Visual Intelligence

RESEARCH Open Access

Panoptic segmentation of 3D point clouds with Gaussian mixture model in outdoor scenes

Yizhou Wang[1], Longguang Wang[2], Qingyong Hu[3], Yan Liu[1], Ye Zhang[1*] and Yulan Guo[4*]

Abstract

Accurate panoptic segmentation of 3D point clouds in outdoor scenes is critical for the success of applications such as autonomous driving and robot navigation. Existing methods in this area typically assume that the differences between instances are greater than the differences between points belonging to the same instance and use heuristic techniques for segmentation. However, this assumption may not hold in real scenes with occlusion and noise. In addition, most of the previous methods formulate point-wise embedding learning and instance clustering as two decoupled steps for separate optimization, making it a challenging task to learn discriminative embeddings. To address these issues, we introduce a framework for modeling points belonging to the same instance using learnable Gaussian distributions and formulate the point cloud as a Gaussian mixture model. Based on this formulation, we introduce a unified loss function that links the embedding learning and instance clustering in an end-to-end manner. Our framework is generic and can be seamlessly incorporated with existing panoptic segmentation networks. By explicitly modeling intra-instance variance and leveraging end-to-end optimization, our framework improves the discrimination capability of point embeddings with higher accuracy and robustness. Extensive experiments on two large-scale benchmarks demonstrate the effectiveness of the proposed method.

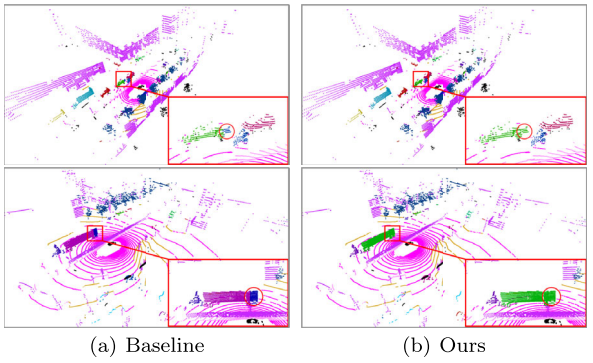
Keywords: Point cloud, Lidar, Panoptic segmentation, Outdoor, Clustering, Gaussian mixture model (GMM)

1 Introduction

As one of the key challenges in autonomous driving [1–3] and robot perception [4], panoptic segmentation of 3D point clouds aims to unify semantic and instance segmentation, further achieving fine-grained 3D scene perception. Specifically, each 3D point is expected to be classified into background (stuff) or foreground (things) classes with a specific instance ID. Due to the irregular and disordered nature of 3D point clouds, coupled with the effects of occlusion, noise and incomplete scanning, achieving effective panoptic segmentation remains a major challenge.

Recently, Behley et al. [5] have first explored panoptic segmentation on the SemanticKITTI dataset. This pioneering method enriches the dataset with instance-level annotations and leverages both semantic segmentation [6] and object detection [7] techniques for panoptic segmentation. Inspired by this work, several dedicated network architectures have been developed with improved accuracy. These approaches can be divided into proposal-based and proposal-free methods. In particular, proposal-based methods [3, 5, 8–10] explicitly predict bounding boxes or binary masks to split instances from backgrounds for segmentation. In contrast, proposal-free methods [11–15] learn discriminative point-wise embedding and adopt clustering techniques to group individual points. Considering the simplicity in terms of network architecture and

*Correspondence: zhangy2658@mail.sysu.edu.cn; yulan.guo@nudt.edu.cn
[1]School of Electronics and Communication Engineering, the Shenzhen Campus of Sun Yat-sen University, Sun Yat-sen University, Shenzhen, 510275, China
[4]College of Electronic Science and Technology, National University of Defense Technology, Changsha, Hunan 410073, China
Full list of author information is available at the end of the article



(a) Baseline (b) Ours

Figure 1 Panoptic results on the nuScenes dataset. We set random colors for the points of each predicted instance to distinguish them

low inference cost, proposal-free methods have drawn increasing interest.

Although substantial progress has been achieved in recent years, existing proposal-free methods still have two limitations. First, previous methods commonly use heuristic techniques (e.g., L2 distance) to distinguish different instances without considering intra-instance variance (the measurement of differences between points belonging to the same instance). In practice, intra-instance variance can be larger than inter-instance variance (the measurement of differences between instances), making these methods sensitive to outliers and prone to dividing an object into fragments (Fig. 1(a)). Second, existing methods formulate point-wise embedding learning and instance clustering as two decoupled steps for separate optimization. Consequently, discriminative embedding cannot be well learned, which hinders further performance improvement.

To address the above issues, we propose a method for panoptic segmentation of 3D point clouds in outdoor scenes using a Gaussian mixture model (GMM), termed GMM-PanopticSeg, by explicitly modeling the intra-instance variance of an object with a 2D Gaussian distribution. Furthermore, the point cloud can be simplified to a GMM [16] for panoptic segmentation. First, we formulate intra-instance variance in the embedding space as a Gaussian distribution. Specifically, we develop a distribution estimation module to predict the covariance of the Gaussian distribution to capture the intra-instance variance for diverse objects. Second, as the intra-instance variance is modeled as a Gaussian distribution, we further introduce a unified loss function to achieve joint optimization of embedding learning and instance clustering. Our framework is generic and can be seamlessly integrated with existing approaches to enable panoptic segmentation. Moreover, the proposed method can be integrated with existing panoptic segmentation networks to achieve consistent performance improvements. For example, with Panoptic-PolarNet [12] and DS-Net [13] serving as the backbone, our framework achieves an average improvement of 9.6%/5.9% in the PQ^{Th}/PQ score on the nuScenes dataset.

Overall, the contributions of this paper can be summarized as follows.

1) We propose modeling the intra-instance variance of an object in the embedding space as a 2D Gaussian distribution and employing a Gaussian mixture model to represent the point cloud. To the best of our knowledge, our framework is the first work that explicitly considers intra-instance variance during panoptic segmentation.

2) We introduce a unified loss function to integrate embedding learning and instance clustering for end-to-end joint optimization.

3) Our framework improves the discrimination capability of the embedding and can further boost the performance of previous state-of-the-art approaches on benchmark datasets.

2 Related work

In this section, we first review several point cloud instance segmentation methods. Then, we discuss recent advances in point cloud panoptic segmentation.

2.1 Instance segmentation of 3D point clouds

Existing point cloud instance segmentation techniques can be categorized into boundary-based and grouping-based methods.

Boundary-based methods. This category of methods commonly follows a two-stage pipeline. Specifically, bounding boxes are first predicted as the initial boundaries of instances and then further refined through bounding box regression or binary classification. For example, GSPN [17] uses an analysis-by-synthesis strategy to generate bounding boxes from shape proposals, and subsequently refines these boxes using R-PointNet. 3D-SIS [18] extracts geometry and color features from multi-views to generate bounding boxes and binary masks to segment instances. 3D-BoNet [19] leverages global features to directly regress bounding boxes and then matches proposals to instances using the Hungarian algorithm [20]. GICN [21] follows a bottom-up paradigm to first select center points and then predict corresponding bounding boxes. Although these methods produce promising results, a two-stage pipeline with costly post-processing technique (e.g., non-maximum suppression) introduces considerable overhead.

Grouping-based methods. Unlike boundary-based methods, grouping-based methods directly learn discriminative point-wise embeddings and adopt clustering techniques for instance segmentation. Specifically, SGPN [22] makes the points belonging to the same instance close to each other in the feature embedding space, and leverages a similarity matrix for grouping. Recently, several

works [23–26] have used the averaged embedding of the points that belong to an instance as the optimization goal of the embedding learning process. Using this method, point-wise embeddings tend to be similar to their corresponding averaged embeddings, but dissimilar to others. MASC [27] iteratively merges neighbor nodes into instance groups and clusters points using learnable multi-scale affinity. DyCo3D [28] and Mask3D [29] predict binary masks to assign instance IDs. Moreover, the other methods [30–33] leverage positions of objects as additional information and adopt instance centers as the optimization goal of embedding learning. However, due to the occlusion and noise, predicted centers from incomplete partial point clouds usually deviate from the instance centroids, thereby resulting in limited performance.

2.2 Panoptic segmentation of 3D point clouds

Point cloud panoptic segmentation aims to provide unified semantic segmentation and unique instance segmentation results. Due to the advantages in handling instance ID conflicts, grouping-based methods have attracted increasing interest from researchers for the task of panoptic segmentation. Considering the irregular nature of point clouds, several methods transform them into other representations for panoptic segmentation. Specifically, LPSAD [11] encodes point clouds into a range-view representation to extract point-wise embeddings and extracts and utilizes a learnable radius for clustering. DS-Net [13] uses Cylinder3D [34] as the backbone and clusters different instances via dynamic shifting to address the issues of inconsistent accuracy of predicted centers from different instances. Panoptic-PolarNet [12] first projects point clouds onto the bird's-eye view (BEV) plane and then predicts a 2D heatmap to conduct clustering. Panoptic-PHNet [14] modifies the BEV encoder and employs voxel features to improve segmentation performance. In addition, a k-nearest neighbor (KNN) transformer is used to predict a pseudo heatmap to avoid inconsistency between the heatmap and the offset branches. Since transformation inevitably introduces information loss, recent works have directly implemented panoptic segmentation on point clouds. PVCL [35] uses contrastive learning to learn stable and discriminative features. GP-S3Net [36] first excessively segments foreground points and then proposes graph convolutional neural networks (GCNNs) to merge fragments from the same instance. PolarStream [37] uses polar coordinate system and leverages wedge-shaped point cloud sectors to improve inference efficiency. Recently, mask-based methods [38, 39] have achieved outstanding performance on leaderboard. They use instance prototypes from learnable parameters matched with point-wise embeddings, and perform instance segmentation by predicting binary masks.

3 The proposed method

In this section, we first introduce the overview of our framework. Then, we present our distribution estimation module, distribution-instance matching strategy, and loss function in detail.

3.1 Overview

Given a point cloud with N points $\mathcal{P} \in \mathbb{R}^{N \times d_{\text{in}}}$, our method aims to predict a semantic label and an instance ID (0 for stuff classes) for each point. Here, d_{in} refers to the input attributes, including 3D coordinates and the intensity of reflection. As illustrated in Fig. 2, our GMM-PanopticSeg method consists of a backbone module, a distribution estimation module (DEM) and a voting-based post-process module. Note that, we follow Ref. [40] to generate instance centers from the heatmap branch (see reference [21] for point-based methods). In addition, the heatmap branch can be replaced by other instance center generation modules [13, 14]. Our framework is generic and can be applied to extend different semantic segmentation networks to the panoptic segmentation task.

Our GMM-PanopticSeg method is composed of four major steps, as shown in Fig. 2. First, 3D point clouds are fed to the backbone network [34, 41] to predict the point-wise offset to move each point towards its corresponding instance center, resulting in predicted centers $\mathcal{P}^{\text{ctr}}$. Moreover, point-wise semantic labels and a BEV heatmap are produced. The resultant heatmap is employed to predict N^{inst} instance centers $\mathcal{I}^{\text{ctr}}$ using a window-based search strategy [12, 40]. Second, $\mathcal{I}^{\text{ctr}}$, $\mathcal{P}^{\text{ctr}}$ and point-wise features in the backbone are passed to the proposed distribution estimation module to produce a Gaussian distribution per instance and model the whole scene as a Gaussian mixture model [16] (a probabilistic model that assumes that all points are generated from a mixture of Gaussian distributions). Third, we calculate the likelihood probabilities that one point belongs to different Gaussian distributions and assign each point to the instance with the maximum probability. Finally, we merge the instance segmentation result and semantic segmentation result. Specifically, we accumulate the semantic segmentation results of points assigned to the same instance ID and set semantic labels to the highest scoring category.

3.2 Distribution estimation module

Our distribution estimation module called DEM aims to use a 2D Gaussian distribution to model the intra-instance variance of point-wise embeddings on BEV. As Gaussian distributions are characterized by their covariance matrices, we parameterize each covariance matrix using three hyperparameters, which are learned with neural networks. As illustrated in Fig. 3, our distribution estimation module consists of three steps, including hyperparameter prediction, covariance matrix generation and distribution learning.

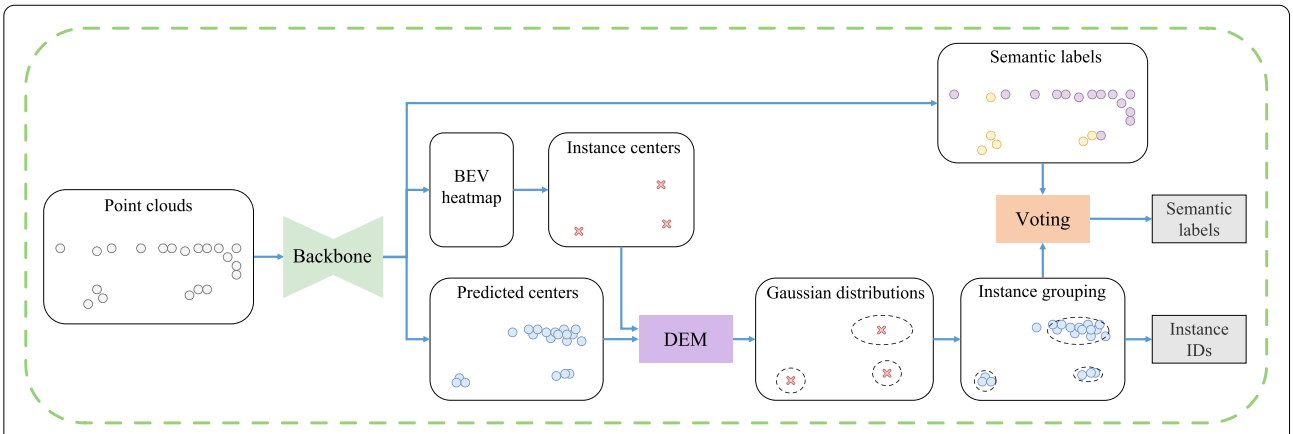


Figure 2 The overall framework of our GMM-PanopticSeg model. The distribution estimation module (DEM) uses the likelihood probability of each point to model the intra-instance variance. BEV refers to the bird's eye view

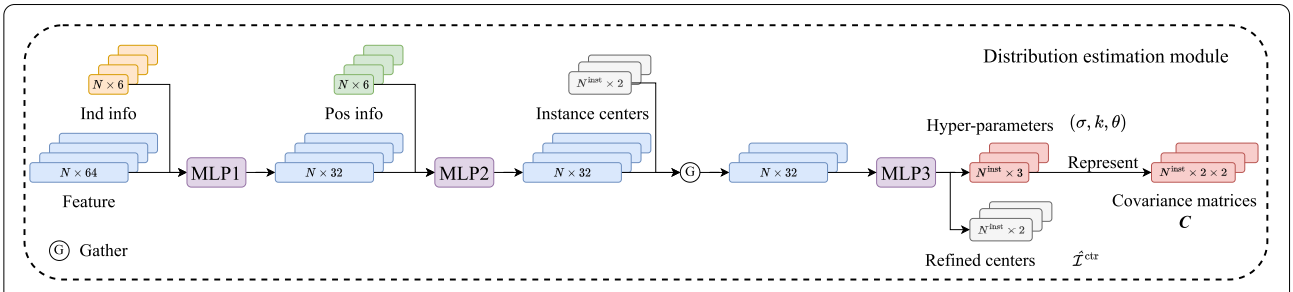


Figure 3 Network architecture of our distribution estimation module. N is the number of input points and N^{inst} is the number of predicted instances. σ, k and θ are parameters of covariance matrices $\boldsymbol{C}$. $\hat{\mathcal{I}}^{\text{ctr}}$ is the refined instance center. MLP1, MLP2 and MLP3 are multi-layer perceptron networks. Ind info is the information of 2D coordinates. Pos info is the information of 2D position

Hyperparameter prediction. Intuitively, the relationship between instance predicted centers $\mathcal{I}^{\text{ctr}}$ and point-wise predicted centers $\mathcal{P}^{\text{ctr}}$ reflects the intra-instance variance and can be used to generate the hyperparameters of Gaussian distributions. For each point $\mathcal{P}_i^{\text{ctr}}$, we first find its nearest instance predicted center $\mathcal{I}_j^{\text{ctr}}$, and concatenate the center feature with it. Next, the 2D indices of $\mathcal{P}_i^{\text{ctr}}$ and $\mathcal{I}_j^{\text{ctr}}$ with $\mathcal{P}_i^{\text{ctr}} - \mathcal{I}_j^{\text{ctr}}$ are concatenated with the point feature, which is subsequently fed into a two-layer multilayer perceptron (MLP) (i.e., MLP1 in Fig. 3) for point-wise aggregation. Similarly, we further concatenate the 2D coordinates of $\mathcal{P}_i^{\text{ctr}}$ and $\mathcal{I}_j^{\text{ctr}}$ with their difference from the point feature and pass the concatenation to another MLP (i.e., MLP2). After aggregating the relationships between instance predicted centers and point-wise predicted centers for each point, we gather features for points belonging to the same predicted instance by indices and use a pooling operation to produce an instance representation. Finally, the hyperparameters σ, k, and θ and the refined instance center $\hat{\mathcal{I}}^{\text{ctr}}$ are regressed for each instance using a three-layer MLP (i.e., MLP3).

Covariance matrix generation. A 2D Gaussian distribution can be characterized using a covariance matrix $\boldsymbol{C} \in \mathbb{R}^{2\times 2}$. With the predicted hyperparameters σ, k, and θ in the previous step, where σ is the minor axis variance, k measures the ratio of the major axis and minor axis variance, and θ represents the rotation angle of the distribution. The covariance matrix of the Gaussian distribution for the i-th predicted instance can be obtained as

$$\boldsymbol{C}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & k_i^2\sigma_i^2 \end{bmatrix} \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix}. \tag{1}$$

Without loss of generality, we keep $k > 1$ so that σ represents the minor axis variance. Here, we use the softplus activate function to keep k and σ positive. Compared with directly calculating the covariance matrix, our method decouples the distribution parameters and reduces the instability during training. Note that, Eq. (1) can also be generalized to higher-dimensional cases. Our ultimate goal is not to perfectly fit the predicted center but to distinguish

points of different instances through the predicted distribution. Intuitively, points closer to the center of the distribution are more likely to be the same instance, and the variance represents the degree of confidence in the distribution. The degree of confidence of the distribution in different dimensions is different and correlated. We adopt a formulation of 2D Gaussian distribution in this paper due to its simplicity in the formulation of point clouds and low computational complexity.

Distribution learning. After the above steps, each predicted instance is modeled as a Gaussian distribution. However, how to fit these 2D Gaussian distributions, characterized by learnable hyperparameters, to diverse objects in point clouds still remains a challenge because the ground truth covariance matrix is not available in real-world scenes to provide supervision. To address this issue, we alternatively maximize the probability that each point in the instance belongs to the corresponding Gaussian distribution. Mathematically, the probability for the i-th predicted instance is calculated as follows:

$$\begin{aligned} P_i &= \prod_{j \in S_i} P_{ij} \\ &= \prod_{j \in S_i} \frac{1}{\sqrt{2\pi|\boldsymbol{C}_i|}} \mathrm{e}^{-\frac{1}{2}(\hat{\mathcal{I}}_i^{\mathrm{ctr}} - \mathcal{P}_j^{\mathrm{ctr}})^{\mathrm{T}} \boldsymbol{C}_i^{-1} (\hat{\mathcal{I}}_i^{\mathrm{ctr}} - \mathcal{P}_j^{\mathrm{ctr}})}, \end{aligned} \tag{2}$$

where P_{ij} is the probability that the j-th point belongs to the i-th predicted instance. $\boldsymbol{C}_i$ and $\hat{\mathcal{I}}_i^{\mathrm{ctr}}$ are the covariance matrix and the center of the Gaussian distribution for the i-th predicted instance, respectively. S_i is the point set for the i-th instance and $\mathcal{P}_j^{\mathrm{ctr}}$ is the predicted center of the j-th point. We assume that the probability of each instance is independent and the negative log-likelihood (NLL) is used as the loss function to help Gaussian distributions tend to fit the intra-instance variance:

$$\mathcal{L}_{\mathrm{dl}} = -\sum_{i=0}^{N^{\mathrm{inst}}-1} \log(P_i). \tag{3}$$

Due to the effects of occlusion and noise, the mismatches between points and corresponding instances usually lead to unstable training. To remedy this, points with scores lower than a threshold τ are filtered out in $\mathcal{L}_{\mathrm{dl}}$: $P_{ij}/\sum_k P_{kj} < \tau$. Specifically, for the j-th point in an instance, the denominator is the probability that a point j belongs to the i-th Gaussian distribution. Moreover, the denominator represents its summed probability with all Gaussian distributions. The smaller the value of $P_{ij}/\sum_k P_{kj}$, the lower the confidence that the j-th point belongs to the i-th Gaussian distribution. Consequently, this point is excluded from Eq. (3) to increase the stability of our Gaussian distributions.

3.3 Distribution-instance matching

With each predicted instance modeled by a Gaussian distribution, the whole point cloud can be considered to be a Gaussian mixture model. In practice, the number of the predicted instances (i.e., Gaussian distributions) may not be consistent with the ground truth number of objects in the scene, and this matching imposes challenges to the optimization. To address this issue, we propose a distribution-instance matching method, which consists of three steps. First, we calculate the probabilities that associate each point with each predicted distribution (i.e., P_{ij} in Eq. (2) that associates the j-th point with the i-th distribution). To prevent missing instances, we first pre-define Gaussian distributions with the identity covariance matrices as padding distributions. Then, for each point, we calculate its probabilities of belonging to different Gaussian distributions, including both predicted and pre-defined distributions. Second, for the i-th Gaussian distribution, we aggregate the mean probabilities of points belonging to the k-th ground truth instance to calculate the distribution-instance matching probability P_{ik}^{match}. Third, we use the Hungarian algorithm [20] to obtain the optimal matching with the highest probability. The PyTorch-style distribution-instance matching algorithm is displayed in Alg. 1.

Note that, we also match the pre-defined distribution from the ground truth to avoid the instance missing problem. We set the selected probability P^{s} of the distribution

Algorithm 1: Distribution-Instance Matching

Input: distribution centers: $\hat{\mathcal{I}}^{\mathrm{ctr}} \in \mathbb{R}^{N^{\mathrm{inst}} \times 2}$
covariance matrices: $\boldsymbol{C} \in \mathbb{R}^{N^{\mathrm{inst}} \times 2 \times 2}$
GT centers: $\mathcal{I}^{\mathrm{gt}} \in \mathbb{R}^{N^{\mathrm{gt}} \times 2}$
GT instance masks: $M^{\mathrm{inst}} \in \mathbb{R}^{N^{\mathrm{gt}} \times N}$
selected probability: $P^{s} \in \mathbb{R}^{(N^{\mathrm{inst}}+N^{\mathrm{gt}}) \times 1}$
predicted centers: $\mathcal{P}^{\mathrm{ctr}} \in \mathbb{R}^{N \times 2}$
Output: match result: $R^{\mathrm{match}} \in \mathbb{R}^{N^{\mathrm{gt}} \times 1}$

Add ground truth centers: $\hat{\mathcal{I}}^{\mathrm{ctr}} = \mathrm{concat}(\hat{\mathcal{I}}^{\mathrm{ctr}}, \mathcal{I}^{\mathrm{gt}})$
Add unit diagonal matrices:
$\boldsymbol{C} = \mathrm{concat}(\boldsymbol{C}, \mathrm{eyes}(2).\mathrm{repeat}(N^{\mathrm{gt}})) \in \mathbb{R}^{(N^{\mathrm{inst}}+N^{\mathrm{gt}}) \times 2 \times 2}$
Reshape: $\hat{\mathcal{I}}^{\mathrm{ctr}} \in \mathbb{R}^{1 \times (N^{\mathrm{inst}}+N^{\mathrm{gt}}) \times 2}$, $\mathcal{P}^{\mathrm{ctr}} \in \mathbb{R}^{N \times 1 \times 2}$
Calculate the distance: $X = \hat{\mathcal{I}}^{\mathrm{ctr}} - \mathcal{P}^{\mathrm{ctr}}$
Calculate the point-distribution probability:
$P^{\mathrm{prob}} = \frac{1}{\sqrt{2\pi|\boldsymbol{C}|}} \mathrm{e}^{-\frac{X \times \boldsymbol{C}^{-1} \times X^{\mathrm{T}}}{2}} \in \mathbb{R}^{N \times (N^{\mathrm{inst}}+N^{\mathrm{gt}})}$
Count the points of each mask:
$N^{\mathrm{mask}} = \mathrm{sum}(M^{\mathrm{inst}}, dim = 1)$
Calculate the instance-distribution probability:
$P^{\mathrm{match}} = M^{\mathrm{inst}} \times P^{\mathrm{prob}} \cdot \frac{1}{N^{\mathrm{mask}}} \cdot P^{s} \in \mathbb{R}^{N^{\mathrm{gt}} \times (N^{\mathrm{inst}}+N^{\mathrm{gt}})}$
Match by the Hungarian algorithm:
$R^{\mathrm{match}} = \mathrm{Hungarian}(P^{\mathrm{match}})$

from the DEM to 1 and set the selected probability P^{s} of the i-th pre-defined distribution to $0.1 \cdot F_{\text{heat}}^{F_i^{\text{gt}}}$, where F^{gt} is the ground truth center and F_{heat} is the heatmap value. Considering that the ground truth centers are potential predicted centers, we associate the selection probability with F_{heat} and encourage proposing ground truth centers when instances are missing.

To achieve trainable distribution-instance matching, we introduce a matching loss to consider matched and missed instances separately:

$$\mathcal{L}_{\text{prob}} = -\frac{1}{N^{\text{gt}}} \sum_{k} \begin{cases} \frac{P_{ak}^{\text{match}}}{\sum_{i=0}^{N^{\text{inst}}-1} P_{ik}^{\text{match}}}, & \text{if matched,} \\ \frac{P_{bk}^{\text{match}}}{\sum_{i=N^{\text{inst}}}^{N^{\text{inst}}+N^{\text{gt}}-1} P_{ik}^{\text{match}}}, & \text{if missed.} \end{cases} \tag{4}$$

If the k-th ground truth instance matches the a-th Gaussian distribution predicted by our distribution estimation module, the upper term is calculated to maximize the matching probability P_{ak}^{match} among all the predicted Gaussian distributions. If the k-th ground truth instance has no matched predicted distribution, it is associated with predefined Gaussian distributions centered at ground truth instance centers to calculate P_{bk}^{match}. Then, the lower term is calculated and our distribution estimation module is used to predict an additional Gaussian distribution to cover this missed instance.

3.4 Loss function

Previous methods consider point-wise embedding learning and instance clustering as two sequential steps and use two independent losses for separate optimization [13]. To address this issue, we introduce a unified loss function $\mathcal{L}_{\text{prob}}$ for end-to-end joint optimization of the whole framework. The sum of loss is as follows:

$$\mathcal{L} = a\mathcal{L}_{\text{cls}} + b\mathcal{L}_{\text{heatmap}} + c\mathcal{L}_{\text{offset}} + d\mathcal{L}_{\text{dl}} + e\mathcal{L}_{\text{prob}}, \tag{5}$$

where we empirically set $a = 1$, $b = 100$, $c = 0.01$, $d = 5$, and $e = 10$. Note that, the first three loss terms are already used in existing panoptic segmentation methods ($\mathcal{L}_{\text{cls}}$ is the loss of the semantic branch, $\mathcal{L}_{\text{offset}}$ is the absolute loss of the offset branch, $\mathcal{L}_{\text{heatmap}}$ is the mean square loss of the heatmap branch). $\mathcal{L}_{\text{dl}}$ and $\mathcal{L}_{\text{prob}}$ incorporate the learning of embedding and clustering into unified losses, thereby allowing for joint optimization of the whole framework.

4 Experiments

In this section, we first present the experimental setups. Then, we compare our method with previous state-of-the-art methods on two benchmark datasets. Finally, we perform ablation experiments to investigate the effectiveness of our framework.

4.1 Experimental setups

Datasets. In our experiments, we evaluate our method on two widely-used large-scale datasets, namely SemanticKITTI [2] and nuScenes [3].

SemanticKITTI. This dataset is composed of 22 sequences with 43,552 sparse LiDAR scans. Specifically, Sequences 00-07 and 09-10 are used for training (19,130 scans), sequence 08 with 4071 scans is used for validation, and the rest are used for online testing (20,351 scans). For the task of panoptic segmentation, the original annotations are remapped to 19 classes, of which there are 8 thing classes and 11 stuff classes. Each point is labeled with a unique semantic label and instance ID, where ID is set to 0 if it belongs to the stuff classes.

nuScenes. This dataset consists of 1000 sequences. We use 28,130 frames for training, 6019 frames for validation, and 6008 frames for testing. The nuScenes dataset contains 16 classes, 10 of which are things. There are more sparse point clouds and denser objects in nuScenes than in SemanticKITTI, which makes instance segmentation more difficult. Moreover, nuScenes has a more balanced distribution of categories than SemanticKITTI, which facilitates the learning of semantic segmentation.

Metrics. Following Alexander et al. [42], we use the widely used panoptic quality (PQ) as the main metric to evaluate the performance of panoptic segmentation.

$$\text{PQ} = \underbrace{\frac{\sum_{(p,g)\in \text{TP}} \text{IoU}(p,g)}{|\text{TP}|}}_{\text{Segmentation quality (SQ)}} \underbrace{\frac{|\text{TP}|}{|\text{TP}| + \frac{1}{2}|\text{FP}| + \frac{1}{2}|\text{FN}|}}_{\text{Recognition quality (RQ)}}, \tag{6}$$

where PQ can be decomposed into the product of recognition quality (RQ) and segmentation quality (SQ). RQ can be used to measure the recognition quality, and SQ represents the segmentation quality when the object is recognized. Considering that PQ over-penalizes errors for stuff, we also follow Porzi et al. [43] to use $\text{PQ}^{\dagger}$ as evaluation metrics. In addition, we use the mean intersection-over-union (mIoU) as the metric for semantic segmentation performance.

Implementation details. Since the latest methods are not open source, we choose two representative methods as our baseline. For a fair comparison with the previous method, we use the same hyperparameter settings as in Ref. [12] for training and inference (all parameters not mentioned are the same). τ is empirically set to 0.01 in our experiments. As Dropblock [44] affects the generation of the heatmap, we do not activate Dropblock when training DEM. We also employ several data augmentation techniques, including instance oversampling, random rotation ($[-\pi, \pi]$), random flipping (both the x-axis and the y-axis), and random scaling ($[0.95, 1.05]$). The default Adam optimizer for the backbone with a learning rate of 0.001 and the customized

Adam optimizer for the DEM are used to train our method on RTX3090 GPUs.

Incorporate DEM to DS-Net. Unlike Panoptic-Polarnet, DS-Net does not have a heatmap branch, hence we project the sparse 3D feature from the backbone to a dense BEV and use U-Net style upsampling layers as the heatmap branch to predict prototypes. Prototypes and point-wise embedding are fed into the DEM to generate predicted distributions.

4.2 Comparison with the state-of-the-arts

Quantitative evaluation on nuScenes. Table 1 shows the quantitative comparison of different methods on the nuScenes test set. The proposed method significantly improves the performance of DS-Net and Panoptic-PolarNet in terms of almost all the metrics. Since the additional loss function only focuses on things, the segmentation performance of stuff may be slightly degraded. In other words, stuff receives less attention in relative terms. Notably, the most significant improvement lies in the PQ^{Th} score (an average improvement of 9.6%), indicating that our method can substantially improve the instance segmentation accuracy of the baseline network. Following recent works, we merge instance segmentation and semantic segmentation results via a major-voting strategy. Specifically, points with the same instance ID are updated with the same semantic label such that the semantic segmentation performance is improved. Moreover, Panoptic-PHNet has the best performance because it uses a stronger non-open source backbone and test time augmentation technology on the nuScenes test dataset, which increases the cost of inference. We compare our method with Panoptic-PHNet in ablation studies with the same backbone and post-processing.

We further provide quantitative comparisons on the validation set of the nuScenes dataset in Table 2. It is observed that the combination of the baseline networks with our method significantly improves their overall performance. In particular, by utilizing Eq. (4) to handle mismatches between instances and Gaussian distributions, a much higher accuracy is achieved. This can be reflected by the substantial increase in RQ^{Th} (an average increase of 8.7%). Additionally, the Gaussian distributions predicted by our DEM enable the baseline network to model intra-instance variance, which improves its robustness to occlusion and noise.

Quantitative evaluation on SemanticKITTI. We have conducted experiments on the SemanticKITTI dataset, and the quantitative results on the validation set are presented in Table 3. For a fair comparison with corresponding baseline networks, their officially released pre-trained models are used for the initialization of our backbones. It can be observed that our framework significantly improves the performance of Panoptic-PolarNet. Since the SemanticKITTI dataset has a lower instance density and relatively rich point information, the main challenge on this dataset is not instance segmentation but semantic segmentation, as also noted in Panoptic-PolarNet [12]. Therefore, the performance improvements achieved on this dataset are relatively smaller than those achieved on the nuScenes dataset. Nevertheless, our GMM-PanopticSeg still improves the PQ^{Th} on SemanticKITTI from 65.7% to 68.6%, demonstrating the effectiveness of our framework.

Qualitative results. We provide qualitative comparisons between the baseline and our GMM-PanopticSeg method on the SemanticKITTI and the nuScenes datasets in Fig. 4 and Fig. 5, respectively. Note that stuff is assigned a unique color according to the semantic label and each thing is assigned a random color according to the instance ID. We use black points to represent points that are mapped to the noise. Two important observations in Fig. 4 and Fig. 5 are noted here. First, large instances are segmented into multiple fragments by the baseline network. Since the point clouds on the surface of these large objects are far from their instance centers, high intra-instance variance limits the accuracy of previous methods. Second, dense objects are prone to being assigned wrong instance IDs. This is because the inter-instance differences for dense objects are relatively small, making previous methods sensitive to occlusion and noise. By explicitly modeling intra-instance variance and conducting embedding learning with instance clustering in an end-to-end framework, our method produces more accurate segmentation results for both large and dense objects.

4.3 Ablation studies

To verify the effectiveness of the proposed components in our framework, we perform ablation studies in this section. Specifically, we start by reproducing our GMM-PanopticSeg method from Panoptic-PolarNet step by step, as shown in Table 4.

Learnable vs. pre-defined Gaussian distribution. One of the major contributions of our method is to model the intra-instance variance using Gaussian distributions characterized with learnable parameters. A straightforward alternative is to use pre-defined Gaussian distributions. To validate the effectiveness of our approach, we design model 1 and model 2 by introducing pre-defined and learnable Gaussian distributions to the baseline to model intra-instance variance, respectively. Note that models 1 and 2 use the same heuristic technique as the baseline for instance segmentation during inference and Gaussian distributions that are used only for embedding learning during the training phase. It is found that pre-defined Gaussian distributions boost the performance of the baseline, with PQ/mIoU scores improving from 63.2%/67.9% to 64.9%/66.8%. When learnable Gaussian distributions are employed, model 2 achieves further gains (68.5%/69.1%).

Table 1 Quantitative results (%) of different approaches on the nuScenes [3] test dataset. PQ means panoptic quality. RQ denotes recognition quality. SQ represents segmentation quality. mIoU denotes intersection over union. Th represents foreground classes. St denotes stuff classes. § means our reproduced results. The blue number represents the growth compared to the baseline. The red number denotes the reduction compared to the baseline

	PQ	PQ†	RQ	SQ	PQTh	RQTh	SQTh	PQSt	RQSt	SQSt	mIoU
EfficientLPS [9]	62.4	66.0	74.1	83.7	57.2	68.2	83.6	71.1	84.0	83.8	66.7
Panoptic-PHNet [14]	80.1	82.8	87.6	91.1	82.1	88.1	93.0	76.6	86.6	87.9	80.2
DS-Net§ [13]	58.8	62.7	68.8	83.7	51.9	60.8	82.9	70.4	82.1	85.0	68.5
Ours (DS-Net+DEM)	65.3$_{+6.5}$	69.5$_{+6.8}$	75.0$_{+6.2}$	86.2$_{+2.5}$	62.7$_{+10.8}$	71.1$_{+10.3}$	87.2$_{+4.3}$	69.7$_{-0.7}$	81.5$_{-0.6}$	84.7$_{-0.3}$	71.4$_{+2.9}$
Panoptic-PolarNet [12]	63.6	67.1	75.1	84.3	59.0	69.8	84.3	71.3	83.9	84.2	67.0
Ours (Panoptic-PolarNet+DEM)	68.9$_{+5.3}$	72.4$_{+5.3}$	78.2$_{+3.1}$	87.7$_{+3.4}$	67.4$_{+8.4}$	75.1$_{+5.3}$	89.4$_{+5.1}$	71.4$_{+0.1}$	83.3$_{-0.6}$	84.8$_{+0.6}$	67.6$_{+0.6}$

Table 2 Quantitative results (%) of different approaches on the nuScenes [3] validation dataset. PQ means panoptic quality. RQ denotes recognition quality. SQ represents segmentation quality. mIoU denotes intersection over union. Th represents foreground classes. St denotes stuff classes. § means our reproduced results. The blue number represents the growth compared to the baseline. The red number denotes the reduction compared to the baseline

	PQ	PQ†	RQ	SQ	PQTh	RQTh	SQTh	PQSt	RQSt	SQSt	mIoU
PanopticTrackNet [8]	51.4	56.2	63.3	80.2	45.8	55.9	81.4	60.4	75.5	78.3	58.0
EfficientLPS [9]	62.0	65.6	73.9	83.4	56.8	68.0	83.2	70.6	83.6	83.8	65.6
MaskPLS-M [38]	57.7	60.2	66.0	71.8	64.4	73.3	84.8	52.2	60.7	62.4	62.5
GP-S3Net [36]	61.0	67.5	72.0	84.1	56.0	65.2	85.3	66.0	78.7	82.9	75.8
SCAN [45]	65.1	68.9	85.7	75.3	60.6	85.7	70.2	72.5	85.7	83.8	77.4
SMAC-Seg [15]	68.4	73.4	79.7	85.2	68.0	77.2	87.3	68.8	82.1	83.0	71.2
Panoptic-PHNet [14]	74.7	77.7	84.2	88.2	74.0	82.5	89.0	75.9	86.9	86.8	79.7
DS-Net [13]	42.5	51.0	50.3	83.6	32.5	83.1	38.3	59.2	84.4	70.3	70.7
DS-Net§ [13]	58.4	62.3	69.2	82.9	51.0	60.9	81.9	70.9	83.2	84.6	69.5
Ours (DS-Net+DEM)	65.8$_{+7.4}$	69.9$_{+7.6}$	76.2$_{+7}$	85.7$_{+2.8}$	63.0$_{+12}$	72.2$_{+11.3}$	86.5$_{+4.6}$	70.4$_{-0.5}$	82.7$_{-0.5}$	84.3$_{-0.3}$	71.6$_{+2.1}$
Panoptic-PolarNet [12]	63.4	67.2	75.3	83.9	59.2	70.3	84.1	70.4	83.5	83.6	66.9
Ours (Panoptic-PolarNet+DEM)	69.6$_{+6.2}$	70.6$_{+3.4}$	79.2$_{+3.9}$	87.4$_{+3.5}$	68.3$_{+9.1}$	76.4$_{+6.1}$	89.0$_{+5.9}$	71.8$_{+1.4}$	83.8$_{+0.3}$	84.7$_{+0.9}$	71.3$_{+4.4}$

Table 3 Quantitative results (%) of different approaches on the SemanticKITTI [2] validation dataset. PQ means panoptic quality. RQ denotes recognition quality. SQ represents segmentation quality. mIoU denotes intersection over union. Th represents foreground classes. St denotes stuff classes. The blue number represents the growth compared to the baseline. The red number denotes the reduction compared to the baseline

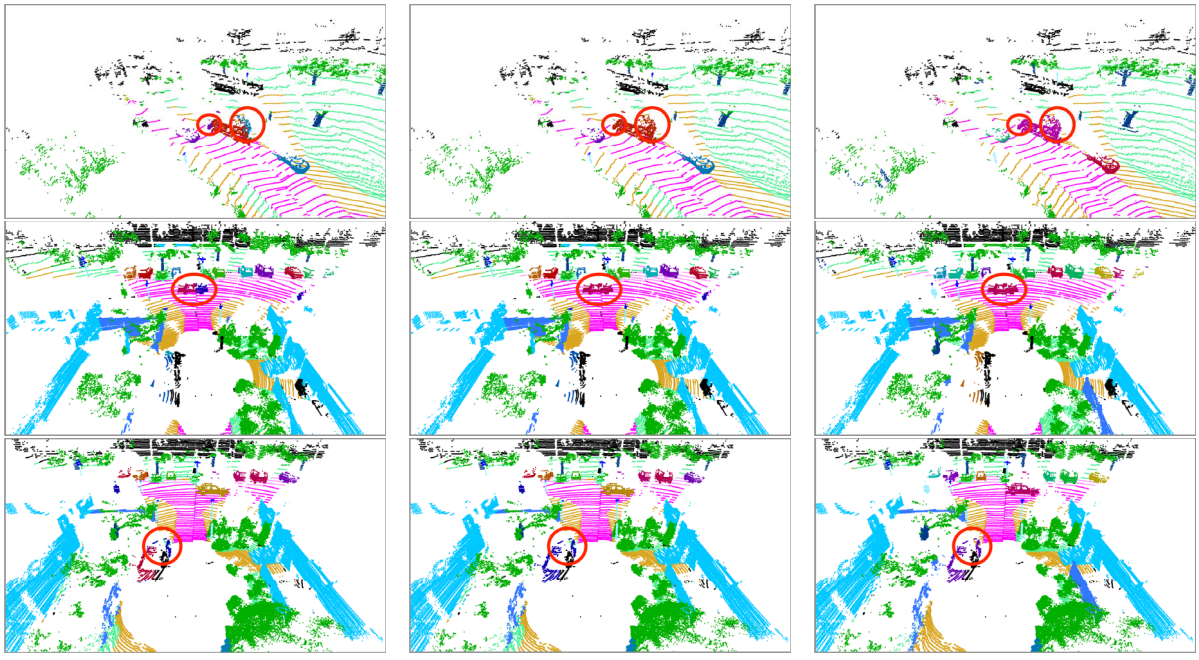
	PQ	PQ†	RQ	SQ	PQTh	RQTh	SQTh	PQSt	RQSt	SQSt	mIoU
PanopticTrackNet [8]	40.0	-	48.3	73.0	29.9	33.6	76.8	47.4	70.3	59.1	53.8
DS-Net [13]	57.7	63.4	68.0	77.6	61.8	68.8	78.2	54.8	67.3	77.1	63.5
Panoster [46]	55.6	-	66.8	79.9	56.6	65.8	-	-	-	-	61.1
EfficientLPS [9]	59.2	65.1	69.8	75.0	58.0	68.2	78.0	60.9	71.0	72.8	64.9
Panoptic-PHNet [14]	61.7	-	-	-	69.3	-	-	-	-	-	65.7
GP-S3Net [36]	63.3	71.5	75.9	81.4	70.2	80.1	86.2	58.3	72.9	77.9	73.0
Panoptc-PolarNet [12]	59.1	64.1	70.2	78.3	65.7	74.7	87.4	54.3	66.9	71.6	63.9
Ours (Panoptc-PolarNet+DEM)	60.3$_{+1.2}$	64.5$_{+0.4}$	70.8$_{+0.6}$	79.2$_{+0.9}$	68.6$_{+2.9}$	76.2$_{+1.5}$	89.7$_{+2.3}$	54.3$_{+0.0}$	66.9$_{+0.0}$	71.6$_{+0.0}$	64.2$_{+0.3}$

This demonstrates that learnable Gaussian distributions can better improve the discrimination capability of point embeddings by explicitly modeling the intra-instance variance of an object.

With learnable Gaussian distributions, we can also replace the heuristic technique in the baseline (i.e., L2 distance) with the likelihood probability to make better use of the modeled intra-instance variance. Using the likelihood probability for instance segmentation during inference, model 3 further surpasses model 2 with notable improvements (69.6%/71.3%). Due to occlusion and noise in real-world scenarios, the intra-instance variance may be larger than inter-instance difference such that heuristic techniques produce limited accuracy. By modeling the intra-instance variance with Gaussian distributions, the likelihood probability measure is more robust than the other methods and can better distinguish different instances.

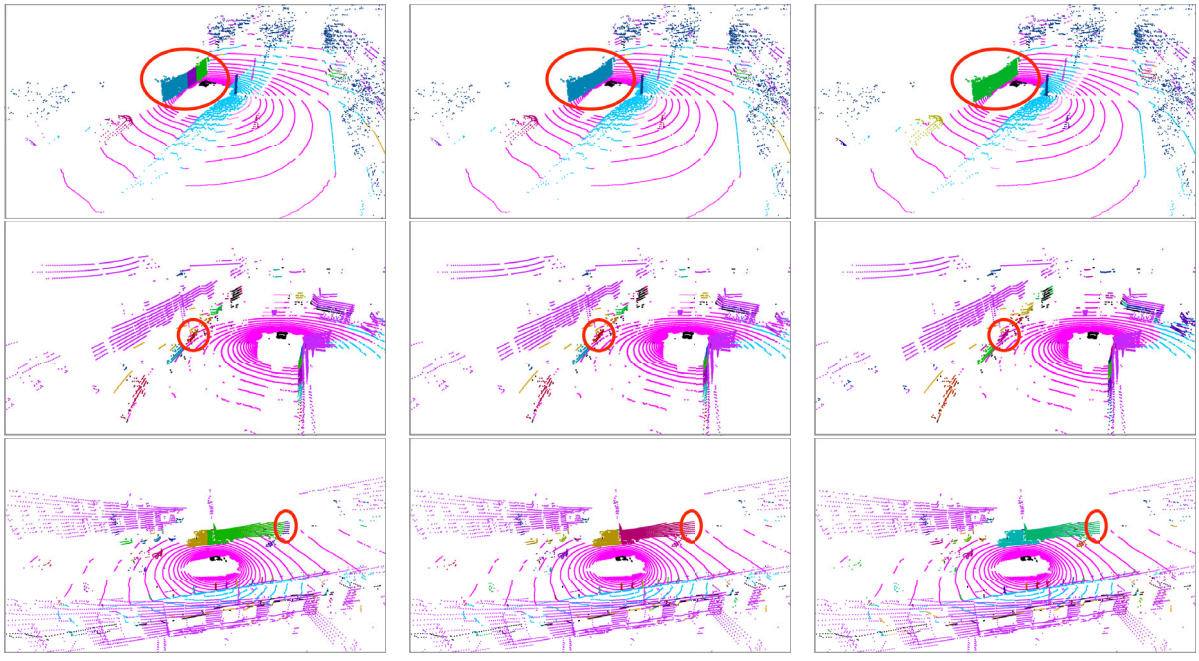
Visualization of learned Gaussian distributions. We further visualize learned Gaussian distributions for different objects in Fig. 6 and two important observations are reported here. First, larger objects (e.g., vehicles) with higher intra-instance variance have higher variance in their predicted Gaussian distributions than smaller objects (e.g., persons and bicycles). Second, the major axis of the predicted Gaussian distribution is usually along the long side of the object (e.g., cars and motorcycles). In summary, our predicted Gaussian distributions can model the intra-instance variance well for diverse instances.

End-to-End vs. decoupled optimization. Another major contribution of our framework is the unified loss function



(a) Baseline (b) Ours (c) Ground truth

Figure 4 Qualitative results on SemanticKITTI. Note that stuff is assigned a unique color according to the semantic label and each thing is assigned a random color according to the instance ID. We use black points to represent points that are mapped to the noise. The errors made by the baseline method are indicated by the red circle



(a) Baseline (b) Ours (c) Ground truth

Figure 5 Qualitative results on nuScenes. Note that stuff is assigned a unique color according to the semantic label and each thing is assigned a random color according to the instance ID. We use black points to represent points that are mapped to the noise. The errors made by the baseline method are indicated by the red circle

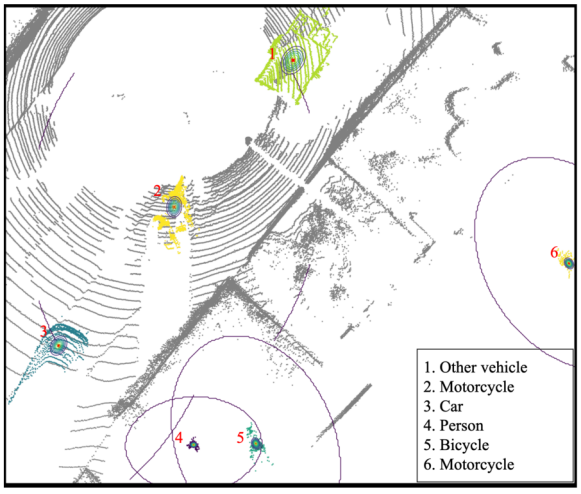


Figure 6 Visualization of the predicted Gaussian distributions on the SemanticKITTI dataset

Table 4 Ablation studies on the nuScenes validation set. PQ means panoptic quality. mIoU denotes intersection over union. L2 means Euclidean norm

	Training phase		Inference phase	PQ(%)	mIoU(%)
	Gaussian	$\mathcal{L}_{prob}$	Measure		
Baseline			L2	63.2	67.9
1	Predefined	✓	L2	64.9	66.8
2	Learnable	✓	L2	68.5	69.1
3	Learnable	✓	Likelihood	69.6	71.3

Table 5 Ablation studies on the SemanticKITTI validation set. All values are in [%]. PQ means panoptic quality. mIoU denotes intersection over union. Th represents foreground classes. St denotes stuff classes

	Optimization	PQ	PQ^{Th}	PQ^{St}	mIoU
Baseline	Decoupled	58.9	65.2	54.3	63.9
4	Decoupled	59.6	67.0	54.3	64.0
5	End-to-End	60.3	68.6	54.3	64.2

that optimizes embedding learning and instance clustering in an end-to-end manner. To validate its effectiveness, we have trained our method using two different optimization strategies. Specifically, the backbone of model 4 is first optimized and then frozen to train the subsequent modules. In contrast, in model 5, all modules are trained together via end-to-end fusion. Table 5 shows that model 5 outperforms model 4 on most metrics. This demonstrates the superiority of our end-to-end optimization paradigm for panoptic segmentation.

Comparison of clustering methods. We compare our method with Meanshift, PHM [14] and LHM [12] in Table 6. The result of using ground truth instance labels is also provided. The same semantic branch is used for fair

Table 6 Results on the SemanticKITTI validation set. PQ means panoptic quality

	MeanShift	LHM [12]	PHM [14]	Ours	GT
PQ(%)	56.2	59.1	59.8	60.0	60.1

Table 7 Computational consumption on SemanticKITTI. Params means parameters. FLOPs denotes floating point operations

With DEM	Params(M)	FLOPs(G)
	13.12	123.7
✓	13.17	131.3

comparison. Our method outperforms existing clustering-based methods while being very close to the ground truth. We can also observe that even if the instance labels are replaced by ground truth labels, the PQ does not change much. This also explains why our method does not significantly improve the PQ score on the SemanticKITTI dataset.

Computational consumption. In our design, we only feed points predicted as things to the DEM. This design has the same effect as feeding the complete points but helps minimize the computational cost. The inference cost is presented in Table 7. We measure the average inference cost of our method with Panoptic-PolarNet as the backbone on SemanticKITTI. The DEM is tiny, and the additional computational consumption is focused mainly on calculating the probability of each point in each distribution.

5 Conclusion

In this paper, we introduce a Gaussian mixture model for 3D panoptic segmentation and employ learnable Gaussian distributions to capture the intra-instance variance of different objects. In addition, we propose an end-to-end loss function for the joint optimization of embedding learning and instance clustering. Extensive experiments on different benchmark datasets and backbones validate the effectiveness of the proposed method.

Abbreviations

BEV, bird's-eye-view; DEM, distribution estimation module; FLOP, floating point operations per second; GCNN, graph convolutional neural networks; GMM, Gaussian mixture model; GPU, graphics processing unit; GT, ground truth; L2, Euclidean norm; LHM, learnable heatmap; mIoU, mean intersection over union; MLP, multi-layer perceptron; NLL, negative log-likelihood; PHM, pseudo heatmap; PQ, panoptic quality; RQ, recognition quality; SOTA, state-of-the-art; SQ, segmentation quality.

Data availability

The datasets analyzed during the current study are available: 1) SemanticKITTI: http://www.semantic-kitti.org/ 2) nuScenes: https://www.nuscenes.org/

Declarations

Competing interests

The authors declare no competing interests.

Author contributions

YW conceived the initial ideas, conducted detailed experiments, and drafted the paper. LW revised the manuscript and improved the experimental design. QH revised the manuscript and made improvements to the experimental design. YL has improved the research ideas. YZ and YG systematically refined the research framework and guided the writing of the paper. All authors read and approved the final manuscript.

Author details

[1]School of Electronics and Communication Engineering, the Shenzhen Campus of Sun Yat-sen University, Sun Yat-sen University, Shenzhen, 510275, China. [2]University of Air Force, Changchun, China. [3]Military Academy Science, Beijing, China. [4]College of Electronic Science and Technology, National University of Defense Technology, Changsha, Hunan 410073, China.

Received: 31 October 2023 Revised: 10 March 2024
Accepted: 11 March 2024 Published online: 07 April 2024

References

1. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., et al. (2020). RandLA-Net: efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11105–11114). Piscataway: IEEE.
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., et al. (2019). SemanticKITTI: a dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9296–9306). Piscataway: IEEE.
3. Fong, W. K., Mohan, R., Hurtado, J. V., Zhou, L., Caesar, H., Beijbom, O., et al. (2022). Panoptic nuscenes: a large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, *7*(2), 3795–3802.
4. Rusu, R. B., & Cousins, S. (2011). 3D is here: point cloud library (PCL). In *Proceedings of the IEEE international conference on robotics and automation*, Piscataway: IEEE.
5. Behley, J., Milioto, A., & Stachniss, C. (2021). A benchmark for lidar-based panoptic segmentation based on KITTI. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 13596–13603). Piscataway: IEEE.
6. Thomas, H., Qi, C. R., Deschaud, J., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). KPConv: flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6410–6419). Piscataway: IEEE.
7. Lang, A. H., Vora, S., Caesar, H., Zhou, L., Yang, J., & Beijbom, O. (2019). 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). Piscataway: IEEE.
8. Hurtado, J. V., Mohan, R., & Valada, A. (2020). MOPT: multi-object panoptic tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshop on scalability in autonomous driving*.
9. Sirohi, K., Mohan, R., Büscher, D., Burgard, W., & Valada, A. (2022). EfficientLPS: efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, *38*(3), 1894–1914.
10. Ye, D., Zhou, Z., Chen, W., Xie, Y., Wang, Y., Wang, P., et al. (2023). LidarMultiNet: towards a unified multi-task network for lidar perception. In B. Williams, Y. Chen, & J. Neville (Eds.), *Proceedings of the 37nd AAAI conference on artificial intelligence* (pp. 3231–3240). Palo Alto: AAAI Press.
11. Milioto, A., Behley, J., McCool, C., & Stachniss, C. (2020). Lidar panoptic segmentation for autonomous driving. In *Proceedings of the IEEE international conference on intelligent robots and systems* (pp. 8505–8512). Piscataway: IEEE.
12. Zhou, Z., Zhang, Y., & Foroosh, H. (2021). Panoptic-polarnet: proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13194–13203). Piscataway: IEEE.
13. Hong, F., Zhou, H., Zhu, X., Li, H., & Liu, Z. (2021). Lidar-based panoptic segmentation via dynamic shifting network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13090–13099). Piscataway: IEEE.
14. Li, J., He, X., Wen, Y., Gao, Y., Cheng, X., & Zhang, D. (2022). Panoptic-PHNet: towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11799–11808). Piscataway: IEEE.
15. Li, E., Razani, R., Xu, Y., & Liu, B. (2022). SMAC-Seg: lidar panoptic segmentation via sparse multi-directional attention clustering. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 9207–9213). Piscataway: IEEE.
16. Reynolds, D. A. (2009). Gaussian mixture models. In S. Z. Li & A. K. Jain (Eds.), *Encyclopedia of biometrics* (pp. 659–663). Cham: Springer.
17. Yi, L., Zhao, W., Wang, H., Sung, M., & Guibas, L. J. (2019). 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). Piscataway: IEEE.
18. Hou, J., Dai, A., & Nießner, M. (2019). 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4421–4430). Piscataway: IEEE.
19. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., et al. (2019). Learning object bounding boxes for 3D instance segmentation on point clouds. In H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. (Eds.), *Proceedings of the 32rd international conference on neural information processing systems* (pp. 6737–6746). Red Hook: Curran Associates.
20. Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1–2), 83–97.
21. Liu, S., Yu, S., Wu, S., Chen, H., & Liu, T. (2020). Learning Gaussian instance segmentation in point clouds. arXiv preprint. arXiv:2007.09860.
22. Wang, W., Yu, R., Huang, Q., & Neumann, U. (2018). SGPN: similarity group proposal network for 3D point cloud instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2569–2578). Piscataway: IEEE.
23. Wang, X., Liu, S., Shen, X., Shen, C., & Jia, J. (2019). Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4096–4105). Piscataway: IEEE.
24. Lahoud, J., Ghanem, B., Oswald, M. R., & Pollefeys, M. (2019). 3D instance segmentation via multi-task metric learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9255–9265). Piscataway: IEEE.
25. Liang, Z., Yang, M., Li, H., & Wang, C. (2020). 3D instance embedding learning with a structure-aware loss function for point cloud segmentation. *IEEE Robotics and Automation Letters*, *5*(3), 4915–4922.
26. Pham, Q., Nguyen, D. T., Hua, B., Roig, G., & Yeung, S. (2019). 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* (pp. 8827–8836). Piscataway: IEEE.
27. Liu, C., & Furukawa, Y. (2019). MASC: multi-scale affinity with sparse convolution for 3D instance segmentation. arXiv preprint. arXiv:1902.04478.
28. He, T., Shen, C., & van den Hengel, A. (2021). DyCo3D: robust instance segmentation of 3D point clouds through dynamic convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 354–363). Piscataway: IEEE.
29. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., & Leibe, B. (2023). Mask3D: mask transformer for 3D semantic instance segmentation. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 8216–8223). Piscataway: IEEE.
30. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C., & Jia, J. (2020). PointGroup: dual-set point grouping for 3D instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4866–4875). Piscataway: IEEE.
31. Chen, S., Fang, J., Zhang, Q., Liu, W., & Wang, X. (2021). Hierarchical aggregation for 3D instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 15447–15456). Piscataway: IEEE.
32. Liang, Z., Li, Z., Xu, S., Tan, M., & Jia, K. (2021). Instance segmentation in 3D scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 2763–2772). Piscataway: IEEE.
33. Vu, T., Kim, K., Luu, T. M., Nguyen, T., & Yoo, C. D. (2022). Softgroup for 3D instance segmentation on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2698–2707). Piscataway: IEEE.
34. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., et al. (2021). Cylindrical and asymmetrical 3D convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9939–9948). Piscataway: IEEE.

35. Liu, M., Zhou, Q., Zhao, H., Li, J., Du, Y., Keutzer, K., et al. (2022). Prototype-voxel contrastive learning for lidar point cloud panoptic segmentation. In *Proceedings of the IEEE international conference on robotics and automation* (pp. 9243–9250). Piscataway: IEEE.
36. Razani, R., Cheng, R., Li, E., Taghavi, E., Ren, Y., & Liu, B. (2021). GP-S3Net: graph-based panoptic sparse semantic segmentation network. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16056–16065). Piscataway: IEEE.
37. Chen, Q., Vora, S., & Beijbom, O. (2021). PolarStream: streaming object detection and segmentation with polar pillars. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, et al. (Eds.), *Proceedings of the 34th international conference on neural information processing systems* (pp. 26871–26883). Red Hook: Curran Associates.
38. Marcuzzi, R., Nunes, L., Wiesmann, L., Behley, J., & Stachniss, C. (2023). Mask-based panoptic lidar segmentation for autonomous driving. *IEEE Robotics and Automation Letters*, *8*(2), 1141–1148.
39. Su, S., Xu, J., Wang, H., Miao, Z., Zhan, X., Hao, D., et al. (2023). PUPS: point cloud unified panoptic segmentation. In B. Williams, Y. Chen, & J. Neville (Eds.), *Proceedings of the 37th AAAI conference on artificial intelligence* (pp. 2339–2347). Palo Alto: AAAI Press.
40. Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., et al. (2020). Panoptic-DeepLab: a simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12472–12482). Piscataway: IEEE.
41. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., et al. (2020). PolarNet: an improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9598–9607). Piscataway: IEEE.
42. Kirillov, A., He, K., Girshick, R. B., Rother, C., & Dollár, P. (2019). Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9404–9413). Piscataway: IEEE.
43. Porzi, L., Bulò, S. R., Colovic, A., & Kontschieder, P. (2019). Seamless scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8277–8286). Piscataway: IEEE.
44. Ghiasi, G., Lin, T., & Le, Q. V. (2018). DropBlock: a regularization method for convolutional networks. In S. Bengio, H. M. Wallach, H. Larochelle, et al. (Eds.), *Proceedings of the 31st international conference on neural information processing systems* (pp. 10750–10760). Red Hook: Curran Associates.
45. Xu, S., Wan, R., Ye, M., Zou, X., & Cao, T. (2022). Sparse cross-scale attention network for efficient lidar panoptic segmentation. In *Proceedings of the 36th international joint conference on artificial intelligence* (pp. 2920–2928). Palo Alto: AAAI Press.
46. Gasperini, S., Mahani, M. N., Marcos-Ramiro, A., Navab, N., & Tombari, F. (2021). Panoster: end-to-end panoptic segmentation of lidar point clouds. *IEEE Robotics and Automation Letters*, *6*(2), 3216–3223.

Publisher's Note