Hierarchical Aggregation for 3D Instance Segmentation

Shaoyu Chen\textsuperscript{1}  Jiemin Fang\textsuperscript{2,1}  Qian Zhang\textsuperscript{3}  Wenyu Liu\textsuperscript{1}  Xinggang Wang\textsuperscript{1,†}

\textsuperscript{1}School of EIC, Huazhong University of Science & Technology
\textsuperscript{2}Institute of AI, Huazhong University of Science & Technology
\textsuperscript{3}Horizon Robotics

\{shaoyuchen,jaminfong,liuwy,xgwang\}@hust.edu.cn \{qian01.zhang\}@horizon.ai

Abstract

Instance segmentation on point clouds is a fundamental task in 3D scene perception. In this work, we propose a concise clustering-based framework named HAIS, which makes full use of spatial relation of points and point sets. Considering clustering-based methods may result in oversegmentation or under-segmentation, we introduce the hierarchical aggregation to progressively generate instance proposals, i.e., point aggregation for preliminarily clustering points to sets and set aggregation for generating complete instances from sets. Once the complete 3D instances are obtained, a sub-network of intra-instance prediction is adopted for noisy points filtering and mask quality scoring. HAIS is fast (only 410ms per frame) and does not require non-maximum suppression. It ranks 1st on the ScanNet v2 benchmark\textsuperscript{1}, achieving the highest 69.9\% AP\textsubscript{50} and surpassing previous state-of-the-art (SOTA) methods by a large margin. Besides, the SOTA results on the S3DIS dataset validate the good generalization ability. Code will be available at \url{https://github.com/hustvl/HAIS}.

1. Introduction

With the rapid development and popularization of commodity 3D sensors (Kinect, RealSense, Velodyne laser scanner, etc.), 3D scene understanding has become a hot research topic in the field of computer vision. Instance segmentation on point cloud, as the basic perception task of 3D scene understanding, is the technical foundation of a wide range of real-life applications, e.g., robotics, augmented/virtual reality, and autonomous driving.

Instance segmentation on 2D images has been exhaustively studied in the past few years [7, 24, 16, 27, 18, 4, 5, 11]. Top-down methods dominate 2D instance segmentation. They first generate instance-level proposals and then predict the mask for each proposal. Though existing 2D instance segmentation methods can be directly extended to 3D scenes, most existing 3D methods adopt a totally different bottom-up pipeline [42, 41, 22, 15, 20], which generates instances through clustering.

However, directly clustering a point cloud into multiple instances is very difficult for the following reasons: (1) A point cloud usually contains a large number of points; (2) The number of instances in a point cloud has large variations for different 3D scenes; (3) The sizes of instances vary significantly; (4) Each point has a very weak feature, i.e., 3D coordinate and color. The semantic gap between point and instance identity is huge. Thus, over-segmentation or under-segmentation are common problems and are prone to exist.

We propose a hierarchical aggregation scheme in bottom-up 3D instance segmentation networks to cope with these problems. We first aggregate points to sets with low

\textsuperscript{1}http://kaldir.vc.in.tum.de/scannet_benchmark/semantic_instance_3d
\textsuperscript{†}Xinggang Wang is the corresponding author.
bandswidth to avoid over-segmentation and then set aggregation with dynamic bandwidth is adopted to form complete instances. Set aggregation may absorb noisy point sets into predictions, making the aggregated instances incomplete. Thus, we design a sub-network for outlier filtering and mask quality scoring. Based on the hierarchical aggregation and the sub-network for intra-instance prediction, we propose a novel bottom-up framework, named HAIS. HAIS achieves the state-of-the-art (SOTA) performance on both the ScanNet v2 benchmark [6] and the S3DIS [1] dataset. Beyond this, HAIS is efficient, only requiring a concise single-forward inference without any post-processing steps. Compared with all the existing methods, HAIS takes the lowest inference latency.

Our contributions can be summarized as follows.

- We propose a novel bottom-up framework with the hierarchical aggregation for instance segmentation on 3D point cloud. The hierarchical aggregation strategy makes up the defects of bottom-up clustering. Besides, an intra-instance prediction network is designed for generating more fine-grained instance predictions.

- Our method ranks 1st on the leaderboard of ScanNet v2 [6]. HAIS also achieves the state-of-the-art result on S3DIS [1]. We significantly promote the performances on various challenging datasets and demonstrate the generalization of the proposed methods.

- Our method achieves the highest efficiency among all existing methods. HAIS keeps a concise single-forward inference pipeline without any post-processing steps. The average per-frame inference time on ScanNet v2 is only 410 ms, much faster than other methods.

2. Related Works

Deep Learning on Point Clouds Extracting features from point clouds is the foundation of 3D scene understanding. Deep learning equipped methods mainly include point-based ones and voxel-based ones. Point-based methods, e.g., PointNet [35] and PointNet++ [36], directly operate on unstructured sets of points. Voxel-based methods [14, 38, 40, 30] transform the unordered and unstructured point sets into ordered and structured volumetric grids, and then perform 3D sparse convolutions on the grids. We adopt voxel-based methods for more efficient feature extraction.

Proposal-based Instance Segmentation Proposal-based approaches directly generate object proposals and predict masks inside each proposal. In the 2D domain, proposal-based methods employ 2D object detectors [13, 37, 8, 25] to generate region proposals and then predict masks inside each proposal. Mask R-CNN [16] extends Faster R-CNN [37] by adding a mask prediction. EmbedMask [45] introduces proposal embedding and pixel embedding so that pixels are assigned to instance proposals according to their embedding similarity. In the 3D domain, GSPN [44] proposes a generative shape proposal network for 3D object proposals following an analysis-by-synthesis strategy. 3D-SiS [17] takes both 3D geometry and 2D color images as input and combines 2D and 3D features through the back projection for a better prediction. 3D-BoNet [43] regresses a fixed set of bounding boxes and designs a novel association layer to match predicted boxes and ground truth boxes. 3D-MPA [9] predicts centers of instances and employs a graph convolutional network to refine proposal features. GICN [28] approximates the distributions of instance centers as Gaussian center heatmaps and uses a center selection mechanism for choosing candidates.

Clustering-based Instance Segmentation Clustering-based approaches first predict point-wise labels and then use clustering methods to generate instance predictions. In the 2D domain, metric learning is widely used to group pixels. Fathi et al. [12] compute likelihoods of pixels and group similar pixels together within an embedding space. Bai and Urtasun [2] adopt energy maps to distinguish among individual instances. Kong and Fowlkes [21] assign all pixels to a spherical embedding for clustering. Neven et al. [33] introduce a learnable clustering bandwidth instead of learning the embedding using hand-crafted cost functions. Bert et al. [3] propose a discriminative loss function which encourages the network to map each pixel to a point in feature space so that pixels belonging to the same instance lie close together while different instances are separated by a wide margin. In the 3D domain, SGPN [41] proposes to learn a similarity matrix for all point pairs and merges similar points to generate instances. JSIS3D [34] adopts multi-value conditional random fields to form instance predictions. MTML [22] introduces a multi-task learning strategy for grouping points. OccuSeg [15] adopts learnt occupancy signals to guide clustering. PointGroup [20] proposes to cluster points based on dual coordinate sets and designs ScoreNet to predict scores for instances.

Our HAIS follows the clustering-based paradigm but differs from existing clustering-based methods in two terms. First, most clustering-based methods require complicated and time-consuming clustering procedures, but our HAIS adopts a much more concise pipeline and keeps high efficiency. Second, previous methods usually group points according to point-level embeddings, without the instance-level correction. Our HAIS introduces the set aggregation and intra-instance prediction to refine the instance at the object level.
3. Method

The overall architecture of HAIS is depicted in Fig. 2, which consists of four main parts. The point-wise prediction network (Sec. 3.1) extracts features from point clouds and predicts point-wise semantic labels and center shift vectors. The point aggregation module (Sec. 3.2) forms preliminary instance predictions based on point-wise prediction results. The set aggregation module (Sec. 3.3) expands incomplete instances to cover missing parts, while the intra-instance prediction network (Sec. 3.4) smooths instances to filter out outliers.

3.1. Point-wise Prediction Network

The point-wise prediction network takes the point cloud \( P \in \mathbb{R}^{N \times K} \) as input, where \( N \) is the number of points and \( K \) is the number of channels. \( K \) is normally set as 6 for colors \( r, g, b \) and locations \( x, y, z \). The submanifold sparse convolution [14] is widely used in 3D perception methods [20, 22, 26, 9] to extract features from point clouds. Following the common practice, we first convert the point cloud data into regular volumetric grids. Then a UNet-like structure with submanifold sparse convolution [39, 14] for point-wise feature learning. Then, we use the spatial constraint of points to perform point aggregation with fixed bandwidth. Based on point aggregation results, set aggregation with dynamic bandwidth is performed to form instance proposals. The intra-instance prediction is designed for outlier filtering and mask quality scoring.

\[
L_{\text{shift}} = \sum_{P_i \in P} \mathbb{I}(p_i \in P_{fg}) \cdot \sum_{P_i \in P} L(p_i),
\]

\[
L(p_i) = w(p_i) \cdot (\parallel \Delta x_i^\text{shift} - \Delta x_i^\text{pred} \parallel_1 \cdot \mathbb{I}(p_i \in P_{fg})),
\]

\[
w(p_i) = \min(\parallel \Delta x_i^\text{shift} \parallel_2, 1).
\]

\( \mathbb{I}(\cdot) \) is the indicator function. \( P \) and \( P_{fg} \) denote the whole point set and the foreground point set respectively. Background points are ignored in \( L_{\text{shift}} \). \( w(p_i) \) serves as a point-wise weighted term. Points closer to the instance center less rely on the center shift vectors and should contribute less to the loss.

3.2. Point Aggregation

In the 3D space, points of the same instance are inherently adjacent to each other. It’s intuitive to utilize this spatial constraint for clustering. Thus, based on the semantic label and the center shift vector, we use a basic and compact clustering method to get preliminary instances. First, as shown in Fig. 3(b), according to the point-wise center shift vector \( \Delta x_i \), we shift every point \( x_i^\text{origin} \) toward its instance center, making the points of the same instance spatially closer to each other. The shifted coordinate is computed as,

\[
x_i^\text{shift} = x_i^\text{origin} + \Delta x_i.
\]

Second, we ignore background points and regard each foreground point as a node. For every pair of nodes, if they have the same semantic label and their spatial distance is smaller than a fixed spatial clustering bandwidth \( r_{\text{point}} \), an edge between these two nodes is created. After traversing...
all the pairs of node and establishing edges, the whole point cloud is separated into multiple independent sets, as shown in Fig. 3(c). Each set can be viewed as a preliminary instance prediction.

3.3. Set Aggregation

Fig. 4 shows the distribution of instance size (the number of points in an instance) of the ground truth and the point aggregation results. Compared with the sizes of ground truth instances, point aggregation generates a much larger number of instance predictions with small sizes. It is because center shift vectors are not totally accurate. Point aggregation cannot guarantee that all the points in an instance are grouped together. As illustrated in Fig. 3(d), most points with accurate center shift vectors can be clustered together to form incomplete instance predictions. We call these instances “primary instances”. But a minority of points with poor center shift vector predictions split from the majority and form fragmentary instances with small sizes, which we call “fragments”. Fragments are too small in size to be regarded as complete instances, but are possible to be the missing part of primary instances. Considering the large number of fragments, it’s not appropriate to directly filter out fragments with a hard threshold. Intuitively, we can aggregate primary instances and fragments at set level to generate complete instance predictions.

Based on the above clues, we propose the set aggregation to smooth the instance predictions generated by point aggregation, which is shown in Fig. 3(d). The detailed procedure is provided in Alg. 1. Briefly, if the following two conditions are satisfied, we consider the fragment \( m \) to be a part of the primary instance \( n \). First, among all the primary instances that have the same semantic label with the fragment \( m \), the primary instance \( n \) is the one whose geometric center is closest to the fragment \( m \). Secondly, for the fragment \( m \) and the primary instance \( n \), the distance between their geometric center should be smaller than \( r_{\text{set}} \), which is the dynamic clustering bandwidth defined as,

\[
r_{\text{set}} = \max(r_{\text{size}}, r_{\text{cls}}),
\]

\[
r_{\text{size}} = \alpha \sqrt{S_{n_{\text{prim}}}},
\]

where \( S_{n_{\text{prim}}} \) is the primary instance \( n \)’s size, \( r_{\text{cls}} \) denotes the class-specific bandwidth, which is the statistical average instance radii of the specific class. The distribution of instance size after the set aggregation is shown in Fig. 4. A large amount of fragments are combined together with primary instances to form instances with higher quality.

Figure 3. Illustrations of hierarchical aggregation. Points with different colors belong to different categories. Black points belong to background. (a): Points distributed in real 3D space. (b): After applying the center shift vector to each point, points belong to the same instance are closer in 3D space. (c): Point aggregation. Aggregating points into sets based on fixed spatial clustering bandwidth. (d): Set aggregation. Primary instances absorb surrounding fragments with dynamic clustering bandwidth to form complete instances.

Figure 4. Distribution of instance size. The instance size is defined as the number of points inside an instance. Blue, green and red correspond to point aggregation results, set aggregation results and ground truth, respectively. The statistics are based on the ScanNet v2 [6] validation set.
Algorithm 1: Set aggregation. \( N_{\text{frag}} \) is the number of fragments. \( N_{\text{prim}} \) is the number of primary instances.

Data:
- Fragments: \( \{ I^1_{\text{frag}}, I^2_{\text{frag}}, \ldots, I^{N_{\text{seg}}}_{\text{frag}} \} \)
- Primary instances: \( \{ I^1_{\text{prim}}, I^2_{\text{prim}}, \ldots, I^{N_{\text{min}}}_{\text{prim}} \} \)
- Centers of fragments: \( \{ c^1_{\text{frag}}, c^2_{\text{frag}}, \ldots, c^{N_{\text{inst}}}_{\text{frag}} \} \)
- Centers of primary instances: \( \{ c^1_{\text{prim}}, c^2_{\text{prim}}, \ldots, c^{N_{\text{prim}}}_{\text{prim}} \} \)
- Class labels of fragments: \( \{ L^1_{\text{frag}}, L^2_{\text{frag}}, \ldots, L^{N_{\text{seg}}}_{\text{frag}} \} \)
- Class labels of primary instances: \( \{ L^1_{\text{prim}}, L^2_{\text{prim}}, \ldots, L^{N_{\text{prim}}}_{\text{prim}} \} \)
- Dynamic set aggregation bandwidths: \( \{ r^1_{\text{set}}, r^2_{\text{set}}, \ldots, r^{N_{\text{prim}}}_{\text{set}} \} \)

Result:
A set of refined instances: \( \{ I^1_{\text{prim}}, I^2_{\text{prim}}, \ldots, I^{N_{\text{prim}}}_{\text{prim}} \} \)

1. for \( m = 1 \rightarrow N_{\text{frag}} \) do
   2. \( \text{index} = -1 \)
   3. \( d_{\text{min}} = +\infty \)
   4. for \( n = 1 \rightarrow N_{\text{prim}} \) do
      5. if \( L^m_{\text{frag}} = L^n_{\text{prim}} \) and
         6. \( ||c^m_{\text{frag}} - c^n_{\text{prim}}|| < d_{\text{min}} \) then
            7. \( \text{index} = n \)
            8. \( d_{\text{min}} = ||c^m_{\text{frag}} - c^n_{\text{prim}}|| \)
      9. end
   10. end
   11. if \( d_{\text{min}} < r^{\text{index}}_{\text{set}} \) then
       12. \( I^1_{\text{prim}} = I^\text{index}_{\text{prim}} \cup I^m_{\text{frag}} \)
   13. end
14. end
15. return \( \{ I^1_{\text{prim}}, I^2_{\text{prim}}, \ldots, I^{N_{\text{prim}}}_{\text{prim}} \} \)

Figure 5. The details of the intra-instance prediction network.

3.4. Intra-instance Prediction Network

The hierarchical aggregation may mistakenly absorb fragments belonging to other instances, generating inaccurate instance predictions. Thus, we propose the intra-instance prediction network for further refining the instances, as shown in Fig. 5. First, we crop instance point cloud patches as input and use the 3D submanifold sparse convolution network to extract features inside instances. After intra-instance feature extraction, the mask branch predicts binary masks to distinguish the instance foreground and background. For every predicted instance, we choose the best matched GT (ground truth) as the mask supervision. The overlapped parts between the predicted instance and the GT are assigned with positive labels, and others are assigned with negative labels. Low quality instances (low IoU with GT) contain little instance-level information and are valueless to optimizing the mask branch. Thus, only the instances with IoU higher than 0.5 are used as training samples, while others are ignored. For mask prediction, the loss is formulated as,

\[
L_{\text{mask}} = - \frac{1}{N_{\text{ins}}} \sum_{i=1}^{N_{\text{ins}}} \{ \mathbb{1}(iou_i > 0.5) \cdot \sum_{i=1}^{N_{\text{inst}}} \left[ y_j \cdot \log(\hat{y}_j) + (1 - y_j) \cdot \log(1 - \hat{y}_j) \right] \},
\]

where \( N_{\text{ins}} \) denotes the number of instances and \( N_i \) denotes the point number of instance \( i \).

Besides mask prediction, the instance certainty score is needed for ranking among instances. We utilize masks for better scoring instances, as illustrated in Fig. 5. Firstly, masks are used to filter out features of background point, which would be noise for scoring. The remained foreground features are sent into a MLP with a sigmoid layer to predict the instance certainty scores. Secondly, inspired by [18, 19, 23], we regard the IoUs between predicted masks and GT masks as the mask quality and use them to supervise instance certainty. For the score prediction, the loss is formulated as,

\[
L_{\text{score}} = - \frac{1}{N_{\text{ins}}} \sum_{i=1}^{N_{\text{ins}}} \left[ iou_i \cdot \log(\hat{s}_i) + (1 - iou_i) \cdot \log(1 - \hat{s}_i) \right].
\]

Ablation studies in Sec. 4.4 further demonstrate that using masks to assist score prediction boosts performance.

3.5. Multi-task Training

The whole network is trained from scratch in an end-to-end manner and optimized by a joint loss consisting of several loss terms,

\[
L = L_{\text{seg}} + L_{\text{shift}} + L_{\text{mask}} + L_{\text{score}},
\]

where \( L_{\text{seg}} \) is the cross-entropy loss of semantic scores, and \( L_{\text{shift}}, L_{\text{mask}} \) and \( L_{\text{score}} \) are defined in Eq. 1, 4 and 5 respectively.

3.6. NMS-free and Single-forward Inference

Proposal-based methods [9, 28] usually require dense proposals for better covering instances. And many
pipeline with higher efficiency. A single-forward design, HAIS keeps a much more concise algorithm. It only requires a compact single-forward inference procedure to generate accurate predictions. With the NMS-free and fast inference, HAIS achieves the SOTA performance, outperforming all other methods by a large margin.

| Method     | Ap50 | bed   | bookshelf | cabinet | chair | counter | curtain | desk | door | otherfu. | picture | fridge. | sink | sofa | table | toilet | window |
|------------|------|-------|-----------|---------|-------|---------|---------|------|------|----------|---------|---------|------|------|-------|--------|--------|
| DPC [10]   | 35.5 | 50.0  | 51.7      | 46.7    | 22.8  | 42.2    | 13.3    | 40.5 | 11.1 | 20.5     | 24.1   | 7.5     | 23.3 | 30.6 | 44.5  | 43.9   | 45.7   | 97.4   | 23.0  |
| 3D-SIS [17]| 38.2 | 100.0 | 43.2      | 24.5    | 19.0  | 57.7    | 1.3     | 26.3 | 3.3  | 32.0     | 24.0   | 7.5     | 42.2 | 85.7 | 11.7  | 69.9   | 27.1   | 88.3   | 23.5  |
| MASC [26]  | 44.7 | 52.8  | 55.5      | 38.1    | 38.2  | 63.3    | 0.2     | 50.9 | 26.0 | 36.1     | 43.2   | 32.7    | 45.1 | 57.1 | 36.7  | 63.9   | 38.6   | 98.0   | 27.6  |
| PanopticFusion [32] | 47.8 | 66.7  | 71.2      | 59.5    | 25.9  | 55.0    | 0.0     | 61.3 | 17.5 | 25.0     | 43.4   | 43.7    | 41.1 | 85.7 | 48.5  | 59.1   | 26.7   | 94.4   | 35.0  |
| 3D-BoNet [43] | 48.8 | 100.0 | 67.2      | 59.0    | 30.1  | 48.4    | 9.8     | 62.0 | 30.6 | 34.1     | 25.9   | 12.5    | 43.3 | 79.6 | 40.2  | 49.9   | 51.3   | 93.0   | 43.9  |
| MTML [22]  | 54.9 | 100.0 | 80.7      | 58.8    | 32.7  | 64.7    | 0.4     | 81.5 | 18.0 | 41.8     | 36.4   | 18.2    | 44.5 | 100.0| 44.2  | 68.8   | 57.1   | 100.0  | 39.6  |
| 3D-MPA [9] | 61.1 | 100.0 | 83.3      | 76.5    | 52.6  | 75.6    | 13.6    | 58.8 | 18.0 | 41.8     | 36.4   | 18.2    | 44.5 | 100.0| 44.2  | 68.8   | 57.1   | 100.0  | 39.6  |
| PointGroup [20] | 63.6 | 100.0 | 75.4      | 62.4    | 50.5  | 79.7    | 11.6    | 69.6 | 38.4 | 44.1     | 55.9   | 47.6    | 59.6 | 100.0| 66.6  | 75.6   | 55.6   | 99.7   | 51.3  |
| GICN [28]  | 63.8 | 100.0 | 89.5      | 80.0    | 48.0  | 67.6    | 14.4    | 73.7 | 35.4 | 44.7     | 40.0   | 36.5    | 70.0 | 100.0| 56.9  | 83.6   | 59.9   | 100.0  | 47.3  |
| HAIS       | 69.9 | 100.0 | 84.9      | 82.0    | 67.5  | 80.8    | 27.9    | 75.7 | 46.5 | 51.7     | 59.6   | 55.9    | 60.0 | 100.0| 65.4  | 76.7   | 67.6   | 99.4   | 56.0  |

Table 1. Quantitative comparison on the testing set of ScanNet v2 [6] benchmark. Our HAIS achieves the SOTA performance, outperforming all other methods by a large margin.

| Method     | Cov   | W-Cov | Prec   | Rec   |
|------------|-------|-------|--------|-------|
| SGPN† [41] | 32.7  | 35.5  | 36.0   | 28.7  |
| ASIS† [42] | 44.6  | 47.8  | 55.3   | 42.4  |
| PointGroup† [20] | - | - | 61.9 | 62.1 |
| HAIS       | 64.3  | 66.0  | 71.1   | 65.0  |

Table 2. Quantitative comparison on S3DIS [1]. Methods marked with † are evaluated on Area 5 and those marked with ‡ are on the 6-fold cross validation. Our method significantly outperforms previous methods in terms of mCov (coverage), mWCov (weighted coverage), mean precision (mPrec) and mean recall (mRec).

clustering-based methods [22, 20] adopt multiple clustering strategies to generate redundant instance predictions. Thus, non-maximum suppression (NMS) or other post-processing steps which function as NMS, are widely required for removing duplicated instance predictions. But in our HAIS, one point is only clustered into a single instance in point aggregation, resulting in no overlap among instance predictions. We can directly use instance certainty scores to rank the instances and take the ones with the highest scores as final predictions, not requiring any post-processing steps. Besides, iterative clustering procedure is widely used in clustering based methods [15, 34], which refines predictions step by step but is time-consuming. In contrast, HAIS only requires a compact single-forward inference procedure to generate accurate predictions. With the NMS-free and single-forward design, HAIS keeps a much more concise pipeline with higher efficiency.

4. Experiments

In this section, we first present our experimental settings (Sec. 4.1). Then, we provide both quantitative (Sec. 4.2) and qualitative evaluations (Sec. 4.3) to demonstrate the effectiveness of HAIS. To better validate each component of our method, we provide detailed ablation studies (Sec. 4.4). And evaluation on inference speed (Sec. 4.5) is offered to prove the efficiency of HAIS.

4.1. Experimental Settings

ScanNet v2 The ScanNet v2 [6] dataset is the most accepted and robust dataset in 3D instance segmentation. It contains 1,613 scans with 3D object instance annotations. The dataset is split into the training, validation and testing set, each with 1,201, 312, and 100 scans, respectively. 18 object categories are used for instance segmentation evaluation. To fairly compare with other works, we report results on the testing set which come from the official evaluation server. For ablation studies, we report results on the validation set. Keeping the same with the ScanNet v2 benchmark, we use the mean average precision with an IoU threshold of 0.5 (AP50) as the main evaluation metric. We also report the mean average precision at the overlap 0.25 (AP25) and overlaps from 0.5 to 0.95 (AP) in the ablation study.

S3DIS To validate the generalization of HAIS, we also conduct experiments on the S3DIS [1] dataset. S3DIS has 3D scans across six areas with 271 scenes in total. Each point is assigned with one label out of 13 semantic classes. All the 13 classes are used in instance evaluation. We report results evaluated on both Area 5 and the 6-fold cross validation. We use coverage (mCov), weighted coverage (mWCov), mean precision (mPrec) and mean recall (mRec) with the IoU threshold of 0.5 as evaluation metrics.

Experimental details Our model is trained on one single Titan X GPU with a batch size of 4 for 120k iterations. The
initial learning rate is 0.001 and decays with a cosine anneal schedule [29]. For stability and efficiency, we do not adopt set aggregation during the training phase. And this does not affect the effectiveness of set aggregation during inference. We set the voxel size as 0.02m following the common practice [26, 20]. The bandwidth \( r_{\text{point}} \) for point aggregation is set as 0.03. \( \alpha \) in Eq. 3 is set as 0.01. The final predictions containing less than 100 points are filtered out before evaluation.

### 4.2. Quantitative Evaluation

**ScanNet v2** In Tab. 1, we compare our HAIS with other methods on the unreleased testing set of ScanNet v2 [6] benchmark. HAIS achieves the highest \( AP_{50} \) of 69.9\%, ranking the first place on the leaderboard of ScanNet v2 and surpassing the previous state-of-the-art (SOTA) work [28] by 6.1\%. For results on each class, our method achieves the best performance in 12 out of 18 classes.

**S3DIS** In Tab. 2, we present the results on S3DIS. HAIS achieves much higher results than other methods in terms of all the widely-used metrics (mCov, mWCoV, mPre and mRec). ScanNet v2 and S3DIS are quite different in terms of category, scene style and point cloud density. The SOTA performances of HAIS on both ScanNet v2 and S3DIS prove the high generalization ability.

### 4.3. Qualitative Evaluation

Fig. 6 visually shows the effectiveness of the hierarchical aggregation and intra-instance prediction. For objects with large sizes and fragmentary point clouds, grouping all points together is quite challenging. We can observe that with the proposed hierarchical aggregation and intra-instance prediction, precise instance segmentation masks are obtained.

### 4.4. Ablation Study

To validate the design of HAIS, we perform a series of ablation studies on the ScanNet validation set.

**Ablation on the Hierarchical Aggregation and Intra-instance Prediction Network** Tab. 3 proves the effectiveness of the hierarchical aggregation and intra-instance prediction network. The intra-instance prediction promotes results by 2.7\% \( AP \), 2.4\% \( AP_{50} \) and 0.3\% \( AP_{25} \). And the set aggregation further improves results by 1.0\% \( AP \), 0.7\% \( AP_{50} \) and 0.7\% \( AP_{25} \).

**Using Masks to Filter Points and Calculate IoU** In the intra-instance prediction network, masks are used to assist certainty score predictions, i.e., filtering out features of background and calculating IoU which is the supervision signal of the instance certainty, as shown in Fig. 5. An alternative method is directly using the whole instance features without filtering background points to predict scores and using the IoU between the original input instance and the GT to supervise the instance certainty. Ablation experiments in Tab. 4 show that, using masks to filter points and calculate IoU improves the results. Filtering features with the mask can avoid the influence of the background noise. And masks are much more accurate than original input instances. The IoU between the mask and GT is more suitable for the supervision signal of the certainty score.

**Filtering Mask Training Samples** As shown in Tab. 5, compared with using all the instances as mask training...
samples, it’s better to filter out low quality instances with the IoU threshold of 0.5. Low quality instances usually covers few foreground points but a large amount of background points. These instances may bring ambiguity to the instance-level refinement. It’s beneficial to filter them out during training.

4.5. Inference Speed

For real-life applications, e.g., mixed reality and autonomous driving, the inference speed of the whole network is of critical importance. We evaluate the efficiency of HAIS and compare it with other methods, as shown in Tab. 6. The single scene inference time is highly correlated to the number of points in the point cloud and varies a lot. Following the evaluation method of [43, 15, 28], we use the whole validation set inference time of ScanNet v2 for fair comparison on the efficiency. HAIS only takes 128 seconds to infer all the 312 scans in the validation set, achieving the highest efficiency among all methods. On average, per scan inference latency of HAIS is 410 ms. Point-wise prediction network, point aggregation, set aggregation and the intra-instance prediction network takes 172, 125, 4, 109 ms, respectively.

5. Conclusion

We propose HAIS, a concise bottom-up approach for 3D instance segmentation. We introduce the hierarchical aggregation to generate instance predictions in a two-step manner and the intra-instance prediction for more fine-grained instance predictions. Experiments on ScanNet v2 and S3DIS demonstrate the effectiveness and generalization of our method. HAIS also retains much better inference speed than all existing methods, showing its practicability in most scenarios, especially latency-sensitive ones.

Acknowledgement: This work was in part supported by NSFC (No. 61876212 and No. 61733007) and the Zhejiang Laboratory under Grant 2019NB0AB02.

References

[1] Iro Armeni, Ozan Sener, Amir Roshan Zamir, Helen Jiang, Ioannis K. Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In CVPR, 2016.

[2] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In CVPR, 2017.
Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. \textit{arXiv:1708.02551}, 2017.

Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In \textit{CVPR}, 2019.

Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask R-CNN. In \textit{ECCV}, 2020.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In \textit{CVPR}, 2017.

Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In \textit{ECCV}, 2016.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: Object detection via region-based fully convolutional networks. In \textit{NeurIPS}, 2016.

Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In \textit{CVPR}, 2020.

Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field size of point convolutions on 3D point clouds. In \textit{ICRA}, 2020.

Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. \textit{arXiv:2105.01928}, 2021.

Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P. Murphy. Semantic instance segmentation via deep metric learning. \textit{arXiv:1703.10277}, 2017.

Ross B. Girshick. Fast R-CNN. In \textit{ICCV}, 2015.

Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In \textit{CVPR}, 2018.

Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3D instance segmentation. In \textit{CVPR}, 2020.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. \textit{PAMI}, 2020.

Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3D semantic instance segmentation of RGB-D scans. In \textit{CVPR}, 2019.

Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring R-CNN. In \textit{CVPR}, 2019.

Boru Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yunming Jiang. Acquisition of localization confidence for accurate object detection. In \textit{ECCV}, 2018.

Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Ch-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3D instance segmentation. In \textit{CVPR}, 2020.

Shu Kong and Charless C. Fowlkes. Recurrent pixel embedding for instance grouping. In \textit{CVPR}, 2018.
In this supplementary material, we provide the detailed inference latency of main individual components of HAIS and compare it with other methods. Besides, we provide more qualitative results to demonstrate the effectiveness of HAIS.

A. Detailed Inference Time

Tab. 7 shows the inference time of main components of different methods. Compared with other methods which require time-consuming clustering and post processing procedures, our HAIS keeps a much more efficient pipeline. The point-wise prediction network, point aggregation, set aggregation and intra-instance prediction network takes 172, 125, 4 and 109 ms, respectively.

B. Additional Qualitative Results

We show more qualitative results on the validation split of the ScanNet v2 dataset in Fig. 7. The predicted center shift vectors of some points are not accurate and a large amount of instance fragments come into being. By introducing the hierarchical aggregation and intra-instance prediction, HAIS generates fine-grained instance predictions.
Table 7. Inference time of main components of different methods on the ScanNet v2 validation set. For fair comparison, data in this table is measured on the same type of GPU (Titan X).

| Method       | Component inference time (msec)                  | Per frame inference time (msec) |
|--------------|--------------------------------------------------|---------------------------------|
| SGPN [41]    | backbone (GPU): 2080                             | 158439                          |
|              | group merging (CPU): 149000                      |                                 |
|              | block merging (CPU): 7119                        |                                 |
| ASIS [42]    | backbone (GPU): 2083                             | 181913                          |
|              | mean shift (CPU): 172711                         |                                 |
|              | block merging (CPU): 7119                        |                                 |
| GSPN [44]    | backbone (GPU): 1612                             | 12702                           |
|              | point sampling (GPU): 9559                       |                                 |
|              | neighbour search (CPU): 1500                     |                                 |
| 3D-BoNet [43]| backbone (GPU): 2083                             | 9202                            |
|              | SCN (GPU): 667                                   |                                 |
|              | block merging (CPU): 7119                        |                                 |
| OccuSeg [15] | backbone GPU): 189                              | 1904                            |
|              | supervoxel (CPU): 1202                           |                                 |
|              | clustering (GPU+CPU): 513                        |                                 |
| PointGroup [20]| backbone (GPU): 128                             | 452                             |
|              | clustering (GPU+CPU):221                         |                                 |
|              | ScoreNet (GPU): 103                              |                                 |
| GICN [28]    | backbone (GPU): 1497                             | 8615                            |
|              | SCN (GPU): 667                                   |                                 |
|              | block merging(CPU): 7119                         |                                 |
| HAIS         | point-wise prediction(GPU): 172                  | 410                             |
|              | point aggregation(GPU+CPU): 125                  |                                 |
|              | set aggregation(GPU): 4                          |                                 |
|              | intra-instance prediction(GPU): 109              |                                 |
Figure 7. Qualitative results on ScanNet v2 [6]. From left to right: input point cloud, ground truth semantic label, predicted semantic label, shifted coordinate, ground truth instance mask and predicted instance mask. Best viewed in color.