Implicit-Part Based Context Aggregation for Point Cloud Instance Segmentation

Xiaodong Wu[1] and Ruiping Wang[1] and Xilin Chen[1]

Abstract*— Context information is important for instance segmentation on point clouds. Existing methods either only use local surroundings by stacking multiple convolution layers or use non-local methods to model long-range interactions. However, they usually directly operate on points which is an unstructured and low-level representation and is highly dependent on context. To address this issue, we propose an effective framework named Implicit-Part Context Aggregation (IPCA), which adopts implicit parts as an intermediate representation and achieves context aggregation through message passing along the implicit part graph. Specifically, we first organize unstructured points into geometrically consistent implicit parts and construct the implicit part graph according to the geometric adjacency. Then, an initial part embedding is extracted using the proposed Implicit Part Network (IPN) which can aggregate point features and capture the intrinsic geometric shape of the part. We further refine the part embedding by a graph reasoning module named Context Aggregation Network (CAN), which helps to make a more precise prediction by well exploiting the context information. Instance proposals are then generated by grouping implicit parts. Finally, we propose an additional step to attribute the entire instance proposal to a Semantic Criterion Net (SCN) to infer the semantics of the instance. The purpose is to correct the semantic prediction errors caused by not knowing the boundary and overall shape of the object in the previous steps. Extensive experiments on two large datasets, ScanNet and 3RScan, demonstrate the effectiveness of our method. To our knowledge, it yields the highest performance on the ScanNet test benchmark and its AP@50 is 9.5 points higher than the baseline. The code is available at https://github.com/xiaodongww/IPCA

I. INTRODUCTION

3D instance segmentation is a fundamental task in scene understanding. Given an input point cloud, the goal is to assign a semantic label and an instance ID for all points on the objects. As the technical cornerstone of many real-world applications like augmented reality [1], [2] and robot perception [3], [4], it is drawing more and more attention.

One of the popular solutions of instance segmentation is to cluster points into instances in a bottom-up manner [5], [6], [7]. They usually predict point semantics and instance embedding vector point-wisely and use them to cluster points into instances directly. Although effective, such methods also suffer from the per-point prediction manner as shown in the top branch of Fig. 1. A single point is meaningless unless it forms a geometric structure together with neighboring points. Since these methods ignore the underlying geometric structure and make predictions independently, there is quite a chance that two points of the same object have different semantic predictions. This may happen even if the two points are close and on the same flat board (as shown in Fig. 1 a).

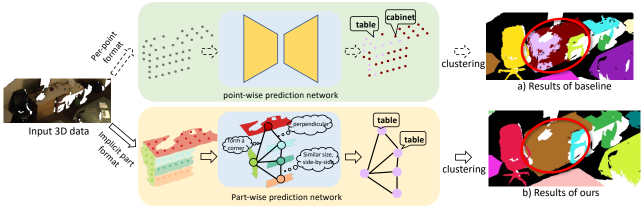

Fig. 1. Comparison between per-point format method (top branch) and our implicit-part based method (bottom branch). Different colors in a) and b) represent individual instances and the red circle areas indicate that our result is more precise and more regular.

Considering the limitations of per-point format methods, we try to find an intermediate representation that is more structural than the bottom-level point representation. It is natural to think of using object parts. However, explicit object parts that have specific semantic names (e.g. table surface, table leg, chair back) require expensive annotations and have limited generalizability across different object categories. For this reason, we relax the requirement and use geometrically consistent areas as implicit parts. There are two types of information that are important on how to identify objects from implicit parts, namely understanding the geometric shape of implicit parts and modeling the relationships between them. For example, we can identify a chair because we see two square boards perpendicular to each other and one of the boards is perpendicular to the four sticks. There are two prior works [8], [9] that also organize points into parts. However, the part geometric shape and relationships between parts in such works are ignored which we believe are of crucial importance for instance segmentation. The bottom branch of Fig. 1 shows the main idea of our implicit part based method.

Specifically, we propose a new framework named Implicit Part Context Aggregation (**IPCA**) for instance segmentation. Given an input point cloud, we first extract point-wise features by a U-Net style backbone. Then it is followed by a two-branch network. The semantic branch is used to predict semantics part-wisely. Concretely, we employ a normal-based graph cut method [10], [11] to group the input point cloud into implicit parts and construct an implicit part graph according to the geometric adjacency between parts. The

*This work is partially supported by National Key R&D Program of China (2021ZD0111901), Natural Science Foundation of China under contracts Nos. U21B2025, U19B2036, 61922080.

[1] Xiaodong Wu, Ruiping Wang and Xilin Chen are with Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China and with University of Chinese Academy of Sciences, Beijing, 100049, China. xiaodong.wu@vipl.ict.ac.cn, {wangruiping, xlchen}@ict.ac.cn

part embeddings are initialized by an Implicit Part Network (**IPN**) which can capture the intrinsic geometric shape of the implicit part and aggregate the backbone features of points on the part. The part embeddings are further refined by our proposed Context Aggregation Network (**CAN**) which can model the relationships between implicit parts and aggregate the context information. After fully exploring the surrounding environment, the semantics for each implicit part is predicted. Parallel to the semantic branch is a center offset branch similar to PointGroup [5] which forms a center voting space where points of the same instance are close to each other. After the two-branch network, we can group implicit parts into instance proposals using a clustering module. Finally, we need to assign a semantic class for each instance proposal. Since the extent and boundaries of the instance proposal are known, we propose a Semantic Criterion Net (**SCN**) to infer the instance semantics in a holistic view by voxelizing the whole instance again and feeding it into the network. The advantage is that it can correct the previous wrong semantic prediction caused by not knowing the overall object shape.

We evaluate our method on two large datasets ScanNet [12] and 3RScan [13]. Results show that our proposed method achieves state-of-the-art performance. Especially, it is 9.5 points higher on AP_{50} of ScanNet's test set than the baseline method PointGroup. From the visualization, our results are more regular and refined, which shows that our proposed method is effective.

II. RELATED WORK

In this section, we first briefly review 3D instance segmentation methods, of which most fall into two paradigms: *Top-down methods* and *Bottom-up clustering methods*, [14]. Then we introduce works using over-segmentation in both 2D and 3D recognition tasks.

Top-down methods. The top-down methods are similar to the 2D instance segmentation methods [15], [16] by predicting 3D proposals together with a foreground mask inside each proposal. Hou *et al.* [17] use 3D convolutions to generate 3D anchor bounding box proposals and use 3D-RPN and 3D-RoI to infer object bounding box locations, class labels, and per-voxel instance masks. Yang *et al.* [18] propose a single-stage, anchor-free, NMS-free method to speed up the inferencing speed. Yi *et al.* [19] use reconstructed shapes instead of 3D BBox as instance proposals. Engelmann *et al.* [20] adopt a similar method as VoteNet [21] to generate proposals by predicting object centers.

Bottom-up clustering methods. Bottom-up methods usually contain two parallel branches. One is a semantic segmentation branch predicting per-point semantic classes. The other is an instance points grouping branch which generates a per-point instance embedding vector or object center prediction or both. The instance masks are generated by clustering in the embedding space or voting space. SGPN [22] groups instances by constructing the similarity matrix of points in the embedding space. ASIS [7] and JSIS3D [23] propose to address the problems of semantic and instance segmentation by simultaneously predicting per-point semantics and per-point instance embedding features. MTML [24] proposes to predict instance centers in addition to instance embedding features. OccuSeg [8] claims that the auxiliary task of predicting instance occupancy size is helpful and it also designs a new method to group over-segments instead of points. PointGroup [5] proposes to cluster in both the Euclidean space and the voted center space. PE [25] proposes a probabilistic embedding space for point cloud embedding. HAIS [26] designs a hierarchical point aggregation method. Our method is also classified to bottom-up type. We use PointGroup as the baseline and also adopt a two-branch architecture. One branch predicts semantics part-wisely and the other branch predicts object center votes which is used to cluster implicit parts into instances.

3D Over-segmentation. There is an increasing interest in exploiting over-segmentation in 3D scene understanding tasks. SPG [27] uses over-segments as superpoints to achieve large scale point cloud semantic segmentation. OccuSeg [8] mentioned above chooses to group over-segments instead of points to achieve instance segmentation. SSTNet [9] proposes to build a semantic superpoint tree (SST) for superpoints clustering. In this work, we try to construct parts from unstructured point cloud inspired by the Recognition-by-component (**RBC**) theory [28]. Since explicitly defined parts are expensive to annotate and have limited generalizability across different object categories, we adopt geometrically consistent areas (i.e. over-segments) as implicit parts. There are two main differences between our approach and existing superpoint-based methods SSTNet and OccuSeg. First, due to the use of pooling operations, existing methods have lost part shape information that is important for recognition. Our method additionally pays attention to capturing the geometric shape information of the part when deriving part features from point features. Second, existing methods neglect to model the relationships between implicit parts. We believe that relationship modeling is important for understanding objects, so a dedicated module is designed in our method to model the part relationship and collect context information from the relationship.

III. METHOD

The architecture of our method is depicted in Fig 2. First we adopt a U-Net style backbone (Sec. III-A) to extract point-wise features. It is then followed by a two-branch network. In the semantic branch (Sec. III-B), we aggregate points into implicit parts and forecast in a coherent manner by transferring information along the edges of the implicit part graph. The center offset branch (Sec. III-C) generates a voting space where points belonging to the same object are close to each other. Output of the two branches is fed into a clustering module (Sec. III-D) to generate instance proposals. Finally, we use the Semantic Criterion Net and IoU Criterion Net to infer the semantics of instance proposals and get the evaluation score to rank the proposals (Sec. III-E).

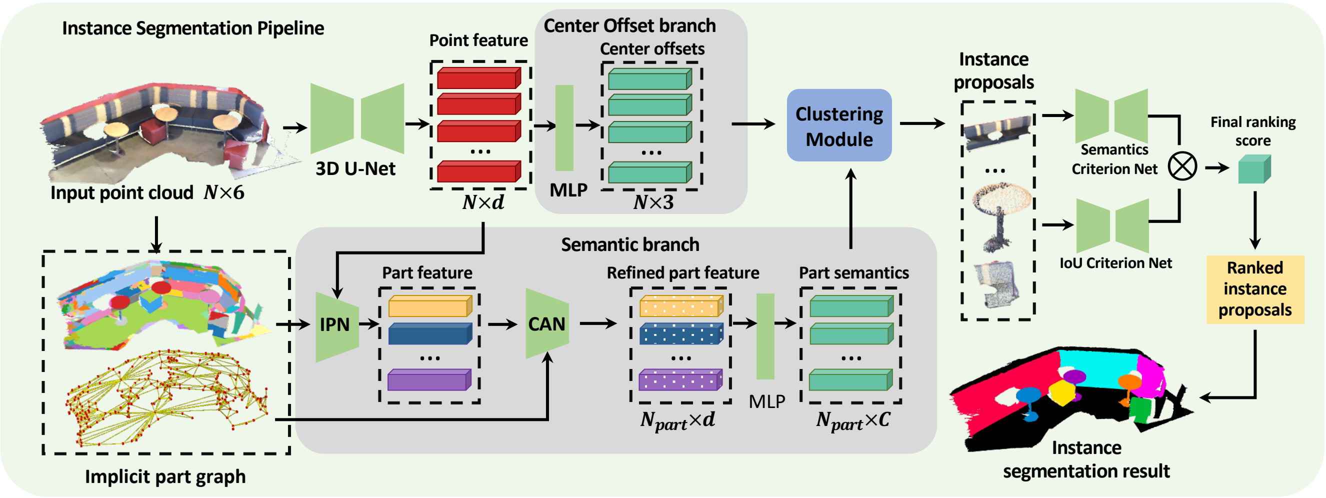

Fig. 2. The overall architecture of our proposed method. Given unstructured point clouds as input, we feed them into a U-Net style backbone to extract point-level features. It is then followed by a two-branch network. The semantic branch predicts implicit part semantics using backbone features and the implicit part graph. Details of the **IPN** and **CAN** are shown in Fig. 3. The center offset branch predicts offsets from points to their instance centers. The output of the two branches is fed into a clustering module to generate instance proposals. The Semantics Criterion Net and the IoU Criterion Net are used to infer the semantic and get the final ranking score for each proposal.

A. Backbone

The input 3D point cloud $\boldsymbol{P}$ can be described as a set of N points. Each point $p_i \in \boldsymbol{P}$ has a 3D location coordinate $\boldsymbol{x}_i$ and a 3D color vector $\boldsymbol{r}_i$ and they are concatenated as a 6D input feature. To process the unordered and unstructured point clouds, we voxelize the points into grids with average pooling. Then they are fed into a U-Net style backbone which consists of multiple 3D Submanifold Sparse Convolution (SSC) layers [29]. After this, we map the voxelized features back to point features and get a point feature set $\boldsymbol{F} = \{\boldsymbol{f}_i \in \mathbb{R}^d\}_{i=1}^{N}$ where d is the feature dimension.

B. Semantic Branch

1) Implicit Part Network: As mentioned before, we try to represent objects with implicit parts instead of bottom-level points. Specifically, given an input point set $\boldsymbol{P}$, we use a normal-based graph cut method [10], [11] to generate a set of over-segments. Each segment is regarded as an implicit part. In this way, the original point set $\boldsymbol{P}$ is partitioned into $\boldsymbol{N}_{part}$ non-overlapped implicit parts $\mathcal{S} = \{\boldsymbol{P}_1, ..., \boldsymbol{P}_{N_{part}}\}$.

To characterize the geometric shape, we get the local coordinates of points on the part by moving the coordinate system to the center of the part. In this way, the m-th implicit part P_m is represented by $\boldsymbol{P}_m = \{(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i, \boldsymbol{f}_i)\}_{i=1}^{N_m}$, where N_m is the number of points and $\sum_{m=1}^{N_{part}} N_m = N$, $\boldsymbol{x}_i$ is the original global coordinate of point p_i, $\hat{\boldsymbol{x}}_i$ is the local coordinate on the implicit part, and $\boldsymbol{f}_i \in \boldsymbol{F}$ is the point feature learned by the U-Net backbone.

To capture the geometric shape information of the implicit part and aggregate the point features into part embeddings, we design an Implicit Part Network (**IPN**) module. As shown in Fig. 3 (a), we first sub-sample each implicit part to 256 points on-the-fly. Then we concatenate the global coordinate $\boldsymbol{x}_i$, local coordinate $\hat{\boldsymbol{x}}_i$ and feature $\boldsymbol{f}_i$ and feed the sampled points into a small PointNet++ [30] network. Finally, the initial implicit part feature set is represented with $\boldsymbol{H} = \{\boldsymbol{h}_m \in \mathbb{R}^d\}_{m=1}^{N_{part}}$.

2) Context Aggregation Network: So far, we have grouped points into implicit parts and have generated part embeddings by the IPN module. The problem is that a single part does not have rich enough context information without knowing the location or shape of other parts which is important for recognition. For example, given two adjacent planes perpendicular to each other and four similar parts supporting one of the planes, they are very likely to form a chair. This requires the model to provide the ability to transmit information among implicit parts. To this end, we elaborately design a Context Aggregation Network (**CAN**) to further refine the initial implicit part embeddings $\boldsymbol{H}$ by collecting context information. The main idea is to construct an implicit part graph and achieve information transmitting along graph edges using graph convolution (as shown in Fig. 3 (b)).

Specifically, we use implicit part set $\mathcal{S}$ as the node set and the adjacency between parts as edge set $\mathcal{E}$ to construct an undirected graph $G = (\mathcal{S}, \mathcal{E})$. Each node is an implicit part represented by its initial embedding $\boldsymbol{h}_m \in \boldsymbol{H}$. The edge $\boldsymbol{e}_{mn} \in \mathcal{E}$ is a binary vector indicating whether the implicit part P_m is geometrically adjacent to P_n. Then a graph convolution network is designed to pass messages along the edges and merge context information into the implicit part embeddings. Concretely, given an implicit part P_m , we first collect the neighbor messages $\boldsymbol{h}_{\mathcal{N}(P_m)}$ passed along the edges from its neighboring implicit parts $\mathcal{N}(P_m)$, and then use the collected message to update its embedding and get the refined embedding $\boldsymbol{h}'_m$:

$$\boldsymbol{h}_{\mathcal{N}(P_m)} = maxpool(\{\boldsymbol{h}_u, \forall u \in \mathcal{N}(P_m)\}) \tag{1}$$

$$\boldsymbol{h}'_m = \boldsymbol{W}_1 \cdot \boldsymbol{h}_m + \boldsymbol{W}_2 \cdot \boldsymbol{h}_{\mathcal{N}(P_m)} \tag{2}$$

$\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are the parameters of two linear projection layers. We stack 2 graph convolution layers in this module.

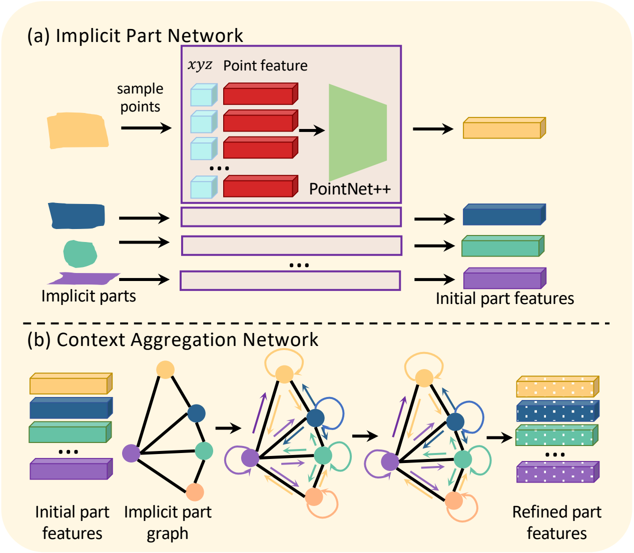

Fig. 3. Illustration of the Implicit Part Network (IPN) (a) and the Context Aggregation Network (CAN) (b). In IPN, we first sample points uniformly and apply a PointNet++ to extract the initial part feature. In CAN, given the initial part features and the implicit part graph, we get the refined part features by transmitting messages between parts using the Graph Convolution Network.

After this process, we can get the refined implicit part embedding $\boldsymbol{H}' = \{\boldsymbol{h}'_m \in \mathbb{R}^d\}_{m=1}^{N_{part}}$. Finally, we apply a 2-layer MLP and a softmax layer to get the semantic probability distribution $\mathcal{A} = \{\boldsymbol{a}_m \in \mathbb{R}^C\}_{m=1}^{N_{part}}$ on C classes for implicit parts:

$$\boldsymbol{a}_m = softmax(MLP(\boldsymbol{h}'_m)) \tag{3}$$

Point-wise semantic predictions are easily generated by mapping the semantics from implicit parts $\mathcal{A}$ to their corresponding points. The learning process of the semantic branch is supervised by the standard cross-entropy loss $\mathcal{L}_{seg}$.

C. Center Offset Branch

Suppose $\mathcal{P}$ is the point set of points on objects (i.e. except points on *wall*, *floor*) and $p_i \in \mathcal{P}$ is a point on an object, the target of this branch is to predict a 3D offset vector $\triangle\boldsymbol{x}_i \in \mathbb{R}^3$ between its coordinate $\boldsymbol{x}_i$ and the relevant object center $\boldsymbol{c}_i^*$. We use a 2-layer MLP network upon the backbone feature $\boldsymbol{f}_i \in \boldsymbol{F}$ to regress the offset vector:

$$\triangle\boldsymbol{x}_i = MLP(\boldsymbol{f_i}) \tag{4}$$

The learning process of the center offset branch is supervised by an L_1 regression loss and a direction loss as [5]:

$$\mathcal{L}_{reg} = \frac{1}{|\mathcal{P}|}\sum_{i=1}^{|\mathcal{P}|} \|\boldsymbol{x}_i + \triangle\boldsymbol{x}_i - \boldsymbol{c}_i^*\| \tag{5}$$

$$\mathcal{L}_{dir} = \frac{1}{|\mathcal{P}|}\sum_{i=1}^{|\mathcal{P}|} \frac{\triangle\boldsymbol{x}_i}{\|\triangle\boldsymbol{x}_i\|_2} \cdot \frac{\boldsymbol{c}_i^* - \boldsymbol{x}_i}{\|\boldsymbol{c}_i^* - \boldsymbol{x}_i\|_2} \tag{6}$$

where $|\mathcal{P}|$ is the number of points in $\mathcal{P}$ and $\boldsymbol{c}_i^*$ is the relevant object center of point p_i. The center offsets are finally predicted as $\triangle\boldsymbol{X} = \{\triangle\boldsymbol{x}_i\}_{i=1}^{|\mathcal{P}|}$

D. Clustering of implicit parts

We describe how to group implicit parts into instance proposals in this sub-section. For now, each point in the point cloud is labeled with a semantic label $c \in \mathcal{C}$ (by mapping the semantics from parts to points) and an offset vector $\triangle\boldsymbol{x}_i \in \mathbb{R}^3$ indicates object centers. For each object class (i.e. except *wall*, *floor*), we shift the relevant points towards their predicted object centers and generate a center voting space where points belonging to the same object are close to each other. We adopt the BFS algorithm [5] to cluster points into L clusters $\mathcal{Q} = \{\boldsymbol{Q}_1, ..., \boldsymbol{Q}_L\}$ in the voting space. Then, for implicit part P_i, it is assigned to cluster $\boldsymbol{Q}_l$ if most points in it are clustered to $\boldsymbol{Q}_l$. Finally, clusters composed of several implicit parts can be regarded as instance proposals.

E. Instance Proposals Scoring

1) Semantic Criterion Net: We believe that the semantic prediction of points or parts is less reliable since it is predicted without knowing the boundaries or extent of the objects. Since such information is known after obtaining instance proposals, we propose to predict the instance semantics in a holistic view, i.e. considering all points of the instance. Specifically, we use a small U-Net named Semantic Criterion Net (**SCN**) to predict instance semantics and give a semantic score.

In the training stage, we use the ground truth instance masks and generate N_{obj} instance point sets $\{\boldsymbol{P}_1, ..., \boldsymbol{P}_{N_{obj}}\}$. For instance o_j with N_{o_j} points $\boldsymbol{P}_{o_j} = \{(\boldsymbol{x}_i, \boldsymbol{f}_i)\}_{i=1}^{N_{o_j}}$ we concatenate its coordinate $\boldsymbol{x}_i$ and its backbone feature $\boldsymbol{f}_i$ as the input point feature and voxelize the ground truth instance as we do in the backbone network. A max-pooling layer is used to get instance-wise feature $\boldsymbol{f}_{o_j} \in \mathbb{R}^d$ and an MLP layer is used to get the classification probability of C classes. We use a cross-entropy loss $\mathcal{L}_{obj\_sem}$ to supervise the learning of object semantics prediction.

In the evaluation stage, we adopt a similar process except using the instance proposal mask instead of the ground truth mask. Each instance proposal has C semantic scores $[s_1^{sem}, ..., s_C^{sem}]$ indicating the probability distribution on C semantic classes. The instance proposal is duplicated and assigned with semantic label c as long as its relevant semantic score s_c^{sem} is larger than a pre-defined threshold τ_{sem}.

2) IoU Criterion Net: We adopt an IoU Criterion Net to predict the IoU between the instance proposal and the ground truth instance mask. Formally, given instance proposal $\boldsymbol{Q}_l$ with N_{Q_l} points $\boldsymbol{Q}_l = \{(\boldsymbol{x}_i, \boldsymbol{f}_i)\}_{i=1}^{N_{Q_l}}$, we voxelize these points and feed them into a small U-Net. By max-pooling the U-Net output of the instance, we get an instance descriptor $\boldsymbol{f}_{Q_l} \in \mathbb{R}^d$. The IoU score is calculated as:

$$s_l^{iou} = Sigmoid(MLP(\boldsymbol{f}_{Q_l})) \tag{7}$$

Suppose $\hat{s}_l^{iou}$ is the maximum IoU between proposal Q_l and ground truth instance masks, the score loss is calculated

TABLE I

3D instance segmentation results on ScanNet(V2) test benchmark. All numbers are from the official website.

Method	AP@50	bath	bed	bkshf	cab	chair	cntr	curt	desk	door	ofurn	pic	fridg	showr	sink	sofa	table	toilet	wind
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 [31]	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 [18]	48.8	100.0	67.2	59.0	30.1	48.4	9.8	62.0	30.6	34.1	25.9	12.5	43.4	79.6	40.2	49.9	51.3	90.9	43.9
MTML [24]	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
PointGroup [5]	63.6	100.0	76.5	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 [33]	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
OccuSeg [8]	67.2	100.0	75.8	68.2	57.6	**84.2**	**47.7**	50.4	52.4	**56.7**	58.5	45.1	55.7	100.0	75.1	79.7	56.3	100.0	46.7
SSTNet [9]	69.8	100.0	69.7	**88.8**	55.6	80.3	38.7	62.6	41.7	55.6	58.5	**70.2**	60.0	100.0	**82.4**	72.0	69.2	100.0	50.9
HAIS [26]	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
Ours	**73.1**	**100.0**	78.8	88.4	**69.8**	78.8	25.2	76.0	**64.6**	51.1	**63.7**	66.5	**80.4**	**100.0**	64.4	77.8	**74.7**	**100.0**	**56.1**

using a binary cross-entropy loss:

$$\mathcal{L}_{obj\_iou} = -\frac{1}{L}\sum_{l=1}^{L}\left(\hat{s}_l^{iou}\log\left(s_l^{iou}\right)+\left(1-\hat{s}_l^{iou}\right)\log\left(1-s_l^{iou}\right)\right) \tag{8}$$

where L is the number of instance proposals.

Finally, we score the instance proposals from two different perspectives. The *Semantics Score* from the *Semantic Criterion Net* judges the instance proposal in a semantic confidence view and the *IoU Score* from the *IoU Criterion Net* judges the instance in an object completeness view. We multiply these two scores and use the product as the final ranking score to rank the instance proposals for evaluation.

F. Network Training and Inference

1) Training: We train the model from scratch with multi-task loss as follows:

$$\mathcal{L} = \mathcal{L}_{seg} + \mathcal{L}_{reg} + \mathcal{L}_{dir} + \mathcal{L}_{obj\_sem} + \mathcal{L}_{obj\_iou} \tag{9}$$

2) Inference: Since the instance proposals have no overlap, we do not need to apply NMS. We directly use the scores computed by *Semantic Criterion Net* and *IoU Criterion Net* in Sec. III-D to rank all instance proposals.

IV. EXPERIMENTS

To demonstrate the effectiveness of our method, we conduct extensive experiments on two large datasets ScanNet [12] and 3RScan [13]. In this section, we first introduce the experimental settings in Sec. IV-A. Then we compare our method with previous state-of-the-art methods (Sec. IV-B). To analyze the effectiveness of our proposed modules, we provide an ablation study in Sec. IV-C. Finally, we show qualitative instance segmentation results in Sec. IV-D.

A. Experimental Setting

1) Datasets: The ScanNet [12] dataset contains 1,613 3D indoor scene reconstructions labeled with 18 object classes. The training, validation, and testing sets have 1,201, 312, and 100 scans respectively. For fair comparison with other methods, we report the performance of the testing set returned by the official evaluation server. We also report the performance of the validation set for the ablation study.

The 3RScan [13] dataset contains 1,482 3D reconstructions of 478 naturally changing indoor environments. There are 27 semantic classes and 24 classes among them (i.e. except *wall*, *floor*, and *ceiling*) are used for instance segmentation evaluation. Following [13], we split the 478 environments into training/validation/testing set with 385/47/46 environments (corresponding to 1,178/157/147 3D scans) respectively. Since the label annotation for the test set is not publicly available, we train on the training set and report the performance on the validation set.

2) Evaluation Metrics: We use the mean average precision (mAP) of different IoU thresholds as the evaluation metric. Specifically, we report AP@50 and AP@25 with IoU threshold 50% and 25% respectively. We also report AP which averages the scores with IoU thresholds from 50% to 95% with step size 5%.

3) Experimental Details: Our model is trained with an initial learning rate of 0.001 and decays with the OneCycle policy [34]. We set the input voxel size as $0.02m$ following the common practice [26], [5]. Instance proposals whose size is smaller than 10 or ranking scores smaller than 0.001 are filtered out. To speed up the training process and reduce the memory requirement, we load the parameters of the U-Net backbone from a pre-trained baseline model [5] and freeze it in the training process.

B. Comparison with State-of-the-art Methods

We submit our result to the testing server[1] (on the official website our method is denoted as ”IPCA-Inst”) of the ScanNet benchmark and compare the performance of our method on the unreleased test set with previous published works. Table I shows per-class AP@50 performance collected from the ScanNet website with the most updated results upon our paper submission. Our method outperforms all exiting methods on AP@50 by a large margin and is 3.2 points higher than previous best method HAIS [26]. Compared with OccuSeg [8] and SSTNet [9] that also use over-segmentation, our method outperforms them by 5.9 and 3.3 points respectively. Note that our method is built on the basis of PointGroup [5] and our proposed modules improve the performance by a significant margin (9.5 points). Among all these categories, our method ranks the 1st place in 9 out of 18 classes in total, and it performs exceptionally well on classes (e.g. Fridge, Desk, Table) that have relatively regular and square shapes. The reason is that the implicit parts of these class are usually flat board and the relationships between them are simple, usually perpendicular or parallel to each other. Benefited from the IPN and CAN module, our method can better capture the underlying geometric shape and model the relationships between them and ultimately obtain more accurate predictions. We also report the performance of AP

[1] http://kaldir.vc.in.tum.de/scannet_benchmark/semantic_instance_3d

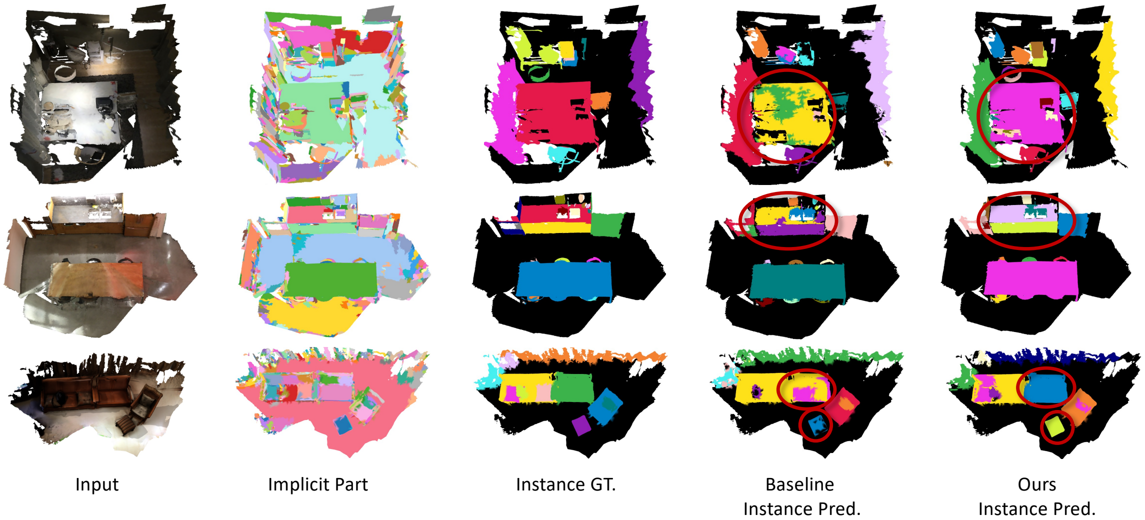

Fig. 4. Qualitative results on the validation set of ScanNet v2 (top two rows) and 3RScan (bottom row). The areas circled in red show that instance masks of our method are more regular and complete.

TABLE II

3D INSTANCE SEGMENTATION RESULTS ON SCANNET(V2) TEST BENCHMARK IN TERMS OF AP, AP@50, AP@25.

Method	AP	AP@50	AP@25
3D-SIS [17]	16.1	38.2	55.8
MASC [31]	25.4	44.7	61.5
PanopticFusion [32]	21.4	47.8	69.3
3D-BoNet [18]	25.3	48.8	68.7
MTML [24]	28.2	54.9	73.1
PointGroup [5]	40.7	63.6	77.8
GICN [33]	34.1	63.8	78.8
OccuSeg [8]	48.6	67.2	74.2
SSTNet [9]	50.6	69.8	78.9
HAIS [26]	45.7	69.9	80.3
Ours	**52.0**	**73.1**	**85.1**

and AP@25 in Table II. Our method outperforms all existing methods on all three metrics. In terms of computational cost, considering that more computations are introduced, given pre-computed implicit part and implicit part graph, inference process of our model (230ms) is 39ms longer than PointGroup's 191ms (without NMS).

On 3RScan validation dataset, we compare our method with PointGroup and report the performances in Table IV. By comparing the first and last row of Table IV, we can see that our method outperforms PointGroup by 7.8 points on AP, 7.8 points on AP@50 and 7.9 points on AP@25.

C. Ablation Study

We conduct ablation studies on ScanNet and 3RScan to demonstrate the effectiveness of our proposed modules. Table III and Table IV show the ablation results on the validation set of ScanNet and 3RScan respectively.

1) Effectiveness of Implicit Part Context Aggregation: We apply implicit-part based context aggregation on the baseline method (i.e. PointGroup [5]) and compare the results on three metrics. By comparing the first row and the second row in Table III, we can observe a large improvement on ScanNet validation set, i.e. AP (+6.3), AP@50 (+3.7), and AP@25(+0.6). This shows that grouping points into implicit parts and collecting context part-wisely are effective. Note that the performance of our method improves greatly when the IoU threshold is high, i.e. we can largely improve high-quality instance masks with implicit parts. We can observe even larger improvement on a more difficult dataset 3RScan. Table IV shows that after applying implicit-part based context aggregation, we improve AP (+8.0), AP@50 (+5.2), and AP@25(+3.2) by a large margin on 3RScan.

TABLE III

ABLATION RESULTS USING OUR PROPOSED MODULES ON THE SCANNET V2 VALIDATION SET. IPCA MEANS USING IMPLICIT PART CONTEXT AGGREGATION. SCN MEANS USING SEMANTIC CRITERION NET.

	IPCA	SCN	AP	AP@50	AP@25
Baseline			40.7	60.3	73.0
Baseline+IPCA	✓		47.0	64.0	73.6
Baseline+SCN		✓	41.0	62.2	76.1
Baseline+IPCA+SCN	✓	✓	49.0	67.6	79.4

TABLE IV

ABLATION RESULTS ON THE 3RSCAN VALIDATION SET. IPCA MEANS USING IMPLICIT PART CONTEXT AGGREGATION. SCN MEANS USING SEMANTIC CRITERION NET.

	IPCA	SCN	AP	AP@50	AP@25
Baseline			27.0	41.1	49.3
Baseline+IPCA	✓		35.0	46.3	52.5
Baseline+SCN		✓	28.9	44.9	56.0
Baseline+IPCA+SCN	✓	✓	34.8	48.9	57.2

2) Effectiveness of Semantic Criterion Net: The main idea of Semantic Criterion Net (SCN) is to infer the semantics of instance proposal again after knowing the boundary and extent of the instance. We believe that this can mitigate the errors of point-wise or part-wise semantic predictions and largely improve the recall. As shown in the third row of Table III, applying SCN on Baseline can improve 3.1 points on AP@25. By comparing the second row and last row of Table III, it can be observed that SCN improves AP (+2.0), AP@50 (+3.6), and AP@25 (+5.8) by a large margin on ScanNet. The performance gets a huge boost for low-quality instance masks which matches our expectations that SCN can largely improve the recall. Similar conclusions can be drawn on 3RScan as shown in Table IV.

D. Qualitative Evaluation

We show the instance segmentation results on the validation set of ScanNet and 3RScan visually in Fig 4. The second column shows the implicit parts used in our method. The last two columns compare the results between PointGroup and our implicit part based method. Key parts are circled in red lines. We can see that by organizing points into implicit parts and collecting context through transmitting information between parts, instance masks of our method are more regular and complete. There are fewer cases that falsely split one ground truth object into multiple instances caused by inaccurate point-wise semantic prediction.

V. CONCLUSIONS

In this work, we propose a new framework for point cloud instance segmentation. Implicit part is used as an intermediate representation which is more structural and meaningful than point cloud. By fully exploiting the context through transmitting information among implicit parts, our method can generate a more accurate and precise instance mask. We also propose to infer instance semantics after knowing the boundary and extent of the instance. Experiments on ScanNet and 3RScan demonstrate the effectiveness of our method.

REFERENCES

[1] P. Milgram, S. Zhai, D. Drascic, and J. Grodski, "Applications of augmented reality for human-robot communication," in *International Conference on Intelligent Robots and Systems (IROS)*, 1993, pp. 1467–1472.

[2] C. P. Quintero, S. Li, M. K. X. J. Pan, W. P. Chan, H. F. M. V. der Loos, and E. A. Croft, "Robot programming through augmented trajectories in augmented reality," in *International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1838–1844.

[3] L. Zhao, J. Lu, and J. Zhou, "Similarity-aware fusion network for 3d semantic segmentation," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1585–1592.

[4] J. Choi, Y. Song, and N. Kwak, "Part-aware data augmentation for 3d object detection in point cloud," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3391–3397.

[5] L. Jiang, H. Zhao, S. Shi, S. Liu, C. Fu, and J. Jia, "Pointgroup: Dual-set point grouping for 3d instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4866–4875.

[6] T. He, C. Shen, and A. van den Hengel, "Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution," *CoRR*, vol. abs/2011.13328, 2020.

[7] X. Wang, S. Liu, X. Shen, C. Shen, and J. Jia, "Associatively segmenting instances and semantics in point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4096–4105.

[8] L. Han, T. Zheng, L. Xu, and L. Fang, "Occuseg: Occupancy-aware 3d instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2937–2946.

[9] Z. Liang, Z. Li, S. Xu, M. Tan, and K. Jia, "Instance segmentation in 3d scenes using semantic superpoint tree networks," *CoRR*, vol. abs/2108.07478, 2021.

[10] A. Karpathy, S. D. Miller, and F. Li, "Object discovery in 3d scenes via shape analysis," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 2088–2095.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[12] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2432–2443.

[13] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Nießner, "Rio: 3d object instance re-localization in changing indoor environments," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[14] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8759–8768.

[15] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[16] P. H. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015, pp. 1990–1998.

[17] J. Hou, A. Dai, and M. Nießner, "3d-sis: 3d semantic instance segmentation of RGB-D scans," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4421–4430.

[18] B. Yang, J. Wang, R. Clark, Q. Hu, S. Wang, A. Markham, and N. Trigoni, "Learning object bounding boxes for 3d instance segmentation on point clouds," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 6737–6746.

[19] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "GSPN: generative shape proposal network for 3d instance segmentation in point cloud," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3947–3956.

[20] F. Engelmann, M. Bokeloh, A. Fathi, B. Leibe, and M. Nießner, "3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9028–9037.

[21] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9276–9285.

[22] W. Wang, R. Yu, Q. Huang, and U. Neumann, "SGPN: similarity group proposal network for 3d point cloud instance segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2569–2578.

[23] Q. Pham, D. T. Nguyen, B. Hua, G. Roig, and S. Yeung, "JSIS3D: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8827–8836.

[24] J. Lahoud, B. Ghanem, M. R. Oswald, and M. Pollefeys, "3d instance segmentation via multi-task metric learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9255–9265.

[25] B. Zhang and P. Wonka, "Point cloud instance segmentation using probabilistic embeddings," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8883–8892.

[26] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang, "Hierarchical aggregation for 3d instance segmentation," *CoRR*, vol. abs/2108.02350, 2021.

[27] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4558–4567.

[28] I. Biederman, "Recognition-by-components: a theory of human image understanding." *Psychological review*, vol. 94, no. 2, p. 115, 1987.

[29] B. Graham, M. Engelcke, and L. van der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9224–9232.

[30] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5099–5108.

[31] C. Liu and Y. Furukawa, "MASC: multi-scale affinity with sparse convolution for 3d instance segmentation," *CoRR*, vol. abs/1902.04478, 2019.

[32] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "Panopticfusion: Online volumetric semantic mapping at the level of stuff and things," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4205–4212.

[33] S. Liu, S. Yu, S. Wu, H. Chen, and T. Liu, "Learning gaussian instance segmentation in point clouds," *CoRR*, vol. abs/2007.09860, 2020.

[34] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006, 2019, p. 1100612.