Identifying Unknown Instances for Autonomous Driving

Kelvin Wong[1,2], Shenlong Wang[1, 2], Mengye Ren[1, 2], Ming Liang[1], Raquel Urtasun[1, 2]
Uber Advanced Technologies Group[1], University of Toronto[2]
{kelvin.wong, slwang, mren3, ming.liang, urtasun}@uber.com

Abstract: In the past few years, we have seen great progress in perception algorithms, particular through the use of deep learning. However, most existing approaches focus on a few categories of interest, which represent only a small fraction of the potential categories that robots need to handle in the real-world. Thus, identifying objects from unknown classes remains a challenging yet crucial task. In this paper, we develop a novel open-set instance segmentation algorithm for point clouds which can segment objects from both known and unknown classes in a holistic way. Our method uses a deep convolutional neural network to project points into a category-agnostic embedding space in which they can be clustered into instances irrespective of their semantics. Experiments on two large-scale self-driving datasets validate the effectiveness of our proposed method.

Keywords: Open-Set Perception, Autonomous Driving, Instance Segmentation

1   Introduction

One relaxing summer weekend, I drove my family on an excursion to the zoo. All of a sudden, I saw a tiny black creature in front of my car. It was too far away for me to tell what it was, but as an experienced driver, I rolled out a series of moves without hesitation: I performed a shoulder check, I signaled, and then I switched lanes. In the end, I still had not figured out what it was until my daughter told me it was a raccoon crossing the street. The ability to recognize an object without knowing its semantics seems innate to us humans. However, this is in fact one of the holy grails that we strive to develop in our robotic perception systems.

A common paradigm in robotics perception is to train and deploy a machine-learned model under the *closed-set* condition; *i.e.*, the robot is only trained to identify instances from known classes. In this paper, we argue that this is not enough for a practical perception system, since in real-world applications, robots often have to operate in an open environment interacting with surrounding objects that were not seen during training. Thus, an ideal perception system should be capable of recognizing and localizing objects from both known and unknown classes. This is referred to as the *open-set* setting.

We are not the first to realize the importance of identifying and interacting with unknown instances. In the pioneering work by Saxena et al. [1], the authors proposed to grasp a novel object by identifying good positions to grasp; this could be trained on known instances and then generalized to unknowns. Also, cognitive scientists have studied the underlying mechanism by which the human vision system detects novel objects. In the 1980s, experiments on novel objects were conducted on rats to reveal how long/short term memory influences recognition [2, 3]. In the computer vision community, researchers approached the open-set recognition problem by first defining the open space as the space sufficiently far from any known positive training samples, measured by a multi-class classification function, where unknowns would carry all zero values in the classifier outputs [4, 5]. However, these approaches are either restricted to a classification task or specific to downstream robotics tasks. We generalize this idea to open-set instance segmentation with the additional capability to group observations into the same instance.

Recognizing and segmenting an object without seeing its category during training is fundamentally challenging for modern deep networks. It makes the networks unable to exploit shape, appearance,
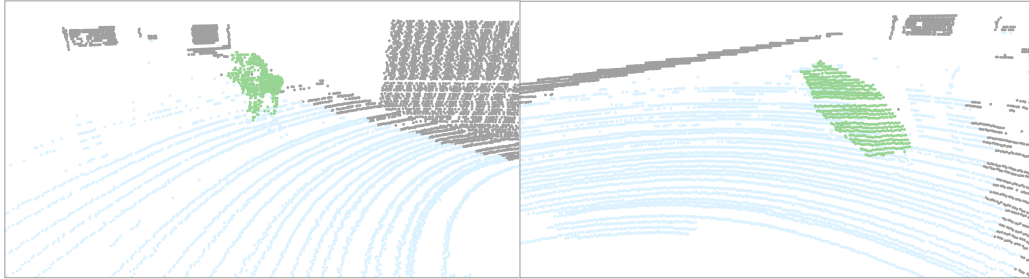
Figure 1: Exemplar output of open-set instance segmentation. The left figure depicts a horse on the road while the right figure depicts an umbrella. In day-to-day driving, these objects are not commonly found on the road. However, an ideal robotics perception system should still be able to recognize and localize such objects, and determine whether they belong to one of the known classes.

and other information about the category during training. However the aforementioned capability critically influences deep models' success. Back to the mid-20th century, vision scientists identified a mechanism in the human vision system which groups visual elements belong together into an object. This mechanism is called *perceptual grouping* [6] and it contributes to our ability to recognize novel objects. Motivated by the success of the human vision system, our goal is to empower robots with a similar capability; *i.e.*, we would like them to learn to perceptually group visual elements into a "thing", and then classify whether it belongs to one of the known classes.

Towards this goal of jointly recognizing both known and unknown instances, we propose a novel perception algorithm for LiDAR point clouds: the Open-Set Instance Segmentation (OSIS) network. The high-level idea is to use a deep convolutional network to identify uncertain points and group them into novel unknown instances. Specifically, a category-agnostic instance embedding network is proposed to project points from the same instance to be close together in the embedding space irrespective of their semantics. As a result, the network learns to group observations into a thing without knowing the thing's category. Our open-set inference procedure is straightforward: We first compute prototypical features from each known-class instance and then associate points with them according to their embedding feature distance. Finally, we cluster the rest of the points to form new unknown-class instances.

We validate our model's performance on two large-scale self-driving datasets with unknown objects. Our experiments show that OSIS outperforms other competing methods in terms of identifying both known and unknown instances.

2 Related Work

The problem of segmentation originates from the concept of perceptual grouping [7], which argues that human perception organizes perceptual signals into objects and meaningful clusters instead of raw pixels. Early segmentation approaches [8, 9] mainly deal with low-level regions and often do not capture the notion of objects. Recently, with the growing availability of high-quality segmentation labels, several benchmarks [10, 11, 12] have become very popular for both semantic and instance segmentation tasks. Panoptic segmentation [13] was proposed to combine the two problems together by jointly reasoning about instances and background.

Standard instance and panoptic segmentation approaches, however, fail to capture unknown instances that have never appeared in the training set. Towards the goal of explaining all pixels in the scene, open-set or out-of-distribution detection has been studied in the classification settings [14, 15]. Typically a threshold is learned such that predictions below the threshold are classified as unknown. [15] proposed to use generative models to help calibrate the confidence level. Open-set recognition is also closely related to zero-shot learning [16, 17, 18]; however, the latter puts more emphasis on bootstrapping novel concepts from cross modality inputs (*e.g.* natural languages).

Recently, several approaches have been proposed to address the open-set instance segmentation scenario. [19] proposes a Bayesian framework that combines an instance segmentation network [20] for known classes with an of-the-shelf contour detection algorithm [21] for unknown classes.
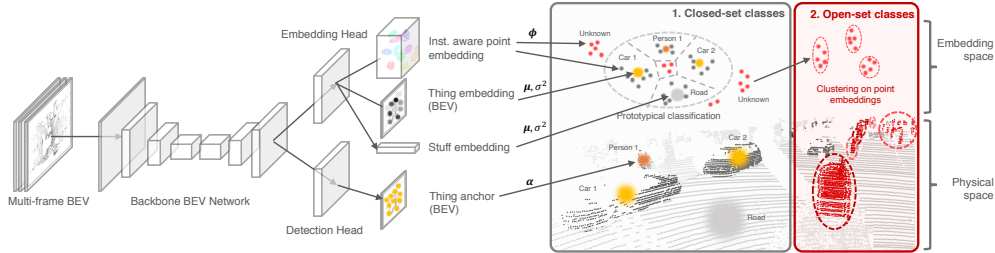
Figure 2: Our OSIS model contains two branches: a) a detection head to detect anchors of known classes; and b) an embedding head to extract instance-aware features. These instance aware features consist of: i) per-point embeddings; ii) thing embeddings for each anchor from a known thing class; and iii) stuff embeddings for the known stuff classes. The thing and stuff embeddings jointly filter out points from known classes in the first stage. In the second stage, we cluster the remaining unknown points into instances using their instance-aware embeddings.

This approach can potentially be limited by the capacity of the offline contour detection algorithm. [22] extends the standard instance segmentation task with thousands of extra visual concepts in the form of weak labels [23], covering a wide range of rare objects. This is, however, still closed-set recognition with weak labels. In [24], a category-agnostic object proposal network is trained and applied on video sequences. Due to its "proposal + classification" nature, the model may learn to suppress unknown objects that are present but not labeled in the training examples. In the 3D point clouds domain, [25] proposed to leverage connected components, which could be less robust to cluttered scenes.

Next we review existing literature on instance segmentation. One mainstream approach for instance segmentation is based on object detection boxes [20, 26, 27], where object segmentations are produced within detection boxes, referred to as "two-stage" joint detection and segmentation models. [28, 29, 30] output object instance proposals directly from each pixel. For 3D point cloud, [31, 32] also use similar two-stage architectures to perform point cloud detection and segmentation. As segmentation happens after detection, unknown objects are often left unrecognized. Unless the object detector is trained to recognize unknown classes, these approaches are likely unsuitable for our open-set instance segmentation problem.

Another line of work for instance segmentation is based on bottom-up grouping of pixels. [33] predicts energy of each instances and obtain instance segmentations using flood fill. [34, 35] cluster pixels by their predicted centroid locations. [36] predicts breaking points on vertical and horizontal directions and segment objects using line scanning. In 3D point cloud instance segmentation, several bottom-up approaches have also been proposed. [37] predicts point affinity to make segmentation proposals for each point. When the number of points is large, as in the case of LiDAR point clouds, this approach can end up with too many proposals to process.

Our method is most similar to a line of work in bottom-up segmentation that learns instance-aware embeddings [38, 39, 40, 41, 42, 43]. Our method and these approaches all use clustering (*e.g.* mean-shift [44], DBSCAN [45], *etc.*) to aggregate the points into instances based on their embedding similarities. Despite having a similar instance-aware embedding component, our proposed method is distinguished by two major differences. First, we leverage an object detection head to propose anchors against which points can be clustered, thus resulting in a more efficient and effective algorithm with top-down guidance. Second, we propose to directly predict prototypical features for each anchor to account for the spatial sparsity and non-uniform density LiDAR point clouds.

3   Identifying the Unknowns

In this paper, we propose the Open-Set Instance Segmentation (OSIS) network for identifying known and unknown objects from point clouds. In the following, we first formally define the problem of open-set instance segmentation in Sec. 3.1. Then, we discuss our full inference framework in Sec. 3.2. Finally, we provide details on how to train our model in Sec. 3.3.

3.1 Problem Formulation

Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ be an input set of N points, where each $\mathbf{x}_i \in \mathbb{R}^D$ is the input feature for point i. Given a set of instance ids $\mathbb{I}$ and a set of open-set semantic labels $\mathbb{O}$, we want a function f mapping each input feature $\mathbf{x}_i \in \mathcal{X}$ to a tuple $(y_i, z_i) \in \mathbb{I} \times \mathbb{O}$. Note that $\mathbb{O}$ may be partitioned into two disjoint subsets $\mathbb{C}$ and $\{\bot\}$, where $\mathbb{C}$ is the set of known classes and $\bot$ is the semantic label for the unknown class. The known classes $\mathbb{C}$ can be further divided into $\mathbb{C}_{\text{thing}}$ and $\mathbb{C}_{\text{stuff}}$, which correspond to the known thing classes (*e.g.*, vehicle and pedestrian) and the known stuff classes (*e.g.*, road) respectively. As in [13], we require that every point with the same instance id have the same semantic label. Furthermore, we ignore the instance ids of stuff points.

Our problem formulation differs from standard panoptic segmentation [13] with regards to how the unknown (void) class is handled. In the standard setting, we do not require instance labels for points with a void semantic label. By contrast, in our setting, we want to identify individual instances for the unknown class as well. Fig. 1 shows an example output for this task.

3.2 Open-Set Instance Segmentation

In this subsection, we describe our proposed approach for open-set instance segmentation. Our approach is based on learning a category-agnostic embedding space in which points can be clustered into instances irrespective of their semantics. To this end, we design a convolutional neural network that consists of three components: 1) a shared backbone feature extractor; 2) a detection head to detect anchors of known things; and 3) an embedding head to predict instance-aware features.

Our inference procedure consists of two stages. First, we perform closed-set perception by associating points to prototypes of known things and stuff using the learned embedding space. Next, we perform open-set perception by classifying points with uncertain associations as unknown, and then clustering them into instances using their instance-aware embeddings and 3D coordinates as features. We refer the reader to Fig. 2 for an illustration of our full inference pipeline.

Input representation: Our model takes as input a bird's eye view (BEV) rasterized image of a LiDAR point cloud $\mathcal{X} = \{(x_i, y_i, z_i)\}_{i=1}^N$ centered on the ego-car. Specifically, we voxelize $\mathcal{X}$ into a 3D occupancy grid using reversed trilinear interpolation [46] and treat its vertical axis as multi-dimensional features. This yields a compact yet effective representation of $\mathcal{X}$ on which we can use 2D convolutions [47]. Our model can also exploit temporal information by taking multiple BEV LiDAR frames stacked along the feature channel as input. To alleviate misalignment across frames due to the ego-car's movements, we use localization to compensate the ego-motion.

Backbone network: We use a custom 2D convolutional feature pyramid network to extract multi-scale features from the input BEV LiDAR frame. In this network, we stack several residual blocks to compute a feature hierarchy consisting of three scales of the input resolution: $1/4$, $1/8$, and $1/16$. These multi-scale features are then upsampled to the $1/4$ scale and fused via residual connections to output a $C \times H \times W$ feature map, where C is the number of feature channels, and H and W is the height and width of the feature map respectively. This feature map is subsequently used as input to the detection and embedding heads.

Detection head: Our detection head consists of four 3×3 convolution layers, followed by a 1×1 convolution layer. For each BEV pixel and for each class in $\mathbb{C}_{\text{thing}}$, it predicts $(\alpha, dx, dy, w, l, \sin(2\theta), \cos(2\theta))$, where α is the anchor confidence score, (dx, dy) is the position offsets to the its object center, and the rest parameterize the geometry of its bounding box [48]. During inference, we remove anchors with scores less than τ to obtain the set of anchors $\mathcal{A}_\tau$. Note that the bounding box parameters are predicted only to exploit additional supervision signals.

Embedding head: The embedding head forms the core of our open-set instance segmentation model: it learns a category-agnostic embedding space in which points can be clustered into instances irrespective of their semantics. Specifically, the embedding head is a four-layer CNN with 3×3 filters followed by three distinct branches:

1. The point branch computes features $\Phi_{\text{point}} \in \mathbb{R}^{(F \times Z) \times H \times W}$ via a 1×1 convolution, where F is the dimension of the embedding space, and Z is the number of bins along the gravitational z-axis. For each point i in $\mathcal{X}$, we extract an embedding ϕ_i from Φ_{point} via trilinear interpolation.

2. The thing branch computes features $\Phi_{\text{thing}} \in \mathbb{R}^{(F+1) \times H \times W}$ via a 1×1 convolution. For each anchor k in $\mathcal{A}_\tau$, we extract its prototype $(\boldsymbol{\mu}_k, \sigma_k^2) \in \mathbb{R}^F \times \mathbb{R}$ by bilinearly interpolating Φ_{thing} around the anchor's object center. This yields a set of thing prototypes $\mathcal{P}_{\text{thing}}$.

3. The stuff branch performs global average pooling to obtain features $\Phi_{\text{stuff}} \in \mathbb{R}^{C \times 1 \times 1}$. For each stuff class $c \in \mathbb{C}_{\text{stuff}}$, we apply a linear layer on Φ_{stuff} to predict its prototype $(\boldsymbol{\mu}_c, \sigma_c^2) \in \mathbb{R}^F \times \mathbb{R}$. This yields a set of stuff prototypes $\mathcal{P}_{\text{stuff}}$.

Closed-set perception: Our closed-set perception algorithm draws inspiration from prototypical networks for few-shot learning [49]. First, we apply non-maximum suppression to $\mathcal{P}_{\text{thing}}$ to obtain a unique set of thing prototypes $\mathcal{P}'_{\text{thing}}$. Let us denote $\mathcal{P}_{\text{all}} = \mathcal{P}'_{\text{thing}} \cup \mathcal{P}_{\text{stuff}}$ as the final set of all thing and stuff prototypes. Then, given a point i in $\mathcal{X}$, we compute its point-to-prototype association score with respect to every prototype k in $\mathcal{P}_{\text{all}}$ as follows:

$$\hat{\boldsymbol{y}}_{i,k} = -\frac{\|\boldsymbol{\phi}_i - \boldsymbol{\mu}_k\|^2}{2\sigma_k^2} - \frac{F}{2}\log\sigma_k^2 \tag{1}$$

Additionally, we have a learnable global constant U corresponding to its score $\hat{\boldsymbol{y}}_{i,|\mathcal{P}_{\text{all}}|+1}$ of not associating with any prototype in $\mathcal{P}_{\text{all}}$. Thus, its instance label can be computed by taking the argmax over its association scores $\hat{\boldsymbol{y}}_i$. Furthermore, its semantic label is simply the class of its instance, or unknown if it is not associated with any prototypes in $\mathcal{P}_{\text{all}}$. Note that, in practice, we compute each point's scores only with the prototypes of its k-nearest thing anchors and all $|\mathbb{C}_{\text{stuff}}|$ stuff classes; this helps to accelerate inference speed.

Identifying unknown instances: We assign instance labels to unknown points via DBSCAN [45] clustering. Specifically, for two points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$, their pairwise distance used in DBSCAN is a convex combination of their point embedding squared distance and their 3D location squared distance; *i.e.*,

$$d_{ij}^2 = \beta\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2 + (1-\beta)\|\boldsymbol{\phi}_i - \boldsymbol{\phi}_j\|^2 \tag{2}$$

Combining the instance labels obtain from this stage with results from closed-set perception, we obtain our final open-set instance segmentation predictions.

3.3 Learning

Our model is optimized with respect to a combination of detection and embedding losses:

$$\mathcal{L} = \lambda_{\text{det}}\ell_{\text{det}} + \lambda_{\text{emb}}\ell_{\text{emb}} \tag{3}$$

where ℓ_{det} is the detection loss, ℓ_{emb} is the embedding loss, and λ's are their associated loss weights. In our experiments, we set λ's to 1. Since $\mathcal{L}$ is fully differentiable with respect to the network parameters, we train our model using the standard back-propagation algorithm.

Detection loss: We use a standard multi-task loss function to train the detection head. In particular, for object classification, we use binary cross-entropy with online negative hard mining, where positive and negative BEV pixels are determined by their distances to an object center [48]. For bounding box regression, we use a combination of IoU loss for box locations and sizes and SmoothL1 loss for box orientations on predictions at positive pixels. It is worth noting that box sizes and orientations are not used during inference, and we predict them only for a stronger supervision signal.

Embedding loss: We use a standard cross-entropy loss function to encourage points to be assigned to the correct prototype. In particular, during training we first gather a set of prototypes $\mathcal{P}_{\text{gt}}$, which is the union of $\mathcal{P}_{\text{stuff}}$ and the set of thing prototypes obtained by bilinearly interpolating Φ_{thing} around ground truth object centers. Next, we compute point-to-prototype association scores $\{\hat{\boldsymbol{y}}_i\}_{i=1}^N$ with respect to $\mathcal{P}_{\text{gt}}$, and normalize each $\hat{\boldsymbol{y}}_i$ using the softmax function. Finally, we calculate the cross-entropy loss as follows:

$$\ell_{\text{proto}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{|\mathcal{P}_{\text{gt}}|+1}\boldsymbol{y}_{i,k}\log\hat{\boldsymbol{y}}_{i,k} \tag{4}$$

where each $\boldsymbol{y}_i$ is a one-hot vector indicating ground truth associations. We also apply a discriminative loss function [38] on the point embeddings $\{\boldsymbol{\phi}_i\}_{i=1}^N$, which we found improves performance.
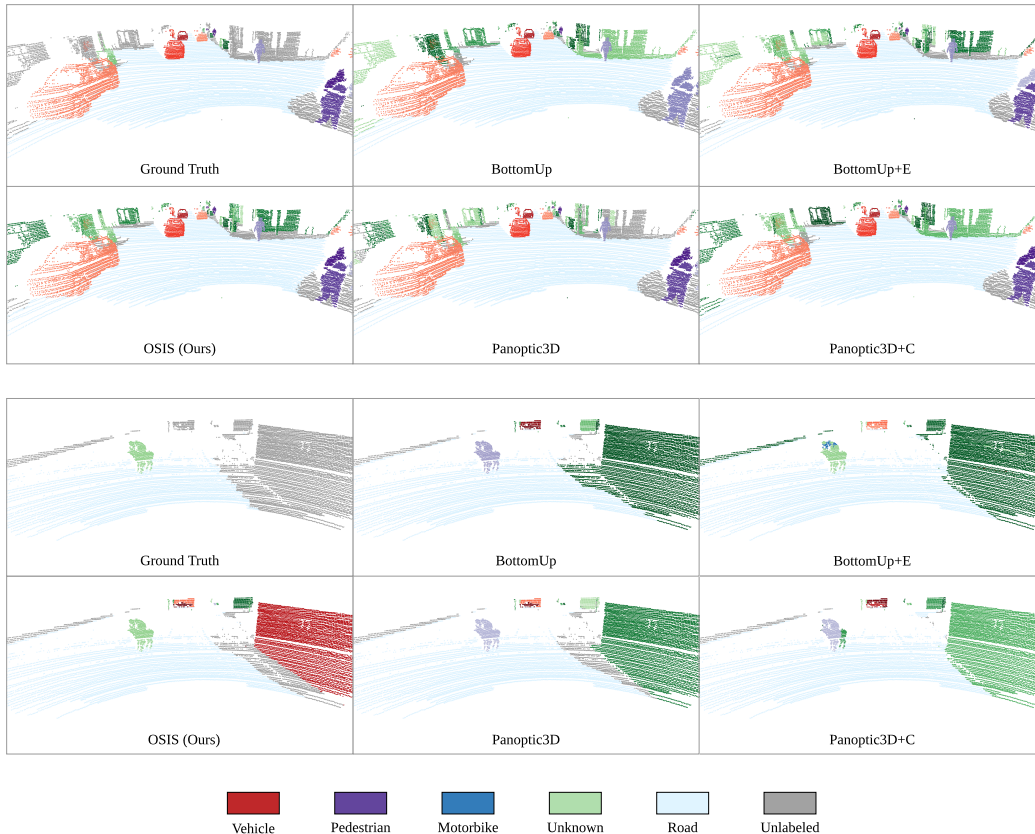
Figure 3: Qualitative results of open-set instance segmentation on TOR4D and Rare4D.

| | Rare4D | | | TOR4D | | | | | | | | |
| | Unknown | | | Unknown | | | Known Thing | | | Known Stuff | | |
	UQ	RQ	SQ	UQ	RQ	SQ	PQ	RQ	SQ	PQ	RQ	SQ
MT-PNet [40]	10.9	11.9	91.9	27.7	31.6	87.8	27.6	29.4	91.9	94.4	100.0	94.4
BottomUp [47]	48.7	53.7	90.7	49.2	54.8	89.8	56.0	60.1	92.8	**98.2**	100.0	**98.2**
BottomUp+E [47]	57.9	62.8	91.0	62.7	69.1	90.7	64.1	67.2	94.9	**98.2**	100.0	**98.2**
Panoptic3D [13]	41.7	49.2	84.8	43.8	51.9	84.4	74.5	77.2	96.3	**98.2**	100.0	**98.2**
Panoptic3D+C [13]	50.8	56.2	90.3	51.5	57.3	89.9	78.8	81.4	**96.6**	97.9	100.0	97.9
OSIS (Ours)	**62.5**	**66.5**	**94.0**	**66.0**	**71.3**	**92.5**	**81.5**	**84.3**	**96.6**	97.7	100.0	97.7

Table 1: Open-set panoptic segmentation results on the TOR4D and Rare4D test sets.

4 Experiments

In this section, we showcase the effectiveness of our proposed model OSIS on two large-scale self-driving datasets. We first describe our experimental setup and then discuss the results we obtained.

4.1 Experimental Setup

Datasets:

- **TOR4D** [48] is a large-scale self-driving dataset collected from cities across North America. This dataset consists of 6500 distinct driving scenarios, each containing 250 sweeps of LiDAR point clouds. We partition TOR4D into a training set of 5000 scenarios, a validation set of 500, and a test set of 1000. Furthermore, we subsample every five frames across all three splits.

6

Each frame in TOR4D is annotated with per-point semantic and instance labels according to four classes: vehicle, pedestrian, motorbike, and road. Points not belonging to one of those classes are unlabled and regarded as unknown. To evaluate OSIS in the open-set setting, we annotate 5,702 and 10,127 unique unknown objects with instance labels in the validation and test sets respectively.

- **Rare4D** is a dataset of curated self-driving scenarios containing 289 unique rare objects such as forklifts, tractors, and even horses (see Fig. 1). In our experiments, Rare4D is not used for training but for evaluation of unknown object identification only.

Evaluation metrics: For known classes, we report the panoptic quality (PQ), recognition quality (RQ), and segmentation quality (SQ) metrics proposed in [13]. Since the labels in our dataset consider only things that are removeable to be separate objects (*e.g.*, flags attached to a building will not be labeled), we decide not to measure precision; instead, we modify PQ into the *unknown quality* (UQ), a recall-based metric that measures performance on annotated instances only:

$$\text{UQ} = \underbrace{\frac{\sum_{(p,g)\in\text{TP}}\text{IoU}(p,g)}{|\text{TP}|}}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|\text{TP}|}{|\text{TP}|+|\text{FN}|}}_{\text{recall quality (RQ)}} \tag{5}$$

where TP is the set of true positives and FN is the set of false negatives. As in [13], a predicted unknown instance p matches with the ground truth unknown instance g if and only if their intersection over union exceeds 0.5.

Baselines: Due to a lack of prior work in open-set instance segmentation for point clouds, we adapt several deep learning based instance segmentation algorithms to the open-set setting to serve as baselines. Note that all baselines except for MT-PNet use the same backbone network and input representations.

- **MT-PNet** [40] is a state-of-the-art joint 3D semantic and instance segmentation algorithm[1]. We adapt MT-PNet to the open-set setting as follows: 1) we augment its semantic header to predict an additional unknown class; and 2) we use DBSCAN [45] to cluster unknown points into instances based on their embedding distances.

- **BottomUp** first runs a state-of-the-art point cloud semantic segmentation algorithm [47] with an additional unknown class, and then uses DBSCAN [45] to cluster points of the same class into instances. We evaluate two versions of this baseline: 1) **BottomUp** clusters points using their 3D locations; and 2) **BottomUp+E** clusters points using embeddings learned via a discriminative loss function [38].

- **Panoptic3D** is similar to the pioneering panoptic segmentation algorithm proposed for 2D images [13]. We first perform 3D detection and segmentation, and then apply heuristics to merge the outputs into a panoptic segmentation of the scene. Unlike [13], we train a single network with both a 3D detection and a semantic segmentation header. Similar to BottomUp, Panoptic3D also predicts an additional unknown class. We compare two versions of this baseline: 1) **Panoptic3D** performs class-agnostic detections; and 2) **Panoptic3D+C** performs class-aware detections and uses DBSCAN to cluster unknown points into instances based on their 3D locations.

Implementation details: In our BottomUp, Panoptic3D, and OSIS experiments, we use a $160 \times 160 \times 5$ meters region of interest centered on the ego-vehicle. Points within this area are rasterized into a BEV image using reversed trilinear interpolation [46] at a discretization resolution of 0.15625. We use five frames of LiDAR as input and align them using ego-motion. This yields an input tensor of size $C \times H \times W = 160 \times 1024 \times 1024$. We use the Adam optimizer [50] with a batch size of 32 and an initial learning rate of $4\text{e}{-}3$, which we decay by 0.1 after every five epochs for a total of ten epochs. Note that experiments for MT-PNet follow a similar setup, with the exception that we feed raw LiDAR point clouds as input to the model.

4.2 Results

As shown in Tab. 1, OSIS outperforms the baselines on known and unknown things on all metrics across both datasets. Our method is also comparable to state-of-the-art semantic segmentation mod-

---

[1]The CRF post-processing stage is not included.

DL	BR	σ^2	Unknown			Known Things			Known Stuff		
			UQ	RQ	SQ	PQ	RQ	SQ	PQ	RQ	SQ
			59.8	68.4	87.5	78.2	81.9	95.0	**97.7**	100.0	**97.7**
✓			68.1	73.2	93.0	80.6	83.9	95.8	**97.7**	100.0	**97.7**
✓	✓		68.5	**73.8**	92.8	81.9	85.0	96.2	**97.7**	100.0	**97.7**
✓	✓	✓	**68.6**	73.7	**93.1**	**82.6**	**85.5**	**96.5**	**97.7**	100.0	**97.7**

Table 2: Ablation study of model components on TOR4D validation set.

	Unknown w/ Oracle		
	UQ	RQ	SQ
Points	58.1	63.9	91.0
Center	73.7	78.6	**93.8**
Semantic	16.1	18.9	85.3
Embedding	70.3	75.8	92.8
Embedding + Points	**74.3**	**79.6**	93.3

Table 3: Unknown instance segmentation performance using different features for clustering. "w/ oracle" indicates that we use ground truth to remove known points prior to DBSCAN.
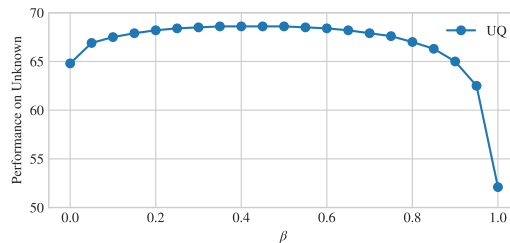
Figure 4: Ablation study on varying the relative weight β of using 3D location distance versus embedding distance for identifying unknown instances.

els for known stuff classes. Interestingly, BottomUp+E is the best baseline for unknown things while Panoptic3D+C is the best baseline for known things. As our results suggest, OSIS achieves the best of both worlds by marrying a bottom-up approach with top-down guidance.

Qualitative results in Fig. 3 further higlight our method's ability to correctly segment instances from both known and unknown classes. In particular, OSIS is the sole method that correctly segmented the horse and identified it as an unknown object; by contrast, the baseline methods suffer from misclassification errors and noise in instance segmentation.

4.3 Ablation Studies

Model design choices: We first conduct an ablation study on three components of our model: 1) whether we optimize the discriminative loss (**DL**); 2) whether we perform bounding box regression (**BR**); and 3) whether we predict per-prototype scalar variances (σ^2). Tab. 2 shows our results on the TOR4D validation set. From this table, we can see that all three components contribute towards the overall performance of our model.

Effectiveness of instance-aware embeddings: We also study the effectiveness of using instance-aware embeddings to group unknown points into instances. In particular, we compare our embeddings against other per-point features, namely 3D location (**Points**), predicted instance center (**Center**), and semantic features (**Semantics**). From Tab. 3, we see that our instance-aware embeddings acheive the best results among the alternatives. Fig. 4 also indicates that using a combination of instance-aware embeddings and geometry features yields further improvements.

5 Conclusion

We have presented a novel and effective open-set instance segmentation method for point clouds. In particular, we proposed a deep convolutional neural network to encode points into a category-agnostic embedding space in which they can be clustered into instances. As a result, our method is able to perceptually group points into instances, irrespective of whether they belong to a known or unknown class. We validate our method on two large-scale self-driving datasets and achieve state-of-the-art performance in the open-set setting. In the future we plan to explicitly reason about motion as a cue for better instance segmentation of moving objects.

References

[1] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *IJRR*, 2008.

[2] A. Ennaceur and J. Delacour. A new one-trial test for neurobiological studies of memory in rats. 1: Behavioral data. *Behavioural brain research*, 31(1):47–59, 1988.

[3] M. Antunes and G. Biala. The novel object recognition memory: neurobiology, test procedure, and its modifications. *Cognitive processing*, 2012.

[4] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *TPAMI*, 2013.

[5] A. Bendale and T. E. Boult. Towards open set deep networks. In *CVPR*, 2016.

[6] S. E. Palmer. Organizing objects and scenes. *Foundations of cognitive psychology: Core readings*, 2002.

[7] S. E. Palmer. *Vision science: Photons to phenomenology*. MIT press, 1999.

[8] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[10] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[11] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[13] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In *CVPR*, 2019.

[14] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

[15] K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

[16] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[17] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[18] Y. Xian, Z. Akata, G. Sharma, Q. N. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016.

[19] T. Pham, B. G. V. Kumar, T. Do, G. Carneiro, and I. D. Reid. Bayesian semantic instance segmentation in open set world. In *ECCV*, 2018.

[20] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. In *ICCV*, 2017.

[21] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

[22] R. Hu, P. Dollár, K. He, T. Darrell, and R. B. Girshick. Learning to segment every thing. In *CVPR*, 2018.

[23] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[24] A. Osep, P. Voigtlaender, M. Weber, J. Luiten, and B. Leibe. 4d generic video object proposals. *CoRR*, abs/1901.09260, 2019.

[25] A. J. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen. Efficient organized point cloud segmentation with connected components. *Semantic Perception Mapping and Exploration (SPME)*, 2013.

[26] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *CVPR*, 2017.

[27] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.

[28] X. Chen, R. B. Girshick, K. He, and P. Dollár. Tensormask: A foundation for dense object segmentation. *CoRR*, abs/1903.12174, 2019.

[29] P. H. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *NIPS*, pages 1990–1998, 2015.

[30] P. O. Pinheiro, T. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016.

[31] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in RGB-D images. In *CVPR*, 2016.

[32] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *CVPR*, 2018.

[33] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

[34] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan. Proposal-free network for instance-level object segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2978–2991, 2018.

[35] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *CVPR*, 2019.

[36] S. Liu, J. Jia, S. Fidler, and R. Urtasun. SGN: sequential grouping networks for instance segmentation. In *ICCV*, 2017.

[37] W. Wang, R. Yu, Q. Huang, and U. Neumann. SGPN: similarity group proposal network for 3d point cloud instance segmentationeccv. In *CVPR*, 2018.

[38] B. D. Brabandere, D. Neven, and L. V. Gool. Semantic instance segmentation with a discriminative loss function. *CoRR*, abs/1708.02551, 2017.

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

[40] Q. Pham, D. T. Nguyen, B. Hua, G. Roig, and S. Yeung. JSIS3D: joint semantic-instance segmentation of 3d point clouds with multi-task pointwise networks and multi-value conditional random fields. In *CVPR*, 2019.

[41] C. Elich, F. Engelmann, J. Schult, T. Kontogianni, and B. Leibe. 3D-BEVIS: birds-eye-view instance segmentation. *CoRR*, abs/1904.02199, 2019.

[42] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance segmentation via deep metric learning. *CoRR*, abs/1703.10277, 2017.

[43] S. Kong and C. C. Fowlkes. Recurrent pixel embedding for instance grouping. In *CVPR*, pages 9018–9028, 2018.

[44] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8): 790–799, 1995.

[45] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. Density-based spatial clustering of applications with noise. In *KDD*, 1996.

[46] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *3DV*, 2017.

[47] C. Zhang, W. Luo, and R. Urtasun. Efficient convolutions for real-time semantic segmentation of 3d point clouds. In *3DV*, 2018.

[48] B. Yang, M. Liang, and R. Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *CoRL*, 2018.

[49] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017.

[50] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.