Segment-Fusion: Hierarchical Context Fusion for Robust 3D Semantic Segmentation

Anirud Thyagharajan[1], Benjamin Ummenhofer[2], Prashant Laddha[1], Om Ji Omer[1], Sreenivas Subramoney[1]
{[1]Processor Architecture Research Lab India, [2]Autonomous Agents Lab Germany}, Intel Labs

Abstract

3D semantic segmentation is a fundamental building block for several scene understanding applications such as autonomous driving, robotics and AR/VR. Several state-of-the-art semantic segmentation models suffer from the part-misclassification problem, wherein parts of the same object are labelled incorrectly. Previous methods have utilized hierarchical, iterative methods to fuse semantic and instance information, but they lack learnability in context fusion, and are computationally complex and heuristic driven. This paper presents Segment-Fusion, a novel attention-based method for hierarchical fusion of semantic and instance information to address the part misclassifications. The presented method includes a graph segmentation algorithm for grouping points into segments that pools point-wise features into segment-wise features, a learnable attention-based network to fuse these segments based on their semantic and instance features, and followed by a simple yet effective connected component labelling algorithm to convert segment features to instance labels. Segment-Fusion can be flexibly employed with any network architecture for semantic/instance segmentation. It improves the qualitative and quantitative performance of several semantic segmentation backbones by upto 5% *on the ScanNet and S3DIS datasets.*

1. Introduction

The growing resolution and availability of 3D visual sensors in recent years (e.g. Kinect, RealSense, Xtion) have enabled high fidelity representation of real world scenes using 3D data, allowing machines to understand 3D scenes with higher accuracy. One of the most important tasks in scene understanding includes 3D semantic segmentation, which aims to recognize the object class that each point in the scene belongs to. 3D semantic segmentation is fundamental to various applications, such as autonomous driving, robotic navigation and AR/VR [11, 28, 31, 38].

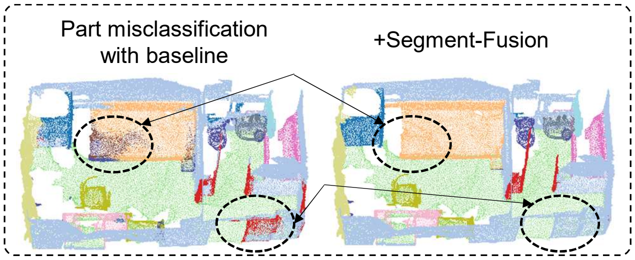

Figure 1. Semantic labels with baseline [37] and our Segment-Fusion approach on a sample point cloud from ScanNet Validation set [5]. The circled regions highlight how well Segment-Fusion addresses the part misclassification problem prevailing in many state-of-the-art semantic segmentation methods.

Recent advancements in 3D scene understanding have been greatly influenced by deep neural networks achieving state-of-the-art results [3, 9, 10, 15–17, 20, 23, 34, 37] on multiple benchmarks and wide range of datasets [1, 5]. However, many of these methods are often plagued by the part-misclassification problem, where parts of the same object are labelled incorrectly as shown in Figure 1. This problem can be better addressed if object instance boundaries can be correctly estimated, which includes annotating points with object instance identifiers. This helps to group all points associated with an object instance and utilize consensus to rectify semantic mispredictions.

Jointly solving semantic and instance segmentation has been explored by several previous works [2, 7, 12, 13, 18, 22, 24, 35, 36]. Many of these methods [13, 18, 24, 35, 36] focus on employing models for instance and semantic segmentation, before fusing pointwise features. However, they require higher compute resources as they fuse at a fine-grained level and do not exploit local continuities in scene structure. Methods such as [2, 7, 12, 22] include hierarchical pooling of features over a spatial region addressing the part misclassification problem by fusing these regions, with an assumption that these regions do not stretch over object boundaries. Such hierarchical methods are more efficient since they work with regions and not directly with points. While such methods are more efficient, they employ itera-

[*1,2] Contact us at: {anirud.thyagharajan, benjamin.ummenhofer, prashant.laddha, om.j.omer, sreenivas.subramoney}@intel.com

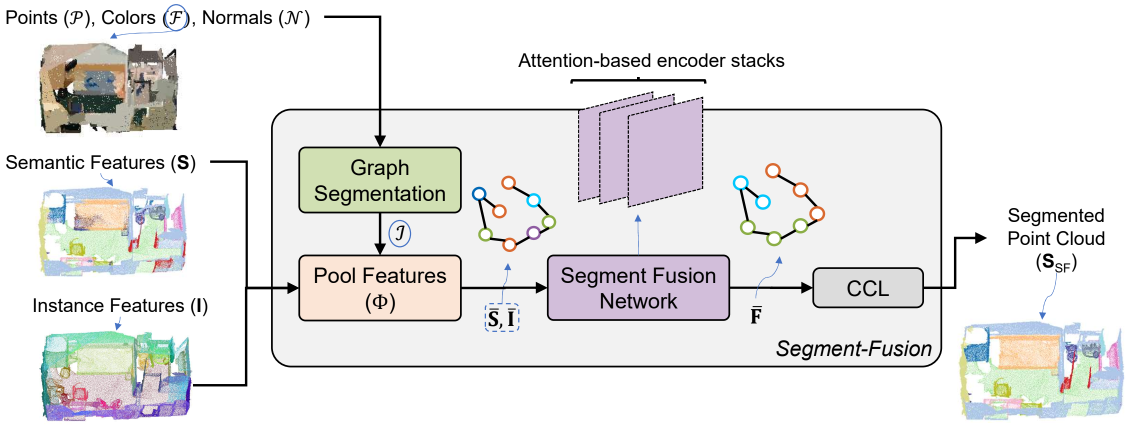
Points ($\mathcal{P}$), Colors ($\mathcal{F}$), Normals ($\mathcal{N}$)
Semantic Features (**S**)
Instance Features (**I**)
Attention-based encoder stacks
Graph Segmentation
$\mathcal{J}$
Pool Features (Φ)
$\bar{\mathbf{S}}, \bar{\mathbf{I}}$
Segment Fusion Network
$\bar{\mathbf{F}}$
CCL
Segment-Fusion
Segmented Point Cloud ($\mathbf{S}_{SF}$)

Figure 2. Overview of the Segment-Fusion approach. The Graph Segmentation module (GS) groups points into segments and enables pooling pointwise semantic and instance features $\mathbf{S}, \mathbf{I}$ into segment-wise features $\bar{\mathbf{S}}, \bar{\mathbf{I}}$. The Segment Fusion Network maps these features to a common feature space $\bar{\mathbf{F}}$ which is then used in the connected component labelling algorithm (CCL) to create the final set of segments.

tive clustering and aggregation techniques that are known to be heuristic-driven, non-learnable and specific to the backbone models used for extracting semantic and instance features. They also use complex post-processing [6, 27] to transform instance features to labels. Thus, there is a need for hierarchical, learnable methods for fusing such regions with simple post-processing.

Our prime objective is to improve semantic segmentation performance of a generic semantic segmentation model by using a hierarchical fusion of semantic and instance information. We use a graph segmentation algorithm to group points into regions and pool region-wise semantic and instance features but downstream, we propose a learnable clustering algorithm. It is desirable that the method should be learnable and agnostic to dataset specific heuristics.

- We propose a graph segmentation algorithm optimized for semantic segmentation across datasets, which is used to compose semantic and instance features at a coarser level (which we term as *segments*).
- We propose a learnable attention based network, Segment Fusion, to hierarchically fuse segments based on the similarity of their semantic and instance features, thus understanding the appropriate granularity of context (local to global: points to instance to scene).
- Our approach offers the advantages of adapting to inputs from other datasets and other backbones, as well as offering the possibility to perform end-to-end training and maintain efficiency (while working on a hierarchically smaller representation, and not working on individual points). These proposals help ameliorate the problem of part-misclassification by improving the performance (mIoU) of multiple semantic backbones on datasets like ScanNet [5] and S3DIS [1] upto 5%. We also compare the impact of our proposed method with the iterative clustering method proposed by Occuseg [12] and report 1-2% mIoU improvement in semantic segmentation.

2. Related Work

3D Semantic Segmentation. Semantic segmentation techniques can be broadly classified into 2D projection based and 3D methods. Methods that use 3D processing are known to provide superior accuracy because they can naturally overcome occlusion or scale ambiguity [12]. Several 3D methods have been proposed for semantic segmentation, which can be further categorized into point-based [25, 26] and volumetric methods [3, 10, 15–17, 34]. Methods from the PointNet family [25, 26] work on unstructured point clouds directly, employing pointwise networks to extract features and using ball queries and hierarchical grouping to encode spatial locality.

On the other hand, volumetric methods voxelize point clouds into regular grids and process them with regular structured computations. SparseConvNet [10] proposed submanifold sparse convolutions to process only the active voxels in a scene, while MinkowskiNet [3] generalized this concept with the help of hybrid kernels to express different types of local structure. Apart from these classes of methods, hybrid methods that utilize deformable/parameteric convolutions [30] on points have been proposed, in addition to employing graph convolutions [14, 33, 39] on points. All the above models apply voxel/point-wise cross entropy loss to optimize the model weights. Eventhough they employ network topologies such as U-Nets that are intended to comprehend scale and local-to-global structure, they exhibit part-misclassification issues as observed in Figure 3. Therefore, the model needs additional information about which instance of the object each point is a part of.

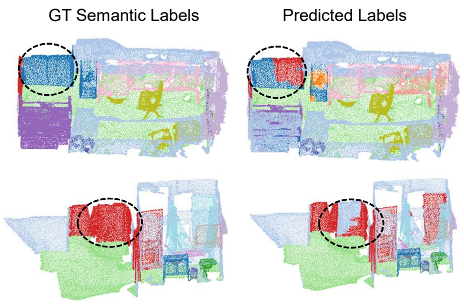

Figure 3. Illustrations of the part misclassification problem while using SCN [10] (1st row) and MinkowskiNet [3] (2nd row) as backbones for semantic predictions on the ScanNet dataset [5].

Joint Semantic-Instance Segmentation. Instance segmentation techniques aim to predict features for every point that indicate the object with which it is associated. Several methods have been proposed to perform semantic and instance segmentation jointly, since there is shared context between two similar tasks. Methods such as ASIS [36], SGPN [35] and JSIS3D [24] exercise models to fuse semantic and instance features pointwise, and subsequently compute the instance labels through clustering algorithms; ASIS [36] uses mean-shift clustering, SGPN [35] uses NMS to filter proposals and JSIS3D [24] uses a CRF to cluster instance embeddings into labels. However, these methods are heuristic driven (eg. mean-shift clustering [6] depends upon the window size) and are computationally complex (since they work with the pointwise features).

Works such as Occuseg, 3D-MPA, HAIS and SPG [2, 7, 21] simplify this complexity by grouping features at a higher abstractive level of surfaces (than points), thus proposing hierarchically fusing these features to form instances. Occuseg [12] computes instance and semantic features per supervoxel, where supervoxels are composed using a graph segmentation scheme [8], which are fed to an iterative clustering algorithm to merge supervoxels into instances. 3D-MPA [7] samples keypoints through a deep voting scheme, uses a graph convolutional network to refine the keypoint features, which are fed to DBSCAN [27] to obtain instance labels. HAIS [2] proposed a hierarchical method to generate point sets (based on the semantic features) and employ an iterative set aggregation algorithm to aggregate these point sets into instances. SST-Net [22] propose a tree classifier to hierarchically build a subtree of superpoint proposals. SPG [21] propose a graph convolutional network to learn relationships between abstracted surfaces.

While these methods may be effective for certain datasets, they propose iterative algorithms, which are not learnable. This makes it difficult for the algorithm to generalize for multiple sets of input features and/or datasets. Eventhough 3D-MPA uses a graph convolutional layer to refine the higher-level features, (a) they work with a fixed number of proposals in a scene (due to the deep voting scheme) which does not scale with scenes, and (b) they eventually use a heuristic driven classifier (DBSCAN [27]) after the refinement. SPG [21] utilizes only abstracted information at the level of surfaces and does not incorporate instance-level information.

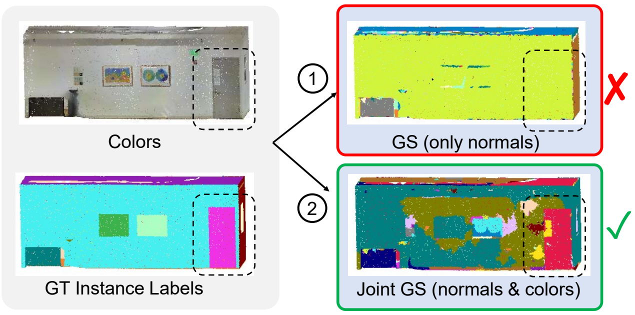

Figure 4. Impact of proposed J-GS method on an example S3DIS point cloud. As observed in ① [8], graph segmentation purely based on difference in normals fails to recognise the boundary between the door and the wall, while augmenting it with color information helps in discerning this (J-GS ②).

Graph Segmentation for hierarchical processing. As discussed above, graph segmentation algorithms [8] have been used previously to group points into larger regions (based on normal-similarity) and work at the abstraction level of these regions. It is observed that representations in higher-level abstractions such as these segments offer geometric continuity, since they ensures that the variation in point normals in a segment is bounded. Graph segmentation solely based on normals is effective but in some datasets, normals across object boundaries are not distinctive enough, as seen in Figure 4. This problem cannot be solved by lowering threshold for discriminating normals, because it leads to a highly over-segmented graph and defeats the purpose of hierarchical processing. Previous work [29] aims to solve this problem by performing segmentation using dissimilarity in normals and colors, but does not generalize for other kinds of features, or propose methods for selecting the thresholds for normals/colors.

Ideally, a semantic-instance fusion algorithm should have the following desirable properties:

- **Hierarchical.** To be efficient, the system should abstract features from the points level to a higher-level spatial representation (*segments*: eg. surfaces, groups of points) and process at this level.
- **Learnable.** The fusion of features from segments to instances should incorporate a learnable model to adapt to a variety of scenarios across datasets and underlying backbones for semantic and instance features.
- **Non-heuristic labelling.** The task of converting instance features to labels should be done simplistically

to avoid imparting any heuristic driven processing and absorb learning to the learnable component.

3. Methods

We present an overview of our method in Figure 2 that incorporates the desirable properties discussed in Section 2. The proposed method assumes a (i) semantic model that emits per-point semantic features $\mathbf{S}$ and (ii) an instance model that emits per-point instance features $\mathbf{I}$. Our method applies equivocally to those networks as well where these models could be sharing a common backbone network. For instance models like Occuseg [12] that emit multiple types of embeddings (eg. instance embeddings, centroid and occupancy estimates), $\mathbf{I}$ can be thought of as a concatenation of such features pointwise. In general, if $\mathbf{X}$ denotes any point-wise features, the corresponding segment-wise features are denoted by $\overline{\mathbf{X}}$.

3.1. Joint Graph Segmentation (J-GS)

We propose a hierarchical strategy, segmenting points into 3D segments (surfaces or super-voxels) using efficient graph segmentation algorithms such as [8, 19, 29]. The input graph is defined by vertices (points $\mathcal{P}$) joined by edges (neighbors E), where the goal of the segmentation is to find a mapping $\mathcal{J}$ that assigns the same segment ID to a set of grouped points. It also records the connections ($\mathcal{E}$) between the segments, represented by an adjacency matrix $\mathbf{A}$.

To cover a gamut of datasets, we generalize our graph segmentation algorithm to work across two options:

- **Point clouds without mesh information.** Datasets like ScanNet [5] have mesh information available, which allows us to compute per vertex normals $\mathcal{N}$ from the adjacent polygonal faces. However, for datasets like S3DIS [1] which do not have such connectivity information, we propose obtaining normals using plane-fitting techniques, as obtaining mesh information would incur additional expensive preprocessing (Figure 5).

- **Indiscriminative Normals.** We propose to generalize this method across datasets over previous work [29] by (a) allowing to use arbitrary pointwise features $\mathcal{F}$ (not specifically colors) in addition to normals, and then performing a joint segmentation over two spaces ($\mathcal{F}$ and $\mathcal{N}$), as shown in Algorithm 1. An example using colors as $\mathcal{F}$ for the S3DIS dataset is shown in Figure 4. (b) we generalize sorting orders ($s : \{norm, feats\}$, choosing to sort edges along similarity in $\mathcal{N}$ or $\mathcal{F}$), and (c) we employ a voting-based methodology (see Algorithm 2) to decide the optimal similarity threshold for a generic dataset for the semantic segmentation task.

In Algorithm 1, `computeEdgeWeights` computes dissimilarity scores along $\mathcal{N}$ (cosine product) and $\mathcal{F}$ (ℓ_1/ℓ_2 norms). `updateThresh` is similar to previous work [29] where the threshold slightly increases to accommodate a growing segment, where it approaches the largest intra-segment weight.

Though the proposed method (shown in Figure 4) may lead to over-segmentation of input scene, it ensures that points belonging to different objects are allotted to different segments. It avoids segment boundaries to cross-over object instance boundaries.

Algorithm 1 Proposed Graph Segmentation

Inputs: Normals $\mathcal{N}$, Features $\mathcal{F}$, Polygonal edges E, feature and normal similarity thresholds (f_{th}, n_{th}), sort-flag (s)

```
function SEGMENT_GRAPH($\mathcal{N}, \mathcal{F}, E, n_{th}, f_{th}, s$)
    $E_{wn}, E_{wf} \leftarrow$ computeEdgeWeights($\mathcal{N}, \mathcal{F}$)
    $E \leftarrow$ sortEdgesByNormalsOrFeats($s$)
    $th_f, th_n \leftarrow f_{th}, n_{th}$                ▷ initialize threshold vectors
    $\mathcal{J} \leftarrow$ initUnionFind($E$)
    for $e_{ij} \in E$ do                          ▷ edge joining ith and jth points
        $S_i \leftarrow \mathcal{J}$.find(i)
        $S_j \leftarrow \mathcal{J}$.find(j)
        $e_{wn}, e_{wf} \leftarrow E_{wn}[i,j], E_{wf}[i,j]$          ▷ edge weights
        if ($e_{wn} < th_n[i], th_n[j]$) and ($e_{wf} < th_f[i], th_f[j]$) then
            $newID \leftarrow \mathcal{J}$.join($i, j$)
            updateThresh($e_{wn}, e_{wf}, th_n, th_f, newID$)
        end if
    end for
    return $\mathcal{J}$
end function
```

Algorithm 2 Graph Segmentation based Voting algorithm

Inputs: List of J-GS Params $g_j = (n_{th}, f_{th}, s)$, predicted semantic labels $\mathbf{S}$, groundtruth labels $\mathbf{G}$

```
function J-GSV($GP$, $\mathbf{S}$)
    init scores
    for $g_j$ in $GP$ do
        $\mathcal{J}_j \leftarrow$ SEGMENT_GRAPH($g_j$)
        $\mathbf{S}_{j,maj} \leftarrow$ majorityVoting($\mathcal{J}_j$, $\mathbf{S}$)
        scores[j] $\leftarrow$ getIoU($\mathbf{S}_{j,maj}$, $G$)
    end for
    return argmin(scores)
end function
```

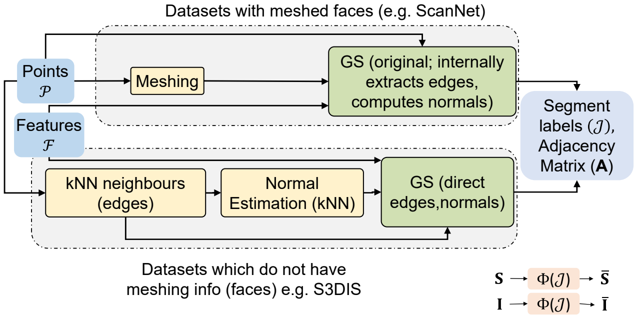
Datasets with meshed faces (e.g. ScanNet)
Points $\mathcal{P}$
Meshing
GS (original; internally extracts edges, computes normals)
Features $\mathcal{F}$
Segment labels ($\mathcal{J}$), Adjacency Matrix ($\mathbf{A}$)
kNN neighbours (edges)
Normal Estimation (kNN)
GS (direct edges,normals)
Datasets which do not have meshing info (faces) e.g. S3DIS
$\mathbf{S} \rightarrow \Phi(\mathcal{J}) \rightarrow \bar{\mathbf{S}}$
$\mathbf{I} \rightarrow \Phi(\mathcal{J}) \rightarrow \bar{\mathbf{I}}$

Figure 5. Proposed flow for generalizing graph segmentation across a variety of point cloud datasets.

To select the right variant for given dataset, we enable quantitative comparison across the above options by proposing the J-GSV score (Joint Graph Segmentation-based Voting) (Algorithm 2). Across point clouds from the training set, for the j-th variant, we obtain a list of sets of points $\mathcal{J}_j$, with the i-th set $\mathcal{J}_{j,i}$ containing points pertaining to the same segment. We compute the semantic predictions for the j-th variant by pooling the per-point semantic predictions $\mathbf{S}$ for each segment, and broadcasting the majority label to all points in the segment. The mean Intersection-over-Union (mIoU) for the modified semantic predictions $\mathbf{S}_{j,maj}$ is the score associated with the j-th variant. For the dataset, we select the variant with the highest score and denote by $\mathcal{J}^*$. Consequently, we employ a differentiable function ϕ, to transform pointwise features to segment-wise features ($\overline{\mathbf{X}} = \phi(\mathbf{X}, \mathcal{J}^*)$). To simplify, our implementation of $\phi(.)$ is the mean function, average-pooling across pointwise semantic and instance features $(\mathbf{S}, \mathbf{I})$ to obtain segment-wise features $(\bar{\mathbf{S}}, \bar{\mathbf{I}})$, similar to [12]. For pointwise features $\mathbf{X}$ we get the feature for a segment as

$$\overline{\mathbf{X}_i} = \frac{1}{|\mathcal{J}_i|} \sum_{j \in \mathcal{J}_i} \mathbf{X}_j. \tag{1}$$

Thus, the proposed joint graph segmentation hierarchically composes features of a scene at a coarse level to enable efficient fusion decisions of these segments.

3.2. Segment-Fusion Network

Having formed segments out of the underlying points, the next goal is to understand how to form objects out of these segments. To this effect, we train a network to jointly associate the instance-level and semantic-level information to essentially form decisions if a pair of segments belong to the same object (*fusable*) or not (*separable*).

We project the instance and semantic features per segment $\{\bar{\mathbf{S}}, \bar{\mathbf{I}}\}$ to a transformed set of features $\bar{\mathbf{F}}$ in the joint semantic-instance space using the Segment-Fusion network. We compose the Segment-Fusion network using a stack of attention encoder blocks (taking cues from a Transformer [32] block). Attention blocks are known to capture larger contextual information among different segments by extracting useful intermediate representations using scaled-dot-product attention. Additionally, in each block, we propose multiplying the attention-matrix element-wise with the adjacency matrix $\mathbf{A}$ describing the connections between the segments. This helps to constrain the interactions only between the segment pairs that are spatially connected. A more detailed description of the network architecture is provided in the supplementary material.

To supervise this learning process, we propose two sets of losses - (i) Instance loss and (ii) Segment Loss:

$$\mathcal{L}_{SF} = \mathcal{L}_{\text{instance}} + \mathcal{L}_{\text{segment}} \tag{2}$$

3.2.1 SF-Instance Loss

We propose attraction and repulsion instance losses at the segment level. These losses ensure that segments of the same instance are clustered together (Equation (4)), whereas the centroids of the instance features are repelled from each other (Equation (5)). While this appears similar to the instance losses employed at the per-point level [12, 36] in principle, but in our case, the losses are applied at the segment-level features. Additionally, we parameterise the thresholds for repulsion and attraction to be Δ_D and Δ_V respectively, essentially to unify the thresholds used in the SF-Segment loss and CCL algorithm.

$$\mathcal{L}_{\text{instance}} = \mathcal{L}_{\text{attract}} + \mathcal{L}_{\text{repel}} + \mathcal{L}_{\text{reg}} \tag{3}$$

$$\mathcal{L}_{\text{attract}} = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{N_i} \sum_{j=1}^{N_i} H(d(\bar{\mathbf{F}}_j, \mu_i) - \Delta_V)^2 \tag{4}$$

$$\mathcal{L}_{\text{repel}} = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j=1; i \neq j}^{K} H(\Delta_D - d(\mu_i, \mu_j))^2 \tag{5}$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{K} \sum_{i=1}^{K} ||\mu_i||_1 \tag{6}$$

where K denotes the number of ground truth instances in the scene; $||.||$ is the ℓ_1 norm; μ_i is the average of the segment features across the segments belonging to the ith instance; $d(f_i, f_j)$ indicates feature distance using a suitable norm (ℓ_1/ℓ_2), and $H(..)$ is the hinge loss; we set the values of the thresholds Δ_V and Δ_D to be 0.10 and 1 respectively.

3.2.2 SF-Segment Loss

The SF-instance losses aid in clustering segment features appropriately, but relying solely on them requires iterative post-processing clustering algorithms such as DBSCAN [27], mean-shift clustering [6] etc. In this work, we propose a much simpler the clustering algorithm for projecting features to labels (Section 3.3), by using penalties on pairwise distances in the segment feature metric space. To this end, we propose a loss function that focuses on fusable and separable edges independently (Equations (7), (8) and (9)). We reuse the same threshold Δ_V here to place constraints along the edges of the graph.

$$\mathcal{L}_{\text{segment}} = w_{\text{fuse}} \mathcal{L}_{\text{fuse}} + w_{\text{sep}} \mathcal{L}_{\text{sep}} \tag{7}$$

$$\mathcal{L}_{\text{sep}} = \frac{1}{|\mathcal{E}_{\text{sep}}|} \sum_{e_{ij} \in \mathcal{E}_{\text{sep}}} H(\Delta_V - d(\bar{\mathbf{F}}_i, \bar{\mathbf{F}}_j)) \tag{8}$$

Table 1. Performance impact (mIoU) of Segment-Fusion on state-of-the-art semantic segmentation backbones on the ScanNet test set.

Model	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	fridge	shower	toilet	sink	bathtub	furniture	**mean**
SparseConvNet	82.9	95	64.4	76.9	79.5	64.4	55.4	57.7	54.1	78.6	27.6	39.5	54.1	65.6	66.6	79.4	92.7	71	78.1	48	66.6
SparseConvNet + SF	85.4	97.3	67	79.9	86	75.7	60	60.5	61.4	79.6	28.1	44.8	61.9	79.7	70.5	88.7	95.7	79	85.9	51.3	**71.9**
PointConv	72.6	94.2	51.8	64	77.4	70	51.2	37.1	54	61	20.6	40	46.1	57.3	55.9	58.2	87.8	56.1	70	38.9	58.2
PointConv + SF	75.9	96.8	57	66.5	79.9	73.1	54.1	40.5	55.9	63.5	21.4	45.3	48.7	60.5	59.9	62	92.8	62.7	81.7	41.4	**62.0**
MinkNet42	83.9	95.4	71.1	82.4	84.5	76.1	66	59	66.8	80	19.7	50.5	59.4	82.1	19.7	89.1	88.7	72.8	91.2	54.5	72.4
MinkNet42 + SF	85.6	97.5	72.5	83.2	86.3	77.4	67.4	63	67.7	81.2	22	54.2	59.3	82.9	22	93.4	92.6	77.6	97.4	56.1	**74.7**
VMNet	85.9	95.4	74.5	82.1	84.2	81.9	61.8	62.3	71.6	83.9	24	49.6	61.4	88.4	74.7	90.3	93.1	78.1	81.8	58	74.2
VMNet + SF	88.3	97.5	77.2	83.3	85.7	83.8	62.5	67.1	72.6	84.9	28	55.1	61.6	89.9	77.1	93.6	96.9	85.2	85.4	59.2	**76.7**

$$\mathcal{L}_{\text{fuse}} = \frac{1}{|\mathcal{E}_{\text{fuse}}|} \sum_{e_{ij} \in \mathcal{E}_{\text{fuse}}} H(d(\bar{\mathbf{F}}_i, \bar{\mathbf{F}}_j) - \Delta_V) \qquad (9)$$

where $\mathcal{E}_{\text{sep}}$ and $\mathcal{E}_{\text{fuse}}$ denote the set of edges which should be kept separate and fused respectively, decided using ground truth instance information.

Since the number of objects in a scene are finite, the number of edges between dissimilar objects are much lower than the edges within the same object. This influences the relative weighing of w_{fuse} and w_{sep}, which we set to be 1 and 0.01 respectively, to counter this imbalance.

3.3. Connected Component Labelling (CCL)

To convert the segment features $\bar{\mathbf{F}}$ to fusion decisions, we employ a connected component labelling algorithm over the feature space that $\bar{\mathbf{F}}$ resides in. Pairwise-distances are computed $d(\bar{\mathbf{F}}_i, \bar{\mathbf{F}}_j) \; \forall \; e_{ij} \in \mathcal{E}$ and thresholded using Δ_V, depicted through a piecewise function as in Equation 10. Thus, each positive entry of $\mathbf{B}_{ij}$ indicates a fusing decision between segments i and j.

$$\mathbf{B}_{ij} = \begin{cases} 1 & d(\bar{\mathbf{F}}_i, \bar{\mathbf{F}}_j) < \Delta_V \; \forall \; e_{ij} \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

For every pair of indices for which $\mathbf{B}_{ij} = 1$, we use an efficient disjoint-set forest to implement Union-Find to record the connectivity information [4]. For every connected component composed of a set of segments, we compute the mean semantic probability vector of the component and identify the most probable semantic label for the entire component, thus forming the final set of predictions $\mathbf{S}_{SF}$. This procedure enables us to (i) identify and assign a single label to an entire object, and (ii) allow the Segment-Fusion block to override erroneous point-wise semantic predictions for parts of the object by using consensus information across larger parts of the object.

4. Evaluation

4.1. Training Setup

We apply and evaluate our method with multiple state-of-the-art backbones of semantic segmentation across ScanNet [1] and S3DIS datasets [1] using the mean Intersection-over-Union (mIoU) metric. To assess qualitative improvements, we also compare results with and without Segment-Fusion (and show its effect in regions with part misclassification).

4.2. ScanNet Dataset

We pick four state-of-the-art semantic segmentation backbones – SparseConvNet [10], PointConv [37], MinkNet [3] and VMNet [16]. We augment each of these semantic backbones with an instance backbone trained with the losses proposed in Occuseg [12]. On the ScanNet test set, as shown in Table 1, we observe significant improvements of 5.3%, 3.8%, 2.3% and 2.5% respectively in mIoU scores. We notice a consistent upswing in IoU scores across all classes. Qualitative improvements are shown in Figure 6. Thus, we observe that the improvements provided by Segment-Fusion do not depend specifically on the choice of the semantic segmentation backbone used. The improvements in mIoU help move the MinkNet42 + SF and VMNet + SF variants to the 5^{th} and 2^{nd} positions on the leaderboard respectively.

4.3. S3DIS Dataset

To demonstrate the effectiveness of our approach on datasets that do not have mesh information, we evaluate Segment-Fusion on the Area 5 set from S3DIS. We choose three semantic backbone networks: MinkNet18 [3], KPConv [30] and ASIS [36] to show improvements with Segment-Fusion. Table 2 presents semantic segmentation results on the S3DIS Area 5 set on a variety of semantic backbones. We augment each of these semantic backbones with instance features from ASIS [36]. The results show that we obtain 1.5%, 0.6% and 0.8% improvement in mIoU on using Segment-Fusion over MinkNet18 [3], KPConv [30] and ASIS [36] respectively.

4.4. Comparisons with Occuseg clustering

We also assess Segment-Fusion with heuristic driven graph clustering approach as employed in OccuSeg [12].

[1] License: CC BY-NC

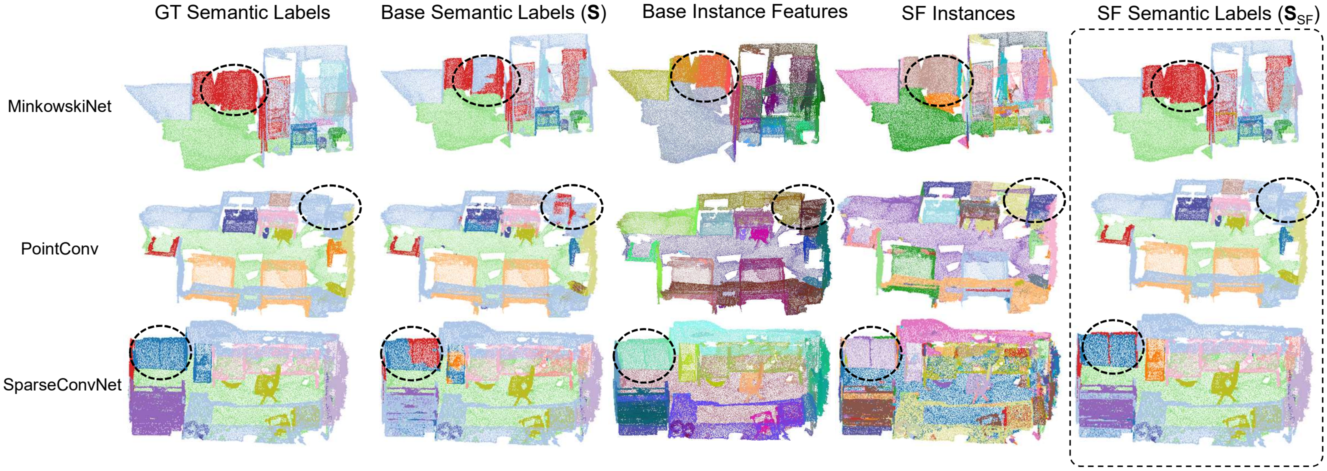

Figure 6. Qualitative results of Segment-Fusion on some sample point clouds of the ScanNet validation set using the MinkowskiNet-42 [3], PointConv [37] and SparseConvNet [10] semantic backbones. The highlighted regions indicate mispredictions in the base semantic model (**S**). Note how Segment-Fusion builds consensus using the instance features to correct the semantic predictions in these areas (without degrading the performance in the other regions).

Table 2. Impact of Segment-Fusion on state-of-the-art semantic segmentation backbones on the S3DIS Area 5

Model	clutter	beam	board	bookcase	ceiling	chair	column	door	floor	sofa	table	wall	window	mean
MinkNet18	51.4	0.0	66.5	67.4	92.3	87.1	34.9	68.4	95.9	59.8	76.0	81.0	49.2	63.8
MinkNet18+SF	52.4	0.0	69.3	67.2	93.0	88.4	37.6	67.7	96.7	67.7	76.5	81.6	51.2	**65.3**
KPConv	54.3	0.0	60.9	72.5	92.3	89.7	20.2	73.2	96.8	70.1	77.9	80.1	52.0	64.6
KPConv+SF	55.2	0.0	61.8	73.3	92.4	89.5	20.7	75.6	96.9	70.0	77.9	80.7	53.4	**65.2**
ASIS	41.5	0.0	52.5	0.0	89.6	1.4	7.8	17.2	96.5	0.0	1.4	73.1	39.9	32.4
ASIS+SF	43.2	0.0	61.6	0.0	89.7	0.0	2.9	16.9	96.6	0.0	0.0	74.5	43.4	**33.2**

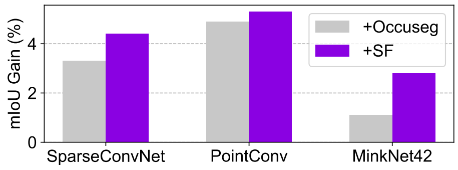

Figure 7. Performance comparison between Occuseg's clustering and Segment-Fusion's learnable fusion algorithm on ScanNet/val.

For this we use semantic and instance backbones similar to OccuSeg, and compare the improvement in semantic segmentation performance obtained through the iterative clustering algorithm they propose, with that of Segment-Fusion. Figure 7 shows that Segment-Fusion outperforms the heuristic driven iterative clustering algorithm across semantic backbones; lending the conclusion that a learnable algorithm like Segment-Fusion may be more scalable and generic to apply on generic semantic and instance backbones as compared to a hand-crafted algorithm.

4.5. Ablation Studies

4.5.1 Impact of learning beyond J-GS

To better understand the effectiveness of our approach, we study the individual contributions of the components of our system. Table 3 compares the mIoU of the 3 semantic segmentation backbones employing (i) only graph segmentation based voting (J-GSV) without any fusion network, (ii) J-GS + SF, but trained only with instance losses ($\mathcal{L}_{\text{instance}}$), and (iii) J-GS + SF trained with both segment and instance losses ($\mathcal{L}_{\text{segment}} + \mathcal{L}_{\text{instance}}$). In J-GSV, we obtain semantic labels by performing majority voting within the segment; while in SF, semantic labels are computed as described in Section 3.3. Note that employing only $\mathcal{L}_{\text{instance}}$ degrades performance, essentially because we use a simplistic CCL algorithm to compute labels from the features; using only instance loss results in over-fusion of segments, leading to poorer performance. We observe a significant performance improvement on adding segment losses to constrain individual edges. Thus, noting steady improvements in mIoU on account of J-GS and SF, we conclude that (i) a hierarchical approach consisting of abstracting pointwise features to segment-wise features and (ii) learnable methods to fuse these segments are both required and effective.

Table 3. Impact of individual components of Segment-Fusion on overall mIoU (ScanNet validation set) ($\mathcal{L}_{\text{ins}} = \mathcal{L}_{\text{instance}}$, $\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{segment}}$)

Model	**Base**	**Base + J-GSV**	**Base + SF($\mathcal{L}_{\text{ins}}$)**	**Base + SF($\mathcal{L}_{\text{ins}}$+$\mathcal{L}_{\text{seg}}$)**
SparseConvNet	67.1	70.4	69.7	71.5
PointConv	58.3	61.5	61.4	63.6
MinkNet42	72.4	74.6	73.11	75.2

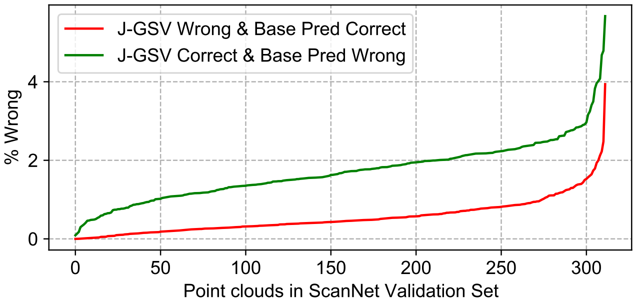

Figure 8. Top: Fraction of points in a point cloud where J-GSV is incorrect and the base semantic predictor is correct. Bottom: Fraction points in a point cloud where J-GSV is correct and the base semantic predictor is incorrect.

4.5.2 Does J-GS result in over-fusion?

While graph segmentation aids in abstracting the problem to a higher level spatially, it runs the risk of over-fusing points belonging to two different classes into one segment. We evaluate the impact of such occurrences on overall performance using the proposed J-GSV score. For each point cloud, we analyse the base semantic predictions $\mathbf{S}$ with the predictions after J-GSV $\mathbf{S}_{maj}$(without Segment-Fusion Network) and count the number of points for which they disagree with each other. We define overfusion to occur when the J-GSV predictions are incorrect while the base predictor is correct. In Figure 8 we observe that J-GS based voting helps in an improvement of atleast 2% in $\sim 32\%$ of the point clouds in the validation set of the ScanNet dataset, while being 2% incorrect in $\sim 1\%$ of the point clouds. This indicates that J-GSV is consistently beneficial across point clouds of the dataset and provides higher improvement than the degradation due to overfusion.

4.5.3 J-GS on S3DIS

We present an ablative study to describe the methodology of selecting the thresholds for the proposed J-GS method. Figure 9 illustrates the performance of J-GSV on the training and validation sets of the S3DIS dataset. As discussed in Section 3.1, we evaluate the performance of various J-GSV variants on the training set to select the J-GS parameters and apply them to the validation set. It is observed that on an average, sorting of edge weights along feature similarity performs better than sorting along similarity along normals. Also, relying solely on normals ends up reducing performance as compared to the base validation performance (without J-GSV). We observe that $(s : feats, f_{th} : 0.5)$ provides optimal performance for the training set, and use this to apply J-GS on the validation set and generate segment-wise data for Segment-Fusion Network. To evaluate how well these parameters generalize from the training to the validation set, we evaluate the same parameter set on the validation set as well. Though $(s : feats, f_{th} : 2)$ performs better on the validation set, the difference with $f_{th} : 0.5$ is low (lower than the gains we see alongwith SF, Table 2). On the validation set, the sorting parameter s has a higher sensitivity than the feature-threshold parameter. The ScanNet dataset has more distinctive normals, hence we did not observe much sensitivity with the f_{th} threshold.

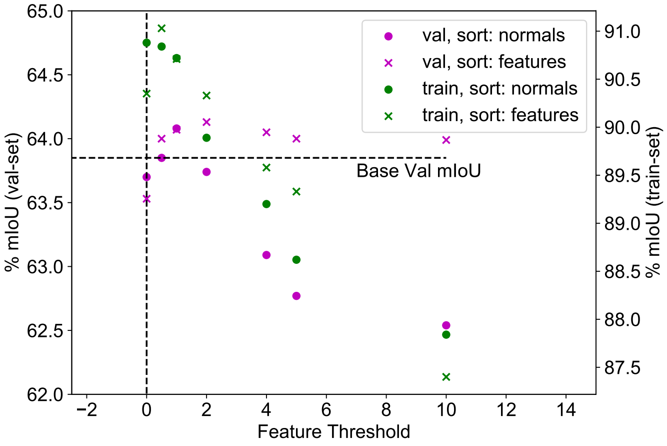

Figure 9. J-GS Voting Score results for the S3DIS validation set across a variety of feature thresholds. The threshold used in the normals space n_{th} is 0.01; results along this axes are omitted here due to lack of sensitivity.

5. Limitations and Future Work

While we have demonstrated the effectiveness of our approach on multiple semantic and instance backbones, the method still requires supervision. There is also potential in exploring deeper and more complex network architectures in the Segment-Fusion network to increase improvements. In the future, we plan to extend this work by using similar strategies to fuse learnings from problems with shared context, along with focusing on semi/self-supervised learning.

6. Conclusion

We presented Segment-Fusion, a learnable, hierarchical method to fuse segments, aimed at improving semantic segmentation performance of generic base models. We proposed a voting based algorithm to select optimal graph segmentation hyperparameters for varying datasets. Our learnable method can be flexibly used in conjunction with existing semantic/instance models. Finally, we performed comprehensive evaluations to demonstrate the effectiveness of our methods on multiple types of semantic backbones.

7. Acknowledgements

We would like to thank Chris Choy for his help in setting up the MinkowskiNet baseline for semantic segmentation.

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 1, 2, 4

[2] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15467–15476, 2021. 1, 3

[3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 2, 3, 6, 7

[4] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009. 6

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2, 3, 4

[6] Konstantinos G Derpanis. Mean shift clustering. *Lecture Notes*, page 32, 2005. 2, 3, 5

[7] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9031–9040, 2020. 1, 3

[8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 3, 4

[9] Jingyu Gong, Jiachen Xu, Xin Tan, Haichuan Song, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Omni-supervised point cloud segmentation via gradual receptive field component reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11673–11682, 2021. 1

[10] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 1, 2, 3, 6, 7

[11] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *International Journal of Computer Vision*, 112(2):133–149, 2015. 1

[12] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2940–2949, 2020. 1, 2, 3, 4, 5, 6

[13] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019. 1

[14] Hanzhe Hu, Deyi Ji, Weihao Gan, Shuai Bai, Wei Wu, and Junjie Yan. Class-wise dynamic graph convolution for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 1–17. Springer, 2020. 2

[15] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14373–14382, 2021. 1, 2

[16] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew-Lan Tai. Vmnet: Voxel-mesh network for geodesic-aware 3d semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15488–15498, 2021. 1, 2, 6

[17] Zeyu Hu, Mingmin Zhen, Xuyang Bai, Hongbo Fu, and Chiew-lan Tai. Jsenet: Joint semantic segmentation and edge detection network for 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 222–239. Springer, 2020. 1, 2

[18] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020. 1

[19] Klaas Klasing, Dirk Wollherr, and Martin Buss. A clustering method for efficient segmentation of 3d laser data. In *2008 IEEE international conference on robotics and automation*, pages 4043–4048. IEEE, 2008. 4

[20] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *European Conference on Computer Vision*, pages 518–535. Springer, 2020. 1

[21] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 3

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

[23] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. *arXiv preprint arXiv:2110.02210*, 2021. 1

[24] Quang-Hieu Pham, Thanh Nguyen, Binh-Son Hua, Gemma Roig, and Sai-Kit Yeung. Jsis3d: Joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8827–8836, 2019. 1, 3

[25] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification

and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[26] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[27] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017. 2, 3, 5

[28] Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, Martin Jagersand, and Hong Zhang. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 587–597, 2018. 1

[29] Johannes Strom, Andrew Richardson, and Edwin Olson. Graph-based segmentation for colored 3d laser point clouds. In *2010 IEEE/RSJ international conference on intelligent robots and systems*, pages 2131–2136. IEEE, 2010. 3, 4

[30] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 2, 6

[31] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4644–4651. IEEE, 2017. 1

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 5

[33] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019. 2

[34] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017. 1, 2

[35] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2569–2578, 2018. 1, 3

[36] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019. 1, 3, 5, 6

[37] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019. 1, 6, 7

[38] Chao Yu, Zuxin Liu, Xin-Jun Liu, Fugui Xie, Yi Yang, Qi Wei, and Qiao Fei. Ds-slam: A semantic visual slam towards dynamic environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1168–1174. IEEE, 2018. 1

[39] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. *arXiv preprint arXiv:1909.06121*, 2019. 2