Novel Class Discovery for 3D Point Cloud Semantic Segmentation

Luigi Riz[1] Cristiano Saltori[2] Elisa Ricci[1,2] Fabio Poiesi[1]

[1]Fondazione Bruno Kessler [2]University of Trento

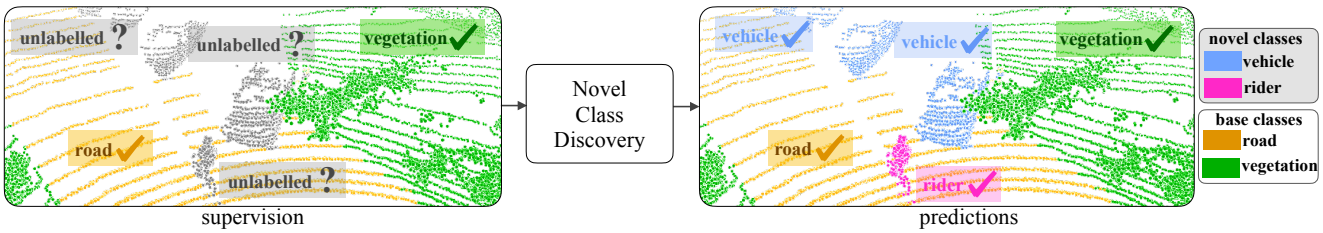

Figure 1. Novel Class Discovery for 3D point cloud semantic segmentation seeks to recognise novel classes by clustering unlabelled novel points with similar semantic features and by exploiting only the knowledge of a set of labelled samples corresponding to the base classes.

Abstract

Novel class discovery (NCD) for semantic segmentation is the task of learning a model that can segment unlabelled (novel) classes using only the supervision from labelled (base) classes. This problem has recently been pioneered for 2D image data, but no work exists for 3D point cloud data. In fact, the assumptions made for 2D are loosely applicable to 3D in this case. This paper is presented to advance the state of the art on point cloud data analysis in four directions. Firstly, we address the new problem of NCD for point cloud semantic segmentation. Secondly, we show that the transposition of the only existing NCD method for 2D semantic segmentation to 3D data is sub-optimal. Thirdly, we present a new method for NCD based on online clustering that exploits uncertainty quantification to produce prototypes for pseudo-labelling the points of the novel classes. Lastly, we introduce a new evaluation protocol to assess the performance of NCD for point cloud semantic segmentation. We thoroughly evaluate our method on SemanticKITTI and SemanticPOSS datasets, showing that it can significantly outperform the baseline. Project page: https://github.com/LuigiRiz/NOPS.

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101058589. This work was also partially supported by the PRIN project LEGO-AI (Prot. 2020TA3K9N), the EU ISFP PROTECTOR (101034216) project and the EU H2020 MARVEL (957337) project and, it was carried out in the Vision and Learning joint laboratory of FBK and UNITN.

1. Introduction

As humans, we are fairly skilled in organising new visual knowledge (novelty) into homogeneous groups, even when we do not know what we are observing. However, machines cannot perform this task satisfactorily without our supervision. The challenge here is mainly in the formulation of discriminative latent representations of the real world and in the quantification of the uncertainty when the novelty is observed [16,40,41]. The work of Han et al. [16] pioneered the formulation of the Novel Class Discovery (NCD) problem by defining it as the task that aims to classify the samples of an unlabelled dataset into different classes, i.e. the *novel samples*, by exploiting the knowledge of a set of labelled samples, i.e. the *base samples*. Note that the classes in the labelled and unlabelled datasets are disjoint.

NCD has been explored in the 2D image domain for classification [12, 16, 41] and later for semantic segmentation [40]. In particular, Zhao et al. [40] presented the first approach for NCD for 2D semantic segmentation. Two key assumptions were made by the authors. Firstly, at most one novel class is allowed in each image. Secondly, the new class belongs to a foreground object that can be found via saliency detection (e.g. a man on a bicycle, where the bicycle is the novel class). Thanks to these assumptions, the authors could pool the features of each image into a single latent representation and cluster the representations of the whole dataset to discover clusters of novel classes. We argue that these two are important constraints that are not applicable to generic 3D data, in particular to point clouds captured with LiDAR sensors in real-world city-scale scenarios. One point cloud can contain more than one novel class, and the saliency for 3D data cannot be leveraged in the same way as that for 2D data. Although they are both

related to the attraction of human fixations, 3D saliency is more related to the regional importance of 3D surfaces rather than the foreground/background distinction [30]. Our motivation in exploring NCD for the 3D setting is mainly driven by addressing these shortcomings.

In this paper, we explore the new problem of NCD for 3D point cloud semantic segmentation (see Fig. 1). Given a partially human-annotated dataset, we jointly learn base and novel semantic classes by clustering unlabelled points with similar semantic features. We adapt the method of Zhao et al. [40] (Entropy-based Uncertainty Modeling and Self-training - EUMS) for point cloud data and use it as our baseline. We go beyond their formulation and, inspired by [5], we integrate batch-level (online) clustering in our method and update prototypes during training in order to make clustering computationally tractable. Cluster assignments are then used as training pseudo-labels. We also exploit over-clustering to achieve a higher clustering accuracy as in EUMS. Because point clouds contain multiple semantic classes, we cannot guarantee that all the classes appear in the point clouds within each batch, some will be missing. Therefore, we design a queuing strategy to store important features over training time, which will be used for pseudo-labelling in the case of missing classes. We further introduce a strategy for exploiting the pseudo-label uncertainty to promote the creation of reliable prototypes that we then exploit to produce higher-quality pseudo-labels. Lastly, we produce two augmented views of the same point cloud and impose pseudo-label consistency amongst them. We evaluate our approach on SemanticKITTI [3, 4, 13] and SemanticPOSS [24], introducing an evaluation protocol for NCD and point cloud segmentation that can be adopted in future works. We empirically show that our approach largely outperforms our baseline in both datasets. We also perform an extensive ablation study to demonstrate the importance of the different components of our method.

To summarise, our contributions are:

- We address the new problem of NCD for 3D semantic segmentation;
- We show that the transposition of the only existing NCD method for 2D semantic segmentation [40] to 3D data is suboptimal;
- We present a new method for NCD based on online clustering and uncertainty estimation, which we name it NOPS (NOvel Point Segmentation);
- We introduce a new evaluation protocol to assess the performance of NCD for 3D semantic segmentation.

2. Related work

Point cloud semantic segmentation can be performed at the point level [26], on range view maps [27], and by voxelising the input points [43]. Point-level networks process the input without intermediate representations. Examples of these include PointNet [25], PointNet++ [26], RandLA-Net [17], and KPConv [33]. PointNet [25] and PointNet++ [26] are based on a series of multi-layer perceptron where PointNet++ introduces global and local feature aggregation at multiple scales. RandLA-Net [17] uses random sampling, attentive pooling, and local spatial encoding. KPConv [33] employs flexible and deformable convolutions in a continuous input space. Point-level networks are computationally inefficient when large-scale point clouds are processed. Range view architectures [23] and voxel-based approaches [10] are more computationally efficient than their point-level counterpart. The former requires projecting the input points on a 2D dense map, processing input maps with 2D convolutional filters [27], and re-projecting predictions to the initial 3D space. SqueezeSeg networks [35, 36], 3D-MiniNet [1], RangeNet++ [23], and PolarNet [39] are examples of this category. Although they are more efficient, these approaches tend to lose information during the projection and re-projection phase. The latter includes 3D quantisation-based approaches that discretise the input points into a 3D voxel grid and employ 3D convolutions [43] or 3D sparse convolutions [10, 15] to predict per-voxel classes. VoxelNet [43], SparseConv [14, 15], MinkowskiNet [10], Cylinder3D [44], and $(AF)^2$-S3Net [9] are architectures belonging to this category. These approaches tackle point cloud segmentation in the supervised settings, whereas we tackle novel class discovery with labelled base classes and unlabelled novel classes.

Novel class discovery (NCD) is explored for 2D classification [12, 16, 19, 20, 28, 34, 37, 41, 42] and 2D segmentation [40]. NCD is more complex than standard semi-supervised learning [31, 32, 38]. In semi-supervised learning, labelled and unlabelled samples belong to the same classes, while in NCD, novel and base samples belong to disjoint classes. Han et al. [16] pioneered the NCD problem for 2D image classification. A classification model is pre-trained on a set of base classes and used as feature extractor for the novel classes. They then train a classifier for the novel classes using the pseudo-labels produced by the pre-trained model. Zhong et al. [41] introduced neighbourhood contrastive learning to generate discriminative representations for clustering. They retrieve and aggregate pseudo-positive pairs with contrastive learning, encouraging the model to learn more discriminative representations. Hard negatives are obtained by mixing labelled and unlabelled samples in the feature space. UNO [12] unifies the two previous works by using a unique classification loss function for both base and novel classes, where pseudo-labels are processed together with ground-truth labels. NCD without Forgetting [20] and FRoST [28] further extend NCD to the incremental learning setting. EUMS [40] is the only approach analysing the NCD problem for semantic segmen-

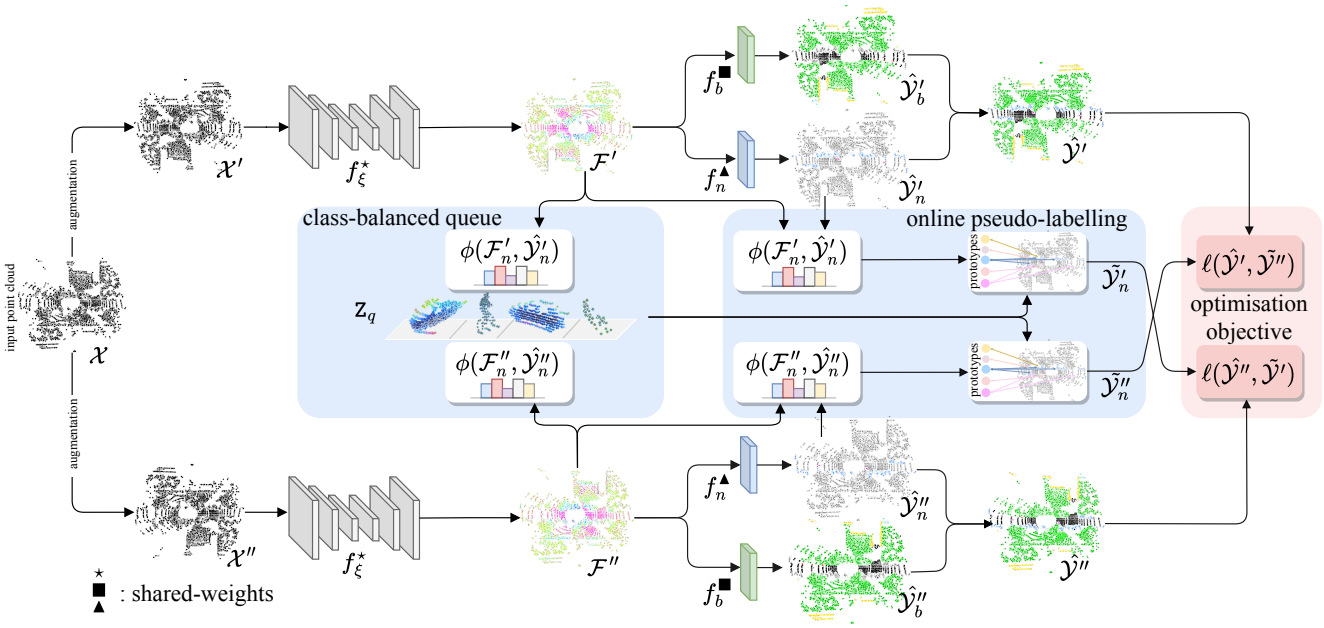

Figure 2. Overview of NOPS. We random augment the input point cloud twice and extract point-level features $\mathcal{F}$ with the shared model f_ξ. $\mathcal{F}$ are used to obtain pseudo-labels in the online pseudo-labelling. We forward $\mathcal{F}$ to a novel f_n and a base f_b segmentation layer to output the novel and base predictions, respectively. We optimise our network by minimising a global objective function based on cross entropy.

tation. Unlike image classification, the model has to classify each pixel and handle multiple classes in each image. EUMS consists of a multi-stage pipeline using a saliency model to cluster the latent representations of novel classes to produce pseudo-labels. Moreover, entropy-based uncertainty and self-training are used to overcome noisy pseudo-labels while improving the model performance on the novel classes. In this work, we tackle the problem of NCD in 3D point cloud semantic segmentation. Unlike previous works, our problem inherits the challenges from the fields of 2D semantic segmentation [7, 8] and 3D point cloud segmentation [10, 23, 29]. From 2D semantic segmentation, it inherits the additional challenges of multiple novel classes in the same image and the strong class unbalance. From 3D point cloud segmentation, we inherit the sparsity of input data, the different density of point cloud regions and the inability to identify foreground and background. The latter are not present in 2D segmentation [40]. Unlike [40] that use K-Means, we formulate clustering as an optimal transport problem to avoid degenerate solutions (i.e. all data points may be assigned to the same label and learn a constant representation) [2, 22]. Lastly, related to EUMS, REAL is proposed for open-world 3D semantic segmentation [6], where both known and unknown points have to be segmented. Unlike NOPS, all the unknown points belong to a single class and it is the task of a human annotator to separately label the novel classes. Then, these labels are used to update the base model by incrementally learning the novel classes.

3. Proposed approach

3.1. Overview

Given an input point cloud, we produce two augmented views that are processed with the same deep neural network to extract point-level features. These features are used to obtain pseudo-labels in the *online pseudo-labelling* step through the Sinkhorn-Knopp algorithm [11] (Sec. 3.3). Concurrently, we process the same features with the last network layers to segment novel and base classes. These features are stored in the *class-balanced queue* to mitigate the problem of batches with missing classes (Sec. 3.4). We exploit pseudo-label values (class probabilities) to filter out uncertain points, thus adding to the queue only high-quality points (Sec. 3.5). Lastly, we train our network by minimising the *optimisation objective* function through a swapped prediction task based on the computed pseudo-labels (Sec. 3.6). Fig. 2 shows the block diagram of NOPS.

3.2. Problem formulation

Let $\mathrm{X} = \{\mathcal{X}\}$ be a dataset of 3D point clouds captured in different scenes. $\mathcal{X}$ is a set composed of a base set $\mathcal{X}_b$ and a novel set $\mathcal{X}_n$, s.t. $\mathcal{X} = \mathcal{X}_b \cup \mathcal{X}_n$. The semantic categories that can be present in our point clouds are $\mathcal{C} = \mathcal{C}_b \cup \mathcal{C}_n$, where $\mathcal{C}_b$ is the set of base classes and $\mathcal{C}_n$ is the set of novel classes, s.t. $\mathcal{C}_b \cap \mathcal{C}_n = \emptyset$. Each $\mathcal{X} \in \mathrm{X}$ is composed of a finite but unknown number of 3D points $\mathcal{X} = \{(\mathbf{x}, c)\}$, where $\mathbf{x} \in \mathbb{R}^3$ is the coordinate of the a point and c is its semantic class. We know the class of the point $(\mathbf{x}, c)$, s.t. $\mathbf{x} \in \mathcal{X}_b$ and

$c \in \mathcal{C}_b$, but we do not know the class of the point $(\mathbf{x}, c)$, s.t. $x \in \mathcal{X}_n$ and $c \in \mathcal{C}_n$. No points in $\mathcal{X}_n$ belong to one of the base classes $\mathcal{C}_b$. As in [16, 40, 41], we assume that the number of classes to discover is known, i.e. $|\mathcal{C}_n| = C_n$. We aim to design a computational approach that trains a deep neural network $f_{\mathbf{\Theta}}$ that can segment all the points of a given point cloud, thus learning to jointly segment base classes $\mathcal{C}_b$ and novel classes $\mathcal{C}_n$. $\mathbf{\Theta}$ are the weights of our deep neural network. $f_{\mathbf{\Theta}}$ is composed of two heads, $f_{\mathbf{\Theta}} = f_\xi \circ \{f_b, f_n\}$, where f_b is the segmentation head for the base classes, f_n is the segmentation head for the novel classes, f_ξ is the feature extractor network and $\circ$ is the composition operator (Fig. 2).

3.3. Online pseudo-labelling

We formulate pseudo-labelling as the assignment of novel points to the class-prototypes learnt during training [5]. Let $\mathtt{P} \in \mathbb{R}^{D \times \rho}$ be the class prototypes, where D is the size of the output features from f_ξ and ρ is the number of prototypes. Let $\mathtt{Z} \in \mathbb{R}^{D \times m}$ be the normalised output features extracted from f_ξ, where m is the number of points of the point cloud. m it is not known a priori and it can differ across point clouds. We aim to find the assignment $\mathtt{Q} \in \mathbb{R}^{\rho \times m}$ s.t. all the points in the batch are equally partitioned across the ρ prototypes. This equipartition ensures that the feature representations of the points belonging to different novel classes are well-separated, thus preventing the case in which the novel class feature representations collapse into a unique solution. Caron et al. [5] employs an arbitrary large number of prototypes ρ to effectively organise the feature space produced by f_ξ. They discard $\mathtt{P}$ after training. In contrast, we learn exactly $\rho = C_n$ class prototypes and propose to use $\mathtt{P}$ as the weights for our new class segmentation head f_n, which outputs the C_n logits for the new classes. In order to optimise the assignment $\mathtt{Q}$, we maximise the similarity between the features of the new points and the learned prototypes as

$$\max_{\mathtt{Q} \in \mathcal{Q}} \ \mathrm{Tr}(\mathtt{Q}^\top \mathtt{P}^\top \mathtt{Z}) + \epsilon H(\mathtt{Q}) \rightarrow \mathtt{Q}^*, \tag{1}$$

where H is the entropy function, ϵ is the parameter that determines the smoothness of the assignment and $\mathtt{Q}^*$ is our sought solution. Asano et al. [2] enforce the equipartioning constraint by requiring $\mathtt{Q}$ to belong to a transportation polytope and perform this optimisation on the whole dataset at once (offline). This operation with point cloud data is computationally impractical. Therefore, we formulate the transportation polytope such that the optimisation is performed online, which consist of processing only the points within the batch being processed

$$\mathcal{Q} = \left\{ \mathtt{Q} \in \mathbb{R}_+^{C_n \times m} | \mathtt{Q} \mathtt{1}_m = \frac{1}{C_n} \mathtt{1}_{C_n}, \mathtt{Q}^\top \mathtt{1}_{C_n} = \frac{1}{m} \mathtt{1}_m \right\}, \tag{2}$$

where $\mathtt{1}_\star$ represents a vector of ones of dimension $\star$. These constraints ensure that each class prototype is selected on average at least m/C_n times in each batch. The solution $\mathtt{Q}^*$ can take the form of a normalised exponential matrix

$$\mathtt{Q}^* = \mathrm{diag}(\alpha) \exp\left(\frac{\mathtt{P}^\top \mathtt{Z}}{\epsilon}\right) \mathrm{diag}(\beta), \tag{3}$$

where α and β are renormalization vectors that are computed iteratively with the Sinkhorn-Knopp algorithm [11, 21]. We then transpose the optimised soft assignment $\mathtt{Q}^* \in \mathbb{R}_+^{C_n \times m}$ to obtain the soft pseudo-labels for each of the m novel points being processed within each batch.

We empirically found that training can be more effective if pseudo-labels are smoother in the first training epochs and peaked in the last training epochs. Therefore, we introduce a linear decay of ϵ during training.

Multi-headed segmentation: A single segmentation head may converge to a suboptimal feature space, thus producing suboptimal prototype solutions. To further improve the segmentation quality, we use multiple novel class segmentation heads to optimise f_Θ based on different training solutions. Different solutions increase the likelihood of producing a diverse partitioning of the feature space as they regularise with each other (they share the same backbone) [18]. In practise, we concatenate the logits of the base class segmentation head with the outputs of each novel class segmentation head and we separately evaluate their loss for each novel class segmentation head at training time.

We task our network to over-cluster novel points, using segmentation heads that output $o \cdot C_n$ logits, where o is the over-clustering factor. Previous studies empirically showed that this is beneficial to learn more informative features [5, 12, 18, 22]. We observed the same and concur that over-clustering can be useful for increasing expressivity of the feature representations. The over-clustering heads are then discarded at inference time.

3.4. Class-balanced queuing

Soft pseudo-labelling described in Sec. 3.3 produces an equipartite matching between the novel points and the class centroids. However, it is highly likely that batches are sampled with point clouds containing novel classes with different cardinalities when dealing with 3D data. It is also likely that some scenes may contain only a subset of the novel classes. Therefore, enforcing the equipartitioning constraint in each batch of the dataset could affect the learning of less-frequent (long-tail) classes. As a solution, we introduce a queue $\mathtt{Z}_q$ containing a randomly extracted portion of the features of the novel points from the previous iterations. We use these additional data to mitigate the potential class imbalance that may occur during training. In practise, we compute $\mathtt{Z} \leftarrow \mathtt{Z} \oplus \mathtt{Z}_q$, where $\oplus$ is the concatenation operator, and execute the Sinkhorn-Knopp algorithm on this

augmented version of $\mathtt{Z}$. Then, we retain only the pseudo-labels for the first m columns of $\mathtt{Q}^*$.

3.5. Uncertainty-aware training and queuing

We propose to carefully select novel points for training f_Θ with fewer but more reliable pseudo-labels and to build a more effective queue $\mathtt{Z}_q$. We perform this selection by applying a threshold to the class probabilities of the novel class pseudo-labels. We found that seeking a fixed threshold for all the novel classes, that is also compatible with the variations of the class probabilities during training, is impractical. Therefore, we employ an adaptive threshold based on the class probabilities within each batch.

Our selection strategy operates as follows. Let τ_c be the adaptive threshold for the points of the novel class $c \in \mathcal{C}_n$. Firstly, we extract the novel points that have the greatest class probability for the class c. Secondly, we compute τ_c as the p-th percentile of the class probabilities of these novel points. Lastly, we retain the novel points of class c whose class probability is above the threshold τ_c. We define this selection strategy as the function

$$\phi : (\mathcal{F}_n, \hat{\mathcal{Y}}_n) \times p \mapsto (\bar{\mathcal{F}}_n), \tag{4}$$

where $\mathcal{F}_n$ is the set of feature vectors extracted from f_ξ and $\hat{\mathcal{Y}}_n$ is the set of class probabilities predicted by the network for these points. $\bar{\mathcal{F}}_n$ are both processed by the Sinkhorn-Knopp algorithm to generate our pseudo-labels and added to $\mathtt{Z}_q$ to make it more effective.

3.6. Optimisation objective

We optimise f_Θ by using the weighted Cross Entropy objective based on the labels $\mathcal{Y}_b$ of the base samples and the pseudo-labels $\tilde{\mathcal{Y}}_n$ of the novel samples. We formulate a swapped prediction task based on these pseudo-labels [5]. Secifically, we begin by generating two different augmentations of $\mathcal{X}$ that we define as $\mathcal{X}'$ and $\mathcal{X}''$. We use the known one-hot labels for $\mathcal{Y}_b$ and the predicted soft pseudo-labels for $\tilde{\mathcal{Y}}_n$. We predict the novel pseudo-labels $\tilde{\mathcal{Y}}'_n$ and $\tilde{\mathcal{Y}}''_n$ of the respective point clouds $\mathcal{X}'$ and $\mathcal{X}''$ with our approach. Then, we enforce prediction consistency between the swapped pseudo-labels of the two augmentations as

$$\mathcal{L}(\mathcal{X}) = \ell(\hat{\mathcal{Y}}', \tilde{\mathcal{Y}}'') + \ell(\hat{\mathcal{Y}}'', \tilde{\mathcal{Y}}'), \tag{5}$$

where $\hat{\mathcal{Y}}' = \hat{\mathcal{Y}}'_b \cup \hat{\mathcal{Y}}'_n$ (same for $\hat{\mathcal{Y}}''$), $\tilde{\mathcal{Y}}' = \mathcal{Y}'_b \cup \tilde{\mathcal{Y}}'_n$ (same for $\tilde{\mathcal{Y}}''$) and ℓ is the weighted Cross Entropy loss. We use separate segmentation heads for base classes and novel classes. The weights of the loss for the base classes are computed based on their occurrence frequency in the training set. The weights of the loss for the novel classes are all set equally as their occurrence frequency in the dataset is unknown.

4. Adapting NCD for 2D images to 3D

Another contribution of this work is to adapt the method proposed by Zhao et al. [40] for NCD for 2D semantic segmentation (EUMS) to 3D data. Our empirical evaluation (see Sec. 5) shows that the transposition of EUMS to the 3D domain has some limitations. In particular, as described in Sec. 1, EUMS uses two assumptions: **I**) the novel classes belong to the foreground and **II**) each image can contain at most one novel class. This allows EUMS to leverage a saliency detection model to produce a foreground mask and a segmentation model pre-trained on the base classes to determine which portion of the image is background. The portion of the image that belongs to both the foreground mask and the background mask is where features are then pooled. EUMS computes a feature representation for each image by average pooling the features of the pixels belonging the unknown portion. The feature representations of all the images in the dataset are clustered with K-Means by using the number of classes to discover as the target number of clusters. EUMS shows that overclustering and entropy-based modelling can be exploited to improve the results. The affiliation of a point to its cluster is used to produce hard pseudo-labels that are in turn used along with the ground-truth labels to fine-tune the pre-trained model.

With 3D point clouds, there is no concept of foreground and background (in contrast with **I**). Our adaptation is designed to discover the classes of all the unlabelled points (in contrast with **II**). Therefore, given the unlabelled points of each point cloud, we randomly extract a subset of these by setting a ratio (e.g. 30%) with upper bound (e.g. 1K) on the number of points to select. We compute and collect their features for all the point clouds in the dataset and apply K-Means on the whole set of features. Note that this clustering step is computationally expensive, and we had to use High Performance Computing to execute it. The subsampling of the points was necessary to fit the data in the RAM (see Sec. 5 for a detailed analysis). Once the cluster prototypes are computed, we produce the hard pseudo-labels. To enrich the set of pseudo-labels, we propagate the pseudo-label of each point to its nearest neighbour in the coordinate space. This allows us to expand the subset of pseudo-labelled randomly selected points. We also implement the other steps of overclustering and entropy-based modelling to boost the results. Lastly, we fine-tune our model with these pseudo-labels. We name our transposition of EUMS as EUMS† and report its block diagram in the Supplementary Material.

5. Experimental results

5.1. Experimental setup

Datasets. We evaluate our approach on SemanticKITTI [3, 4, 13] and SemanticPOSS [24]. SemanticKITTI [4] consists of 43,552 point cloud acquisitions with point-level annota-

Table 1. SemanticKITTI splits, is defined as KITTI-n^i, where n is the number of novel classes and i is the split index.

Split	Novel Classes
KITTI-5^0	*building*, *road*, *sidewalk*, *terrain*, *vegetation*
KITTI-5^1	*car*, *fence*, *other-ground*, *parking*, *trunk*
KITTI-5^2	*motorcycle*, *other-vehicle*, *pole*, *traffic-sign*, *truck*
KITTI-4^3	*bicycle*, *bicyclist*, *motorcyclist*, *person*

Table 2. SemanticPOSS splits, defined as POSS-n^i, where n is the number of novel classes and i is the split index.

Split	Novel Classes
POSS-4^0	*building*, *car*, *ground*, *plants*
POSS-3^1	*bike*, *fence*, *person*
POSS-3^2	*pole*, *traffic-sign*, *trunk*
POSS-3^3	*cone-stone*, *rider*, *trashcan*

tions of 19 semantic classes. Based on the official benchmark guidelines [4], we use sequence 08 for validation and the other sequences for training. SemanticPOSS [24] consists of 2,988 real-world point cloud acquisitions with point-level annotations of 13 semantic classes. Based on the official benchmark guidelines [24], we use sequence 03 for validation and the other sequences for training.

Experimental protocol for 3D NCD. Similarly to what proposed by [40] in the 2D domain, we create different splits of each dataset to validate the NCD performance. We create four splits for SemanticKITTI and SemanticPOSS. We refer to these splits as SemanticKITTI-n^i and SemanticPOSS-n^i, where i indexes the split. In each set, the novel classes and the base classes correspond to unlabelled and labelled points, respectively. Tabs. 1 & 2 detail the splits of our datasets. These splits are selected based on their class distribution in the dataset and on the semantic relationship between novel and base classes, e.g. in KITTI-4^3 the base class *motorcycle* can be helpful to discover the novel class *motorcyclist*. We report additional details about the selection process in the Supplementary Material.

We quantify the performance by using the mean Intersection over Union (mIoU), which is defined as the average IoU across the considered classes [4]. We provide separate mIoU values for the base and novel classes. We also report the overall mIoU computed across all the classes in the dataset for completeness.

Implementation Details. We implement our network based on a MinkowskiUNet-34C network [10]. Point-level features are extracted from the penultimate layer. The segmentation heads are implemented as linear layers, producing output logits for each point in the batched point clouds. We train our network for 10 epochs. We use the SGD optimizer, with momentum 0.9 and weight decay 0.0001. Our learning rate scheduler consists of linear warm-up and cosine annealing, with $lr_{max} = 10^{-2}$ and $lr_{min} = 10^{-5}$. We train with batch size equal to 4. We employ 5 segmentation heads, that are used in synergy with an equal number of over-clustering heads, with $o = 3$. In ϕ, we set $p = 0.5$ for SemanticKITTI-n^i and $p = 0.3$ for SemanticPOSS-n^i. We adapted the implementation of the Sinkhorn-Knopp algorithm [11] from the code provided by [5], with the introduction of the queue and an in-place normalisation steps. Similarly to [5], we set $n_{sk\_iters} = 3$, while we adopt a linear decay for ϵ, with $\epsilon_{start} = 0.3, \epsilon_{end} = 0.05$.

5.2. Quantitative analysis

Segmentation quality. Tabs. 3 & 4 report the quantitative results on SemanticPOSS and SemanticKITTI, respectively. We report the *Full supervision* setting as our upper bound.

On SemanticPOSS, we outperform EUMS† on three out of four splits with an improvement of 18.3 mIoU on POSS-4^0, 9.0 mIoU on POSS-3^1 and 0.6 on POSS-3^2. In these splits, NOPS shows a large improvement on all the novel classes, with the exception of the class *fence* in POSS-3^1 and *traffic-sign* in POSS-3^3. Differently, we deem the lower performance in POSS-3^3 is due to the difficulty and scarce presence of these novel classes. The advantage of EUMS† is the clustering on the whole dataset that enables a complete visibility of all the novel classes. On average, NOPS achieves 21.40 mIoU, improving over EUMS† of 6.5 mIoU.

On SemanticKITTI, we outperform EUMS† on all the four splits, improving by 12.6 mIoU on KITTI-5^0, 1.2 mIoU on KITTI-5^1, 4.2mIoU on KITTI-5^2 and 5.3 mIoU on KITTI-4^3. NOPS outperforms EUMS† by a large margin on the majority of the novel classes. Exceptions are the class *sidewalk* in KITTI-5^0, *car* in KITTI-5^1, *motorcycle* in KITTI-5^2 and *motorcyclist* in KITTI-4^3. On average, NOPS achieves 22.84 mIoU, improving over EUMS† of 5.8 mIoU. Interestingly, NOPS outperforms also the supervised upper bound on the class *trunk* in KITTI-5^1.

Computational time. NOPS outperforms EUMS† in terms of computational time. Firstly, EUMS† requires a pre-training step and a fine-tuning step, i.e. 30 training epochs in total. Then, EUMS† requires a large amount of memory (up to 200 GB memory for KITTI-5^0) to store the data required for clustering, taking several hours (50 hrs) to complete the training procedure. Differently, NOPS achieves superior performance with 10 training epochs, by using less memory (10 GB max) and a lower computational time (up to 25 hrs for KITTI-5^0). We run these tests using one GPU Tesla A40-48GB.

5.3. Qualitative analysis

Fig. 3 shows some segmentation results of NOPS and EUMS† on SemanticPOSS and SemaniticKITTI. We can observe that the predictions of the base classes in the two datasets are correct for both the models, with just minor er-

Table 3. Novel class discovery results on SemanticPOSS. NOPS outperforms EUMS† on three out of four splits. Full supervision: model trained with labels for base and novel classes. EUMS†: baseline described in Sec. 4. Highlighted values are the novel classes in each split.

Split	Model	bike	build.	car	cone.	fence	grou.	pers.	plants	pole	rider	traf.	trashc.	trunk	mIoU Novel	mIoU Base	mIoU All
	Full supervision	43.20	71.30	33.00	32.50	44.60	78.50	61.80	73.90	30.90	54.70	26.70	11.00	19.30	-	-	44.72
POSS-4^0	EUMS† [40]	25.67	3.98	0.56	16.44	29.40	36.76	43.84	28.46	13.13	26.75	18.18	3.34	16.91	17.44	21.52	20.26
	NOPS (Ours)	35.47	**30.35**	**1.24**	13.52	24.13	**69.14**	44.70	**42.07**	19.19	47.65	24.44	8.17	21.82	**35.70**	26.57	29.38
POSS-3^1	EUMS† [40]	15.17	67.98	28.02	23.98	**11.88**	75.07	35.98	74.46	26.91	48.56	26.00	5.60	23.05	21.01	39.96	35.59
	NOPS (Ours)	**29.35**	71.35	28.70	12.21	3.94	78.24	**56.78**	74.21	18.29	38.88	23.31	13.74	23.51	**30.02**	38.24	36.35
POSS-3^2	EUMS† [40]	40.14	69.45	27.67	13.50	34.86	76.03	54.66	75.59	5.27	39.22	**7.79**	8.52	11.85	8.31	43.96	35.74
	NOPS (Ours)	37.16	71.81	29.74	14.64	28.38	77.53	52.09	73.00	**11.51**	47.11	0.54	10.20	**14.79**	**8.95**	44.17	36.04
POSS-3^3	EUMS† [40]	41.17	70.68	28.08	**4.34**	38.27	76.66	38.29	75.35	25.76	**34.34**	28.31	0.36	24.40	**13.01**	44.70	37.38
	NOPS (Ours)	38.55	70.36	30.91	0.00	29.38	76.50	55.98	71.84	17.03	31.87	26.15	**0.95**	22.57	10.94	43.93	36.32
														Avg	EUMS† [40]	14.94	37.54	32.24
															NOPS (Ours)	**21.40**	38.23	34.52

Table 4. Novel class discovery results on SemanticKITTI. NOPS outperforms EUMS† on all four splits. Full supervision: model trained with annotations for base and novel classes. EUMS†: baseline described in Sec. 4. Highlighted values are the novel classes in each split.

Split	Model	bi.cle	b.clst	build.	car	fence	mt.cle	m.clst	oth-g.	oth-v.	park.	pers.	pole	road	sidew.	terra.	traff.	truck	trunk	veget.	mIoU Novel	mIoU Base	mIoU All
	Full supervision	6.30	39.50	85.40	90.00	23.20	20.30	5.70	3.90	18.00	28.90	31.00	40.60	90.90	74.60	62.10	20.50	62.90	46.20	83.90	-	-	43.89
KITTI-5^0	EUMS† [40]	5.28	39.96	15.77	79.20	9.03	16.89	2.52	0.07	11.39	14.40	12.67	29.17	42.58	**26.10**	0.05	10.30	47.37	37.92	38.35	24.57	21.08	23.11
	NOPS (Ours)	5.59	47.76	**52.68**	82.60	13.76	25.55	1.36	1.66	14.52	19.80	25.86	32.12	**56.74**	8.08	**23.84**	14.28	49.41	36.18	**44.17**	**37.10**	24.70	29.62
KITTI-5^1	EUMS† [40]	7.53	42.41	79.97	**76.77**	**8.62**	19.58	1.39	**0.57**	12.03	14.14	13.95	40.74	86.32	66.45	56.29	11.97	44.79	20.94	72.40	24.21	37.06	35.62
	NOPS (Ours)	7.36	51.23	84.53	50.87	7.27	28.93	1.76	0.00	22.20	**19.39**	30.42	37.61	90.07	72.18	60.75	16.78	57.34	**49.25**	85.12	**25.36**	43.09	40.69
KITTI-5^2	EUMS† [40]	8.26	50.78	82.98	88.05	17.88	**2.75**	2.32	0.17	3.16	25.40	24.98	20.20	88.30	71.04	57.85	8.63	**27.16**	38.36	76.95	12.38	42.22	36.59
	NOPS (Ours)	6.72	49.24	86.36	90.79	23.68	2.69	0.58	1.87	**15.46**	29.48	27.92	**36.39**	90.26	73.39	61.21	**17.83**	10.32	46.16	84.29	**16.54**	44.80	39.72
KITTI-4^3	EUMS† [40]	**3.95**	2.47	80.10	87.21	16.81	14.02	**14.98**	0.31	14.13	20.77	6.80	37.59	86.79	66.50	55.26	16.20	40.62	38.37	76.15	7.05	43.39	35.74
	NOPS (Ours)	2.32	**27.83**	86.04	89.89	23.06	24.47	2.92	3.06	18.19	30.09	**16.32**	39.90	90.65	73.51	61.04	17.40	49.76	44.01	83.18	**12.35**	48.95	41.24
																				Avg	EUMS† [40]	17.05	35.94	32.76
																					NOPS (Ours)	**22.84**	40.38	37.73

rors at the edges of the objects. NOPS shows better segmentation capabilities also when dealing with the novel classes, being able to properly segment the correct class for the unknown objects. On the larger objects, such as buildings in the scenes, some mixing of labels can be observed. Differently, EUMS nearly fails in correctly recognising the novel objects, resulting in the inclusion of different classes (either novel or base) into the same object, e.g. the facade of the building in POSS-3^1.

6. Ablation studies

We study the main components of NOPS and its behaviour when varying its parameters on SemanticPOSS.

Method components. Fig. 4 shows the performance on novel and base classes of seven versions of NOPS. The first three versions use the pre-trained model on the base classes, while the last four versions use the model trained from scratch. Each version is defined as follows:

- P: we use a pre-trained model, and we remove Z_q, τ_c and the over-clustering heads.
- OC: P + over-clustering heads.
- Q: OC + Z_q, i.e. our queue without class-balancing.
- NP: Q without pre-training.
- NP+: NP + our selection function ϕ on the queue.
- NP++: NP + τ_c on the features used to derive the pseudo-labels.
- Full: NOPS with all the components activated.

Pre-trained approaches generally underperform their trained-from-scratch counterparts on the novel classes. This is visible in the low performance of P, OC and Q. We have a significant improvement when pre-training is not used (NP), i.e. we achieve 20.26 mIoU. We can see that the queue both with and without pre-training is helpful. When we add the feature selection for the queue and for the training, i.e. NP+ and NP++, we have improvements, i.e. 20.63 mIoU and 20.90 mIoU, respectively. The best performance is achieved with Full. Although we can observe variations on the performance of the base classes, their information is retained by the network when we discover the novel ones.

Parameter analysis. We study the behaviour of the percentile p in our selection function ϕ, when we apply it to the features both for pseudo-labelling and for the class-balanced queue Z_q. Tab. 5 reports the results on each split of SemanticPOSS. For each split, we can observe that the performance depends on the number of points and difficulty of the novel classes. In POSS-4^0 and POSS-3^1, lower values

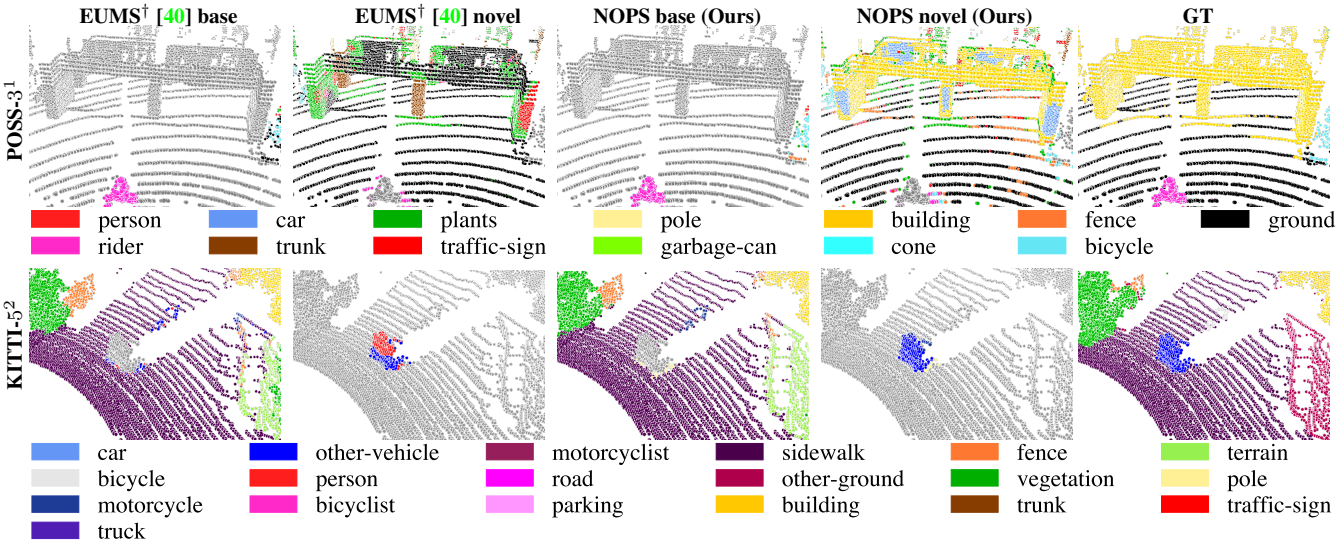

Figure 3. Qualitative comparisons on (top) SemanticPOSS and (bottom) SemanticKITTI. We report results on both base and novel classes. On novel classes, EUMS† fails in recognising novel objects with mixed and cluttered predictions, e.g. the facade of *building* in POSS-3^1. NOPS shows superior segmentation performance on all the novel classes.

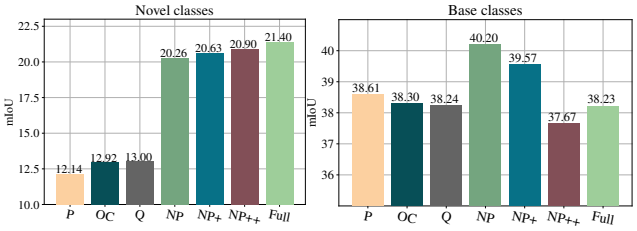

Figure 4. Ablation study with different components and initialisation strategies on SemanticPOSS. In P, OC and Q, we initialise the model after base pre-training, and use different configurations of the over-clustering heads and of our queue balancing. In NP NP+, NP++ and Full, we begin with Q, we avoid pre-training, and we use ϕ and τ_c incrementally. See Sec. 6 for definition of methods.

Table 5. Ablation study showing how different values of p affect the performance on SemanticPOSS. The lower p is, the less severe the selection of the features, resulting in better performances for POSS-4^0. Differently, POSS-3^3 benefits from an higher value of p, which leads to a more vigorous filtering of the features. POSS-3^1 and POSS-3^2 show the best performances with $p = 0.5$

Split	Percentile p				
	0.1	0.3	0.5	0.7	0.9
POSS-4^0	30.81	**35.70**	28.77	30.93	26.69
POSS-3^1	28.33	30.02	**30.43**	23.32	18.91
POSS-3^2	8.07	8.95	**10.32**	10.25	7.76
POSS-3^3	10.55	10.94	11.69	**14.38**	13.42
Avg.	19.44	**21.40**	20.30	19.72	16.70

of p result in less severe selection. We believe this is related to the class distribution within these splits. This is in line with what observed in Tab. 3. In POSS-3^2 and POSS-3^3, we notice a different behaviour, a higher value of p provides better results. We relate this to the difficulty of the novel classes in these splits whose noisy pseudo-labels can benefit from a more rigorous selection of the features.

7. Conclusions

We explored the new problem of novel class discovery for 3D point cloud segmentation. Firstly, we transposed the only NCD method for 2D semantic segmentation to 3D point cloud data, and experimentally found that it has several limitations. We discussed that extending 2D NCD approaches to 3D data (point clouds) is not trivial because the assumptions made for 2D data are not easily transferable to 3D. Secondly, we presented NOPS, to tackle NCD for point cloud segmentation by using online clustering and exploiting uncertainty quantification to produce pseudo-labels for the novel points. Lastly, we introduced a new evaluation protocol to asses the performance of NCD for point cloud segmentation. Experiments on two different segmentation dataset showed that NOPS outperforms the compared baselines by a large margin. Future research directions could investigate the extension of our method when base annotations are fewer and/or weakly labelled.

Limitations NOPS limitations include the prior knowledge on the number of novel classes C_n to discover. This could be a limitation when C_n is not known a-priori and novel classes may appear in an incremental manner. We believe that a solution may be to learn novel classes incrementally. Another limitation is the loss we use to handle class unbalancing. More recent techniques to handle this drawback could be further explored.

References

[1] I. Alonso, L. Riazuelo, L. Montesano, and A.C. Murillo. 3D-MiniNet: Learning a 2D representation from point clouds for fast and efficient 3D LiDAR semantic segmentation. In *IROS*, 2020. 2

[2] Y.M. Asano, C. Rupprecht, and A. Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 3, 4

[3] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss. Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. In *IJRR*, 2021. 2, 5

[4] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *CVPR*, 2019. 2, 5, 6

[5] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2, 4, 5, 6

[6] J. Cen, P. Yun, S. Zhang, J. Cai, D. Luan, M. Tang, M. Liu, and M.Y. Wang. Open-world semantic segmentation for lidar point clouds. In *ECCV*, 2022. 3

[7] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In *T-PAMI*, 2017. 3

[8] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and A. Hartwig. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 3

[9] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu. (AF)2-S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network. In *CVPR*, 2021. 2

[10] C. Choy, JY. Gwak, and S. Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019. 2, 3, 6

[11] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, 2013. 3, 4, 6

[12] E. Fini, E. Sangineto, S. Lathuiliere, Z. Zhong, M. Nabi, and E. Ricci. A unified objective for novel class discovery. In *CVPR*, 2021. 1, 2, 4

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*, 2012. 2, 5

[14] B. Graham, M. Engelcke, and L. van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 2018. 2

[15] B. Graham and L. van der Maaten. Submanifold sparse convolutional networks. *arXiv*, 2017. 2

[16] K. Han, A. Vedaldi, and A. Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *CVPR*, 2019. 1, 2, 4

[17] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *CVPR*, 2020. 2

[18] X. Ji, J.F. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019. 4

[19] X. Jia, K. Han, Y. Zhu, and B. Green. Joint representation learning and novel category discovery on single-and multi-modal data. In *ICCV*, 2021. 2

[20] K. Joseph, S. Paul, G. Aggarwal, S. Biswas, P. Rai, K. Han, and V.N. Balasubramanian. Novel class discovery without forgetting. *ECCV*, 2022. 2

[21] G. Mei, F. Poiesi, C. Saltori, J. Zhang, E. Ricci, and N. Sebe. Overlap-guided gaussian mixture models for point cloud registration. In *WACV*, 2023. 4

[22] G. Mei, C. Saltori, F. Poiesi, J. Zhang, E. Ricci, N. Sebe, and Q. Wu. Data augmentation-free unsupervised learning for 3d point cloud understanding. In *BMVC*, 2022. 3, 4

[23] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019. 2, 3

[24] Y. Pan, B. Gao, J. Mei, S. Geng, C. Li, and H. Zhao. SemanticPOSS: A point cloud dataset with large quantity of dynamic instances. In *IV*, 2020. 2, 5, 6

[25] C.R. Qi, H. Su, K. Mo, and L.J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 2

[26] C.R. Qi, L. Yi, H. Su, and L.J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 2

[27] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[28] S. Roy, M. Liu, Z. Zhong, N. Sebe, and E. Ricci. Class-incremental novel class discovery. In *ECCV*, 2022. 2

[29] C. Saltori, F. Galasso, G. Fiameni, N. Sebe, E. Ricci, and F. Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. In *ECCV*, 2022. 3

[30] R. Song, W. Zhang, Y. Zhao, Y. Liu, and P.L. Rosin. Mesh saliency: An independent perceptual measure or a derivative of image saliency? In *CVPR*, 2021. 2

[31] N. Souly, C. Spampinato, and M. Shah. Semi supervised semantic segmentation using generative adversarial network. In *ICCV*, 2017. 2

[32] X. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *CVPR*, 2016. 2

[33] H. Thomas, C.R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and J. L. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 2

[34] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman. Generalized category discovery. In *CVPR*, 2022. 2

[35] B. Wu, A. Wan, X. Yue, and K. Keutzer. Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In *ICRA*, 2018. 2

[36] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. In *ICRA*, 2019. 2

[37] M. Yang, Y. Zhu, J. Yu, A. Wu, and C. Deng. Divide and conquer: Compositional experts for generalized novel class discovery. In *CVPR*, 2022. 2

[38] L. Zhang and G.J. Qi. Wcp: Worst-case perturbations for semi-supervised deep learning. In *CVPR*, 2020. 2

[39] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh. Polarnet: An improved grid representation for online LiDAR point clouds semantic segmentation. In *CVPR*, 2020. 2

[40] Y. Zhao, Z. Zhong, N. Sebe, and G.H. Lee. Novel class discovery in semantic segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[41] Z. Zhong, E. Fini, S. Roy, Z. Luo, E. Ricci, and N. Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, 2021. 1, 2, 4

[42] Z. Zhong, L. Zhu, Z. Luo, S. Li, Y. Yang, and N. Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *CVPR*, 2021. 2

[43] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3D object detection. In *CVPR*, 2018. 2

[44] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin. Cylindrical and asymmetrical 3D convolution networks for lidar segmentation. In *CVPR*, 2021. 2