Computational Visual Media
https://doi.org/10.1007/s41095-022-0300-x Vol. 9, No. 4, December 2023, 699–715

Research Article

Semantic segmentation-assisted instance feature fusion for multi-level 3D part instance segmentation

Chun-Yu Sun[1], Xin Tong[2], and Yang Liu[2] (✉)

Abstract Recognizing 3D part instances from a 3D point cloud is crucial for 3D structure and scene understanding. Several learning-based approaches use semantic segmentation and instance center prediction as training tasks and fail to further exploit the inherent relationship between shape semantics and part instances. In this paper, we present a new method for 3D part instance segmentation. Our method exploits semantic segmentation to fuse nonlocal instance features, such as center prediction, and further enhances the fusion scheme in a multi- and cross-level way. We also propose a semantic region center prediction task to train and leverage the prediction results to improve the clustering of instance points. Our method outperforms existing methods with a large-margin improvement in the PartNet benchmark. We also demonstrate that our feature fusion scheme can be applied to other existing methods to improve their performance in indoor scene instance segmentation tasks.

Keywords 3D part instance segmentation; feature fusion; 3D deep learning

1 Introduction

3D instance segmentation is the task of distinguishing 3D instances from 3D data at the object or part level and extracting the instance semantics simultaneously [1–4]. It is essential for various applications, such as remote sensing, autonomous driving, mixed reality, 3D reverse engineering, and robotics. However, it is also a challenging task due to the diverse geometry and irregular distribution of 3D instances. Extracting part-level instances like chair wheels and desk legs becomes more difficult than segmenting object-level instances like beds and bookshelves, as the shapes of the parts have large variations in structure and geometry, while part-annotated data are scarce.

A popular learning-based approach to 3D instance segmentation follows the encoder–decoder paradigm, which predicts pointwise semantic labels and pointwise instance-aware features intercurrently [3, 5–10]. Instance-sensitive features can be either 3D instance centers, which have a clear geometric and semantic meaning, or feature vectors embedded in a high-dimensional space, where the feature vectors of the points within the same instance should be similar. The feature vectors of the points belonging to different instances are far apart from each other. Instance-aware features are used to group points into 3D instances via suitable clustering algorithms. Point semantics is usually used only in the clustering step. As the point set with the same semantics in a scene is composed of one or multiple 3D instances, it is natural to think about how to utilize this relation maximally. The works of Refs. [11] and [12] associate semantic features with instance-aware features to improve the learning of semantic features and instance features. However, they only fuse instance features with semantic features in a pointwise manner, without using semantics-similar points to provide nonlocal and robust guidance to instance features.

In this study, we leverage the probability vectors of semantic segmentation to help aggregate the instance features of points in an explicit and nonlocal way. We call our approach *semantic segmentation-assisted instance feature fusion*. The aggregated instance feature combined with the pointwise instance feature

1 Institute for Advanced Study, Tsinghua University, Beijing 100084, China. E-mail: sunchyqd@gmail.com.
2 Microsoft Research Asia, Beijing 100080, China. E-mail: X. Tong, xtong@microsoft.com; Y. Liu, yangliu@microsoft.com (✉).
Manuscript received: 2022-04-08; accepted: 2022-06-03

provides both global and local guidance to improve instance center prediction robustly, whose accuracy is critical to the final quality of instance clustering. Compared to existing feature fusion schemes [11, 12], our feature fusion strategy is more effective and simpler, as verified by our experiments.

Human-made 3D shapes, such as chairs, are composed of a set of meaningful parts and exhibit hierarchical 3D structures (see Fig. 1). Extracting multi-level part instances from the point cloud is challenging, especially for fine-level 3D instances, such as chair wheels. Existing studies independently performed 3D part instance segmentation on each structural level and also suffered from the insufficient labeled-data issue on some shape categories. By utilizing the hierarchy of shape semantics and part instances, we extend our feature fusion scheme in a multi- and cross-level manner, where the probability feature vectors at all levels are used to aggregate instance features.

Furthermore, to better distinguish part instances that are very close to each other, we propose to predict the centers of grouped instances, called *semantic region centers*, and use them to push the predicted instance centers away from them, as the semantic region centers play the role of the centers of a group of semantics-same part instances. On the PartNet dataset [13] in which 3D shapes have 3-level semantic part instances, our approach exceeds all existing approaches on the mean average precision (mAP) part category (IoU $>$ 0.5) by an average margin of +6.6% on 24 shape categories.

Our semantic segmentation-assisted instance feature fusion scheme is simple and lightweight; it is not limited to 3D part instance segmentation and can be extended to 3D instance segmentation for indoor scenes. We integrated several state-of-the-art 3D instance segmentation frameworks with our feature fusion scheme and observed consistent improvements on the benchmark of ScanNet [14] and S3DIS [15], which demonstrate the efficacy and generality of our approach.

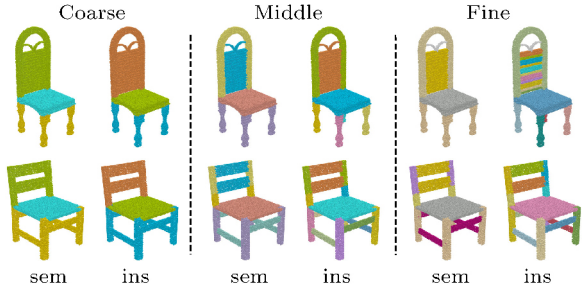

Fig. 1 Illustration of 3D models with fine-grained and hierarchical part structures. Models are selected from PartNet [13]. From left to right: part semantics and part instances at the coarse, middle, and fine level. Point colors are assigned to distinguish different part semantics and part instances.

Contributions. We make two contributions to tackle 3D instance segmentation: (1) We propose an instance feature fusion strategy that directly fuses instance features in a nonlocal way according to the guidance of semantic segmentation to improve instance center prediction. This strategy is lightweight and easily incorporated into many 3D instance segmentation frameworks for both 3D object and part instance segmentation. (2) Our multi- and cross-level instance feature fusion and the use of the semantic region center are effective for multi-level part instance segmentation and achieve the best performance on the PartNet benchmark. Our code and trained models are publicly available at `https://isunchy.github.io/projects/3d_instance_segmentation.html`.

2 Related work

2D instance segmentation. As surveyed by Ref. [16], four typical paradigms exist in the literature. The methods in the first paradigm generate mask proposals and then assign suitable shape semantics to the proposals [17–19]. The second one detects multiple objects using boxes and then extracts object masks within the boxes. Mask R-CNN [20] is one of the representative methods. The third is a bottom–up approach that predicts the semantic labels of each pixel and then groups pixels into 2D instances [21]. Its computation is relatively heavy due to per-pixel prediction. The fourth paradigm suggests using *dense sliding windows* techniques to generate mask proposals and mask scores for better instance segmentation [22, 23]. For detailed surveys, see Refs. [16, 24, 25].

3D instance segmentation. The existing 3D approaches follow the paradigms of 2D instance segmentation (*c.f.* Refs. [26, 27]). *Proposal-based methods* [13, 28] predict a fixed number of instance segmentation masks and match them with the ground truth using the Hungarian algorithm or a trainable assignment module. The learned matching scores are used to group 3D points into instances. *Detection-based methods* [2, 6, 29, 30] generate high-objectness 3D proposals like boxes and then refine them to obtain

instance masks.

Clustering-based methods first produce per-point predictions and then use clustering methods to group points into instances. SGPN [31] predicts the similarity score of any two points and merges points into instance groups according to the scores. MASC [32] predicts the multiscale affinity between neighboring voxels, for instance, clustering. Han et al. [33] regress the instance voxel occupancy for more accurate segmentation outputs. PointGroup [8] uses both the original and offset-shifted point sets to group points into candidate instances. DyCo3D [10] improves pointgroup by introducing a dynamic-convolution-based instance decoder. Observing that non-end-to-end clustered-based methods often exhibit over-segmentation and under-segmentation, Chen et al. [34] and Liang et al. [35] propose mid-level shape representation to generate instance proposals hierarchically in an end-to-end training manner. Liu et al. [7] approximate the distributions of centers to select center candidates for instance prediction. As mentioned in Section 1, most cluster-based methods treat semantic segmentation and instance feature learning as multitasks; only the works of Refs. [11] and [12] fuse the network features of the instance prediction branch and the semantic segmentation branch to improve the performance of both branches. Unlike the pointwise fusion of Refs. [11] and [12], our method fuses instance features in a nonlocal manner guided by semantic outputs, which is more robust and effective.

Part instance segmentation. Different from object-level 3D instance segmentation, part-level 3D instance segmentation is less studied due to limited annotated data and the difficulty brought by geometry-similar but semantics-different shape parts. Mo et al. [13] present PartNet—a large-scale dataset of 3D objects with fine-grained, instance-level, and hierarchical part information. For the part instance segmentation task, they developed a detection-by-segmentation method and trained a specific network to extract part instances per structural level, where the semantic hierarchy was used for part instance segmentation. Other object-level instance segmentation methods, such as Refs. [9, 10], have also been extended to the task of part instance segmentation, but they do not use the semantic hierarchy. Yu et al. [36] further enrich PartNet with information about the binary hierarchy and design a recursive neural network to perform recursive binary decomposition to extract 3D parts. Our multi- and cross-level instance feature fusion uses semantic hierarchy to improve instance center prediction. Furthermore, the use of semantic region centers assists instance grouping. The semantic region centers serve the role of symmetric centers of a group of semantics-same part instances and provide weak supervision to the training.

3 Methodology

In this section, we first introduce our baseline neural network for single-level and multi-level 3D part instance segmentation in Section 3.1, then present the model enhanced by our semantic segmentation-assisted instance feature fusion module in Section 3.2 and the semantic region center prediction module in Section 3.3.

3.1 Baseline network

Our baseline network follows the encoder–decoder paradigm. The input to the encoder is a set of 3D points $\mathcal{S}$ in which each point may be equipped with additional signals such as point normal and RGB color. Two parallel decoders are concatenated after the encoder to predict the point-wise semantic labels and the point offset to its corresponding instance center, named *semantic decoder* D_{sem} and *instance decoder* D_{ins}, respectively. The baseline network is depicted in Fig. 2, where the fusion module

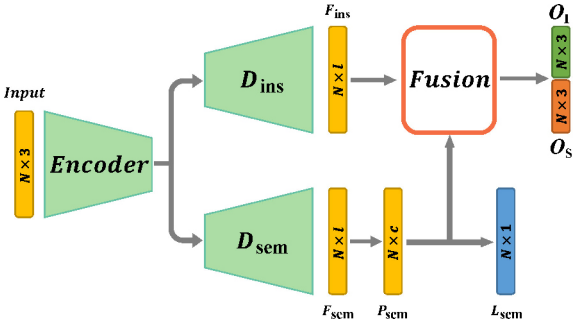

Fig. 2 Illustration of our network architecture for single-level part instance segmentation. The network takes a 3D point cloud as input. N is the point number. A shared encoder and two parallel decoders $D_{\text{sem}}, D_{\text{ins}}$ are used to output the pointwise semantic feature F_{sem} and instance feature F_{ins} to predict the point semantic label L_{sem} and the offset vector O_{I} to the instance center, and the offset vector O_{S} to the semantic region center. The feature fusion module aggregates the instance features of points according to semantic segmentation probability vectors to improve the offset prediction.

and semantic region center will be introduced in Sections 3.2 and 3.3, respectively.

The input points are shifted by the predicted offsets, and the shifted points with the same semantics are clustered into multiple 3D instances via the mean-shift algorithm [37]. In an ideal situation, all input points are shifted to their ground truth instance centers, but in practice, the accuracy of predicted offsets affects the performance of instance clustering.

Network structure. We choose O-CNN-based U-Nets [38, 39] as our encoder–decoder structure. The network is built on octree-based CNNs , and its memory and computational efficiency are similar to those of other sparse convolution-based neural networks [40, 41]. The input point cloud is converted to an octree first, whose non-empty finest octants store the average signal of the points contained by the octants. Both D_{sem} and D_{ins} output point-wise features via trilinear interpolation on sparse voxels: $F_{\text{sem}}, F_{\text{ins}} \in \mathbb{R}^{N\times l}$, where N is the number of points and l is the dimension of feature vectors.

Semantic prediction and offset prediction. A two-layer MLP is used to convert F_{sem} to the segmentation probability $P_{\text{sem}} \in \mathbb{R}^{N\times c}$, where c is the number of semantic classes. The segmentation label L_{sem} is then determined from P_{sem}. The loss for training semantic segmentation is the standard cross-entropy loss.

$$L_{\text{semantic}} = \frac{1}{N}\sum_{i=1}^{N} \text{CE}(p_i, p_i^*) \tag{1}$$

Here, p^* is the semantic label.

Parallel to the semantic branch, another two-layer MLP maps F_{ins} to the offset tensor $O_I \in \mathbb{R}^{N\times 3}$, which is used to shift the input points to the center of the target instance. The loss for predicting the offsets is the L_2 loss between the prediction and the ground-truth offsets.

$$L_{\text{offset}} = \frac{1}{N}\sum_{i=1}^{N} ||o_i - o_i^*||_2 \tag{2}$$

Here, o^* is the ground-truth offset.

Instance clustering. During the test phase, the network outputs pointwise semantics and offset vectors. We use the mean-shift algorithm to group the shifted points with the same semantics into disjointed instances.

Multi-level part instances. For shapes with hierarchical and multi-level part instances, there are two naive way to extend the baseline network: (1) train the baseline network for each level individually; (2) revise the baseline network to output multi-level semantics and multi-level offset vectors simultaneously by adding multi-prediction branches after F_{ins} and F_{sem}. We denote K as the level number, add a superscript k to all the symbols defined above to distinguish features at the k-th level, like $F_{\text{sem}}^{(k)}, F_{\text{ins}}^{(k)}, P_{\text{sem}}^{(k)}, c^{(k)}, O_{\text{I}}^{(k)}$.

3.2 Semantic segmentation-assisted instance feature fusion

3.2.1 Single-level instance feature fusion

As the points within the same instance possess the same instance center, it is essential to aggregate the instance features over these points to regress the offset to the instance center robustly. However, these points are not known during the network inference stage and they are also the objective of the task. The semantic decoder branch can predict the semantic region composed by a set of part instances; we can aggregate the instance features over the semantic parts to provide nonlocal guidance to the input points. We propose a semantic segmentation-assisted instance feature fusion module that contains two steps. In the first step, for each semantic part, we compute the instance feature based on the points associated with this part. Each point is associated with an aggregated instance feature from semantic parts in the second step according to its semantic probability vector. The instance feature fusion pipeline is illustrated in Fig. 3. Our feature aggregation procedure is as follows.

Part instance feature. We first aggregate the instance features with respect to the semantic label $m \in \{1, \cdots, c\}$ over the input:

$$Z_m := \frac{\sum_{\boldsymbol{p}\in\mathcal{S}} P_{\text{sem}}(\boldsymbol{p})|_m \cdot F_{\text{ins}}(\boldsymbol{p})}{\sum_{\boldsymbol{p}\in\mathcal{S}} P_{\text{sem}}(\boldsymbol{p})|_m} \tag{3}$$

where Z_m is the aggregated instance feature for the semantic part with semantic label of m, and $P_{\text{sem}}(\boldsymbol{p})|_m$ is the probability value of point $\boldsymbol{p}$ with respect to the semantic label m.

Aggregated instance feature. For each point $\boldsymbol{p}$, we aggregate the instance feature Z_m using the semantic probability of $\boldsymbol{p}$ as Eq. (4):

$$\hat{F}(\boldsymbol{p}) = \sum_{m=1}^{c} P_{\text{sem}}(\boldsymbol{p})|_m \cdot Z_m \tag{4}$$

The above equations for all points can be written in matrix form: $\boldsymbol{Z} = \left(\boldsymbol{P}_{\text{sem}} / \left(\boldsymbol{I}_1 \boldsymbol{P}_{\text{sem}}\right)\right)^{\text{T}} \boldsymbol{F}_{\text{ins}}, \hat{\boldsymbol{F}} = \boldsymbol{P}_{\text{sem}}\boldsymbol{Z}$, where $\boldsymbol{Z} \in \mathbb{R}^{c\times l}, \boldsymbol{P}_{\text{sem}} \in \mathbb{R}^{N\times c}, \boldsymbol{F}_{\text{ins}} \in$

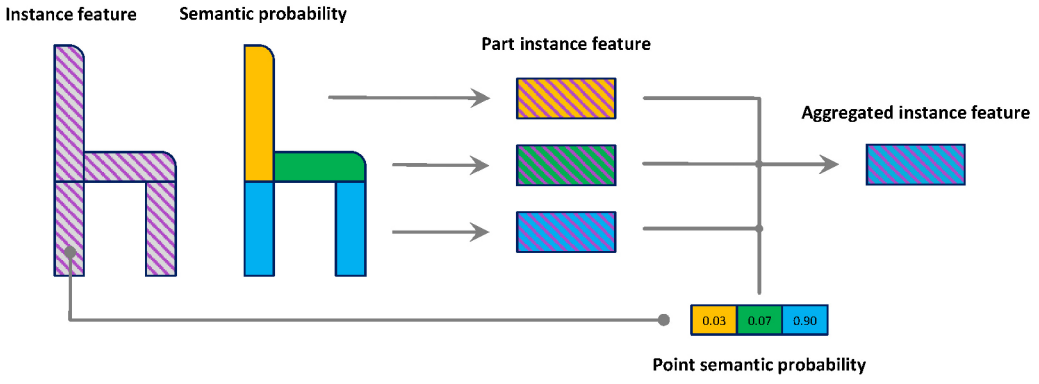

Fig. 3 Semantic segmentation-assisted instance feature fusion pipeline. Given the per-point instance feature and semantic probability, we get the part instance features according to the instance feature of points associated with each semantic part. Then we obtain the aggregated instance feature for each point by combining part instance features using its semantic probability.

$\mathbb{R}^{N\times l}$, $\hat{\boldsymbol{F}} \in \mathbb{R}^{N\times l}$, $\boldsymbol{I}_1$ is an $N \times N$ matrix with all ones, and "/" represents element-wise division.

We concatenate the aggregated instance feature $\hat{F}(\boldsymbol{p})$, the local instance feature $F_{\text{ins}}(\boldsymbol{p})$, and the position of $\boldsymbol{p}$ to form a fused instance feature $F_{\text{fusion}}(\boldsymbol{p}) := [\hat{F}(\boldsymbol{p}), F_{\text{ins}}(\boldsymbol{p}), \boldsymbol{p}]$, and use it to predict the instance center offset. Figure 4(a) illustrates our feature fusion module for a single level. The overall network structure is shown in Fig. 2.

3.2.2 Multi-level instance feature fusion

For shapes with multi-level part instances, our single-level instance feature fusion can be applied to each level individually. The naively extended baseline networks (Section 3.1) can benefit from this kind of instance feature fusion for multi-level part instance segmentation.

3.2.3 Cross-level instance feature fusion

When multi-level part instances and semantic segmentation exhibit a hierarchical relationship, i.e., the fine-level part instances are contained within the coarser-level part instances and can inherit the semantics from their parent level, we leverage the semantic segmentation in multi-levels to fuse instance features at each level, we call our strategy *cross-level instance feature fusion*. The exact fusion procedure is as follows.

Instance feature aggregation. On level k, we aggregate the instance features using semantic probability vectors at the r-th level:

$$Z_m^{(k,r)} := \frac{\sum_{\boldsymbol{q}\in\mathcal{S}} P_{\text{sem}}^{(r)}(\boldsymbol{q})|_m \cdot F_{\text{ins}}^{(k)}(\boldsymbol{q})}{\sum_{\boldsymbol{q}\in\mathcal{S}} P_{\text{sem}}^{(r)}(\boldsymbol{q})|_m}, \; m \in \{1, \cdots, c^{(r)}\} \tag{5}$$

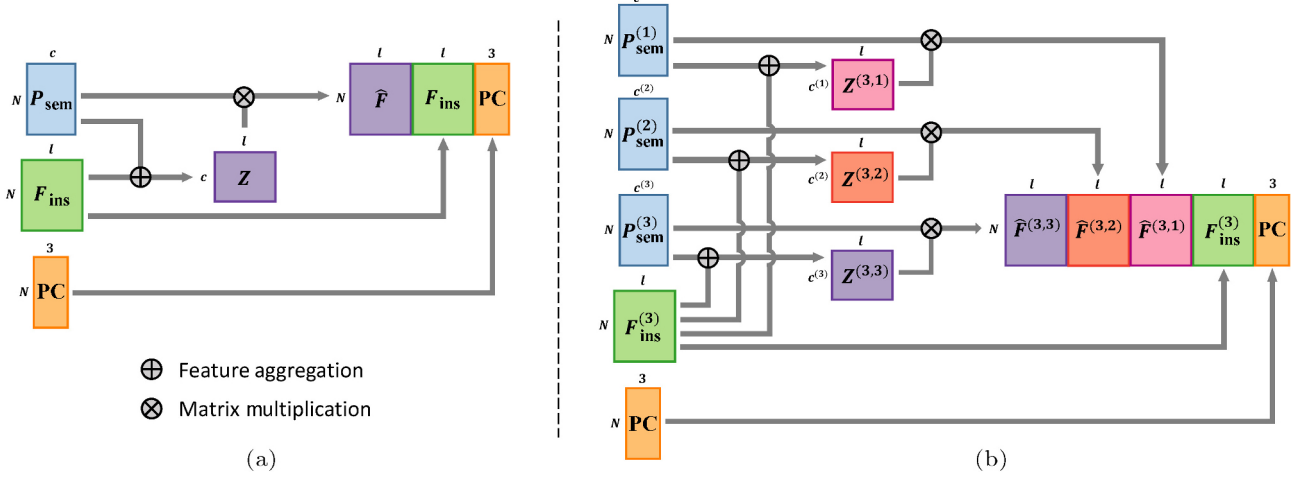

Fig. 4 Semantic segmentation-assisted instance feature fusion for single-level and cross-level. (a) Single-level instance feature fusion. Instance features $\boldsymbol{F}_{\text{ins}}$ are aggregated to $\hat{\boldsymbol{F}}$, with the help of semantic probability vectors $\boldsymbol{P}_{\text{sem}}$. $\hat{\boldsymbol{F}}$, $\boldsymbol{F}_{\text{ins}}$, and the point position **PC** are assembled to form the fused instance features $\boldsymbol{F}_{\text{fusion}}$. (b) Cross-level instance feature fusion for a 3-level part instance segmentation. The fused features at the 3rd level are depicted. For clarity, we omit fused features at other levels.

$Z_m^{(k,r)}$s are then averaged at point $\boldsymbol{p}$ at the k-th level:

$$\hat{F}^{(k,r)}(\boldsymbol{p}) = \sum_{m=1}^{c^{(r)}} P_{\text{sem}}^{(r)}(\boldsymbol{p})|_m \cdot Z_m^{(k,r)} \tag{6}$$

The fused instance feature of $\boldsymbol{p}$ at the k-th level is defined as

$$F_{\text{fusion}}^{(k)}(\boldsymbol{p}) := [\hat{F}^{(k,1)}(\boldsymbol{p}), \cdots, \hat{F}^{(k,K)}(\boldsymbol{p}), F_{\text{ins}}^{(k)}(\boldsymbol{p}), \boldsymbol{p}]$$

It is mapped to offset vectors at the k-th level by an MLP layer. We illustrate the cross-level instance feature fusion in Fig. 4(b).

3.3 Semantic region center

During the test phase, we use the mean-shift algorithm to split the offset-shifted points with the same semantics into different instances. For 3D instances which are close to each other, like two blades of a scissor shown in Fig. 5(a), it is difficult to separate the points belonging to them using mean-shift or other 3D point clustering algorithms, as the instance centers are very close to each other (see Fig. 5(b)). We introduce the concept of semantic region center, which is the center of semantically same instance centers. The semantic region center is usually the center of symmetrically arranged parts for human-made shapes. Figure 5(c) illustrates the semantic region centers. To make instance clustering easy, the instance centers can be further shifted away from the semantic region center, as shown Fig. 5(d). In the offset prediction branch of our network, we also add the offset prediction O_{S} to the center of the semantic region for each point.

In the instance clustering step, we shift the input points as Eq. (7):

$$\hat{\boldsymbol{p}} := \boldsymbol{p} + O_{\text{I}}(\boldsymbol{p}) + \lambda \cdot \frac{O_{\text{I}}(\boldsymbol{p}) - O_{\text{S}}(\boldsymbol{p})}{||O_{\text{I}}(\boldsymbol{p}) - O_{\text{S}}(\boldsymbol{p})||} \tag{7}$$

Here, $\boldsymbol{p} \in \mathcal{S}$, $\lambda > 0$.

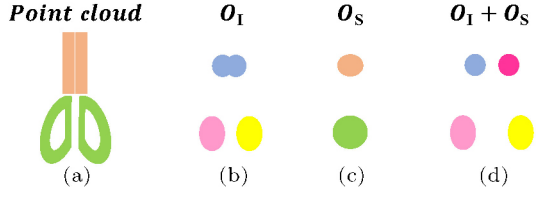

Fig. 5 Illustration of the use of semantic region centers. (a) Input point cloud of a scissor shape. Ground-truth part instances are colored according to their semantics. (b) Predicted instance centers. (c) Predicted semantic region centers. (d) By pushing the predicted instance centers away from the predicted semantic region centers, the shifted instance centers of the scissor blades become more distinguishable than in (b).

4 Experiments and analysis

We design a series of experiments and ablation studies to demonstrate the efficacy of our approach and its superiority to other fusion schemes, including multi-level part instance segmentation on PartNet [13] (Section 4.1), and instance segmentation on indoor scene datasets (Section 4.2): ScanNet [14] and S3DIS [15].

4.1 Part instance segmentation on PartNet

4.1.1 Experiments and comparison

Dataset. PartNet is a large-scale dataset with fine-grained and hierarchical part annotations. It contains more than 570k part instances over 26,671 3D models covering 24 object categories. It provides coarse-, middle-, and fine-grained part instance annotations.

Network configuration. The encoder and decoders of our O-CNN-based U-Net had five levels of domain resolution, and the maximum depth of the octree was six. The dimension of the feature was set to 64. Details of the U-Net structure are provided in Appendix A. We implemented our network in the TensorFlow framework [42]. The network was trained with 100,000 iterations with a batch size of 8. We used the SGD optimizer with a learning rate of 0.1 and decay two times with the factor of 0.1 at the 50,000-th and 75,000-th iterations. Our code and trained models are available.

Data processing. The input point cloud contained 10,000 points and was scaled into a unit sphere. During training, we also augmented each shape by a uniform scaling with the scale ratio of $[0.75, 1, 25]$, a random rotation whose pitch, yaw, and roll rotation angles were less than 10°, and random translations along each coordinate axis within the interval $[-0.125, 0.125]$. The train/test split is provided in PartNet. Note that not all categories have three-level part annotations. During training, we duplicated the labels at the coarser level to the finer level, if the latter was missing, to mimic the three-level shape structure. During the test phase, we only evaluated the output from the levels which exist in the data. The ground-truth instance centers and semantic region centers were pre-computed according to the semantic labels and part instances of PartNet.

Experiment setup. We set $\lambda = 0.05$ for Eq. (7). We used the mean-shift implementation implemented in scikit-learn [43]. The default bandwidth of mean-shift was set to 0.1. All our experiments were

conducted on an Azure Linux server with Intel Xeon Platinum 8168 CPU (2.7 GHz) and Tesla V100 GPU (16 GB memory). Our baseline network with cross-level fusion was the default configuration. In practice, we found that stopping the gradient from the fusion module to the semantic decoder helps maintain the semantic segmentation accuracy and slightly improves the instance segmentation. So, we enabled gradient stopping by default. An ablation study on gradient stopping is provided in Section 4.1.2.

Evaluation metrics. We used *per-category mAP score* with the IoU threshold of 0.25, 0.5, and 0.75 to evaluate the quality of part instance segmentation. They are denoted by AP_{25}, AP_{50}, and AP_{75}. s-AP_{50} is the metric proposed by Ref. [13], which averages the precision over the shapes.

Performance report and comparison. We report AP_{50} of our approach in all 24 shape categories in Table 1. We also report the performance of three comparison approaches: SGPN [31], PartNet [13], and PE [9]. The results are averaged over three levels of granularity. Our method outperformed the best competitor PE [9] by 6.6%, and also achieved the best performance in most categories. Our approach was also the best on other evaluation metrics, as shown in Table 2. Appendix C reports the per-category results of AP_{25}, AP_{75}, and s-AP_{50}. As DyCo3D [10] only performed instance segmentation experiments in four categories of the PartNet dataset, we compare it with our approach on these categories separately in Table 3. Our method outperformed DyCo3D by a large margin.

4.1.2 Ablation study

We validated our network design on PartNet instance segmentation, especially for the fusion module and the semantic region centers. The variants of our network are listed below.

- **Single-level baseline**: the network trained for each level individually without using the fusion module.
- **Multi-level baseline**: the network trained for multi-levels simultaneously without using the fusion module.
- **Single-level fusion**: Single-level baseline with single-level fusion.
- **Multi-level fusion**: Multi-level baseline with single-level fusion on each level.
- **Cross-level fusion**: Multi-level baseline with cross-level fusion.

Table 1 Part instance segmentation results of the test set on PartNet [13]. We report part-category AP_{50} on three instance levels. The results of other methods are reported by PE [9]. Bold numbers are better. Some shape categories, masked by dashed lines, have no middle- and fine-level instances for benchmark

	Level	Avg	*Bag*	*Bed*	*Bottle*	*Bowl*	*Chair*	*Clock*	*Dish*	*Disp*	*Door*	*Ear*	*Faucet*	*Hat*	*Key*	*Knife*	*Lamp*	*Laptop*	*Micro*	*Mug*	*Fridge*	*Scis*	*Stora*	*Table*	*Trash*	*Vase*
SGPN	Coarse	55.7	38.8	29.8	61.9	56.9	72.4	20.3	72.2	89.3	49.0	57.8	63.2	68.7	20.0	63.2	32.7	**100.0**	50.6	82.2	50.6	71.7	32.9	49.2	56.8	46.6
	Middle	29.7	—	15.4	—	—	25.4	—	58.1	—	25.4	—	—	—	—	—	21.7	—	49.4	—	22.1	—	30.5	18.9	—	—
	Fine	29.5	—	11.8	45.1	—	19.4	18.2	38.3	78.8	15.4	35.9	37.8	—	—	38.3	14.4	—	32.7	—	18.2	—	21.5	14.6	24.9	36.5
	Avg	46.8	38.8	19.0	53.5	56.9	39.1	19.3	56.2	84.1	29.9	46.9	50.5	68.7	20.0	50.8	22.9	**100.0**	44.2	82.2	30.3	71.7	28.3	27.6	40.9	41.6
PartNet	Coarse	62.6	64.7	48.4	63.6	59.7	74.4	42.8	76.3	93.3	52.9	57.7	69.6	70.9	43.9	58.4	37.2	**100.0**	50.0	86.0	50.0	80.9	45.2	54.2	71.7	49.8
	Middle	37.4	—	23.0	—	—	35.5	—	62.8	—	39.7	—	—	—	—	—	26.9	—	47.8	—	35.2	—	35.0	31.0	—	—
	Fine	36.6	—	15.0	48.6	—	29.0	32.3	53.3	80.1	17.2	39.4	44.7	—	—	45.8	18.7	—	34.8	—	26.5	—	27.5	23.9	33.7	52.0
	Avg	54.4	64.7	28.8	56.1	59.7	46.3	37.6	64.1	86.7	36.6	48.6	57.2	70.9	43.9	52.1	27.6	**100.0**	44.2	86.0	37.2	80.9	35.9	36.4	52.7	50.9
PE	Coarse	65.1	64.6	51.4	63.1	72.0	77.1	41.1	76.9	95.3	61.2	66.5	73.1	71.8	**48.6**	**76.5**	37.1	**100.0**	50.5	90.9	50.5	**88.6**	47.3	40.3	69.0	48.7
	Middle	40.4	—	31.0	—	—	38.6	—	64.2	—	36.9	—	—	—	—	—	31.0	—	51.2	—	37.3	—	42.0	31.5	—	—
	Fine	39.8	—	26.2	50.7	—	34.7	30.2	50.0	82.0	25.7	43.2	55.6	—	—	44.4	20.3	—	37.0	—	31.1	—	34.2	25.5	37.7	47.6
	Avg	57.5	64.6	36.2	56.9	72.0	50.1	35.6	63.7	88.7	41.3	54.9	64.4	71.8	**48.6**	**60.5**	29.5	**100.0**	46.2	90.9	39.6	**88.6**	41.2	32.4	53.4	48.1
Ours	Coarse	**71.2**	**80.5**	**54.1**	**66.9**	**84.2**	**84.1**	**51.2**	**79.9**	**97.2**	**76.8**	**71.6**	**79.2**	**77.3**	47.0	67.8	**38.2**	100.0	**62.5**	**91.8**	**57.4**	86.8	**56.4**	**65.3**	**79.7**	**53.8**
	Middle	**49.7**	—	**45.5**	—	—	**45.7**	—	**73.2**	—	**52.0**	—	—	—	—	—	**30.9**	—	**62.5**	—	**48.2**	—	**53.3**	**36.2**	—	—
	Fine	**47.8**	—	**40.9**	**55.9**	—	**38.2**	**37.1**	**56.5**	**87.4**	**41.3**	**53.7**	**59.1**	—	—	**48.8**	**21.7**	—	**49.7**	—	**44.1**	—	**44.0**	**28.9**	**51.3**	**54.6**
	Avg	**64.1**	**80.5**	**46.8**	**61.4**	**84.2**	**56.0**	**44.2**	**69.9**	**92.3**	**56.7**	**62.7**	**69.2**	**77.3**	47.0	58.3	**30.3**	100.0	**58.2**	**91.8**	**49.9**	86.8	**51.2**	**43.5**	**65.5**	**54.2**

Table 2 Part instance segmentation on the test set of PartNet. $AP_{25}, AP_{50}, AP_{75}$, s-$AP_{50}$ are averaged over three levels. The results of other methods are reported by PartNet [13] and PE [9]

	AP_{25}	AP_{50}	AP_{75}	s-AP_{50}	mIoU
SGPN [31]	—	46.8	—	64.2	—
PartNet [13]	62.8	54.4	38.9	72.2	—
PE [9]	66.5	57.5	41.7	—	—
Ours	**72.1**	**64.1**	**49.7**	**76.1**	66.1

Table 3 Part instance segmentation on the four categories of PartNet. AP_{50} is reported

	Level	Chair	Lamp	Stora.	Table
DyCo3D	Coarse	81.0	37.3	44.5	55.0
	Middle	41.3	28.8	38.9	32.5
	Fine	33.4	20.5	30.4	24.9
	Avg	51.9	28.9	37.9	37.5
Ours	Coarse	**84.1**	**38.2**	**56.4**	**65.3**
	Middle	**45.7**	**30.9**	**53.3**	**36.2**
	Fine	**38.2**	**21.7**	**44.0**	**28.9**
	Avg	**56.0**	**30.3**	**51.2**	**43.5**

For each variant, we use symbol † to indicate that the predicted semantic region centers are not used for instance clustering. The optimal variant is *cross-level fusion*. The performance of each variant is reported in Table 4.

Single-level baseline versus multi-level baseline. The performances of *single-level baseline* and *multi-level baseline* in the same setting (w. or w/o fusion and semantic region center) are not much different. However, the training effort of *multi-level baseline* is much lower. There are a total of 50 levels for all 24 categories of PartNet. The *single-level baseline* must train 50 networks, while the *multi-level baseline* only needs to train 24 networks.

Fusion module. It is clear that the performance of all baselines with the fusion modules improved. *Single-level fusion* and *multi-level fusion* increase AP_{50} by +3.9 and +4.4 points compared to their baselines, respectively. *Cross-level fusion* surpasses them at AP_{50} by +2.0 and +1.6 points. Here, the network of cross-level fusion has a slightly large network size. On Chair category, the network parameters of cross-level fusion, multi-level fusion, multi-level baselines are 8.13M, 7.98M, and 7.89M, respectively.

Use of semantic region centers. The instance segmentation performance is consistently improved by using semantic region centers. The improvement is also more noticeable when the fusion module is enabled to improve both the instance center prediction and the semantic region center prediction. For example, there is only +(0.2–0.3) improvement when using semantic region centers on *single-level baseline* and *multi-level baseline*, while the improvement over *cross-level fusion*† is +1.0.

Table 4 Ablation studies of our approach on PartNet test data. Methods marked with † use the predicted instance centers only. Our default and optimal network setting is *cross-level fusion*

	AP_{25}	AP_{50}	AP_{75}	s-AP_{50}	mIoU
single-level baseline†	67.3	57.9	45.3	74.4	64.9
single-level baseline	67.4	58.2	45.5	75.0	64.9
single-level fusion†	70.4	61.2	48.8	74.8	65.4
single-level fusion	71.1	62.1	49.0	75.8	65.4
multi-level baseline†	67.1	57.9	45.0	74.1	65.0
multi-level baseline	67.3	58.1	45.1	74.7	65.0
multi-level fusion†	70.9	61.8	48.8	74.8	65.5
multi-level fusion	71.5	62.5	49.2	75.6	65.5
cross-level fusion†	71.3	63.1	48.6	75.2	**66.1**
cross-level fusion	**72.1**	**64.1**	**49.7**	**76.1**	**66.1**
cross-level fusion(gradient)†	71.1	62.2	48.4	75.0	65.2
cross-level fusion(gradient)	71.8	63.3	49.3	75.9	65.2
cross-level fusion(one-hot)†	70.7	62.4	48.1	75.0	65.8
cross-level fusion(one-hot)	71.6	63.5	49.0	75.8	65.8
cross-level fusion(backbone)†	69.6	61.6	46.0	74.7	65.3
cross-level fusion(backbone)	70.2	62.4	47.1	75.3	65.3
cross-level fusion(two-dir)†	71.0	62.6	48.3	75.2	65.7
cross-level fusion(two-dir)	71.8	63.6	48.7	76.0	65.7
ASIS fusion†	68.2	59.0	45.0	74.7	65.1
ASIS fusion	68.6	59.1	45.9	75.0	65.1
JSNet fusion†	68.5	59.2	46.3	75.4	65.4
JSNet fusion	68.8	59.3	46.6	75.6	65.4

In Fig. 6, we present the instance segmentation results of *multi-level baseline*, *cross-level fusion*†, and *cross-level fusion*. The predicted instance centers are more compact and distinguishable when using the fusion module. The use of semantic region centers helps further separate close instances, e.g., the scissor blades in the 1st column, the bag handles in the 2nd column, and the chair back frames in the 7th column.

Stopping gradient. One of the inputs of the fusion module is the semantic segmentation probability. The gradients of the fusion module can backpropagate the errors to the semantic branch. In our experiments, we found that gradient backpropagation impairs semantic segmentation and leads to slightly worse instance segmentation results (see *cross-level fusion*(*gradient*) in Table 4).

Fig. 6 Visual comparison of part instance segmentation on the test set of PartNet. Part instances at the fine level are colored with random colors. 1st row: part instance ground-truth. 2nd row: results of our *multi-level baseline*†. 3rd row: results of our *cross-level fusion*† without using semantic region centers. 4th row: results of our *cross-level fusion* using semantic region centers. The corresponding shifted points are rendered at the top left of each instance segmentation image. Green and red boxes represent good and bad instances, respectively.

Instance feature aggregation. In our instance feature fusion module, we used the semantic probability of the point to aggregate the instance features from different semantic parts. An alternative way is to aggregate the instance features of the part which the point belongs to, i.e., using the one-hot version of semantic probability for each point. We found that our default fusion is better than this alternative (*cross-level fusion*(*one-hot*) in Table 4) because the instance features from different semantic parts can bring more contextual information, especially for points with fuzzy semantic probability.

Network backbone. The O-CNN [38, 39] backbone used in our network is different from the PointNet++ [44] backbone used in Refs. [9, 13]. Therefore, we also replaced the O-CNN backbone with PointNet++ for a fair comparison. As shown in *cross-level fusion*(*backbone*) in Table 4, the performance of the PointNet++ backbone with our fusion scheme is lower than that of the O-CNN backbone by 1.7 points in AP_{50}, but it is still much better than Refs. [13] and [9], by +8.0 and +4.9 points, respectively, in AP_{50}. This experiment further validates the efficacy of our approach.

Fusion scheme of ASIS [11] and JSNet [12]. We compare our fusion module with other fusion schemes proposed in ASIS [11] and JSNet [12]. ASIS jointly fuses the features between the segmentation and instance branches to improve the performance, as shown in Fig. 7(a). It has two fusion directions: one of them maps the semantic feature to the instance feature space using an MLP layer; the other one uses K-nearest neighbors in the instance feature space to aggregate the semantic feature. Similar to ASIS, JSNet also has two fusion directions as shown in Fig. 7(b). One maps the semantic feature

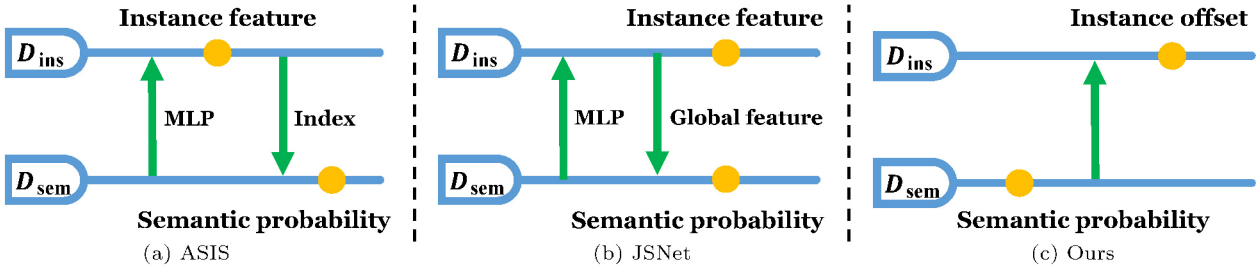

Fig. 7 Concept illustration of the fusion schemes of ASIS [11], JSNet [12], and our method. (a) ASIS has two fusion directions. It maps the semantic feature to the instance feature using an MLP layer and uses the nearest neighbors in the instance feature space to aggregate the semantic features. (b) JSNet also has two fusion directions. It maps the semantic feature to the instance feature using an MLP layer and adds the global instance feature to the pointwise semantic feature. (c) Our fusion module has only one fusion direction: the semantic probability directly helps the aggregation of instance features in a nonlocal manner.

to the instance feature space, and the other adds the global instance feature to the semantic feature. Our fusion scheme differs in two aspects compared to the ASIS and JSNet fusion modules. Firstly, our fusion module has only one fusion direction, as shown in Fig. 7(c), which uses semantic probability to guide the instance feature aggregation. Secondly, our fusion module uses the network output of semantic branch—*semantic probability* to guide the fusion of instance features, while ASIS and JSNet use the intermediate network information to fuse features. The fusion modules of ASIS and JSNet are more like enhancing the two decoders of the network, while our fusion module has a more specific target—to improve the accuracy of the predicted instance offsets. To prove the superiority of our fusion scheme, we replace our fusion module with the ASIS fusion and the JSNet fusion and integrate them with our *single-level baseline* and our loss functions. We observed +0.9 and +1.1 points improvement of AP_{50} over the baselines using semantic region centers (see ASIS fusion and JSNet fusion in Table 4). However, the improvements are minor compared to our *single-level fusion* which has +3.9 points improvement. In Fig. 8, we illustrate some results generated by different fusion methods. The shifted points of our fusion module are more compact and accurate, resulting in a more reasonable segmentation of the part instances. We also insert the other direction fusion into our fusion module by mapping the semantic feature to the instance feature space using an MLP layer. The performance is slightly worse than *cross-level fusion* due to the worse semantic segmentation results, as shown in *cross-level fusion*(*two-dir*) in Table 4.

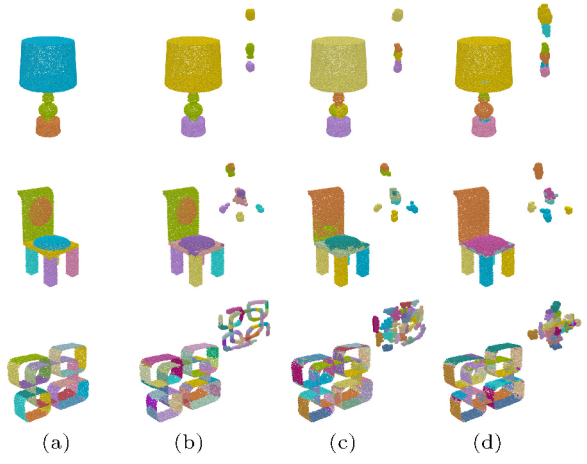

Fig. 8 Visualization comparison of different fusion methods on PartNet. (a) Part instance ground-truth. (b) Results of our fusion module. (c) Results of ASIS fusion module. (d) Results of JSNet fusion module. The corresponding shifted points are rendered at the top right of each instance segmentation image.

Bandwidth of mean-shift. We experienced different bandwidth values for the mean-shift algorithm: $0.05, 0.10, 0.20$, with *cross-level fusion*† setting. Their performance results are slightly different, as shown in the first three rows of Table 5. Mean-shift with bandwidth 0.10 performed better than the other two choices. Therefore, we used 0.10 by default.

Choices of λ. With the default bandwidth of the mean-shift algorithm, we experienced several choices of λ for Eq. (7): 0.025, 0.050, 0.075, under *cross-level fusion*. The last three rows of Table 5 show the results. $\lambda = 0.050$ achieved the best result, while larger λ could damage the centerness of the shifted points and did not comply with the predefined bandwidth. According to our empirical study, λ was set to 0.050 by default.

4.2 Instance segmentation on indoor scenes

Datasets. The ScanNet [14] dataset contains 1613 scans with annotations of 3D object instances. Instance segmentation was evaluated on 18 object categories. We report the results on the validation set. The S3DIS [15] dataset has 272 scenes with 13 categories. It was collected from six large-scale areas, covering more than 6000 m^2 with more than 273 million points. We report the performance on both Area-5 and 6-fold sets.

Evaluation metrics. For ScanNet, we use the widely-adopted evaluation metric, mAP; AP_{25} and AP_{50} denote AP scores with the IoU threshold of 0.25 and 0.5, respectively. In addition, AP averages the scores with the IoU threshold set from 0.5 to 0.95, with a step size of 0.05. For S3DIS, we use the metrics proposed by Ref. [11]: mCov, mWCov, mPrec, and mRec. mCov is the mean instance-wise IoU. mWCov

Table 5 Bandwidth and λ selection. The first three rows are our results for *cross-level fusion*† with different bandwidths. The last three rows are the results for *cross-level fusion* with different λ settings

Bandwidth	λ	AP_{25}	AP_{50}	AP_{75}	s-AP_{50}
0.05	—	70.5	61.8	48.2	75.0
0.10	—	**71.3**	**63.1**	**48.6**	**75.2**
0.20	—	71.1	62.4	**48.6**	74.4
0.10	0.025	71.9	64.0	**49.7**	76.0
0.10	0.050	**72.1**	**64.1**	**49.7**	**76.1**
0.10	0.075	70.0	61.9	47.5	74.8

is the weighted version of mCov, where the weights are determined by the sizes of each instance. mPrec and mRec denote the mean precision and recall with an IoU threshold of 0.5. In both datasets, we also report the semantic segmentation metric mIoU for reference.

Experiment setup. To demonstrate the efficiency of our instance feature fusion module and its applicability to different network designs, we integrated our single-level fusion module into some recent instance segmentation frameworks, which have both the semantic segmentation branch and the instance feature branch: PointGroup [8], DyCo3D [10], HAIS [34], ASIS [11], and JSNet [12]. The settings of the original frameworks, such as loss functions, clustering algorithms, and training protocols, were kept. Our multi- or cross-level fusion is not used here as there are no multi-level instances on the indoor scene datasets. On the ScanNet dataset, we used the original frameworks of PointGroup, DyCo3D, and HAIS as baselines and inserted our fusion module to help in network training. As the work of HAIS and DyCo3D leveraged pretrained network weights to initialize the network weights to obtain high performance, for a fair comparison, we followed their method and used pretrained weights as initialization to train their networks with our fusion module. In Appendix B, we also provided the comparison without using any pretrained weights. On the S3DIS dataset, we retrained ASIS and JSNet with and without their original fusion modules, and trained the networks by replacing their fusion modules with our fusion module for further comparison.

Performance report and time analysis. Table 6 shows the performance results of PointGroup, DyCo3D, and HAIS with and without our fusion module on the validation set of ScanNet. Our fusion module consistently improved these methods: +2.4, +1.1, and +0.8 points on AP, and +1.6, +0.7, and +0.5 points on AP_{50}. In Fig. 9, we present some instance segmentation results by HAIS with and without our fusion module. Without our fusion module, the shifted points have a larger distribution which can lead to wrong clustering results, as highlighted by the red circles. With our fusion module, the shifted points are closer to their instance centers, which helps to achieve more accurate clustering results.

Table 6 Quantitative comparison on ScanNet [14] validation set. Our fusion module is added to each network (marked with $*$) and exhibits consistent performance improvements. The results of other methods are from their released models and checkpoints. We used their pre-trained weights for initialization and training of the whole network with our fusion module

Method	AP	AP_{50}	AP_{25}	mIoU
PointGroup	35.2	57.1	71.4	67.3
PointGroup*	**37.6**	**58.7**	**71.8**	**67.6**
DyCo3D	35.5	57.6	72.9	**69.5**
DyCo3D*	**36.6**	**58.3**	**73.2**	**69.5**
HAIS	44.1	64.4	75.7	72.3
HAIS*	**44.9**	**64.9**	**75.9**	**72.4**

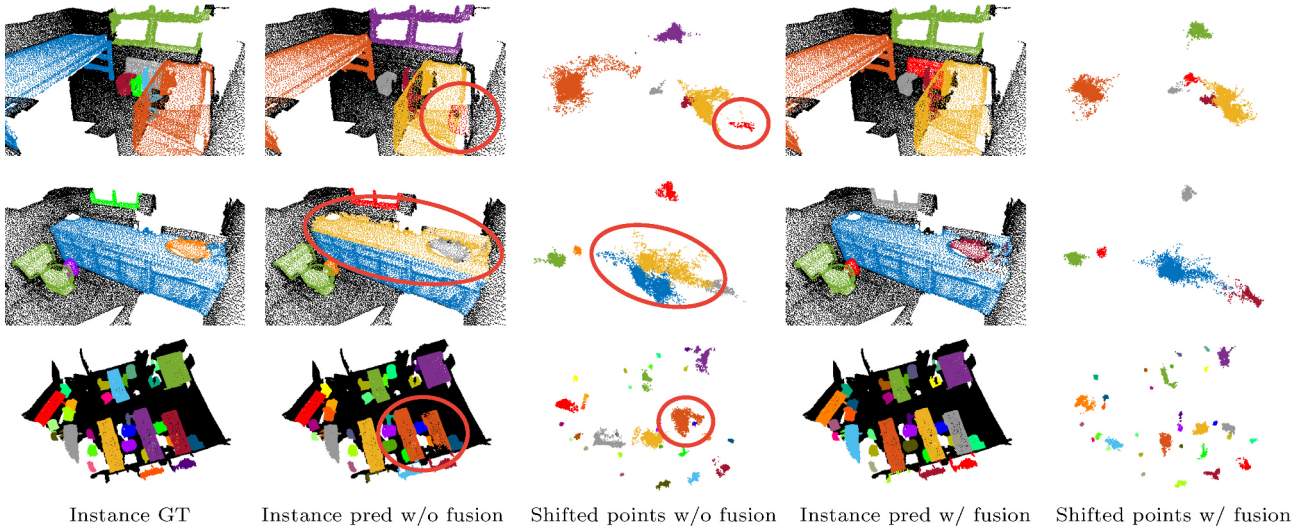

Fig. 9 Visual comparison of instance segmentation on the validation set of ScanNet. Without the fusion module, the shifted points are more dispersive and result in wrong instance segmentation results, as shown in the red circles. Our fusion module can help to get more accurate offsets, and the compact shifted points can get better instance clustering.

On S3DIS, we retrained ASIS and JSNet with and without their original fusion modules, and we also integrated our fusion module with their base networks. As reported in Table 7, the improvement of our fusion module outperformed their original fusion modules.

On the above experiments, the additional inference time caused by our fusion module for each method was small compared to the total time, as reported in Table 8. The additional time of our fusion is also smaller than the fusion modules of ASIS and JSNet. We conclude that our fusion module is a lightweight and an effective plugin to improve the performance of other methods.

Table 7 Quantitative comparison on S3DIS [15]. b-ASIS is the baseline of ASIS, i.e., without the ASIS feature fusion module. Similarly, b-JSNet is the baseline of JSNet. We added our fusion module to each method marked with ∗. The number before parentheses is the metric on Area 5, while the number inside parentheses is the metric on 6-fold cross-validation

Method	mCov	mWCov	mPrec	mRec	mIoU
b-ASIS	45.4(49.0)	48.6(53.0)	53.7(58.8)	42.9(47.3)	52.0(58.4)
ASIS	45.8(49.4)	48.9(53.3)	54.7(59.5)	**43.6**(47.4)	52.3(58.8)
b-ASIS*	**46.1**(**50.4**)	**49.2**(**54.4**)	**55.4**(**63.0**)	43.4(**50.2**)	**53.1**(**59.3**)
b-JSNet	47.9(50.8)	50.7(54.8)	55.6(60.7)	44.8(49.7)	53.5(59.5)
JSNet	48.8(51.7)	51.6(55.5)	56.6(61.1)	46.1(50.6)	53.9(59.9)
b-JSNet*	**49.5**(**51.9**)	**52.6**(**55.8**)	**58.6**(**63.1**)	**46.6**(**51.0**)	**54.7**(**60.4**)

Table 8 Average inference time for a 3D scan. Methods using our fusion module are marked with ∗. The first three methods are measured on ScanNet validation set and the last two methods are measured on Area 5 of S3DIS. The runtime was measured on Tesla V100 GPU

Method	Inference time (ms)
PointGroup	428
PointGroup*	439(+11)
DyCo3D	392
DyCo3D*	400(+8)
HAIS	375
HAIS*	388(+13)
b-ASIS	3405
ASIS	5058(+1653)
b-ASIS*	3646(+241)
b-JSNet	4138
JSNet	4256(+118)
b-JSNet*	4192(+54)

5 Conclusions

We present a novel semantic segmentation-assisted instance feature fusion scheme and an improved instance clustering method via the semantic region center for multi-level 3D part instance segmentation. Our method explicitly utilizes the inherent relationship between semantic segmentation and part instances considering their hierarchy. Its efficacy is well demonstrated on a challenging 3D shape dataset—PartNet. Our feature fusion scheme also integrates well with other state-of-the-art 3D indoor-scene instance segmentation models, which it consistently improve on ScanNet and S3DIS.

Limitation. In our algorithm for PartNet, the bandwidth of the mean-shift algorithm and the shift parameter λ were set empirically. Devising a differentiable clustering algorithm with trainable bandwidth and λ for end-to-end training would help improve the instance segmentation accuracy further. The approach of taking mean-shift iterations as differentiable recurrent functions [45] is a promising direction.

Appendix

A U-Net structure

We used an O-CNN-based U-Net structure with two decoders as our base network. The encoder and decoders have five levels of domain resolution, and the maximum depth of the octree is 6, as illustrated in Fig. 10.

B Training from scratch in ScanNet

For the methods of PointGroup [8], DyCo3D [10], and HAIS [34], we trained their networks using the default setting of their released codes from scratch with and without our fusion module. The results in Table 9 show that our fusion module led to consistent improvements. Note that all methods trained from

Table 9 Quantitative comparison on ScanNet [14] validation set. Our fusion module is added to each network (marked with ∗) and exhibits consistent performance improvements. The other methods are trained from scratch using their released codes. The networks with our fusion module are also trained from scratch

Method	AP	AP_{50}	AP_{25}	mIoU
PointGroup	33.6	55.4	70.0	67.1
PointGroup*	**34.4**	**56.1**	**71.7**	**67.3**
DyCo3D	32.5	53.0	69.0	67.2
DyCo3D*	**34.5**	**55.8**	**70.7**	**67.6**
HAIS	42.5	61.7	73.5	71.0
HAIS*	**43.1**	**62.8**	**74.5**	**71.4**

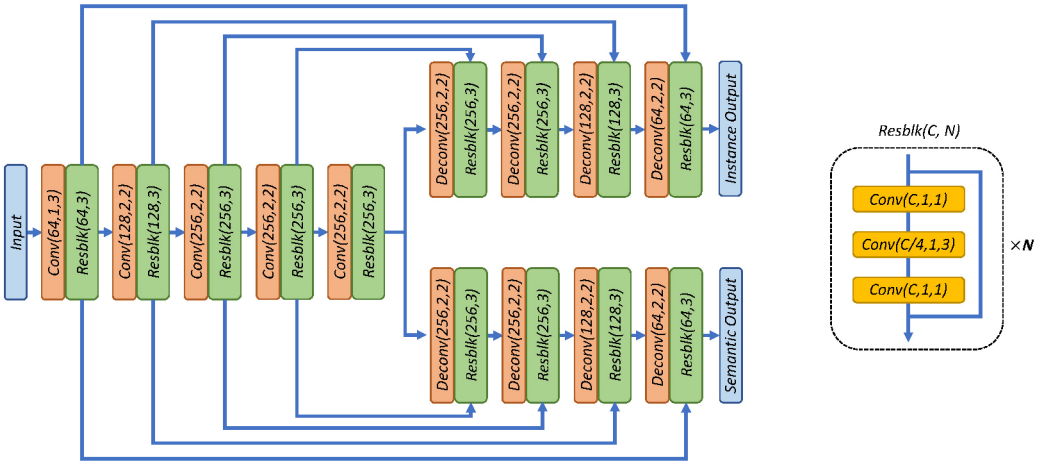

Fig. 10 O-CNN-based U-Net structure for instance segmentation on the PartNet dataset. $Conv(C, S, K)$ and $Deconv(C, S, K)$ represent octree-based convolution and deconvolution. C, S, K are the output channel number, stride, and kernel size, respectively.

Table 10 Part instance segmentation results on the test set of PartNet [13]. We report AP_{25}, AP_{75}, and s-AP_{50} on three instance levelles. Bold numbers are better

AP_{25}

	Level	Avg	Bag	Bed	Bottle	Bowl	Chair	Clock	Dish	Disp	Door	Ear	Faucet	Hat	Key	Knife	Lamp	Laptop	Micro	Mug	Fridge	Scis	Stora	Table	Trash	Vase
PartNet [13]	Coarse	70.2	**89.4**	**82.3**	65.2	63.1	78.1	48.0	79.1	97.1	64.9	64.6	77.3	73.9	58.9	59.2	42.5	**100.0**	50.0	92.9	50.0	96.3	57.7	59.3	82.7	52.6
	Middle	46.7	—	44.5	—	—	43.0	—	71.3	—	49.3	—	—	—	—	—	32.2	—	51.2	—	45.2	—	46.7	36.5	—	—
	Fine	45.6	—	29.0	52.6	—	35.3	39.6	59.9	89.3	27.1	56.9	55.0	—	—	49.0	22.6	—	56.9	—	35.6	—	36.3	28.6	44.8	57.0
	Avg	62.8	89.4	51.9	58.9	63.1	52.1	43.8	70.1	93.2	47.1	60.8	66.2	73.9	58.9	54.1	32.4	**100.0**	52.7	92.9	43.6	96.3	46.9	41.5	63.8	54.8
PE [9]	Coarse	72.7	82.8	79.6	65.6	72.0	82.8	49.1	83.8	98.3	75.5	74.3	83.2	79.5	**59.9**	**78.8**	**45.2**	**100.0**	50.5	95.4	51.6	**96.9**	60.9	44.6	82.9	51.1
	Middle	51.4	—	55.4	—	—	47.1	—	78.0	—	48.1	—	—	—	—	—	**39.3**	—	54.4	—	48.8	—	53.7	37.7	—	—
	Fine	51.6	—	44.4	57.2	—	43.2	45.7	64.8	90.7	34.6	59.3	67.2	—	—	53.0	26.0	—	60.0	—	**51.5**	—	44.4	31.7	50.0	53.9
	Avg	66.5	82.8	59.8	61.4	72.0	57.7	47.4	75.6	94.5	52.7	66.8	75.2	79.5	**59.9**	**65.9**	36.8	**100.0**	55.0	95.4	50.6	**96.9**	53.0	38.0	66.5	52.5
Ours	Coarse	**78.1**	83.8	72.7	**68.2**	**84.2**	**87.3**	**58.5**	**87.2**	**99.0**	**82.8**	**80.5**	**88.2**	**87.9**	53.2	71.3	43.9	**100.0**	**83.4**	**96.7**	**61.3**	96.0	**70.3**	**77.1**	**85.8**	**54.3**
	Middle	**62.1**	—	**67.3**	—	—	**54.0**	—	**82.6**	—	**67.1**	—	—	—	—	—	38.8	—	**80.9**	—	**60.5**	—	**64.4**	**43.4**	—	—
	Fine	**58.4**	—	**58.8**	**62.9**	—	**47.0**	**49.9**	**70.1**	**93.4**	**52.0**	**65.4**	**70.8**	—	—	**54.3**	**30.0**	—	**72.2**	—	51.2	—	**53.8**	**36.7**	**62.0**	**61.8**
	Avg	**72.1**	83.8	**66.3**	**65.6**	**84.2**	**62.8**	**54.2**	**80.0**	**96.2**	**67.3**	**73.0**	**79.5**	**87.9**	53.2	62.8	**37.6**	**100.0**	**78.8**	**96.7**	**57.7**	96.0	**62.8**	**52.4**	**73.9**	**58.1**

AP_{75}

	Level	Avg	Bag	Bed	Bottle	Bowl	Chair	Clock	Dish	Disp	Door	Ear	Faucet	Hat	Key	Knife	Lamp	Laptop	Micro	Mug	Fridge	Scis	Stora	Table	Trash	Vase
PartNet [13]	Coarse	47.4	39.7	14.6	**60.6**	41.4	58.3	28.8	58.3	84.7	35.6	49.1	48.2	66.3	10.7	48.7	**29.6**	**98.0**	47.8	76.1	50.0	35.1	29.9	43.2	42.2	40.5
	Middle	22.0	—	4.2	—	—	21.4	—	37.2	—	22.4	—	—	—	—	—	19.6	—	32.1	—	16.7	—	22.8	22.0	—	—
	Fine	23.5	—	3.9	37.9	—	16.6	17.6	29.8	63.2	8.1	27.6	25.8	—	—	31.0	13.6	—	23.9	—	12.1	—	18.2	16.4	19.7	34.5
	Avg	38.9	39.7	7.6	49.2	41.4	32.1	23.2	41.7	73.9	22.0	38.4	37.0	66.3	10.7	39.8	20.9	**98.0**	34.6	76.1	26.3	35.1	23.6	**27.2**	31.0	37.5
PE [9]	Coarse	50.0	40.3	13.3	60.2	60.2	59.3	28.2	61.9	90.6	39.1	**59.6**	54.2	69.3	7.4	**65.7**	28.5	**98.0**	47.9	77.1	50.5	42.8	30.1	34.8	40.7	41.1
	Middle	23.8	—	7.1	—	—	22.8	—	37.4	—	21.3	—	—	—	—	—	22.0	—	**35.5**	—	20.6	—	26.1	21.4	—	—
	Fine	25.7	—	7.3	**38.8**	—	20.5	17.2	30.0	66.8	10.8	28.2	33.2	—	—	31.5	14.1	—	25.6	—	17.1	—	21.0	17.4	19.4	38.0
	Avg	41.7	40.3	9.2	**49.5**	60.2	34.2	22.7	43.1	78.7	23.7	**43.9**	43.7	69.3	7.4	48.6	21.5	**98.0**	**36.4**	77.1	29.4	42.8	25.7	24.5	30.0	39.6
Ours	Coarse	**57.7**	**61.1**	**22.0**	44.8	**66.1**	**70.6**	**37.4**	**65.0**	**91.7**	**52.3**	55.5	**65.5**	**72.4**	**44.4**	62.1	28.6	**98.0**	**49.0**	**87.9**	**54.0**	**62.8**	**37.7**	**48.7**	**61.0**	**45.3**
	Middle	**31.2**	—	**19.1**	—	—	**32.4**	—	**47.7**	—	**32.5**	—	—	—	—	—	**23.6**	—	33.5	—	**29.4**	—	**36.6**	**25.9**	—	—
	Fine	**31.6**	—	**16.9**	33.2	—	**25.4**	**19.6**	**36.3**	**74.5**	**21.9**	**30.8**	**44.8**	—	—	**35.5**	**16.4**	—	**25.9**	—	**27.8**	—	**30.4**	**20.4**	**33.9**	**43.0**
	Avg	**49.7**	**61.1**	**19.3**	39.0	**66.1**	**42.8**	**28.5**	**49.7**	**83.1**	**35.6**	43.2	**55.2**	**72.4**	**44.4**	**48.8**	**22.9**	**98.0**	36.1	**87.9**	**37.1**	**62.8**	**34.9**	**31.7**	**47.5**	**44.2**

s-AP_{50}

	Level	Avg	Bag	Bed	Bottle	Bowl	Chair	Clock	Dish	Disp	Door	Ear	Faucet	Hat	Key	Knife	Lamp	Laptop	Micro	Mug	Fridge	Scis	Stora	Table	Trash	Vase
SGPN [31]	Coarse	72.5	62.8	38.7	76.7	83.2	91.5	41.5	81.4	91.3	71.2	81.4	82.2	71.9	23.2	**78.0**	60.3	**100.0**	76.2	94.3	60.6	74.9	55.0	80.1	76.1	87.1
	Middle	50.2	—	22.7	—	—	51.1	—	78.7	—	43.3	—	—	—	—	—	49.1	—	68.6	—	42.9	—	51.9	43.7	—	—
	Fine	50.2	—	17.5	66.5	—	42.3	40.7	59.3	83.9	29.0	60.2	61.6	—	—	55.0	37.6	—	53.7	—	30.6	—	45.1	37.8	50.0	82.0
	Avg	64.2	62.8	26.3	71.6	83.2	61.6	41.1	73.1	87.6	47.8	70.8	71.9	71.9	23.2	66.5	49.0	**100.0**	66.2	94.3	44.7	74.9	50.7	53.8	63.0	84.6
PartNet [13]	Coarse	80.3	78.4	62.2	80.8	83.8	**94.9**	74.6	81.4	94.3	76.1	**87.1**	**86.5**	77.8	44.5	76.6	65.0	**100.0**	79.5	95.3	**79.0**	87.6	62.7	88.1	82.3	**89.0**
	Middle	60.5	—	29.4	—	—	64.7	—	75.4	—	61.1	—	—	—	—	—	**56.8**	—	78.2	—	61.7	—	57.4	**59.4**	—	—
	Fine	57.7	—	22.1	68.3	-	58.4	53.7	67.5	84.8	38.0	62.4	66.8	—	—	**63.5**	45.8	—	54.0	—	45.0	—	52.6	**52.5**	58.7	86.4
	Avg	72.2	78.4	37.9	74.6	83.8	72.7	64.2	74.8	89.5	58.4	74.8	76.6	77.8	44.5	**70.1**	55.8	**100.0**	70.6	95.3	61.9	87.6	57.6	66.7	70.5	87.7
Ours	Coarse	**83.3**	**84.6**	**75.8**	**91.0**	**88.7**	94.8	**74.9**	**86.3**	**97.4**	**83.4**	86.3	85.7	**80.1**	**47.4**	76.5	**65.8**	**100.0**	**84.7**	**96.2**	75.9	**88.6**	**73.0**	**90.5**	**83.4**	88.5
	Middle	**66.2**	—	**50.7**	—	—	**65.6**	—	**81.5**	—	**66.5**	—	—	—	—	—	54.8	—	**80.9**	—	**70.9**	—	**66.7**	58.5	—	—
	Fine	**63.9**	—	**39.1**	**70.1**	—	**59.5**	**54.8**	**69.5**	**89.1**	**56.5**	**69.5**	**73.7**	—	—	55.6	**47.4**	—	**67.2**	—	**63.3**	—	**63.9**	51.9	**66.2**	**88.4**
	Avg	**76.1**	**84.6**	**55.2**	**80.6**	**88.7**	**73.3**	**64.9**	**79.1**	**93.3**	**68.8**	**77.9**	**79.7**	**80.1**	**47.4**	66.1	**56.0**	**100.0**	**77.6**	**96.2**	**70.0**	**88.6**	**67.9**	**67.0**	**74.8**	**88.5**

scratch are inferior to their versions using pretrained weights.

Remark. The above networks trained from scratch do not reproduce the performance of the released checkpoints of these works. The authors of DyCo3D and HAIS responded that their released checkpoints used other pre-trained network weights and were not trained from scratch.

C Evaluation and visualization in PartNet

We report AP_{25}, AP_{75}, and s-AP_{50} on the 24 shape categories of PartNet in Table 10. In Fig. 11,

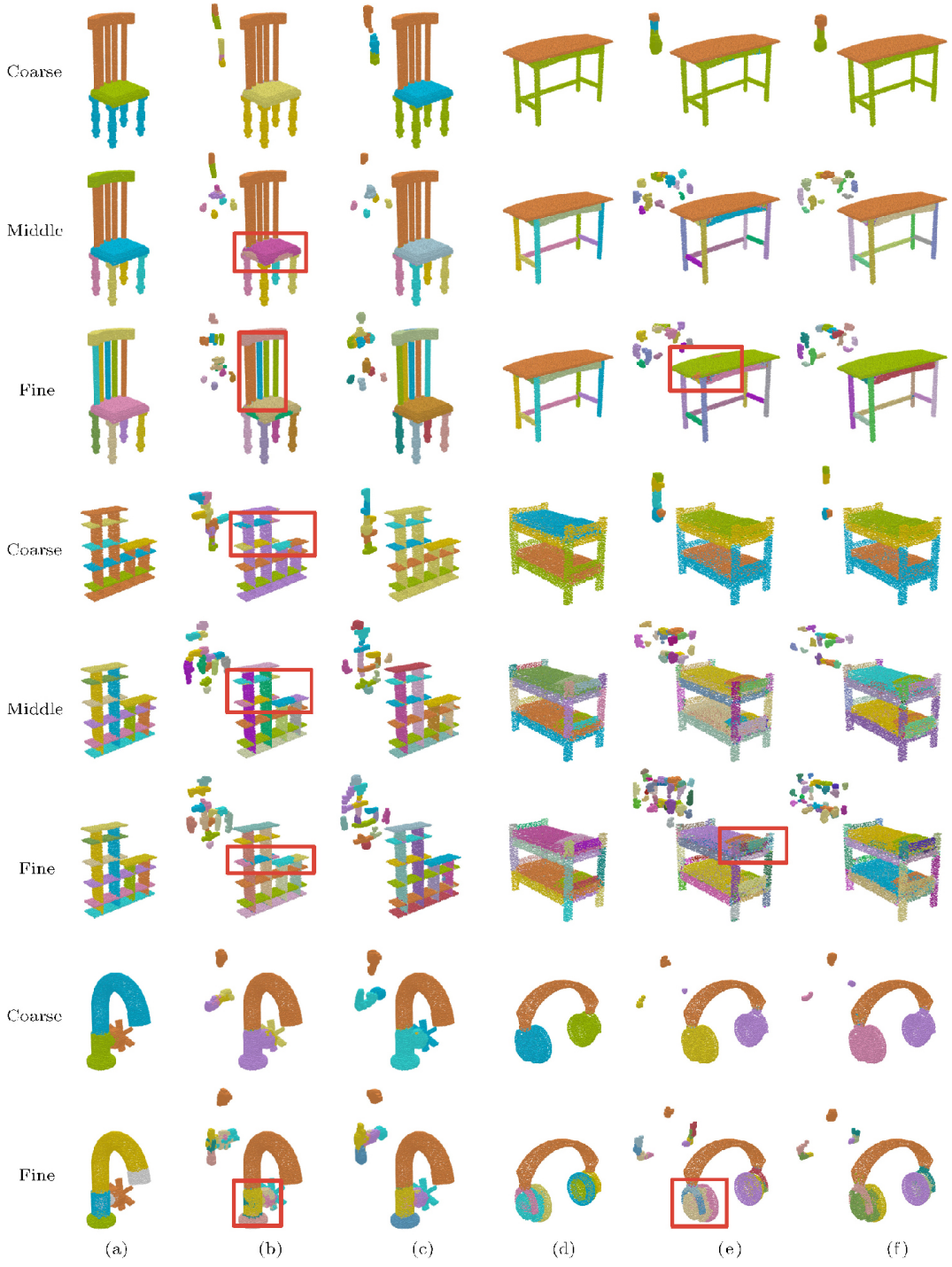

Fig. 11 Visual comparison of part instance segmentation on the test set PartNet. Part instances at each level are colored with random colors. (a, d) Ground truth instance parts. (b, e) Results of our *multi-level baseline*†. (c, f) Results of our *cross-level fusion*. The corresponding shifted points are rendered on the top-left of each instance segmentation image. Red boxes represent wrong instance results.

we illustrate the *multi-level baseline* and *cross-level fusion* instance segmentation results. Our fusion module helps obtain more compact and distinguishable instance centers and yielded better instance segmentation results.

Availability of data and materials

PartNet, ScanNet, and S3DIS are all publicly released datasets.

Author contributions

Chun-Yu Sun proposed and implemented the key idea, conducted the main experiments, and contributed to paper writing. Xin Tong supervised the findings of this work and verified the key concept. Yang Liu led the project and contributed to the key concept, experimental design, and paper writing.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. SEGCloud: Semantic segmentation of 3D point clouds. In: Proceedings of the International Conference on 3D Vision, 537–547, 2017.

[2] Yang, B.; Wang, J. N.; Clark, R.; Hu, Q. Y.; Wang, S.; Markham, A.; Trigoni, N. Learning object bounding boxes for 3D instance segmentation on point clouds. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Article No. 605, 6740–6749, 2019.

[3] Lahoud, J.; Ghanem, B.; Oswald, M. R.; Pollefeys, M. 3D instance segmentation via multi-task metric learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9255–9265, 2019.

[4] Zhang, F. H.; Guan, C. Y.; Fang, J.; Bai, S.; Yang, R. G.; Torr, P. H. S.; Prisacariu, V. Instance segmentation of LiDAR point clouds. In: Proceedings of the IEEE International Conference on Robotics and Automation, 9448–9455, 2020.

[5] Tan, J. G.; Chen, L. L.; Wang, K. R.; Li, J. M.; Zhang, X. L. SASO: Joint 3D semantic-instance segmentation via multi-scale semantic association and salient point clustering optimization. *IET Computer Vision* Vol. 15, No. 5, 366–379, 2021.

[6] Engelmann, F.; Bokeloh, M.; Fathi, A.; Leibe, B.; Nießner, M. 3D-MPA: Multi-proposal aggregation for 3D semantic instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9028–9037, 2020.

[7] Liu, S. H.; Yu, S. Y.; Wu, S. C.; Chen, H. T.; Liu, T. L. Learning Gaussian instance segmentation in point clouds. *arXiv preprint* arXiv:2007.09860, 2020.

[8] Jiang, L.; Zhao, H. S.; Shi, S. S.; Liu, S.; Fu, C. W.; Jia, J. Y. PointGroup: Dual-set point grouping for 3D instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4866–4875, 2020.

[9] Zhang, B.; Wonka, P. Point cloud instance segmentation using probabilistic embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8879–8888, 2021.

[10] He, T.; Shen, C. H.; van den Hengel, A. DyCo3D: Robust instance segmentation of 3D point clouds through dynamic convolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 354–363, 2021.

[11] Wang, X. L.; Liu, S.; Shen, X. Y.; Shen, C. H.; Jia, J. Y. Associatively segmenting instances and semantics in point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4091–4100, 2019.

[12] Zhao, L.; Tao, W. B. JSNet: Joint instance and semantic segmentation of 3D point clouds. *Proceedings of the AAAI Conference on Artificial Intelligence* Vol. 34, No. 7, 12951–12958, 2020.

[13] Mo, K. C.; Zhu, S. L.; Chang, A. X.; Yi, L.; Tripathi, S.; Guibas, L. J.; Su, H. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 909–918, 2019.

[14] Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2432–2443, 2017.

[15] Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1534–1543, 2016.

[16] Hafiz, A. M.; Bhat, G. M. A survey on instance segmentation: State of the art. *International Journal*

of Multimedia Information Retrieval Vol. 9, No. 3, 171–189, 2020.

[17] Girshick, R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 1440–1448, 2015.

[18] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, 91–99, 2015.

[19] Wang, X. L.; Kong, T.; Shen, C. H.; Jiang, Y. N.; Li, L. SOLO: Segmenting objects by locations. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12363*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 649–665, 2020.

[20] He, K. M.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2980–2988, 2017.

[21] Bai, M.; Urtasun, R. Deep watershed transform for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2858–2866, 2017.

[22] Dai, J. F.; He, K. M.; Li, Y.; Ren, S. Q.; Sun, J. Instance-sensitive fully convolutional networks. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 534–549, 2016.

[23] Chen, X. L.; Girshick, R.; He, K. M.; Dollar, P. TensorMask: A foundation for dense object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2061–2069, 2019.

[24] Zhang, H.; Sun, H.; Ao, W.; Dimirovski, G. A survey on instance segmentation: Recent advances and challenges. *International Journal of Innovative Computing, Information and Control* Vol. 17, No. 3, 1041–1053, 2021.

[25] Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 44, No. 7, 3523–3542, 2022.

[26] Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 43, No. 12, 4338–4364, 2021.

[27] He, Y.; Yu, H. S.; Liu, X. Y.; Yang, Z. G.; Sun, W.; Wang, Y. N.; Fu, Q.; Zou, Y. M.; Mian, A. Deep learning based 3D segmentation: A survey. *arXiv preprint* arXiv:2103.05423, 2021.

[28] Jiang, H. Y.; Yan, F. L.; Cai, J. F.; Zheng, J. M.; Xiao, J. End-to-end 3D point cloud instance segmentation without detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12793–12802, 2020.

[29] Hou, J.; Dai, A.; Nießner, M. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4416–4425, 2019.

[30] Yi, L.; Zhao, W.; Wang, H.; Sung, M.; Guibas, L. J. GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3942–3951, 2019.

[31] Wang, W. Y.; Yu, R.; Huang, Q. G.; Neumann, U. SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2569–2578, 2018.

[32] Liu, C.; Furukawa, Y. MASC: Multi-scale affinity with sparse convolution for 3D instance segmentation. *arXiv preprint* arXiv:1902.04478, 2019.

[33] Han, L.; Zheng, T.; Xu, L.; Fang, L. OccuSeg: Occupancy-aware 3D instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2937–2946, 2020.

[34] Chen, S. Y.; Fang, J. M.; Zhang, Q.; Liu, W. Y.; Wang, X. G. Hierarchical aggregation for 3D instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 15447–15456, 2021.

[35] Liang, Z. H.; Li, Z. H.; Xu, S. C.; Tan, M. K.; Jia, K. Instance segmentation in 3D scenes using semantic superpoint tree networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2763–2772, 2021.

[36] Yu, F. G.; Liu, K.; Zhang, Y.; Zhu, C. Y.; Xu, K. PartNet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9483–9492, 2019.

[37] Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 24, No. 5, 603–619, 2002.

[38] Wang, P.-S.; Liu, Y.; Guo, Y.-X.; Sun, C.-Y.; Tong, X. O-CNN: Octree-based convolutional neural networks

for 3D shape analysis. *ACM Transactions on Graphics* Vol. 36, No. 4, Article No. 72, 2017.

[39] Wang, P. S.; Liu, Y.; Tong, X. Deep octree-based CNNs with output-guided skip connections for 3D shape and scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 1074–1081, 2020.

[40] Graham, B.; van der Maaten, L. Submanifold sparse convolutional networks. *arXiv preprint* arXiv:1706.01307, 2017.

[41] Choy, C.; Gwak, J.; Savarese, S. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3070–3079, 2019.

[42] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M., et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint* arXiv:1603.04467, 2016.

[43] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* Vol. 12, 2825–2830, 2011.

[44] Qi, C. R.; Yi, L.; Su, H.; Guibas, L. J. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint* arXiv:1706.02413, 2017.

[45] Sharma, G.; Liu, D. F.; Maji, S.; Kalogerakis, E.; Chaudhuri, S.; Měch, R. ParSeNet: A parametric surface fitting network for 3D point clouds. In: C*omputer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12352*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer Cham, 261–276, 2020.

Chun-Yu Sun received his bachelor degree in computer science and technology from Xidian University in 2015. He is currently a Ph.D. student at Institute for Advanced Study, Tsinghua University. His research interests include computer graphics and 3D vision.

Xin Tong is a principal researcher manager with Microsoft Research Asia, where he leads the Internet Graphics Group. He received his Ph.D. degree from Tsinghua University in 1999. His research interests include computer graphics and computer vision, including texture synthesis, appearance modeling, light transport simulation and acquisition, 3D facial animation, and data-driven geometric processing. He was on the editorial boards of *IEEE Transactions on Visualization and Computer graphics*, *ACM Transactions on Graphics*, and *Computer Graphics Forum*.

Yang Liu is a principal researcher at Microsoft Research Asia. He received his Ph.D. degree from The University of Hong Kong in 2008, master and bachelor degrees in computational mathematics from University of Science and Technology of China in 2003 and 2000, respectively. His recent research focuses on geometry processing and 3D learning. He is on the editorial boards of *IEEE Transactions on Visualization and Computer graphics* and *ACM Transactions on Graphics*.