Top-Down Beats Bottom-Up in 3D Instance Segmentation

Maksim Kolodiazhnyi Anna Vorontsova Anton Konushin Danila Rukhovich
Samsung Research
{m.kolodiazhn, a.vorontsova, a.konushin, d.rukhovich}@samsung.com

Abstract

Most 3D instance segmentation methods exploit a bottom-up strategy, typically including resource-exhaustive post-processing. For point grouping, bottom-up methods rely on prior assumptions about the objects in the form of hyperparameters, which are domain-specific and need to be carefully tuned. On the contrary, we address 3D instance segmentation with a TD3D: the pioneering cluster-free, fully-convolutional and entirely data-driven approach trained in an end-to-end manner. This is the first top-down method outperforming bottom-up approaches in 3D domain. With its straightforward pipeline, it demonstrates outstanding accuracy and generalization ability on the standard indoor benchmarks: ScanNet v2, its extension ScanNet200, and S3DIS, as well as on the aerial STPLS3D dataset. Besides, our method is much faster on inference than the current state-of-the-art grouping-based approaches: our flagship modification is 1.9x faster than the most accurate bottom-up method, while being more accurate, and our faster modification shows state-of-the-art accuracy running at 2.6x speed. Code is available at https://github.com/SamsungLabs/td3d.

1. Introduction

With the emergence of AR/VR, 3D indoor scanning, and household robotics, 3D instance segmentation becomes a key technology facilitating scene understanding. It is a holistic and challenging task of finding objects in 3D point clouds, predicting their semantic labels, and assigning an instance ID for each object.

Two major 3D instance segmentation paradigms have been introduced so far [18]. *Bottom-up* methods learn per-point embeddings and use them to cluster points so that they form a set of proposals. *Top-down* directly predict instance proposals as object proxies, which are then filtered via non-maximum suppression, and refined via mask segmentation individually.

In 2D instance segmentation, most state-of-the-art methods follow the top-down paradigm. Unfortunately, 2D methods that work well on a pixel grid cannot be directly adapted to process unstructured and sparse 3D points, and bottom-up methods dominate the field of 3D point cloud processing. Accordingly, the recent progress in 3D instance segmentation has been associated with improving components of bottom-up approaches: different ways of selecting points to be grouped have been studied [11], advanced feature aggregation strategies have been proposed [3, 15], with estimates being refined via elaborate post-processing schemes [14, 19]. In the meantime, top-down methods have been out of the spotlight.

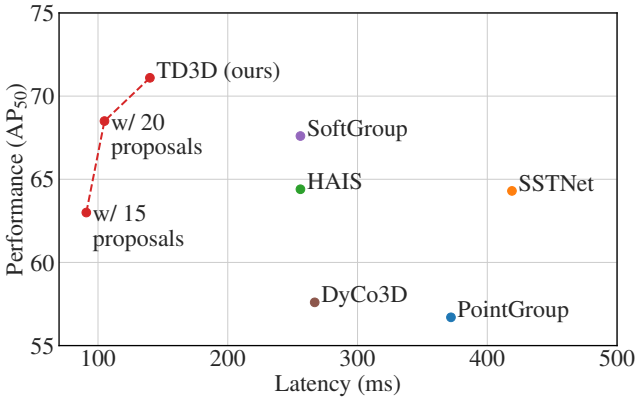

Figure 1. Prediction accuracy on ScanNet against latency. TD3D modifications (marked red) have a different number of proposals. Our top-performing default TD3D model surpasses existing methods in both detection accuracy and latency, while the faster modifications demonstrate an impressive inference speed with a comparable quality.

Nevertheless, bottom-up 3D instance segmentation methods have crucial drawbacks, limiting their performance. Besides being computationally-expensive, bottom-up approaches are sensitive to the values of numerous hyperparameters. Particularly, they might fail to find a proper balance between over-fragmented and accidentally merged masks and have limited generalization ability to complex scenes with objects of varying scales.

Our goal is to prove the top-down paradigm has great potential, which is yet unleashed in the 3D domain. In this

paper, we tackle the challenging 3D instance segmentation task with TD3D, a top-down, fully-convolutional, simple approach trained end-to-end in a fully data-driven way. We conduct extensive experiments on the ScanNet v2, ScanNet200, S3DIS and STPLS3D datasets, and report competitive results for all these benchmarks.

Overall, our contributions are three-fold:

- We develop the world's first fully sparse convolutional cluster-free 3D instance segmentation approach, dubbed TD3D;
- We introduce the first top-down method that supersedes bottom-up competitors, hence questioning the dominating paradigm in 3D instance segmentation;
- We establish a state-of-the-art in both accuracy and speed: our flagship model is 1.9x faster than the best bottom-up approach, and we also show that state-of-the-art accuracy can be achieved with a 2.6x speed-up;

2. Related Work

Bottom-up methods. Up until very recently, grouping-based bottom-up methods have dominated the field. In SGPN [20], a similarity matrix for all 3D point pairs is learned, and the most similar points are assembled into instances. 3D-SIS [12] utilizes RGB images as an additional source of data and merges 3D features from a point cloud with backprojected 2D features extracted from RGB images. ASIS [21] uses spatial discriminative loss to learn point-level embeddings and generates instance masks via a mean-shift algorithm. 3D-MPA [7] predicts instance centers and refines initial instance proposals with a graph convolutional network. Additional estimates are used to guide clustering, e.g., OccuSeg [8] predicts occupancy, while PointGroup [14] assigns 3D points with semantic labels and center votes. In HAIS [3], clustering is performed in a hierarchic manner. SSTNet [15] aggregates point-wise semantic and instance-level features, using a semantic superpoint tree (SST) with superpoints as leaves. SoftGroup [19] leverages a 3D sparse network to group 3D points according to the predicted soft semantic scores and refines the obtained proposals with a 3D U-Net-like network. DyCo3D [11] also employs refinement, yet incorporates dynamic convolutions.

Top-down methods. Top-down 3D instance segmentation methods directly generate object proposals and then predict or refine masks for each proposal. 3D-BoNet [22] applies Hungarian matching and outputs a fixed set of proposals in the form of non-oriented 3D bounding boxes. Instead of regressing 3D bounding boxes, GSPN [23] employs an analysis-by-synthesis strategy to predict instance shapes. NeuralBF [18] generates the affinity of points in the point cloud to a query point and uses coordinate networks representing convex domains to model the spatial affinity in the neural bilateral filter.

3D object detection. Modern 3D object detection methods can be categorized into voting-based, transformer-based, and 3D convolutional. Voting-based methods extract per-point features, merge them into an object proposal, and accumulate features of points within each group; overall, they use point grouping similar to bottom-up 3D instance segmentation methods. Instead of domain-specific heuristics and hyperparameters, transformer-based methods use end-to-end learning and forward pass on inference. Both voting- and transformer-based methods have scalability issues, making them impractical. Differently, top-down 3D convolutional methods represent point clouds as voxels, which makes them more memory-efficient and allows scaling to large scenes without sacrificing point density. Up until very recently, such methods lacked accuracy, yet the last advances in the field allowed developing fast, scalable, and accurate methods [17].

3. Proposed Method

The proposed method runs in two stages. First, it detects objects in a point cloud and extracts corresponding bounding boxes. These bounding boxes are interpreted as initial object proposals and then refined with a lightweight network to obtain final instance masks (Fig. 3). All operations within the pipeline are implemented through 3D sparse convolutions.

3.1. Proposal Generation

3D Bounding Boxes. We employ a 3D object detection method that outputs 3D object bounding boxes alongside object categories and confidence scores. A 3D object bounding box is parameterized as (x, y, z, w, l, h), where x, y, z denote the coordinates of the center of a bounding box, while w, l, h are its width, length, and height, respectively.

Any conventional 3D object detection method can be employed for this purpose. We aim to avoid point grouping and follow a top-down paradigm at each stage of our pipeline, so we narrowed our search to the 3D convolutional-based methods. As the result, we opted for fast and efficient, fully-convolutional FCAF3D [17].

The backbone in FCAF3D is a sparse ResNet [10] with sparse 3D convolutions. In the neck, the features on each level are processed with one sparse transposed 3D convolution and one sparse 3D convolution. To prevent sparsity growth, at most N_{vox} voxels with the highest classification probabilities are selected at each level, where N_{vox} equals the number of input points N_{pts}. The anchor-free FCAF3D head consists of two parallel sparse convolutional layers

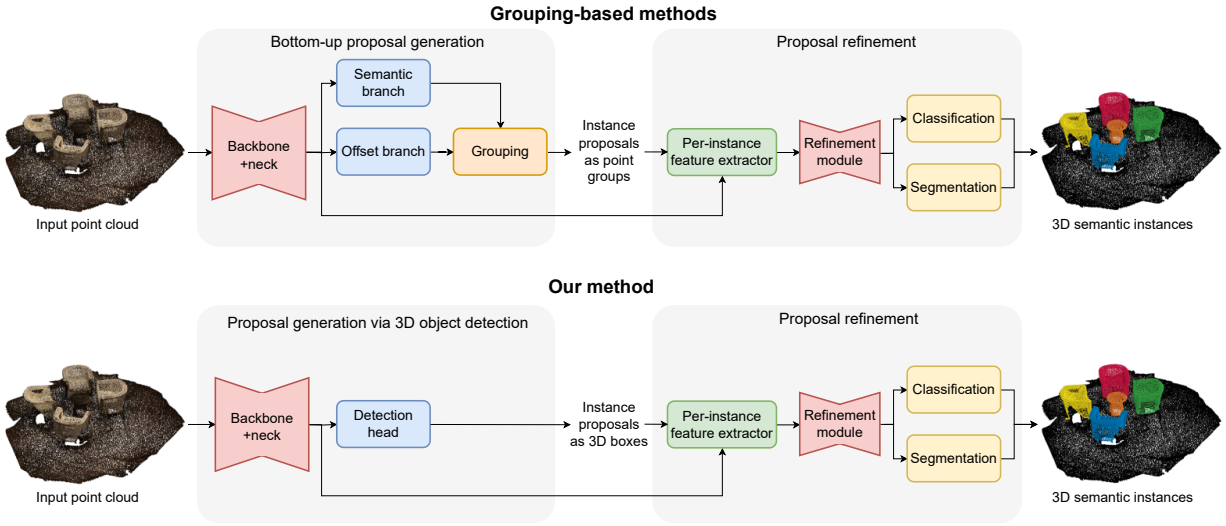

Figure 2. Scheme of TD3D in comparison with state-of-the-art grouping-based methods (e.g., PointGroup [14], SoftGroup [19]). Our proposal refinement is the same as in SoftGroup [19], while proposal generation significantly differs. Specifically, bottom-up proposal generation returns *point groups*, while we leverage 3D object detection that outputs instance proposals defined by *3D object bounding boxes* instead.

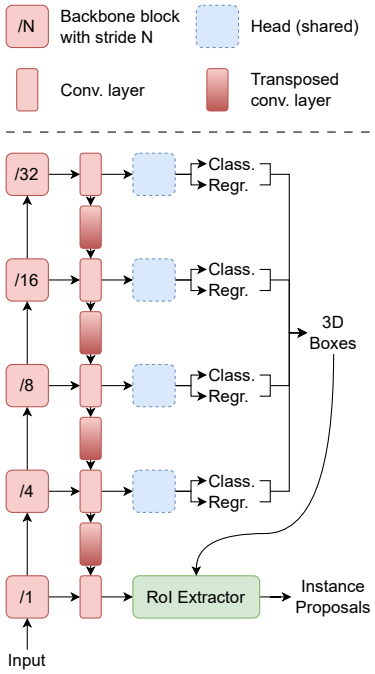

Figure 3. Our proposal generation scheme. 3D bounding boxes are estimated from the downsized 3D feature maps at higher network levels. Then, the predicted bounding boxes are used to select features from the 3D feature map of the original resolution.

with weights shared across feature levels. For each location $(\hat{x}, \hat{y}, \hat{z})$, these layers output classification probabilities $\hat{\boldsymbol{p}}$ and bounding box regression parameters $\boldsymbol{\delta}$ [17].

Object Proposals. Given 3D bounding boxes, we extract the features from the corresponding regions of the 3D feature maps, which consists of voxels, where each voxel is defined by its own coordinate (x, y, z) and feature vector $\vec{f}$. We have five 3D feature maps of decreasing resolution: one map per feature level in the neck. At the first level, the 3D feature map has the same resolution as the input voxelized point cloud. The 3D feature maps at the second, third, fourth, and fifth level are 4x, 8x, 16x, and 32x smaller; they are used to estimate 3D bounding boxes. Then, these 3D bounding boxes and the first-level 3D feature map are processed with a RoI extractor, which selects voxels, whose centers are inside the given bounding box (Fig. 3).

Eventually, these voxels with corresponding 3D features serve as initial object proposals. The pseudocode for RoI extraction is provided below (Algorithm 1).

3.2. Proposal Refinement

Our proposal refinement module takes voxels with features from RoI extractor as inputs and predicts final instance masks. For this purpose, we consider a 3D tiny U-Net network (a U-Net style network with few layers) solving a binary segmentation task, that classifies voxels into *foreground* and *background*. For each voxel, all points inside this voxel are assigned with the same *foreground* or *background* label predicted by U-Net, which gives final per-point instance masks.

Algorithm 1: RoI Extractor

Input:
Feature Map:
$F = \{v_i : (x_{v_i}, y_{v_i}, z_{v_i}, \vec{f_i}) \mid i = 1 \dots n\}$
Bounding Boxes:
$B = \{(x_{b_j}, y_{b_j}, z_{b_j}, w_j, l_j, h_j) \mid j = 1 \dots k\}$
Output:
Proposals:
$P = \{P_t = \{v_0, v_1, \dots v_{s_t}\} \mid t = 1 \dots m,\ m \leq k\}$
for *j=1 ... k* **do**
  $P_j := \emptyset$
for *i=1 ... n* **do**
  for *j=1 ... k* **do**
    $\delta x_1 = x_{v_i} - x_{b_j} + w_j/2$
    $\delta x_2 = x_{b_j} - x_{v_i} + w_j/2$
    $\delta y_1 = y_{v_i} - y_{b_j} + l_j/2$
    $\delta y_2 = y_{b_j} - y_{v_i} + l_j/2$
    $\delta z_1 = z_{v_i} - z_{b_j} + h_j/2$
    $\delta z_2 = z_{b_j} - z_{v_i} + h_j/2$
    if $\min(\delta x_1, \delta x_2, \delta y_1, \delta y_2, \delta z_1, \delta z_2) > 0$ **then**
      $P_j := P_j \bigcup \{v_i\}$
for *j=1 ... k* **do**
  if $|P_j| < threshold$ **then**
    $P := P \backslash P_j$

3.3. Training Procedure

We train our method end-to-end, updating both proposal generation and proposal refinement models simultaneously. The total loss is a sum of two proposal generation losses $\mathcal{L}_{cls}$, $\mathcal{L}_{reg}$ and a proposal refinement loss $\mathcal{L}_{seg}$:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \mathcal{L}_{seg}$$

Proposal generation. Our proposal generation model is inherited from FCAF3D [17], accordingly, we follow the original training procedure and use focal loss $\mathcal{L}_{cls}$ and IoU loss $\mathcal{L}_{reg}$ to penalize classification and regression errors, respectively. During training, this model outputs 3D object bounding boxes parameterized with their centers and sizes (length, width, height), which we consider as initial object proposals.

Proposal refinement. To train the proposal refinement model, it is essential to establish a correspondence between the proposals and the ground truth instances. This process has two stages. First, the centers of the 3D bounding boxes of the ground truth instances are calculated, and the ground truth instances are matched with the predicted 3D bounding boxes using the FCAF3D assigner [17]. For each ground truth bounding box, we select the last feature level where a 3D bounding box covers at least N_{loc} voxels (if there is no such a feature level, the first feature level is chosen). Each voxel covered with a ground truth 3D bounding box is assigned with a semantic label and an index of this 3D bounding box. Respectively, the predicted 3D bounding box encoded with this voxel gets the same label and index.

At the second stage, the IoU assigner is employed. For each predicted 3D bounding box, we calculate IoU with all ground truth bounding boxes, and select the ground truth bounding box with the maximum IoU score. If FCAF3D and IoU assigners assigned the same ground truth 3D bounding box for the predicted 3D bounding box, then this assignment is considered trusted, and the predicted bounding box gets the label and index of the corresponding ground truth 3D bounding box.

In [9], the predicted 3D bounding boxes that do not have a corresponding ground truth 3D bounding box with an IoU score exceeding the given threshold, are filtered out. However, our ablation study reveals this strategy to be suboptimal (Tab. 11).

Finally, based on the predicted 3D bounding box, a proposal is extracted, and each voxel of the proposal is assigned with a semantic label and the index of the predicted 3D bounding box.

Our binary segmentation model is trained via minimizing $\mathcal{L}_{seg}$, which is calculated as a BCE loss between predicted and ground truth instance masks.

4. Experiments

4.1. Experimental Settings

Datasets. The experiments are conducted on ScanNet v2 [6], ScanNet200 [16], S3DIS [1], and recently introduced STPLS3D [2]. ScanNet v2 [6] contains 1613 scans divided into training, validation, and testing splits of 1201, 312, and 100 scans, respectively. 3D instance segmentation is typically evaluated using 18 object classes. We report results on both validation and hidden test splits. ScanNet200 [16] extends the original ScanNet semantic annotation with fine-grained categories with the long-tail distribution. The training, validation, and testing splits are similar to the original ScanNet v2 dataset. The S3DIS dataset [1] features 272 scenes within 6 large areas. Following the standard evaluation protocol, we assess the segmentation quality on scans from Area 5, and via 6 cross-fold validation, using 13 semantic categories in both settings. STPLS3D [2] is a synthetic outdoor dataset emulating aerial photogrammetry. It covers 25 urban scenes of 6 km^2, densely annotated with 14 categories. We use the splits proposed in the original work [2].

Metrics. We use the average precision as a major metric. AP_{50} and AP_{25} are the scores obtained with IoU thresholds of 50% and 25%, respectively. AP is an average score with

Paradigm	Method	Conference	Validation			Test			Runtime
			AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}	(in sec)
Bottom-up	PointGroup [14]	CVPR'20	34.8	56.7	71.3	40.7	63.6	77.8	0.372
	SSTNet [15]	ICCV'21	**49.4**	64.3	74.0	**50.6**	69.8	78.9	0.419
	HAIS [3]	ICCV'21	43.5	64.4	75.6	45.7	69.9	80.3	0.256
	DyCo3D [11]	CVPR'21	35.4	57.6	72.9	39.5	64.1	76.1	0.267
	SoftGroup [19]	CVPR'22	45.8	67.6	78.9	50.4	**76.1**	86.5	0.266
Top-down	3D-SIS [12]	CVPR'19	-	18.7	35.7	16.1	38.2	55.8	>10
	GSPN [23]	CVPR'19	19.3	37.8	53.4	-	30.6	-	>10
	3D-BoNet [22]	NeurIPS'19	-	-	-	25.3	48.8	68.7	9.174
	NeuralBF [18]	WACV'23	36.0	55.5	71.1	35.3	55.5	71.8	-
	TD3D (ours)		47.3	**71.2**	**81.9**	48.9	75.1	**87.5**	**0.140**

Table 1: Results on ScanNet v2. The best results are **bold**, the second best are underlined. The runtime is measured using a single NVidia 3090 GPU. Our approach outperforms the previous state-of-art SoftGroup [19] on the validation subset, while being 1.9 times faster.

Method	AP		
	head	common	tail
CSC [13]	22.3	8.2	4.6
Mink34D [4]	24.6	8.3	4.3
LGround [16]	27.5	10.8	6.0
TD3D (ours)	**33.2**	**17.7**	**10.3**

Table 2: Results on the ScanNet200 test split. AP scores for the most frequent (*head* of distribution), common, and rare (*tail*) object categories are provided. The best results are **bold**. TD3D achieves 1.7x improvement of the previous state-of-the-art AP scores for *common* and *tail* categories.

IoU threshold varying from 50% to 95% with a step of 5%.

Implementation details. Our models are implemented using mmdetection3d framework [5] based on Pytorch. We use MinkUNet14B as a binary segmentation model at the refinement stage. We train for 330 epochs on a single NVidia 3090 GPU with the Adam optimizer. The batch size is 4, and the initial learning rate is set to 0.001 and is reduced by 10 times after 280 and 320 epochs. Other implementation details are similar to FCAF3D [17].

4.2. Comparison to Prior Work

ScanNet v2. Results for validation and test splits of ScanNet v2 are presented in Tab. 1. Overall, TD3D is on par with previous state-of-the-art SoftGroup [19] on the test split and shows superior results on validation. Another advantage of TD3D is its inference speed: according to the reported runtime, it is more than 1.8x faster than any method that performs grouping.

ScanNet200. We evaluate TD3D on the test split of ScanNet200 and report metrics in Tab. 2. For either frequent, common, or rare categories, our method demonstrates a solid superiority over the existing approaches. The gain is especially tangible for less frequent categories, where TD3D improves previous state-of-the-art metrics by approximately 1.7x times (+6.9 and +4.3 AP for *common* and *tail*, respectively).

S3DIS. According to the Tab. 3, TD3D surpasses other methods by at least +5.9 AP and +8.9 $Prec_{50}$ for Area 5 and +2.1 AP and +2.8 $Prec_{50}$ on 6-fold cross-validation. Meanwhile, if being pre-trained on ScanNet and fine-tuned on S3DIS, as proposed in [3,19], it consistently outperforms the previous state-of-the-art SoftGroup [19] in both testing scenarios and in terms of all metrics (Tab. 4).

STPLS3D. We evaluate TD3D behind the indoor domain, scoring unexpectedly high on STPLS3D. In Tab. 6, we compare our approach against strong baselines: evidently, TD3D sets a new state-of-art, superseding SoftGroup by impressive +8.1 AP and +8.0 AP_{50}.

Method	Area 5				6-fold CV			
	AP	AP_{50}	$Prec_{50}$	Rec_{50}	AP	AP_{50}	$Prec_{50}$	Rec_{50}
SGPN [20]	-	-	36.0	28.7	-	-	38.2	31.2
ASIS [21]	-	-	55.3	42.4	-	-	63.6	47.5
3D-BoNet [22]	-	-	57.5	40.2	-	-	65.6	47.6
OccuSeg [8]	-	-	-	-	-	-	72.8	60.3
3D-MPA [7]	-	-	63.1	58.0	-	-	66.7	64.1
PointGroup [14]	-	57.8	61.9	62.1	-	64.0	69.6	69.2
DyCo3D [11]	-	-	64.3	64.2	-	-	-	-
MaskGroup [24]	-	65.0	62.9	64.7	-	**69.9**	66.6	69.2
SSTNet [15]	42.7	59.3	65.5	64.2	54.1	67.8	73.5	73.4
TD3D (ours)	**48.6**	**65.1**	**74.4**	**64.8**	**56.2**	68.2	**76.3**	**74.0**

Table 3. Results on S3DIS. The best results are **bold**, the second best are underlined. Being superior in all metrics in both testing scenarios, our approach sets a new state-of-art in 3D instance segmentation.

Method	Area 5			6-fold CV		
	AP	$Prec_{50}$	Rec_{50}	AP	$Prec_{50}$	Rec_{50}
HAIS [3]	-	71.1	65.0	-	73.2	69.4
SoftGroup [19]	51.6	73.6	66.6	54.4	75.3	69.8
TD3D (ours)	**52.1**	**75.2**	**68.7**	**58.1**	**82.8**	**71.6**

Table 4. Results on S3DIS with the ScanNet v2 pre-training. The best results are **bold**. TD3D shows a solid improvement over SoftGroup [19] in terms of all metrics.

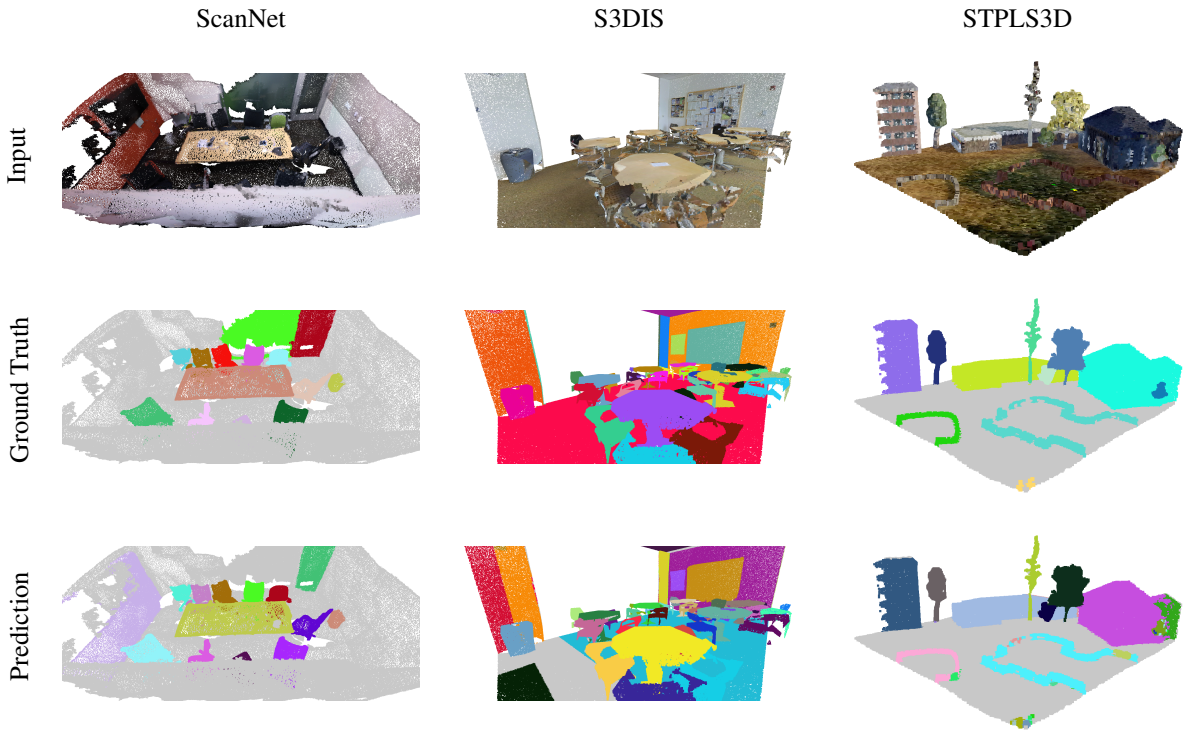

Figure 4. Results of 3D instance segmentation of point clouds from ScanNet, S3DIS, and STPLS3D.The first row is the original point cloud, the second row is the ground truth, the third row is the model predictions.

Method	Component	Device	Component time (ms)	Total (ms)
PointGroup [14]	Backbone	GPU	48	372
	Grouping	GPU+CPU	218	
	ScoreNet	GPU	106	
HAIS [3]	Backbone	GPU	50	256
	Hierarchical aggregation	GPU+CPU	116	
	Intra-instance refinement	GPU	90	
SoftGroup [19]	Backbone	GPU	48	266
	Soft grouping	GPU+CPU	121	
	Top-down refinement	GPU	97	
SSTNet [15]	Superpoint extraction	CPU	179	419
	Backbone	GPU	34	
	Tree Network	GPU+CPU	148	
	ScoreNet	GPU	58	
TD3D, ours	Backbone	GPU	39	**140**
	Proposal generation	GPU	8	
	RoI extraction	GPU	16	
	Proposal refinement	GPU	77	

Table 5. The inference time of TD3D and existing 3D instance segmentation methods, measured component-wise. All intermediate operations are performed on GPU, which allows achieving x1.8 speed-up in comparison with the fastest competitor, HAIS [3] and x1.9 speed-up with the most accurate competitor, SoftGroup [19].

Method	AP	AP_{50}
PointGroup	23.3	38.5
HAIS	35.1	46.7
SoftGroup	46.2	61.8
TD3D, ours	**54.3**	**69.8**

Table 6. Results on STPLS3D. The proposed approach outperforms the competitors by a large margin.

4.3. Qualitative Results

The original and segmented point clouds from ScanNet, S3DIS and STPLS3D datasets are depicted in Fig. 4.

4.4. Performance

To provide an in-depth performance evaluation, we run a profiler to measure the time required to complete each component of our method: extracting 3D features with a 3D CNN, proposal generation, RoI extraction, and proposal refinement. We decompose several competing approaches into components similarly, and report the inference time component-wise in Tab. 5. Contrary to other listed methods, TD3D follows a top-down paradigm, which allows running all operations on GPU in an end-to-end pipeline. As the result, our approach is notably faster than the previous fastest method, HAIS [3].

4.5. Ablation Studies

In this section, we analyze different components of our approach and measure the contribution to the final quality of each component. We do not introduce any changes into the 3D object detection part, but focus on the components that constitute the novelty of our approach: proposal generation and refinement. Namely, we investigate such aspects of proposal generation as the number of initial proposals and the RoI extraction threshold, and study the impact of the number of feature levels and the assigners used in the proposal refinement network. The ablation experiments are conducted on the ScanNet v2 validation set, following the same evaluation protocol as for the qualitative comparison.

Number of feature levels in the proposal refinement network. We study how the size of the proposal refinement network affects the segmentation accuracy. Starting from 0 levels (all points in an initial proposal are included in the instance mask), and using no more than 4 levels (as in the backbone), we select the best option in terms of AP. As can be seen from Tab. 7, AP grows with the number of levels; yet, we do not want our refinement model to be large, so we opt for four levels in the default version.

Number of initial proposals. Furthermore, we investigate the dependency between the number of initial proposals, accuracy, and runtime. Note that the number of proposals are approximate, since they cannot be set explicitly but manipulated through the NMS hyperparameters. Expectedly, the more proposals, the higher is AP (Tab. 8). However, with as many as 60 proposals, our method reaches the plateau in terms of segmentation accuracy. In the meantime, the inference time tends to increase with the growing number of proposals. Overall, we assume that with $\approx$ 60 initial proposals, our method demonstrates a decent trade-off between accuracy and speed, so we use this value by default in our experiments.

U-Net size	AP
0	25.8
1	37.9
2	45.4
3	46.3
4	47.3

Table 7. Results of a study of the tiny U-Net size on the ScanNet v2 validation set. We use four feature levels by default.

#Initial proposals	AP	Runtime (in sec)
$\approx$140	47.6	0.260
$\approx$100	47.5	0.210
$\approx$**60**	47.3	0.140
$\approx$20	45.6	0.105

Table 8. Results of a study of the approximate number of initial proposals on the ScanNet v2 validation set. 60 object proposals are chosen as a default value, as it serves a good balance of accuracy and speed.

Point classification threshold. We also vary the point binary segmentation threshold in 3D tiny U-Net, which is used to identify points either as *foreground* or *background* on inference. The results are presented in the Tab. 9. As can be observed from the Tab. 9, as the point binary seg-

Threshold	AP	AP_{50}
0.10	43.9	69.2
0.15	45.9	70.8
0.20	47.3	71.2
0.25	47.9	70.9
0.30	48.2	70.6
0.40	48.1	70.1

Table 9. Results of TD3D with different point binary segmentation thresholds, obtained on the ScanNet v2 validation set. The threshold of 0.2 allows for the highest quality.

FCAF3D assigner	IoU assigner	AP	AP_{50}
✓		46.3	70.2
	✓	45.5	69.6
✓	✓	**47.3**	**71.2**

Table 10. Results of the proposal refinement model with different assigners. The FCAF3D assigner slightly outperforms the IoU assigner in a single-assigner mode, but the best scores are obtained with their combination. Accordingly, we use FCAF3D+IoU assigners.

Threshold	AP	AP_{50}
0.00	47.3	71.2
0.25	46.2	71.1
0.50	46.0	71.0
0.75	45.6	69.3

Table 11. Results of the IoU assigner with different IoU thresholds, obtained on the ScanNet v2 validation set. Thresholding with 0.0 provides the best results, so we assume that filtering is not needed.

mentation threshold rises from 0.1 to 0.2, the AP_{50} value experiences a noticeable increase, hitting the highest score of 71.2 at a threshold of 0.2. For the larger values, the AP_{50} declines gradually, so the optimal value is defined unambiguously.

Assigners in the proposal refinement model. Tab. 10 presents the results of models trained using two different assigners: FCAF3D assigner and IoU assigner, individually and in combination. Taken individually, the FCAF3D assigner outperforms the IoU assigner, and the combination slightly improves the performance compared to using only one assigner, so we use the two of them by default. The Tab. 11 shows the results obtained by varying IoU threshold value in the IoU assigner on the ScanNet v2 validation set. Evidently, filtering by threshold is redundant, since the highest AP and AP_{50} values are obtained with the threshold of 0.0.

RoI extractor threshold. The RoI extractor algorithm is parameterized with the minimum number of voxels in the proposal. If a proposal contains fewer voxels, it is discarded and not used further at the subsequent stages. The Tab. 12 shows the results of the RoI extractor algorithm with different minimum voxel thresholds on the ScanNet v2 validation set. As can be seen, when the minimum voxel threshold is between 1 and 200, both the AP and AP_{50} scores remain constant at 47.3 and 71.2, respectively. However, when the threshold surpasses 200, its further increasing causes the degradation of the performance, so any number between 1 and 200 can be used as a default option.

Threshold	AP	AP_{50}
1	47.3	71.2
10	47.3	71.2
50	47.3	71.2
100	47.3	71.2
200	47.3	71.2
500	45.6	70.1
700	44.6	68.4
1000	42.6	65.1

Table 12. Results of the RoI extractor algorithm with different minimum voxel thresholds on the ScanNet v2 validation set. Any value between 1 and 200 can be used, since all of them ensure the same final quality.

5. Conclusion

In this work, we introduced TD3D, a novel 3D instance segmentation method following a top-down paradigm. Being fully-convolutional and trained end-to-end in a data-driven way, it does not rely on prior assumptions about the objects, which eases the burden of manually tuning domain-specific hyperparameters. We evaluated our method on the standard benchmarks: indoor ScanNet v2, ScanNet200, S3DIS, and aerial STPLS3D. Our experiments demonstrated that TD3D is on par with state-of-the-art grouping-based 3D instance segmentation methods: but being as accurate, it is more than 1.9x faster on inference.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016. 4

[2] Meida Chen, Qingyong Hu, Zifan Yu, Hugues THOMAS, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. In *British Machine Vision Conference (BMVC)*. BMVA Press, 2022. 4

[3] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *IEEE/CVF International Conference on Computer Vision*, 2021. 1, 2, 5, 6, 7

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 5

[5] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 5

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet:

Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

[7] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 5

[8] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2937–2946, 2020. 2, 5

[9] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *IEEE/CVF International Conference on Computer Vision*, pages 2980–2988, 10 2017. 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[11] Tong He, Chunhua Shen, and Anton van den Hengel. DyCo3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 5

[12] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5

[13] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021. 5

[14] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 5, 6

[15] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *IEEE/CVF International Conference on Computer Vision*, pages 2783–2792, 2021. 1, 2, 5, 6

[16] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *IEEE/CVF European Conference on Computer Vision*, 2022. 4, 5

[17] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: fully convolutional anchor-free 3d object detection. In *IEEE/CVF European Conference on Computer Vision*, pages 477–493, 2022. 2, 3, 4, 5

[18] Weiwei Sun, Daniel Rebain, Renjie Liao, Vladimir Tankovich, Soroosh Yazdani, Kwang Moo Yi, and Andrea Tagliasacchi. Neuralbf: Neural bilateral filtering for top-down instance segmentation on point clouds. In *Winter Conference on Computer Vision (WACV)*, 2023. 1, 2, 5

[19] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. Softgroup for 3d instance segmentation on 3d point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 5, 6

[20] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018. 2, 5

[21] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4091–4100, 2019. 2, 5

[22] 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. In *Advances in Neural Information Processing Systems*, pages 6737–6746, 2019. 2, 5

[23] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3942–3951, 2019. 2, 5

[24] Min Zhong, Xinghao Chen, Xiaokang Chen, Gang Zeng, and Yunhe Wang. Maskgroup: Hierarchical point grouping and masking for 3d instance segmentation. In *International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2022. 5