Semi-Supervised Vehicle Part-Segmentation in LiDAR Point Clouds for Autonomous Driving

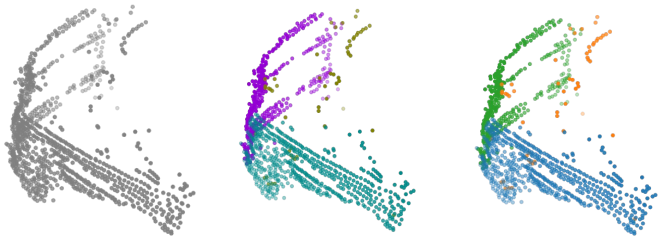
Mingyu Liu*, Ekim Yurtsever*, Huu Tung Nguyen, Magdy Mahmoud, Yutong Hou, Xingcheng Zhou, Yuning Cui, Alois C. Knoll

***Abstract*—A reliable perception system is a critical component of intelligent vehicles. Perception tasks commonly focus on holistic information, like 3D object detection, which treats vehicles as bounding boxes. However, more granular surface-level features are ignored, which can provide additional information for improving perception capability and be utilized for other driving tasks. To this end, we propose a novel semi-supervised vehicle part-segmentation method for autonomous driving to learn surface features from the input point clouds. Specifically, we utilize points within the ground truth bounding boxes to train a feature extractor to learn surface features. Then, after surface normal estimation, these learned surface features concatenated with normals are grouped with an unsupervised clustering method. In the end, we manually annotate a subset of the widely used KITTI dataset to identify each cluster with corresponding vehicle parts. We explore using PointNext and RepSurf as the surface feature extractor, combining with ten different unsupervised clustering approaches. We verify the optimal pipeline stack for surface feature learning and part segmentation through numerous experiments on the labeled data. https://github.com/MingyuLiu1/Surface_Part_Segmentation.git**

I. Introduction

The advance in autonomous driving in recent years has brought the critical need for advanced perception methodologies to the forefront. Traditional perception tasks in autonomous driving, such as LiDAR-based 3D object detection [1]–[6], focus on holistic recognition and localization of objects. However, more granular tasks, such as vehicle part segmentation, are often understudied. We believe a more granular surface-level approach is critical for obtaining additional useful information for dynamic driving tasks. For example, distinguishing the front hood surface from the side-door surfaces of a vehicle can be utilized for better decision-making during emergency situations. This can lead to more informed planning decisions in complicated driving scenarios.

State-of-the-art 3D object detection methods [3] primarily focus on recognizing and localizing objects based on bounding boxes. On the other hand, most point-level segmentation works [7], [8] consider holistic categories, such as car or pedestrian, on a scene level. In both approaches, intra-object features are overlooked under the coarse granularity of these existing tasks.



(a) Input point cloud (b) Clustering (c) Matching

Fig. 1: Illustration for the key processes of the proposed semi-supervised part segmentation approach, including the input point cloud (left), feature clustering (middle), and the feature matching or part segmentation result (right). The results illustrate that our introduced algorithm obtains rich surface features and accurately segments the car parts based on the learned features.

We believe that intra-object, surface-level information can be utilized to improve the robustness and effectiveness of the perception system of intelligent vehicles.

Recognizing the surface features of other objects can lead to better decision-making in emergencies. For example, during an unavoidable collision situation, knowing where to hit can reduce the severity of an accident. Several collision reconfiguration studies focus on reducing collision severity by deciding how to hit another object [9], [10]. These studies indicate that prioritizing collision reconfiguration in an unavoidable crash scenario is crucial for reducing the severity of accidents [9]. However, most of these studies are simulation-based, and they assume vehicle part segmentation can be achieved in real-time.

Existing part-segmentation methods focus on high-density, indoor datasets [11], [12]. However, precise segmentation of vehicle-parts in large, outdoor point cloud data is under-studied and particularly challenging due to the inherent complexities of the problem and the high cost of data-labeling efforts. To alleviate these issues, we propose a novel semi-supervised learning approach for vehicle part-segmentation in outdoor LiDAR point clouds for autonomous driving. We exhibit an example of the part segmentation results in Fig. 1.

The proposed method consists of three main components. First, a backbone is trained for extracting surface features through object classification. This step does not require additional annotations. An ordinary object detection dataset with ground truth bounding box annotation, such as the

* Authors contributed equally.

M. Liu, HT Nguyen, M. Mahmoud, Y. Hou, X. Zhou, Y. Cui, and AC. Knoll are with the Chair of Robotics, Artificial Intelligence and Real-Time Systems, Technical University of Munich, 85748 Garching bei München, Germany (E-mail: {mingyu.liu, hu.nguyen, magdy.mahmoud, yutong.hou, xingchneg.zhou}@tum.de, {yuning.cui, knoll}@in.tum.de)

E. Yurtsever is with the College of Engineering, Center for Automotive Research, The Ohio State University, Columbus, OH 43212, USA (E-mail: yurtsever.2@osu.edu)

KITTI [13] dataset, is adequate to train the first component. Then, these learned surface features are grouped together with unsupervised clustering methods such as Gaussian Mixture Model [14]. Finally, a small subset of the dataset is used to identify each cluster with corresponding vehicle parts. This final step requires a manual labeling effort to associate unsupervised cluster IDs with real vehicle parts.

The rest of the paper details our methods, results, and the broader implications for LiDAR point cloud data processing in autonomous driving.

Our main contributions are as follows:

- **A novel semi-supervised vehicle-part segmentation method for autonomous driving.** We propose a novel two-step approach for semi-supervised vehicle-part segmentation in outdoor point cloud data, emphasizing surface information extracted from the input point clouds.
- **Extending the KITTI dataset with vehicle-part segmentation labels and open-source implementation.** We manually annotate a subset of the KITTI dataset to evaluate the part segmentation performance of our proposed method.
- **Extensive experiments with multiple surface-representation and clustering modules.** We conducted numerous experiments on the labeled data, which included 20 types of combinations of two surface feature extractors with ten unsupervised clustering methods. Through comparisons between each combination, we justified the selection of the final algorithmic pipeline.

II. Related Works

A. Point Cloud Segmentation

Point cloud segmentation enhances understanding of 3D scenes and represents a significant research direction in autonomous driving. PointNet [2] and PointNet++ [15] pioneer multi-layer perceptron networks for processing unstructured point clouds. Subsequently, more methods have been proposed, such as [4]–[6], exploring using convolution neural networks to learn point features. DGCNN [16] proposes using the EdgeConv module to obtain edge features from the KNN-based local graphs. PointNext [17] based on [15] enlarges the receptive fields while keeping rich semantic features. One of the most recent works, RepSurf [18], exploits triangle-based and multi-surface representation for segmentation. These methods achieve promising performance on part segmentation of indoor objects or semantic segmentation in outdoor scenarios. However, the application of part segmentation on objects in the autonomous driving domain has been under-explored, while it is vital for improving the performance of perception systems.

B. Semi-Supervised Learning

Semi-supervised learning allows combining labeled and unlabeled data to perform certain tasks [19], which has significant advantages in eliminating the data-hungry problem and keeping task performance. [20] proposed a semi-supervised learning approach utilizing GANs, which leads to the generation of more effective classifiers and higher-quality samples. FixMatch [21] improved the performance by combining consistency regularization and pseudo-labeling. [22] explores compacting latent space clustering for semi-supervised learning. Considering the benefits of semi-supervised learning, we exploit using it to facilitate part segmentation tasks in autonomous driving.

C. Unsupervised Clustering

Clustering aims to partition similar data into the same group and push dissimilar data far from each other [23], which is widely used in various computer vision tasks, such as anomaly detection, segmentation, and many more. Usually, as no predefined categories provide prior knowledge, it is also called unsupervised clustering. K-Means [24], as one of the most popular algorithms, partitions the input data into K distinct, non-overlapping clusters. K is the predefined number of clusters. Compared to K-Means, the Gaussian Mixture Model [14] follows a probabilistic approach, assuming the data is generated from several Gaussian distributions. In our study, we verify ten types of clustering algorithms to find the most optimal combination for part segmentation and learning surface features.

III. Methodology

In this section, we introduce the overall pipeline of the proposed semi-supervised vehicle part-segmentation pipeline, including surface-feature extractor, surface normal estimation, feature clustering, and cluster matching. Furthermore, in the rest of this section, we explain each core component in detail.

A. Semi-supervised vehicle part-segmentation pipeline

We illustrate our proposed approach in Fig. 2. We extract points within the ground truth bounding boxes as input to train the surface-feature extractor through object classification. After that, the extracted features are fed into the unsupervised part-segmentation module.

To segment objects into several parts, we first calculate the surface normal based on the extracted surface features. Then, we concatenate the surface features with the estimated surface normal for the following feature clustering by unsupervised learning methods. Ultimately, we align the grouped features to the labeled data through cluster matching to evaluate the quality of the part-segmentation pipeline.

B. Surface Feature Extraction

We leverage a learning-based network as the extractor to capture surface features from point clouds. Usually, the network is based on an encoder-decoder architecture for various perception tasks. In our case, we aim to obtain valuable features through classification; hence, we apply a Multiple Perceptron (MLP) as the decoder and keep the encoder as the default.

The smooth cross-entropy loss is used for the training of classification:

$$l(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) \tag{1}$$

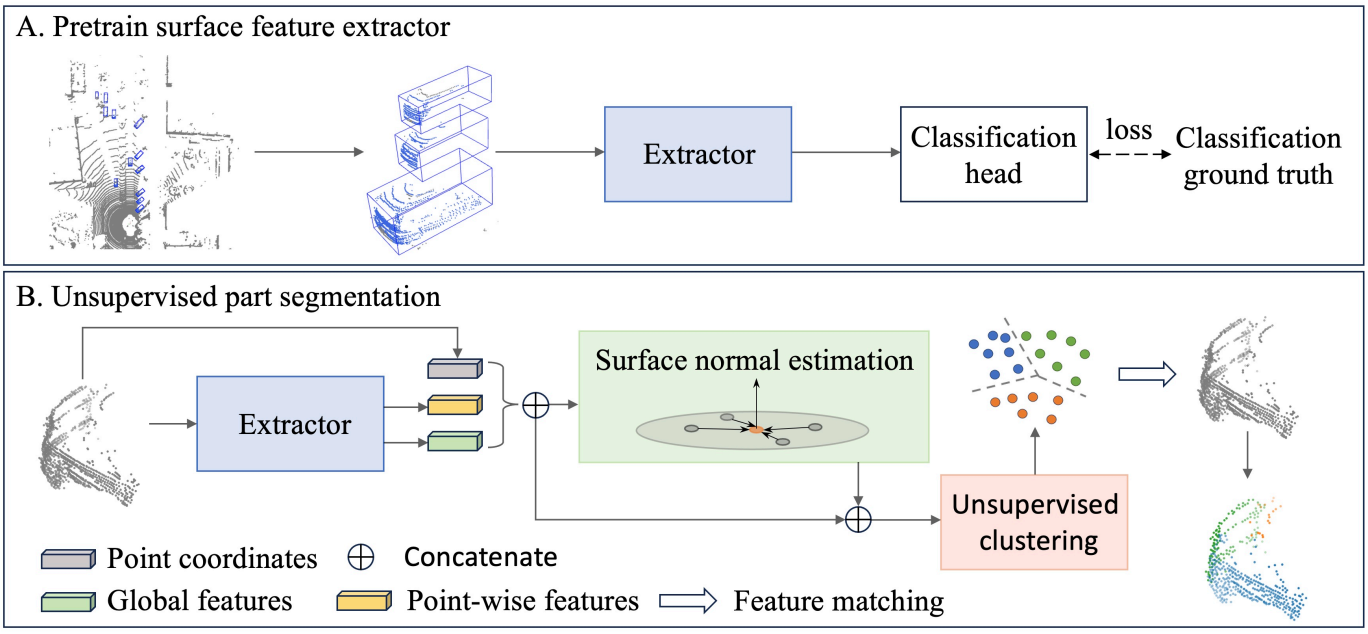


Fig. 2: Overview of the proposed semi-supervised part segmentation pipeline. As part A shows, we first utilize the points within the ground truth bounding boxes as input to train the surface feature extractor through the object classification task. Afterward, we leverage the pretrained extractor to obtain features from the input points, which combine with the point coordinates later. Additionally, we concatenate the estimated surface normals with the features from the previous step to obtain the feature clusters through the unsupervised module. Ultimately, by feature matching, we generate the final part segmentation results.

where p is the predicted possibility and y is the indicator. Since we mainly focus on part segmentation on cars, we set the label of cars to 1 and others to 0, then $y = \{0, 1\}$.

After the classification training, the extractor is able to identify whether an object is a car, which means the extractor effectively learns the distinguished features of cars from others. Therefore, the captured features by the network implicitly encode rich surface information. To fully utilize the surface features and keep the dimension alignment, we combine the features from the first layer of the extractor with the corresponding point coordinate. The combination can be described as $\mathbf{f}_i^{'} = (\mathbf{f}_i, x_i, y_i, z_i)$, where $\mathbf{f}_i \in \mathbb{R}^{1\times d}$ is the point-wise feature of point $i \in \{1, \ldots, n\}$, and (x_i, y_i, z_i) are its coordinate. Additionally, we combine the global features $\mathbf{g} \in \mathbb{R}^{1\times d^{'}}$ from the last layer of the extractor with $\mathbf{f}^{'}$. The features encompass point-wise and global information through concatenating, facilitating an effective part segmentation.

C. Surface Normal Estimation

The essence of surface normal estimation lies in determining the orientation of the surfaces at every data point, effectively giving us geometric and topological insights into the object's shape and structure. In our methodology, the depth and richness of features the extractor obtains are vital in normal estimation, as the network's understanding of broader object geometry assists in refining local surface orientations. Specifically, we explicitly estimate the surface normals using the concatenated features from the first step. After finding the k nearest neighbors of each point via the k-nearest neighbor (KNN), we compute the covariance matrix $\mathbf{M}$ of these neighbors (Eq. 2) and apply Singular Value Decomposition (SVD) to extract the normals from the minor variance direction.

$$\mathbf{M} = \frac{1}{k}\sum_{j=1}^{k}(\mathbf{c}_{ij} - \mathbf{c})(\mathbf{c}_{ij} - \mathbf{c})^T \tag{2}$$

where $\mathbf{c}_{ij} = (x_{ij}, y_{ij}, z_{ij})$ is coordinate of the jth neighbor of point p_i, and $\mathbf{c} = \frac{1}{k+1}(\mathbf{c}_i + \sum_{j=1}^{k}\mathbf{c}_j)$ is the centroid of point set $\{p_i, p_{i1}, \ldots, p_{ik}\}$.

Hence, the SVM can be demonstrated as:

$$\mathbf{M} = \mathbf{U}\Sigma\mathbf{V}^* \tag{3}$$

where $\mathbf{U}$ is the unitary matrix, $\mathbf{\Sigma}$ is rectangular diagonal matrix, and $\mathbf{V}^*$ is the conjugate transpose of the unitary matrix $\mathbf{V} = \{(a_i, b_i, c_i)|i = 1, \ldots, n\}$.

Finally, we follow [18] to describe the surface as Eq. 4. The normals are then normalized and oriented to point outward from the surface.

$$\begin{aligned} a_i(x - x_i) + b_i(y - yi) + c_i(z - z_i) = 0 \\ \Rightarrow a_i x + b_i y + c_i z - (a_i x + b_i y + c_i z) \end{aligned} \tag{4}$$

D. Feature Clustering

Clustering enables the separation of different surface features, resulting in a well-organized and segmented representation of an object. We concatenate the surface features from the extractor and the estimated surface normals. Therefore, the features seamlessly fuse geometric and learned latent

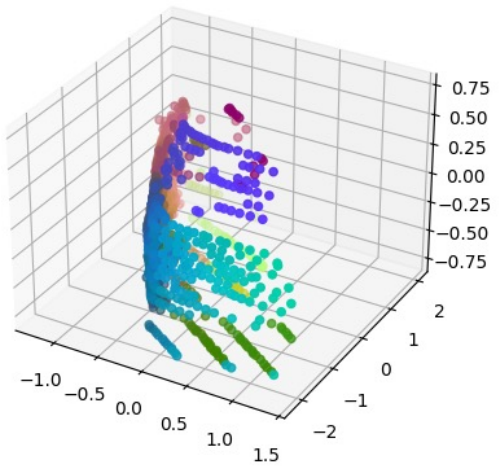


Fig. 3: We illustrate the combined features of a car via UMAP [25]. Points with the same color indicate similar features, such as the front in blue and the back in brown.

information through the combination, resulting in a comprehensive representation of the point clouds. We leverage the dimension reduction technology like UMAP [14] to visualize the combined features, illustrated in Fig. 3. Then, we employ unsupervised clustering algorithms to detect distinct, coherent groups. These groups represent the separated surfaces of objects based on the combined features with the estimated surface normals.

E. Cluster Matching

The core intent behind cluster matching is to map clusters derived from our algorithm to the ground truth clusters in the labeled dataset. This alignment process enables us to gauge the efficacy of our semi-supervised part segmentation technique. We construct a cost matrix by calculating the Intersection over Union (IoU) for each element of the predicted clusters with each class of the ground truth data. The algorithm of the cost matrix is illustrated in Alg. 1. After obtaining the cost matrix, we apply the Hungarian algorithm to find the optimal assignments.

Algorithm 1 Calculate Cost Matrix

Require: Unsupervised labels U, Ground truth labels G, Number of unsupervised clusters n, Number of ground truth labels m
Ensure: Cost matrix C

function IoU(A, B)
  return $\frac{|A \cap B|}{|A \cup B|}$
end function
$C \leftarrow$ matrix of zeros of size $n \times m$
for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $U_i \leftarrow$ set of points in U with label i
    $G_j \leftarrow$ set of points in G with label j
    if G_j is not empty **then**
      $iou \leftarrow$ IoU(U_i, G_j)
      $C_{i,j} \leftarrow 1 - iou$
    end if
  end for
end for
return C

IV. Experiments and Results

In this section, we introduce the datasets and models used in our experiments. Then, we explain the evaluation metrics and experimental setup and analyze both quantitative and qualitative results in depth.

A. Datasets

KITTI. As one of the pioneering datasets in the domain of autonomous driving and renowned for its diverse urban scenarios and rich annotations [26], KITTI 3D object detection dataset [13] primarily focuses on classes such as cars, pedestrians, and cyclists. The KITTI dataset comprises 7,481 annotated images and associated LiDAR point cloud scans. It provides a comprehensive view with approximately 200,000 labeled object instances in total. In our study, we utilize the classification labels and points within the ground truth bounding boxes for the training of feature extractors.

Manually Annotated Data. We manually labeled 64 cars from the KITTI dataset to evaluate the efficiency of each semi-supervised network on learning surface features and performing part segmentation. We only annotate car points within a bounding box, and noise and ground points are not marked during the experiments. Every point is assigned to a specific class, representing different parts of the vehicle: front, rear, left, right, or top. To provide a consistent number of points across the labels, we extract cars, including more than 1024 points, which are then down-sampled with farthest point sampling to obtain a uniform sample.

Evaluation Metrics In our experiments, we leverage several widely used metrics [27] to evaluate the part segmentation performance based on the combination of various feature extractors and clusters. We first utilize mean Intersection over Union (IoU) (see Eq. 5) to measure the overlap between the predicted segmentation and the ground truth.

$$mIoU = \frac{1}{C+1} \sum_{m=0}^{C} \frac{p_{mm}}{\sum_{n=0}^{C} p_{mn} + \sum_{n=0}^{C} p_{nm} - p_{mm}} \quad (5)$$

where C is the number of classes, and p_{mm}, p_{mn}, and p_{nm} represent true positives, false positives, and false negatives, respectively.

Additionally, we also take into account using mean Pixel Accuracy as another evaluation metric. In our case, we calculate the correctly classified points over the total number of points:

$$mPA = \frac{1}{C+1} \sum_{m=0}^{C} \frac{p_{mm}}{\sum_{n=0}^{C} p_{mn}} \quad (6)$$

We also apply F1-Score to do the evaluation, which can be demonstrated as:

$$F1 = \frac{2\sum_{n=0}^{C} p_{mm}}{\sum_{m=0}^{C}(2p_{mm} + p_{nm} + p_{mn})} \tag{7}$$

B. Experimental Setups.

We conduct all experiments using PyTorch 2.0 on one NVIDIA GeForce 4090 (24Gb). All surface-feature extractors are trained with 30 epochs with batch size 2. We use the AdamW optimizer with an initial learning rate of 2e-3 for the training.

Surface Feature Extractors We test two types of feature extractors, PointNext [17] and Repsurf [18], to obtain the surface features from the points within bounding boxes. Based on PointNet++ [15], PointNext follows the hierarchical feature-learning scheme by partitioning point clouds into overlapping local regions at multiple scales and varying resolutions. On the other hand, RepSurf focuses on explicitly depicting the local shape of point clouds, emphasizing surface representation. Both of them follow the encoder-decoder architecture. Because we train the extractor on classification, we replace the decoder with a Multilayer Perceptron (MLP).

Unsupervised Feature Clustering In our experiments, we evaluate ten unsupervised clustering algorithms, which can be categorized into four classes: 1) partitioning methods (K-Means and Gaussian Mixture Model), 2) hierarchical methods (Agglomerative Clustering and BIRCH), 3) density-based method (DBSCAN, OPTICS, and HDBSCAN), 4) graph-based methods (Spectral Clustering and Affinity Propagation), and 5) model-seeking method (Mean Shift). In the main experiments, we set the cluster size n to 3 for those algorithms requiring a fixed number of clusters since three sides of the car are present on average. Furthermore, for a fair comparison, we set top-3 filtering in our evaluation pipeline for the methods with automatically adaptive cluster sizes.

C. Experiment results

In this section, we show and discuss the experiment results of different combinations of surface-feature extractors and unsupervised algorithms on our manually annotated data. The quantitative results of surface-feature extractors PointNext [17] and Repsurf [18] are shown in Tab. I and Tab. II, respectively.

First, we evaluate PointNext on the labeled data regarding the mIoU, mPA, and F1-Score. Overall, almost all approaches with fixed cluster sizes perform around 43% under mIoU and over 60% for mPA, which outperforms the approaches with adaptive cluster sizes in terms of all metrics. For example, K-Means achieves 42.1% on mIoU, which is much higher than Mean Shift (29.3%). Additionally, Affinity Propagation, another adaptive cluster, performs the worst on the three metrics compared to other methods. The reason is the adaptive clusters don't have a target cluster number as prior knowledge for the clustering, causing under-segmentation, such as OPTICS, DBSCAN, HDBSCAN (as shown in Fig. 4).

On the other hand, it is noted that our proposed PointNext combined with the Gaussian Mixture Model stands out from other algorithms with huge improvements on the three metrics: mIoU 62.3%, accuracy 80.8%, and 43.6% on F1-Score. According to the qualitative result in the dotted box in Fig. 4, our approach accurately segments the car into the three targeted parts.

Extractor	Method	mIoU	mPA	F1-Score
PointNext	K-MEANS	42.1	61.0	33.0
	Agglonerative Clustering	41.4	61.6	32.5
	Spectral Clustering	43.7	66.7	32.9
	Affinity Propagation	8.8	30.4	12.6
	Mean Shift	29.3	62.2	22.4
	BIRCH	44.4	65.6	34.2
	OPTICS	24.7	60.5	18.8
	DBSCAN	19.3	58.0	14.6
	HDBSCAN	25.6	58.1	20.2
Ours (PointNext + GMM)		**62.3**	**80.8**	**43.6**

TABLE I: We show the experiment results of PointNext [17] combining with ten unsupervised clustering algorithms in terms of three evaluation metrics, mIoU (%) ↑, mPA(%) ↑, and F1-Score(%) ↑. GMM is the Gaussian Mixture Model.

Furthermore, we explore using RepSurf as the surface-feature extractor (Tab. II). Each unsupervised clustering method exhibits performance similar to that of the combination with PointNext when using RepSurf as the feature extractor. In general, RepSurf slightly increases the performance of fixed-size approaches. However, the Gaussian Mixture Model performs worse than PointNext as the extractor but is still the best. In conclusion, leveraging algorithms based on a fixed number of clusters can perform better than the adaptive clustering methods for effective surface feature learning. Moreover, a feature extractor focusing on detailed geometrical information can improve capturing valuable surface features.

Extractor	Method	mIoU	mPA	F1-Score
RepSurf	K-MEANS	42.7	60.6	33.6
	Agglonerative Clustering	43.2	62.4	34.1
	Spectral Clustering	44.1	64.4	33.4
	Affinity Propagation	10.9	28.5	14.3
	Mean Shift	33.4	64.0	25.5
	BIRCH	48.1	67.5	36.9
	OPTICS	21.0	57.0	16.2
	DBSCAN	18.6	55.8	14.2
	HDBSCAN	21.9	60.0	17.3
	Gaussian Mixture Model	**58.1**	**76.3**	**41.5**

TABLE II: We show the experiment results of RepSurf [18] combining with ten unsupervised clustering algorithms in terms of three evaluation metrics, mIoU (%) ↑, mPA(%) ↑, and F1-Score(%) ↑.

D. Ablation Study

In this section, we study the influence of the cluster size of our proposed approach (PointNext [17] with Gaussian Mixture Model (GMM)) on the segmentation performance. We set 3 different cluster sizes (3, 4, and 5) to qualitatively identify the optimal size for the surface segmentation task. A larger

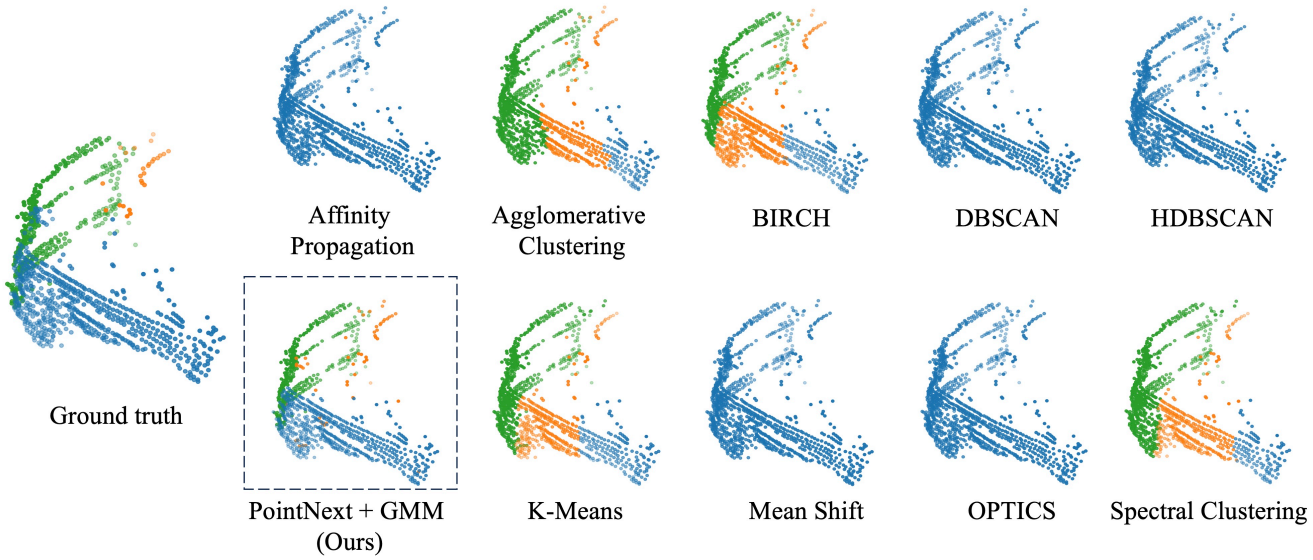


Fig. 4: Qualitative results of part segmentation between different unsupervised clustering algorithms with PointNext [17] as the surface feature extractor. Blue, green, and orange represent the car's right, rear, and top sides for the ground truth points, respectively. The dotted box shows the result of our proposed PointNext combined with the Gaussian Mixture Model (GMM) stack.

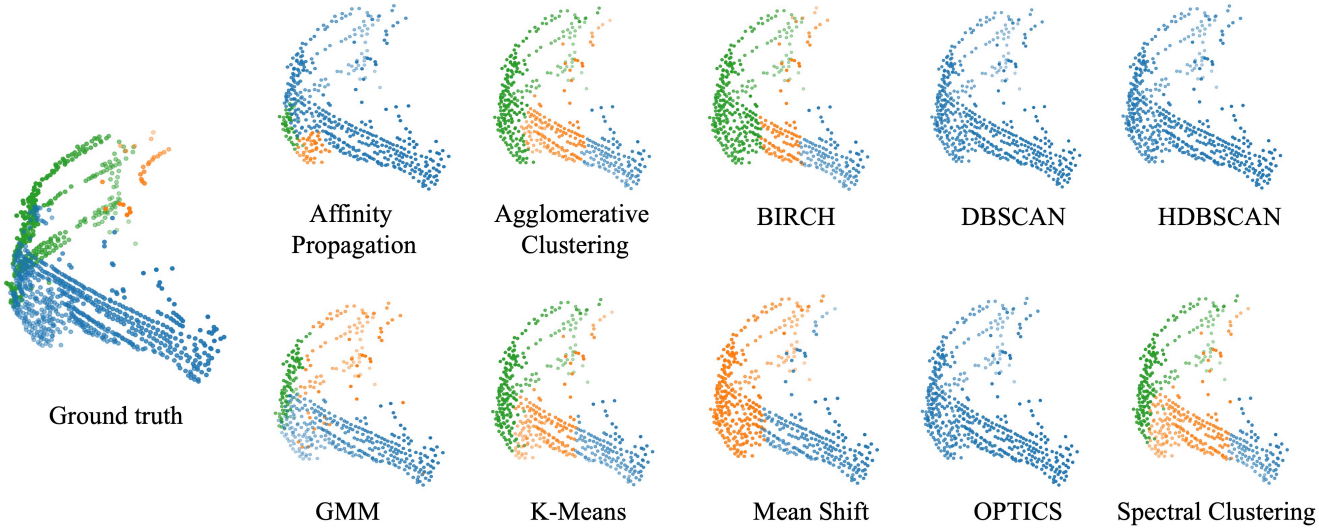
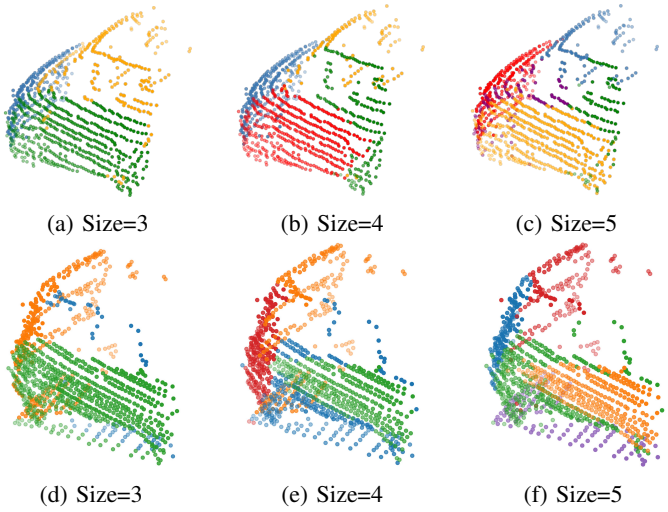


Fig. 5: Qualitative results of part segmentation between different unsupervised clustering algorithms with Repsurf [18] as the surface feature extractor. Blue, green, and orange represent the car's right, rear, and top sides for the ground truth points, respectively.

size means the cluster focuses more on details and prefers to segment an object into more parts. We consider two cases for evaluating the segmentation performance, one without ground and the other one including. We illustrate the experiment results in Fig. 6. When we set the cluster to 3, the algorithm precisely segments the car without ground into three parts: right, rear, and top. In contrast, with a cluster size of 4 or 5, the rear and right sides of the car are divided into several sub-parts while losing global information.

Moreover, if there are ground points or noises, as shown in the second row, although the network does not distinguish object points and the ground points well under cluster size 3, the segmentation performance of the car body is still desired. With the larger cluster sizes, the segmentation results are even worse compared to clean input. Hence, setting the cluster size to 3 suits our proposed semi-supervised part segmentation pipeline.



(a) Size=3 (b) Size=4 (c) Size=5

(d) Size=3 (e) Size=4 (f) Size=5

Fig. 6: Comparison between different cluster sizes of our introduced PointNext [17] combining with GMM approach. The first row ((a), (b), and (c)) shows the segmentation results based on clean input point clouds. The second row exhibits the results considering ground points.

V. Conclusion

In this study, we introduce a semi-supervised vehicle part-segmentation approach consisting of a surface-feature extractor and an unsupervised feature clustering module. The extractor learns surface features from the points within ground truth bounding boxes. Afterward, we estimate the surface normals based on the surface features combined with point coordinates, and we further concatenate the estimated normals with the extracted surface features. Then, the integrated features are grouped with unsupervised clustering algorithms. Moreover, we manually annotate a subset of the KITTI dataset to evaluate the performance of the semi-supervised part segmentation approach. In total, we compared 20 combinations to each other to verify the optimal algorithmic stack.

References

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

[2] C. R. Qi, H. Su, K. Mo, and L. 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*, pp. 652–660, 2017.

[3] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11784–11793, 2021.

[4] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "Pointcnn: Convolution on x-transformed points," *Advances in neural information processing systems*, vol. 31, 2018.

[5] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 87–102, 2018.

[6] M. Xu, R. Ding, H. Zhao, and X. Qi, "Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3173–3182, 2021.

[7] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11108–11117, 2020.

[8] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, "Segcloud: Semantic segmentation of 3d point clouds," in *2017 international conference on 3D vision (3DV)*, pp. 537–547, IEEE, 2017.

[9] M. D. Robinson, C. E. Beal, and S. N. Brennan, "At what cost? how planned collisions with pedestrians may save lives," *Accident Analysis & Prevention*, vol. 141, p. 105492, 2020.

[10] M. Parseh and F. Asplund, "New needs to consider during accident analysis: Implications of autonomous vehicles with collision reconfiguration systems," *Accident Analysis & Prevention*, vol. 173, p. 106704, 2022.

[11] X. Wang, X. Sun, X. Cao, K. Xu, and B. Zhou, "Learning fine-grained segmentation of 3d shapes without part labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10276–10285, 2021.

[12] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "Gspn: Generative shape proposal network for 3d instance segmentation in point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3947–3956, 2019.

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society: series B (methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[16] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[17] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23192–23204, 2022.

[18] H. Ran, J. Liu, and C. Wang, "Surface representation for point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18942–18952, 2022.

[19] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[20] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[21] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[22] K. Kamnitsas, D. Castro, L. Le Folgoc, I. Walker, R. Tanno, D. Rueckert, B. Glocker, A. Criminisi, and A. Nori, "Semi-supervised learning via compact latent space clustering," in *International conference on machine learning*, pp. 2459–2468, PMLR, 2018.

[23] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: a brief survey," *A review of machine learning techniques for processing multimedia content*, vol. 1, no. 2004, pp. 9–16, 2004.

[24] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA, 1967.

[25] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[26] M. Liu, E. Yurtsever, X. Zhou, J. Fossaert, Y. Cui, B. L. Zagar, and A. C. Knoll, "A survey on autonomous driving datasets: Data statistic, annotation, and outlook," *arXiv preprint arXiv:2401.01454*, 2024.

[27] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.