table 6. Ablation study on semantic segmentation.

4.3.2 Ablation on Semantic Segmentation

The ablation results on semantic segmentation are shown in Tab. 6. The model trained on supervoxel-level weak labels achieved 61.9% mIoU. We find that even only with the training on instance pseudo labels, the semantic segmentation accuracy increased by 4.2% as well, which is shown in the second raw of Tab. 6. This is because we use a multi-task framework, where each task can benefit from the others. The training on the semantic pseudo labels further boosted the results by 1.8% mIoU. At the inference stage, for each point, we fuse the semantic probabilities of similar other points. This utilizes the instance information to guide the semantic segmentation, which helps to smooth the predictions of similar points, with an increase of 3.5% mIoU.

5. Conclusion and Discussion

In this paper, we present ClickSeg. We first develop a baseline training method for weakly supervised instance segmentation. To utilize the property of the click-level annotation setting, we further propose a specific training framework termed click annotation version. Extensive experiments prove the superior performance of our method. ClickSeg also has limitations. The baseline version could be applied to some other annotation settings but is not accurate enough compare with fully supervised methods. The click annotation version has higher accuracy but may have performance degradation problem when not all instances are annotated for at least one point. Setting threshold when generating pseudo labels may help to tackle this problem.

ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4981–4990, 2018. 2

[2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 1, 5

[3] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016. 2

[4] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15467–15476, 2021. 2

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 7

[6] Christopher Bongsoo Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3070–3079, 2019. 8

[7] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2432–2443, 2017. 1, 5

[9] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1635–1643, 2015. 2

[10] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017. 3

[11] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 6

[12] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018. 2, 3, 7

[13] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2937–2946, 2020. 2, 3, 7

[14] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4416–4425, 2019. 2, 7

[15] 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. 2, 4, 7, 8

[16] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 92–101, 2016. 1

[17] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7014–7023, 2018. 2

[18] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4866–4875, 2020. 2, 5, 6, 7, 8

[19] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019. 3

[20] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3d instance segmentation via multi-task metric learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9255–9265, 2019. 2, 7

[21] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 6

[22] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2783–2792, 2021. 2, 6, 8

[23] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3159–3167, 2016. 2

ICCV
#8783

ICCV
#8783

ICCV 2023 Submission #8783. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

[24] Chen Liu and Yasutaka Furukawa. Masc: Multi-scale affinity with sparse convolution for 3d instance segmentation. *arXiv preprint arXiv:1902.04478*, 2019. 7

[25] Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. 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*, pages 1726–1736, 2021. 1, 2, 5, 6, 7, 8

[26] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 2

[27] C. Qi, L. Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2, 7, 8

[28] An Tao, Yueqi Duan, Yi Wei, Jiwen Lu, and Jie Zhou. Seggroup: Seg-level supervision for 3d instance and semantic segmentation. *IEEE Transactions on Image Processing*, 31:4952–4965, 2022. 1, 2, 6, 7

[29] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2018. 8

[30] Hugues Thomas, C. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6410–6419, 2019. 2, 7, 8

[31] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2569–2578, 2018. 7

[32] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12275–12284, 2020. 2

[33] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4384–4393, 2020. 1, 2, 7

[34] Wenxuan Wu, Zhongang Qi, and Fuxin Li. Pointconv: Deep convolutional networks on 3d point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9613–9622, 2019. 2, 7

[35] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13706–13715, 2020. 1, 2

[36] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *Advances in neural information processing systems*, 32, 2019. 2, 7

[37] Dongsu Zhang, Junha Chun, Sang Kyun Cha, and Young Min Kim. Spatial semantic embedding network: Fast 3d instance segmentation with deep metric learning. *arXiv preprint arXiv:2007.03169*, 2020. 3