International Journal of Computer Vision
https://doi.org/10.1007/s11263-024-02180-x

Novel Class Discovery Meets Foundation Models for 3D Semantic Segmentation

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

Received: 15 September 2023 / Accepted: 28 June 2024

Abstract

The task of Novel Class Discovery (NCD) in semantic segmentation involves training a model to accurately segment unlabelled (novel) classes, using the supervision available from annotated (base) classes. The NCD task within the 3D point cloud domain is novel, and it is characterised by assumptions and challenges absent in its 2D counterpart. This paper advances the analysis of point cloud data in four directions. Firstly, it introduces the novel task of NCD for point cloud semantic segmentation. Secondly, it demonstrates that directly applying an existing NCD method for 2D image semantic segmentation to 3D data yields limited results. Thirdly, it presents a new NCD approach based on online clustering, uncertainty estimation, and semantic distillation. Lastly, it proposes a novel evaluation protocol to rigorously assess the performance of NCD in point cloud semantic segmentation. Through comprehensive evaluations on the SemanticKITTI, SemanticPOSS, and S3DIS datasets, our approach show superior performance compared to the considered baselines.

Keywords Novel class discovery · Point cloud semantic segmentation · 3D foundation models

1 Introduction

Humans possess a remarkable ability to categorise new information (or novelties) into homogeneous groups, even when they are unfamiliar with what is observed. In contrast, machines can hardly achieve this without guidance. The primary challenges of machine vision lie in crafting discriminative latent representations of the real world and in quantifying uncertainty when faced with novelties (Han et al., 2019; Zhong et al., 2021; Zhao et al., 2022). Han et al. (2019) pioneered the formulation of the Novel Class Discovery (NCD) problem. They defined it as the endeavor to categorise samples from an unlabelled dataset, termed *novel samples*, into distinct classes by leveraging the insights from a set of labelled samples, known as 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 (Han et al., 2019; Fini et al., 2021; Zhong et al., 2021), and subsequently, for semantic segmentation (Zhao et al., 2022). Specifically, Zhao et al. (2022) introduced the first approach to address NCD in the 2D semantic segmentation task. The authors posited two key assumptions: first, each image contains only one novel class; and second, the novel class corresponds to a foreground object detectable

Communicated by Ming-Hsuan Yang.

Project page: https://luigiriz.github.io/SNOPS_website/. This project has received funding from the European Union's Horizon Europe research and innovation programme under the projects AI-PRISM (grant agreement No. 101058589) and FEROX (grant agreement No. 101070440). This work was also partially sponsored by the PRIN project LEGO-AI (Prot. 2020TA3K9N), EU ISFP PRECRISIS (ISFP-2022-TFI-AG-PROTECT-02-101100539), PNRR ICSC National Research Centre for HPC, Big Data and Quantum Computing (CN00000013) and the FAIR - Future AI Research (PE00000013), funded by NextGeneration EU. It was carried out in the Vision and Learning joint laboratory of FBK and UNITN.

✉ Luigi Riz
luriz@fbk.eu

Cristiano Saltori
cristiano.saltori@unitn.it

Yiming Wang
ywang@fbk.eu

Elisa Ricci
e.ricci@unitn.it

Fabio Poiesi
poiesi@fbk.eu

[1] Fondazione Bruno Kessler, Trento, Italy

[2] University of Trento, Trento, Italy

Published online: 07 August 2024

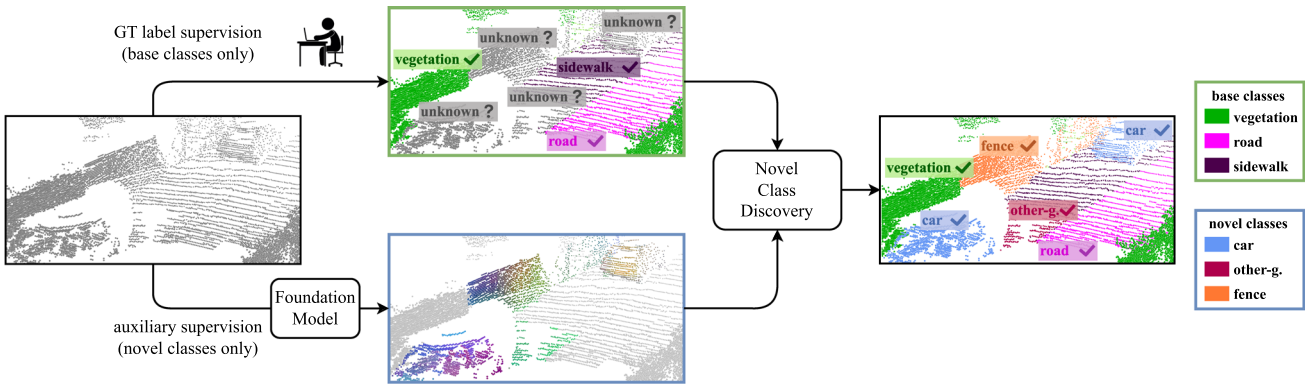

Fig. 1 SNOPS addresses the novel class discovery task in 3D point cloud semantic segmentation by leveraging the knowledge of ground-truth labels (for base classes) and the auxiliary supervision from a foundation model (for novel classes) to learn the correct semantic segmentation of both base and novel points

through saliency detection (e.g., a man on a bicycle, with the bicycle being the novel class). Leveraging these assumptions, the authors were able to pool the features of each image into a single latent representation and group the representations of the entire dataset to identify clusters of novel classes. However, we argue that these assumptions impose significant constraints that are difficult to meet with generic 3D data, especially point clouds obtained from LiDAR sensors in large-scale settings. A single point cloud may contain multiple novel classes, and the concept of saliency in 3D data does not directly translate from its 2D counterpart. While both concepts relate to the focus of human attention, 3D saliency is more about the regional significance of 3D surfaces rather than a simple foreground/background distinction (Song et al., 2021).

Our previous work, NOvel Point Segmentation (NOPS) (Riz et al., 2023) pioneers in NCD for 3D semantic segmentation, with the primary focus on addressing the above discussed limitations. NOPS has shown promising performance in tackling 3D NCD, yet, the recent emergence of 3D foundation models (Peng et al., 2023) offers us new opportunities in terms of the methodology design in NCD for their strong performance in zero-shot recognition. However we empirically show that the accuracy of using the foundation model alone in a zero-shot manner for the NCD task is significantly lower than combining it with method that is specifically designed for NCD on multiple benchmark datasets (Behley et al., 2019; Pan et al., 2020; Armeni et al., 2016), as shown in Tables 4, 5 and 6.

In this work, we present Semantically-aligned Novel Point Segmentation (SNOPS), a method that extends NOPS (Riz et al., 2023) by utilising an additional, unsupervised source of semantic knowledge in the form of a foundation model, such as CLIP (Radford et al., 2021) (Fig. 1). SNOPS, given a dataset partially annotated by humans, concurrently learns base and novel semantic classes by clustering unlabelled points based on their semantic similarities. We have adapted the methodology of Zhao et al. (2022), termed Entropy-based Uncertainty Modelling and Self-training (EUMS), to accommodate point cloud data, thereby establishing it as our baseline. We move beyond their framework and, drawing inspiration from Caron et al. (2020) and Peng et al. (2023), incorporate batch-level (online) clustering and distillation from a foundation model. Batch-level clustering generates prototypes that we utilise to manage large-cardinality 3D point clouds, while distillation is essential to leverage the intrinsic semantic knowledge contained within foundation models. We update prototypes during training to make clustering computationally feasible and introduce a method based on uncertainty to enhance prototype quality. We establish point-cluster assignment to produce pseudo-labels for self-training and also employ over-clustering to ensure precision. Given the diverse semantic classes within point clouds, it is inevitable that not all classes are represented in every batch. To address this issue, we have developed a queuing approach to maintain representative features throughout the training process. These features act as proxies for missing categories during the generation of pseudo-labels, facilitating a more balanced clustering of the novel classes. Lastly, we generate two augmented perspectives of a singular point cloud and enforce consistency in pseudo-labels between them. Our methodology is assessed on SemanticKITTI (Behley et al., 2019; Geiger et al., 2012; Behley et al., 2021), SemanticPOSS (Pan et al., 2020), S3DIS (Armeni et al., 2016), and nuScenes Caesar et al. (2020). We establish an evaluation protocol for NCD and point cloud segmentation, serving as a potential benchmark for subsequent research. Empirical evidence suggests that our method significantly surpasses our baseline and predecessor version of our method (Riz et al., 2023) across all datasets. Additionally, we undertake a comprehensive ablation study to underscore the significance of our method's diverse components.

To summarise, our contributions are:

- Tackling NCD for 3D semantic segmentation, addressing unfit assumptions that are originally imposed on NCD for 2D semantic segmentation;
- Providing empirical proof that zero-shot semantic segmentation with 3D foundation model is not a good enough solution for NCD.
- Presenting a novel method SNOPS that effectively synergises NCD method with semantic distillation through foundation model, advancing the state of the art in NCD for 3D semantic segmentation;
- Introducing a new evaluation protocol to assess the performance of NCD for 3D semantic segmentation.

This paper extends our earlier work (Riz et al., 2023) in several aspects. We extend the original NOPS by leveraging a foundation model (Peng et al., 2023) to improve the accuracy of novel classes. We empirically show that the zero-shot accuracy of the foundation model alone is significantly lower than that achieved by using it in combination of our novel class discovery method. Then, we significantly extend our experimental evaluation and analysis by adding new experiments, new datasets, new comparisons, and new ablation studies to evaluate this new setup. Lastly, we expand the related work by reviewing additional state-of-the-art approaches.

2 Related Work

In this section, we thoroughly discuss recent works on three relevant topics, including point cloud semantic segmentation, 3D representation learning and novel class discovery.

Point cloud semantic segmentation can be performed at point level (Qi et al., 2017), on range view maps (Ronneberger et al., 2015), or by voxelising the input points (Zhou & Tuzel, 2018). Point-level networks process the input without intermediate representations. Examples of these include PointNet (Qi et al., 2017), PointNet++ (Qi et al., 2017), RandLA-Net (Hu et al., 2020), and KPConv (Thomas et al., 2019). PointNet (Qi et al., 2017) and PointNet++ (Qi et al., 2017) are based on a series of multi-layer perceptron where PointNet++ introduces global and local feature aggregation at multiple scales. RandLA-Net (Hu et al., 2020) uses random sampling, attentive pooling, and local spatial encoding. KPConv (Thomas et al., 2019) 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 (Milioto et al., 2019) and voxel-based approaches (Choy et al., 2019) 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 (Ronneberger et al., 2015), and re-projecting predictions to the initial 3D space. SqueezeSeg networks (Wu et al., 2018, 2019), 3D-MiniNet (Alonso, Riazuelo, Montesano, and Murillo, 2020), RangeNet++ (Milioto et al., 2019), and PolarNet (Zhang et al., 2020) are examples of this category. Although they are more efficient, these approaches tend to lose information during projection and re-projection. The latter includes 3D quantisation-based approaches that discretise the input points into a 3D voxel grid and employ 3D convolutions (Zhou & Tuzel, 2018) or 3D sparse convolutions (Graham & van der Maaten, 2017; Choy et al., 2019) to predict per-voxel classes. VoxelNet (Zhou & Tuzel, 2018), SparseConv (Graham & van der Maaten, 2017; Graham et al., 2018), MinkowskiNet (Choy et al., 2019), Cylinder3D (Zhu et al., 2021), and $(AF)^2$-S3Net (Cheng et al., 2021) are architectures belonging to this category. The above-mentioned approaches usually tackle point cloud segmentation in a supervised setting, whereas we address novel class discovery with both labelled base classes and unlabelled novel classes.

3D representation learning refers to learn general and useful point cloud representations from unlabelled point cloud data (Achlioptas et al., 2018; Xiao et al., 2023). Existing methods can be grouped into generative, context similarity based, local descriptor based, and multi-modal approaches. Generative approaches involve the generation of a point cloud as unsupervised task (Yang et al., 2018, 2021, 2019). FoldingNet (Yang et al., 2018), PSG-Net (Yang et al., 2021) and PointFlow (Yang et al., 2019) follow the autoencoder (Hinton & Salakhutdinov, 2006) paradigm and learn to self-reconstruct the input point cloud. Differently, LatentGAN (Achlioptas et al., 2018), Tree-GAN (Shu et al., 2019) and 3D-GAN (Wu et al., 2016) follow a generative adversarial strategy and learn to generate point cloud instances from a sampled vector or a latent embedding. PU-GAN (Li et al., 2019) and PU-GCN (Qian et al., 2021) learn the underlying geometries of point clouds by generating a denser point cloud with similar geometries. On the other hand, PCN (Yuan et al., 2018), SA-Net (Wen et al., 2020), Point-BERT (Yu et al., 2022) and Point-MAE (Pang et al., 2022) learn to complete the input point cloud by predicting the arbitrary missing parts. Context similarity based approaches learn discriminative 3D representations through the underlying similarities between point samples. PointContrast (Xie et al., 2020), DepthContrast (Zhang et al., 2021), ACD (Gadelha et al., 2020) and STRL (Huang et al., 2021) enforce the network to group feature representations through contrastive learning between positive and negative point cloud pairs. Another similarity based technique makes use of coordinate sorting as a unsupervised task. For example, Jigsaw3D (Sauder & Sievers, 2019) and Rotation3D (Poursaeed et al., 2020) follow this idea and train the network to predict either the re-organised version or the rotation angle of the input point clouds. Local descriptor

approaches focus on learning to encode per-point informative features by solving low-level tasks, e.g. point cloud registration. PPF-FoldNet (Deng et al., 2018) and CEM (Jiang et al., 2021) learn compact descriptors by solving the task of point cloud matching and registration, respectively. Differently, GeDi (Poiesi & Boscaini, 2022) employs constrastive learning between canonical point cloud patches to learn compact and generalisable descriptors. Multi-modal approaches follow the recent success from the 2D literature (Radford et al., 2021; Dong et al., 2023) and learn robust and comprehensive representations by modeling the relationships across modalities. Language grounding (Rozenberszki et al., 2022) maps per-point features to text CLIP (Radford et al., 2021) embeddings, providing a robust pre-training for semantic tasks. ConceptFusion (Jatavallabhula et al., 2023) leverages the open-set capabilities of foundation models (Guzhov et al., 2022; Radford et al., 2021; Kirillov et al., 2023) from multiple modalities and fuses their features into a 3D map via traditional integration approaches. More recently, OpenScene (Peng et al., 2023) learns a feature space where text and multi-view image pixels are co-embedded in the CLIP feature space. In this work, we tackle NCD for 3D segmentation and extend our previous method NOPS (Riz et al., 2023) by leveraging the powerful representations of OpenScene (Peng et al., 2023) in our novel class discovery network.

Novel class discovery (NCD) is initially explored for 2D classification (Han et al., 2019; Zhong et al., 2021; Fini et al., 2021; Joseph et al., 2022; Roy et al., 2022; Jia et al., 2021; Zhong et al., 2021; Vaze et al., 2022; Yang et al., 2022) and 2D segmentation (Zhao et al., 2022). NCD is formulated in a different way compared to standard semi-supervised learning (Souly et al., 2017; Zhang & Qi, 2020; Tang et al., 2016). 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. (Han et al., 2019) 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. (Zhong et al., 2021) 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 (Fini et al., 2021) 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 (Joseph et al., 2022) and FRoST (Roy et al., 2022) further extend NCD to the incremental learning setting. EUMS (Zhao et al., 2022) is the only approach analysing NCD for 2D semantic segmentation. 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 focus on NCD for 3D point cloud semantic segmentation. Unlike previous works, our problem inherits the challenges from the fields of 2D semantic segmentation (Chen et al., 2018; Chen, Papandreou, Kokkinos, Murphy, and Yuille, 2017) and 3D point cloud segmentation (Choy et al., 2019; Saltori et al., 2022; Milioto et al., 2019). From 2D semantic segmentation, the main challenges are multiple novel classes in the same image and the strong class unbalance. From 3D point cloud segmentation, we have to tackle the sparsity of input data, the different density of point cloud regions and the inability to identify foreground and background, which are not present in 2D segmentation (Zhao et al., 2022). From related fields in 3D scene understanding, the previous effort REAL (Cen et al., 2022) tackles open-world 3D semantic segmentation by classifying all the unknown points into a single class. Novel classes are then labelled by a human annotator and used for learning incrementally novel classes. Instead, NOPS (Riz et al., 2023) is the first work tackling NCD for 3D semantic segmentation. Unlike (Zhao et al., 2022) that uses K-Means, Riz et al. (2023) 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 (Asano et al., 2020; Mei et al., 2022). On top of NOPS (Riz et al., 2023), this work incorporates the unsupervised semantic knowledge distilled from a 3D foundation model (Peng et al., 2023). We show that the unsupervised knowledge distilled from a 3D foundation model significantly improves NCD performance.

3 Our Approach

3.1 Overview

We use two UNet-like deep neural networks optimised for 3D data tof extract point-level features from an input point cloud. The primary network starts untrained, serving as our target for training to concurrently segment both base and novel classes. The secondary network is auxiliary and pre-trained for task-agnostic open-vocabulary 3D scene understanding. For base class points, we use traditional supervised training, leveraging the available human annotations (ground truth). The training for novel classes pursues two distinct objectives. Firstly, we aim to align the features with the semantic knowledge of the auxiliary network (Sect. 3.6). Secondly,

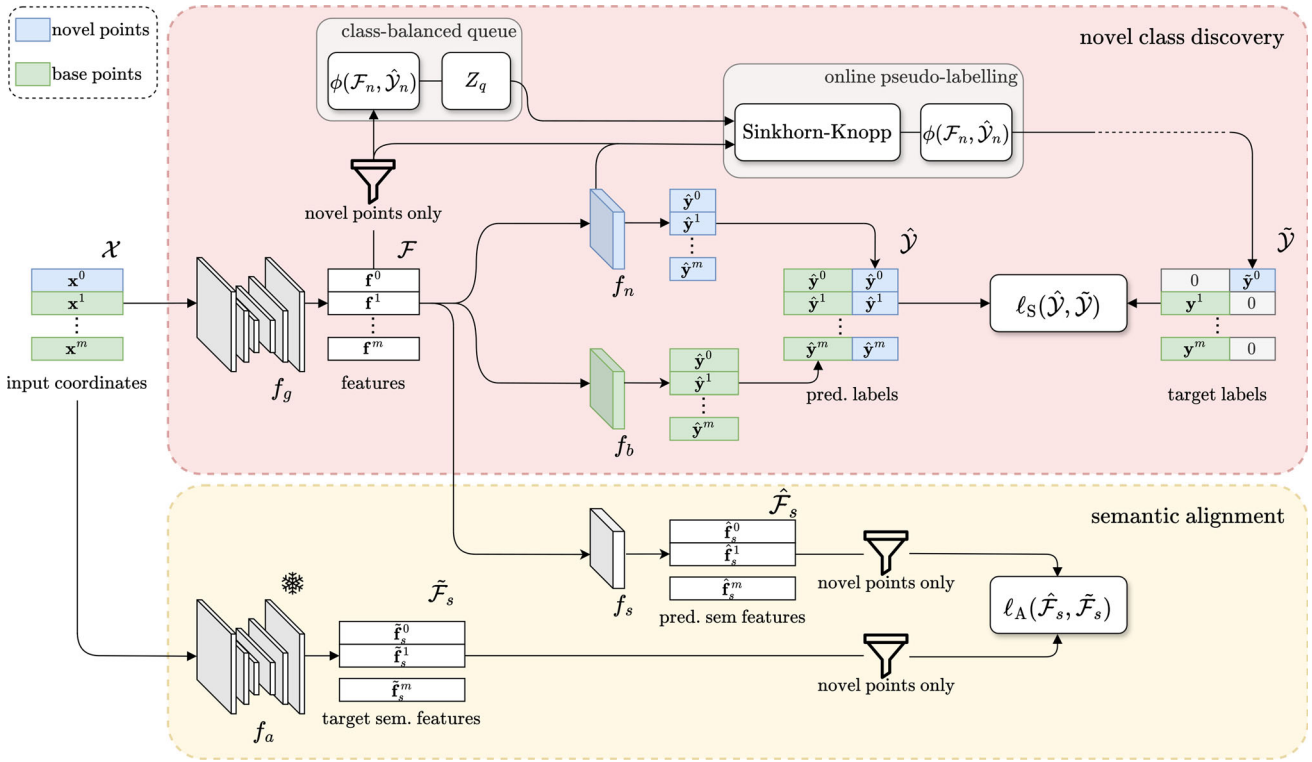

Fig. 2 Overview of SNOPS. We extract point-level features $\mathcal{F}$ with the shared backbone f_g. $\mathcal{F}$ are used to obtain pseudo-labels in the online pseudo-labelling block. We forward $\mathcal{F}$ through a novel f_n and a base f_b segmentation head to obtain point-wise predictions. We also pass $\mathcal{F}$ through a projection layer f_s that produces point-wise features for novel points. We align such point descriptors to the ones output by a frozen auxiliary network f_a, a large 3D vision model. The network is optimised by minimising the sum of a segmentation loss and an alignment loss

we adopt a self-supervised approach that generates pseudo-labels based on our *online pseudo-labelling* method through the Sinkhorn-Knopp algorithm (Cuturi, 2013) (Sect. 3.3). To enable each processed batch to maintain an equal number of novel classes, even if some are absent in point clouds being processed, we use a *class-balanced queue* that stores features during training (Sect. 3.4). We harness the pseudo-label confidences (class probabilities) to sift out uncertain points, thus populating the queue solely with high-quality points (Sect. 3.5). Specifically, our optimisation objective is

$$\mathcal{L} = \ell_S + \gamma \ell_A, \tag{1}$$

where ℓ_S is the segmentation loss involving ground-truth labels and pseudo-labels (Sect. 3.3), ℓ_A is the alignment loss that considers the semantic features extracted with the auxiliary network (Sect. 3.6) and γ is a weighting factor. Figure 2 shows the block diagram of SNOPS.

3.2 Problem Formulation

Let $\mathrm{X} = \{\mathcal{X}\}$ be a dataset of 3D point clouds captured in different scenes. The point cloud $\mathcal{X}$ 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 (Han et al., 2019; Zhong et al., 2021; Zhao et al., 2022), we assume that the number of classes to discover is known, i.e. $|\mathcal{C}_n| = C_n$. We aim to train a deep neural network f_Θ that can segment all the points of a given point cloud, thus learning to jointly segment base classes $\mathcal{C}_b$ and novel classes $\mathcal{C}_n$. Θ are the weights of our deep neural network. f_Θ is composed of a feature extractor network f_g, two segmentation heads f_n and f_b (for novel and base classes, respectively) and a feature projector f_s. $f_\Theta = f_g \circ \{f_b, f_n, f_s\}$, where $\circ$ 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 (Caron

et al., 2020). Let $\mathtt{P} \in \mathbb{R}^{D \times \rho}$ be the class prototypes, where D is the size of the output features from f_g and ρ is the number of prototypes. Let $\mathtt{Z} \in \mathbb{R}^{D \times m_n}$ be the normalised output features for novel points extracted from f_g, i.e. $\mathtt{Z} = f_g(\mathcal{X}_n)$, where m_n is the number of novel points of the point cloud. m_n is unknown a priori and it can differ across point clouds. We define $\mathtt{Q} \in \mathbb{R}^{\rho \times m_n}$ as the assignment between the ρ prototypes and the m_n novel points that equally partitions the novel points in the point cloud across the available 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. (2020) employs an arbitrary large number of prototypes ρ to effectively organise the feature space produced by f_g. They discard $\mathtt{P}$ after training. In contrast, we learn exactly $\rho = C_n$ class prototypes and 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 learnt prototypes as

$$\max_{\mathtt{Q} \in \mathcal{Q}} \operatorname{Tr}(\mathtt{Q}^\top \mathtt{P}^\top \mathtt{Z}) + \epsilon H(\mathtt{Q}) \rightarrow \mathtt{Q}^*, \tag{2}$$

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. (2020) 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 considering only the points within the point cloud being processed

$$\mathcal{Q} = \left\{ \mathtt{Q} \in \mathbb{R}_+^{C_n \times m_n} | \mathtt{Q} 1_{m_n} = \frac{1}{C_n} 1_{C_n}, \mathtt{Q}^\top 1_{C_n} = \frac{1}{m_n} 1_{m_n} \right\}, \tag{3}$$

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

$$\mathtt{Q}^* = \operatorname{diag}(\alpha) \exp\left(\frac{\mathtt{P}^\top \mathtt{Z}}{\epsilon} \right) \operatorname{diag}(\beta), \tag{4}$$

where α and β are renormalisation vectors that are computed iteratively with the Sinkhorn-Knopp algorithm (Cuturi, 2013; Mei et al., 2023). We then transpose the optimised soft assignment $\mathtt{Q}^* \in \mathbb{R}_+^{C_n \times m_n}$ to obtain the soft pseudo-labels for each of the m_n novel points being processed within each point cloud. For simplicity, the procedure described here takes into account only batches composed of a single point cloud. However, the same algorithm can be applied when two or more point clouds are concatenated into a single batch.

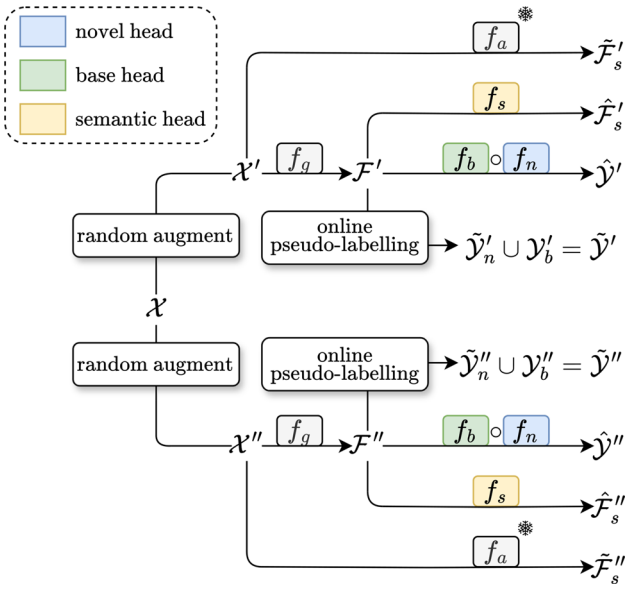

Fig. 3 Overview of the different outputs after the input point cloud $\mathcal{X}$ undergoes two different random augmentations, required for the generation of self-supervised pseudo-labels

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.

Segmentation objective: The segmentation objective ℓ_S is formulated as the weighted Cross Entropy loss and it is based on the ground-truth labels $\mathcal{Y}_b$ for base points and on the pseudo-labels $\tilde{\mathcal{Y}}_n$ for novel points. We formulate a swapped prediction task based on these pseudo-labels (Caron et al., 2020). We begin by generating two different augmentations of the original point cloud $\mathcal{X}$ that we define as $\mathcal{X}'$ and $\mathcal{X}''$ (Fig. 3). For the augmentation $\mathcal{X}'$ we define the segmentation predictions $\hat{\mathcal{Y}}' = f_b(f_g(\mathcal{X}')) \oplus f_n(f_g(\mathcal{X}'))$, where $\oplus$ is the concatenation operator. Analogously, we define $\hat{\mathcal{Y}}''$ as the network output for $\mathcal{X}''$. The segmentation targets are defined as $\tilde{\mathcal{Y}}' = \tilde{\mathcal{Y}}'_n \cup \mathcal{Y}'_b$, where $\tilde{\mathcal{Y}}'_n$ are the pseudo-labels predicted with our approach and $\mathcal{Y}'_b$ are the available targets for base classes (same for $\tilde{\mathcal{Y}}''_n$).

At this point, we enforce prediction consistency between the swapped pseudo-labels of the two augmentations as:

$$\ell_S(\mathcal{X}) = \ell_{\text{wCE}}(\hat{\mathcal{Y}}', \tilde{\mathcal{Y}}'') + \ell_{\text{wCE}}(\hat{\mathcal{Y}}'', \tilde{\mathcal{Y}}'), \tag{5}$$

where ℓ_{wCE} is the weighted Cross Entropy loss. 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.

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) (Ji et al., 2019). 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 (Caron et al., 2020; Fini et al., 2021; Mei et al., 2022; Ji et al., 2019). 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 Sect. 3.3 produces an equipartite matching between the novel points and the class centroids. However, it is likely that batches are sampled with point clouds containing novel classes with different cardinalities when dealing with 3D data. In addition, some scenes may contain only a subset of the novel classes. Therefore, enforcing the equipartitioning constraint for 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 this additional data to mitigate the potential class imbalance that may occur during training. 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 $\mathtt{Z}$. The obtained $\mathtt{Q}^* \in \mathbb{R}^{\mathcal{C}_n \times (m_c + |\mathtt{Z}_q|)}$ represents the assignment between the class prototypes and all the points in the augmented version of $\mathtt{Z}$. Being interested only in the pseudo-labels for the points in the actual batch, we retain only the first m_c columns of $\mathtt{Q}^*$, discarding the additional information related to the points contained in $\mathtt{Z}_q$.

3.5 Uncertainty-Aware Training and Queuing

The optimisation of f_Θ through pseudo-labels and the insertion of the novel points into the queue $\mathtt{Z}_q$ can both benefit from the selection of novel points that are considered reliable by the network. We perform this selection by considering the class assignment probability $\hat{\mathcal{Y}}_n$ for the novel points. In particular, we propose to apply a different threshold τ_c for each novel class $c \in \mathcal{C}_n$. All the novel points predicted by the network as belonging to novel class c with the confidence above τ_c are used during optimisation, and are kept as candidates for the insertion in the queue. All the other novel points are instead discarded. We found that it is impractical to seek a fixed threshold for all the novel classes, while being also compatible with the variations of the class probabilities during training. Therefore, we employ an adaptive threshold based on the class probabilities within each batch.

Our adaptive selection strategy operates as follows. Firstly, we extract the novel points that have been predicted as part of novel class c by the network. Secondly, we compute τ_c as the p-th percentile of the class probabilities of these novel points. Lastly, we retain only the novel points of class c whose class probability is above the threshold τ_c. We define this selection strategy as the function

$$\phi : (\mathcal{F}_n, \hat{\mathcal{Y}}_n) \times p \mapsto (\bar{\mathcal{F}}_n), \tag{6}$$

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

At the first optimisation iterations, the threshold τ_c is low for all the novel classes $c \in \mathcal{C}_n$ due to the network's random initialisation. However, each novel class is discovered during training, each threshold τ_c is expected to increase in an adaptive way to select novel points that are more and more reliable, resulting in a better optimisation of f_Θ. Figure 4 shows the evolution of the adaptive threshold τ_c when discovering four novel classes. The behaviour of the four different thresholds indicates that our method progressively selects more reliable novel points for training, thereby enhancing the optimisation process of f_Θ, leading to effective discovery of the four novel classes.

3.6 Incorporating Semantic Knowledge

The optimisation of f_Θ through the ground-truth labels and the pseudo-labels generated as described in Sect. 3.3 arranges the feature space output by f_g so to effectively separate representations of novel and base classes. However, the supervision provided by ground-truth targets is significantly stronger than the self-supervision of the pseudo-labels. This unbalance could result into a sub-optimal organisation of the feature space, in which base class representations are compact and well-separated while novel class features are poorly clustered with more noise and less compactness. To address this issue, we incorporate additional supervision for novel categories. We employ an auxiliary neural network f_a that

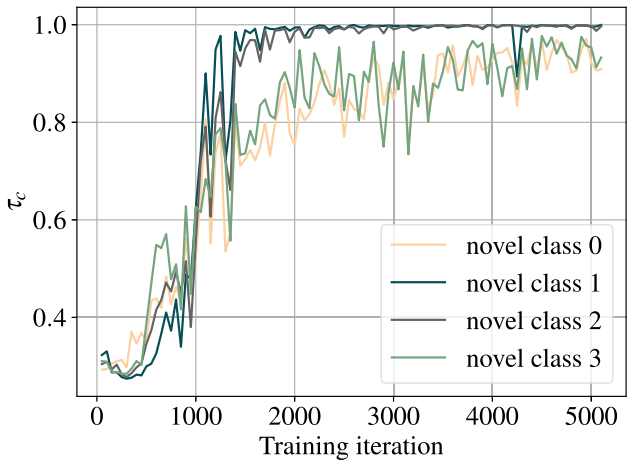

Fig. 4 Evolution of the adaptive selection threshold τ_c when discovering four novel classes on S3DIS

is able to output point-level semantic-aligned features, such as a 3D foundation model (Peng et al., 2023). Such architectures have shown great performance in multiple 3D scene understanding tasks, being able to reach the results of models specifically tailored for each single task. This good generalisation capability suggests that the feature spaces of 3D foundation model are well partitioned and organised according to semantics. So, by guiding our network f_Θ to mimic the point-level features output by f_a, we can enhance the semantic organisation of its feature space.

The most natural choice when aligning the features of our network to the ones of f_a would be to consider the feature space output by our backbone f_g. However, there is the risk that the distillation procedure from f_a interferes with the Sinkhorn-Knopp algorithm in organising the feature space output by f_g. So, we attach an additional projection head f_s on top of the feature extractor f_g. The knowledge distillation from the foundation model is performed on the features $\mathcal{F}_s$ output by f_s, while the features $\mathcal{F}$ are reserved for the Sinkhorn-Knopp algorithm. A well-separated representation of novel classes in the feature space of f_s, achieved through knowledge distillation from the foundation model, can in turn lead to improved separation of novel classes in the feature space of f_g. In fact, the feature space f_s is on top of f_g and they share the same underlying architecture. The distillation procedure is performed by minimising the alignment objective:

$$\ell_{\mathrm{A}}(\mathcal{X}) = \ell_{\cos}(\hat{\mathcal{F}}'_s, \tilde{\mathcal{F}}'_s) + \ell_{\cos}(\hat{\mathcal{F}}''_s, \tilde{\mathcal{F}}''_s), \tag{7}$$

where $\ell_{\cos}$ is the cosine loss, $\hat{\mathcal{F}}'_s = f_s(f_g(\mathcal{X}'))$ and $\tilde{\mathcal{F}}'_s = f_a(\mathcal{X}')$. The same applies for $\hat{\mathcal{F}}''_s$ and $\tilde{\mathcal{F}}''_s$ (Fig. 3). Differently from ℓ_{S}, in this case we do not use a swapped prediction task.

The projection head f_s and features $\mathcal{F}_s$ are only considered during training and we ignore this branch of f_Θ at test time.

4 Baseline Methods for 3D Novel Class Discovery

SNOPS and our earlier method NOPS (Riz et al., 2023) are the first architectures proposed to tackle the task of Novel Class Discovery in point cloud semantic segmentation. So, in this work we also present two baseline methods related to 3D novel class discovery we can compare SNOPS to: the adaptation of EUMS from the image domain to the 3D point cloud domain (referred to EUMS†) and the zero-shot testing of the OpenScene (Peng et al., 2023) model. These approaches hold significant importance in the relatively unexplored domain of 3D NCD, as they offer valuable insights into the challenges of such task. In particular, EUMS† serves as a baseline to highlight the challenges in naively adapting 2D methods to the 3D domain. The zero-shot testing with OpenScene provides instead a reference point, demonstrating the deep scene understanding capabilities of 3D Vision-Language Models and highlighting also the difficulties encountered in their application.

4.1 Adapting NCD for 2D Images to 3D Point Clouds

One of the contributions of this work is to adapt the method proposed by (Zhao et al., 2022) for NCD in 2D semantic segmentation (EUMS) to 3D data. Our empirical evaluation (see Sect. 5) shows that the transposition of EUMS to the 3D domain has some limitations. In particular, as described in Sect. 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

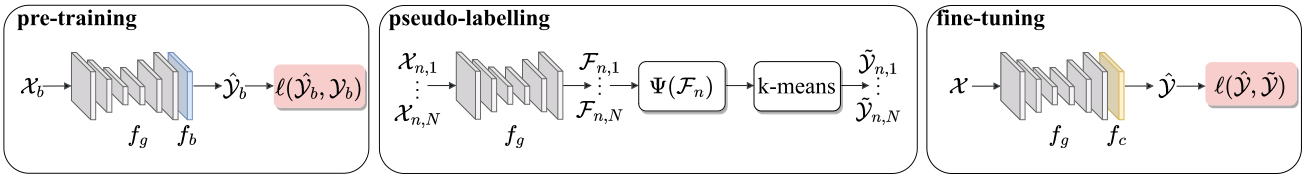

Fig. 5 Overview of EUMS†, our adaptation of the method proposed by (Zhao et al., 2022). We first pre-train f_g and f_b considering only the base points in each point cloud. Using f_g, we extract the features of the novel points in each scene, that are filtered with the selection function $\Psi(\cdot)$. Then, we produce the pseudo-labels for the selected novel points by using the k-means algorithm. Lastly, we plug a new segmentation head f_c into f_g and fine-tune the complete model on both novel and base points, considering pseudo-labels and ground-truth labels respectively

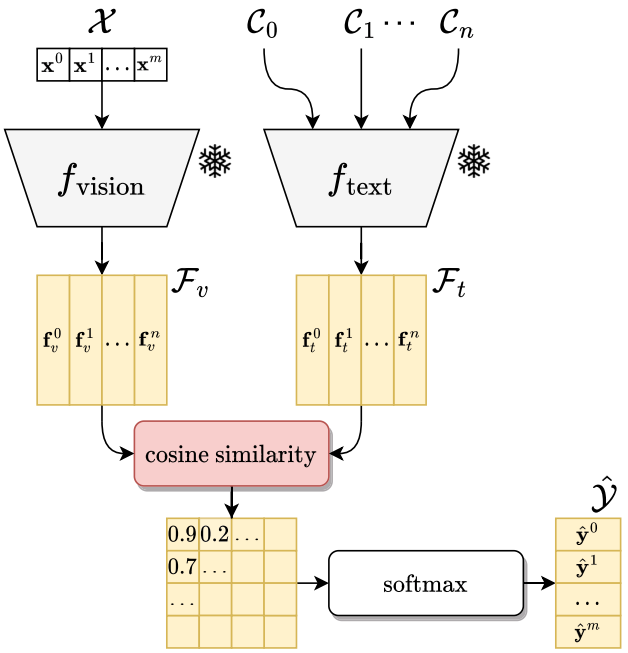

Fig. 6 Zero-shot 3D semantic segmentation. We extract point-level features from the 3D vision encoder f_{vision} and text embedding from the text encoder f_{text}. We assume class vocabularies to be known. Class predictions are assigned by similarity matching between point-level features and text embedding

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 Sect. 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 Fig. 5.

4.2 Zero-Shot Testing with OpenScene

3D Vision-Language Models (3D-VLMs) have shown promising generalization capabilities in scene understanding (Peng et al., 2023), especially in the context of 3D semantic segmentation. Their capabilities can be used off-the-shelf to recognize new objects within scenes by providing the correct text prompt. In this section, we assess the zero-shot semantic segmentation capabilities of the OpenScene models (Peng et al., 2023). Our objective is to evaluate these 3D-VLMs on datasets that differ from the ones they were distilled from. We consider this testing to be our lower bound and with SNOPS we aim to improve over it.

We report in Fig. 6 the implementation of our zero-shot experiment. The input point cloud $\mathcal{X}$ is forwarded into the frozen f_{vision}, resulting in point-wise CLIP-aligned features $\mathcal{F}_{\text{v}}$. Subsequently, the frozen text encoder f_{text} is provided with classical prompts following the template `An image of a <CLASS>` for all the classes present in the dataset. This process yields the output $\mathcal{F}_{\text{t}}$. Finally, we use the pair-wise cosine similarity between each point and class name embedding to assign the predicted classes $\hat{\mathcal{Y}}$.

In its naive version, this experiment would imply the use of dataset class names as input prompts with a simple text template,e.g., `An image of a <CLASS>`. However, we argue that the simple use of the given dataset class names may limit zero-shot capabilities of OpenScene. Dataset names are selected from a (large) set of synonyms with the same meaning, e.g., *person* can also be indicated with the words *pedestrian* or *walker*. Oppositely, the CLIP text encoder has been trained with descriptors that contain different terms for the same concept and consequently it has learnt slightly different embeddings for each of the synonyms of the same concept. So, deriving a single text embedding for each class (e.g. for the dataset class name *person*) does not ensure the coverage of all the variations inside such class (e.g. for *walker* and *pedestrian*). To this extent, we propose to map each class name to four additional synonyms, which capture different meanings under the same class name. Similarly to what proposed for the text templates, we use the five different embeddings for each class as an ensemble. This enable a

better coverage ot the CLIP feature space and increases the zero-shot capabilities of OpenScene. The synonyms for each class are extracted by querying the WordNet dataset (Miller, 1995). When the returned words are not enough, we also make use of *Thesaurus.com* for additional synonyms. To further exploit the sensitivity of f_{text} to text input (Radford et al., 2021), we use an ensemble of text templates to produce robust and stable predictions. In particular, we use the 80 templates originally proposed in (Radford et al., 2021).

Tables 4, 5 & 6 show the OpenScene zero-shot performance on SemanticPOSS, SemanticKITTI, and S3DIS respectively. We report such results as "*OpenScene**" to highlight the usage of the proposed pipeline involving synonyms and templates. On SemanticPOSS, *OpenScene** using ensembles of 3 synonyms reaches 21.18% mIoU, surpassing the standard testing with single class words by around 2.4%. The testing with 5 synonyms shows even better results, showing an overall 23.05% mIoU, with an improvement over the baseline of around 4.3%. On SemanticKITTI, *OpenScene** reaches 19.28% mIoU with 5 synonyms, improving over the testing with 3 synonyms by around 0.7% and gaining 1.0% mIoU over the standard testing with single class words. On S3DIS, we report the results without using ensembles of synonyms, since the furniture class names (e.g. *chair*) of this dataset are too specific to find suitable synonyms for each class label. The zero-shot testing of *OpenScene** on the S3DIS dataset results in an overall 36.76 mIoU.

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*, *veget.*
KITTI-5^1	*car*, *fence*, *other-ground*, *parking*, *trunk*
KITTI-5^2	*motorc.*, *other-v.*, *pole*, *traffic-s.*, *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*

Table 3 S3DIS splits, defined as S3DIS-n^i, where n is the number of novel classes and i is the split index

Split	Novel classes
S3DIS-4^0	*ceiling*, *clutter*, *floor*, *wall*
S3DIS-3^1	*chair*, *door*, *table*
S3DIS-3^2	*beam*, *bookcase*, *column*
S3DIS-3^3	*board*, *sofa*, *window*

5 Experimental Results

5.1 Experiments

Datasets. We evaluate our approach on SemanticKITTI (Behley et al., 2019; Geiger et al., 2012; Behley et al., 2021), SemanticPOSS (Pan et al., 2020) and Stanford 3D Indoor Scene Dataset (S3DIS) (Armeni et al., 2016). SemanticKITTI (Behley et al., 2019) consists of 43,552 point cloud acquisitions with point-level annotations of 19 semantic classes. Based on the conventional benchmark guidelines (Behley et al., 2019), we use sequence 08 for validation and the other sequences for training. SemanticPOSS (Pan et al., 2020) consists of 2,988 real-world point cloud acquisitions with point-level annotations of 13 semantic classes. Based on the conventional benchmark guidelines (Pan et al., 2020), we use sequence 03 for validation and the other sequences for training. S3DIS (Armeni et al., 2016) consists of 271 indoor RGB-D scans with point-level annotations of 13 semantic classes. We follow the official split (Armeni et al., 2016) and use Area_5 for validation and the other areas for training.

Experimental protocol for 3D NCD. Similarly to what proposed by (Zhao et al., 2022) in the 2D domain, we create different splits of each dataset to validate the NCD performance with point cloud data. We create four splits for SemanticKITTI, SemanticPOSS, and S3DIS. We refer to these splits as SemanticKITTI-n^i, SemanticPOSS-n^i, and S3DIS-n^i, where i indexes the split. In each set, the novel classes and the base classes correspond to unlabelled and labelled points, respectively. Tables 1, 2 and 3 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 quantify the performance by using the mean Intersection over Union (mIoU), which is defined as the average IoU across the considered classes (Behley et al., 2019). 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 (Choy et al., 2019). Point-level features are extracted from the penultimate layer. The segmentation heads f_b and f_n are implemented as linear layers, producing output logits for each point in the batched point clouds. The projection head f_s is a sequence of lin-

Table 4 Novel class discovery results on SemanticPOSS. SNOPS outperforms EUMS† and NOPS on all the four splits

Split	Model	bike	build.	car	cone.	fence	grou.	pers.	plant	pole	rider	traf.	trash.	trunk	mIoU Novel	Base	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
	OpenScene* 1 Syn	0.08	50.34	38.71	1.31	6.37	61.12	31.78	48.47	2.45	0.05	0.00	2.74	0.57	–	–	18.77
	OpenScene* 3 Syn	0.06	52.29	39.32	0.34	5.97	61.96	41.79	61.00	3.21	0.00	0.00	4.46	4.98	–	–	21.18
	OpenScene* 5 Syn	0.06	55.60	38.62	0.12	6.63	67.04	42.00	67.81	5.72	2.83	1.61	5.37	6.28	–	–	23.05
POSS-4^0	EUMS†(Zhao et al., 2022)	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 (Riz et al., 2023)	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
	SNOPS (Ours)	34.21	***58.80***	***10.04***	13.20	18.69	***77.25***	45.84	***58.62***	17.27	48.35	22.61	8.72	22.85	***51.18***	25.75	33.57
POSS-3^1	EUMS†(Zhao et al., 2022)	*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 (Riz et al., 2023)	***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
	SNOPS (Ours)	*16.29*	71.37	30.01	19.75	***24.90***	77.14	*54.96*	73.36	15.76	38.43	22.28	15.74	23.59	***32.05***	38.72	37.18
POSS-3^2	EUMS†(Zhao et al., 2022)	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 (Riz et al., 2023)	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
	SNOPS (Ours)	38.37	72.45	27.96	14.47	26.19	78.08	54.73	74.31	*9.99*	48.25	***22.98***	10.16	***17.71***	***16.89***	44.50	38.13
POSS-3^3	EUMS†(Zhao et al., 2022)	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 (Riz et al., 2023)	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
	SNOPS (Ours)	39.40	70.33	30.03	***9.10***	26.84	77.64	54.32	72.54	16.02	***49.89***	28.13	***1.31***	23.51	***20.10***	43.88	38.39
											Avg	EUMS†(Zhao et al., 2022)			*14.94*	37.54	32.24
												NOPS (Riz et al., 2023)			*21.40*	38.23	34.52
												SNOPS (Ours)			***30.05***	38.21	36.82

Full supervision: model trained with labels for base and novel classes. OpenScene*: reference described in Sect. 4.2 ("*n* Syn" indicates the number *n* of synonyms used to build the ensembles). EUMS†: baseline described in Sect. 4.1. Highlighted in *italic* and ***bold italic*** values are the novel classes in each split

ear layer, batch norm, ReLU, and another linear layer. The auxiliary network f_a is the MinkowskiUNet-18 network presented in OpenScene (Peng et al., 2023). We use the version distilled from nuScenes-OpenSeg for SemanticKITTI-n^i and SemanticPOSS-n^i, and the version distilled from ScanNet-OpenSeg for S3DIS-n^i. We train our network for 10 epochs for SemanticKITTI-n^i and SemanticPOSS-n^i, and for 50 epochs for S3DIS-n^i. 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 five 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 and S3DIS-n^i. We set $\gamma = 3.0$ for SemanticKITTI-n^i, and $\gamma = 7.0$ for SemanticPOSS-n^i and S3DIS. We adapted the implementation of the Sinkhorn-Knopp algorithm (Cuturi, 2013) from the code provided by (Caron et al., 2020), with the introduction of the queue and an in-place normalisation steps. Similarly to (Caron et al., 2020), 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

We evaluate SNOPS on both outdoor LiDAR datasets (SemanticPOSS (Pan et al., 2020) and SemanticKITTI (Behley et al., 2019)) and indoor RGB-D datasets (S3DIS (Armeni et al., 2016)). For each setting, we report the upper bound *Full supervision* obtained by supervised training over both base and novel classes. We name with *OpenScene⋆ n Syn.* the zero-shot results achieved by OpenScene as described in Sect. 4.2 using n synonyms when building the ensembles. This baseline is our competitor for the performance achieved on novel classes. EUMS† (Sect. 4.1) and NOPS (Riz et al., 2023) are the NCD approaches that we directly compare against SNOPS.

Outdoor datasets. Tables 4 and 5 report the segmentation results on SemanticPOSS and SemanticKITTI, respectively.

On SemanticPOSS, SNOPS achieves 30.05 IoU on novel classes, improving of +15.11 IoU over EUMS† and +8.65 IoU over NOPS (Table 4). SNOPS outperforms the other methods on all the four dataset splits and on all the classes, except for *bike*, *person*, and *pole*, where NOPS achieves better results. We attribute the significant decline in performance observed in SNOPS for the *bike* class to the alignment procedure with the auxiliary zero-shot model, since the auxiliary zero-shot network exhibits notably poor results on this particular class (0.06 IoU). We consider the decrease in performance for the other two classes (i.e. *person* and *pole)* as simple fluctuations that may happen when SNOPS organizes its feature space differently from the one of NOPS. SNOPS improves over the reference *OpenScene⋆* baseline of +7.00 IoU, outperforming it on all classes, apart for *car*, *plant* and *trashcan*. Interestingly, the *OpenScene⋆* setting outperforms even the *Full supervision* upper bound on the *car* class.

On SemanticKITTI, SNOPS achieves 26.39 IoU on novel classes, improving of +9.34 IoU over EUMS† and +3.55 IoU over NOPS (Table 5). SNOPS outperforms all the compared approaches on all the SemanticKITTI splits, showing a large improvement on novel classes, e.g., *building* and *sidewalk*. Again, SNOPS improves over the reference *OpenScene⋆* baseline of +7.11 IoU, outperforming it on 14 out of 19 classes, surpassing it with a large margin in the classes *bicyclist*, *car*, and *traffic-sign*. Interestingly, SNOPS outperforms the *Full supervision* upper bound on the *traffic-sign* class, with an improvement of +1.81 IoU.

Indoor dataset. Table 6 reports the results on the indoor S3DIS dataset. In this settings, SNOPS achieves 34.05 IoU on novel classes, improving of +24.67 IoU over EUMS† and +13.26 IoU over NOPS. SNOPS outperforms by a large margin EUMS† on all four splits. Compared to NOPS, it improves on three out of four splits, with the remarkably large margins of +29.86 IoU and +23.08 IoU on S3DIS-4^0 and S3DIS-3^1, respectively. Considering the average performance over base and novel classes, SNOPS notably surpasses the results obtained by *Full supervision* on two splits (S3DIS-3^1 and S3DIS-3^3), with 43.45 IoU in average over the four splits (only −0.98 as compared to *Full supervision*).

Discussion. SNOPS consistently outperforms the compared baselines across most of the splits within the three datasets. While the superiority of SNOPS over other NCD methods is empirically clear, understanding the underlying factors contributing to this improvement is essential. Compared to EUMS†, SNOPS achieves superior performance, thanks to online pseudo-labelling (shared with NOPS). This enables precise refinement and adaptation of the model's predictions and leads to better results. Compared to NOPS, SNOPS incorporates an alignment procedure that injects unsupervised semantic knowledge into our architecture. To assess the impact of the semantically-aligned branch, we compare the tSNE (Van der Maaten & Hinton, 2008) dimensionality reduction of the embedding spaces of NOPS and SNOPS, as shown in Fig. 7. We randomly selected eight point clouds from the validation set of each dataset and processed them through the feature extractors of both NOPS and SNOPS, resulting in point-wise features. From these, we retained 5000 random novel points along with their respective features, applied t-SNE reduction, and visualized the points with colors corresponding to their ground-truth labels. As illustrated in Fig. 7, SNOPS exhibits a more refined organization of the feature space, characterized by compact and well-separated class clusters. This contributes to the observed performance enhancement between SNOPS and NOPS. SNOPS also significantly improves over the *OpenScene⋆* baseline on two out of three datasets, highlighting that relying solely on the

Table 5 Novel class discovery results on SemanticKITTI

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
	OpenScene* 1 Syn	0.00	5.20	40.59	55.57	8.12	11.22	0.50	0.05	4.26	0.10	17.16	4.05	62.89	34.74	0.00	0.04	41.39	0.31	61.25	–	–	18.29
	OpenScene* 3 Syn	0.04	0.00	42.56	44.53	8.01	8.55	5.25	0.09	0.27	0.25	23.80	6.19	45.57	36.71	11.48	0.01	41.30	5.48	72.45	–	–	18.55
	OpenScene* 5 Syn	0.18	1.53	48.51	32.89	7.99	9.29	4.34	0.00	0.06	0.22	21.19	8.08	38.20	35.62	34.18	1.36	41.20	6.82	74.60	–	–	19.28
KITTI-5^0	EUMS†(Zhao et al., 2022)	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 (Riz et al., 2023)	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
	SNOPS (Ours)	6.64	43.88	***71.95***	83.34	13.63	24.74	2.47	2.40	15.12	18.67	24.61	31.60	*49.47*	***43.15***	***27.36***	15.68	42.12	38.52	*37.46*	***45.88***	25.96	31.20
KITTI-5^1	EUMS†(Zhao et al., 2022)	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 (Riz et al., 2023)	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
	SNOPS (Ours)	7.58	43.48	85.12	*68.70*	***18.98***	24.42	3.48	*0.00*	23.86	*19.09*	27.00	36.50	89.30	71.92	61.99	17.16	55.85	*29.42*	84.37	***27.24***	45.15	40.43
KITTI-5^2	EUMS†(Zhao et al., 2022)	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 (Riz et al., 2023)	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
	SNOPS (Ours)	6.79	48.30	86.08	89.88	22.20	***9.27***	0.56	3.55	*10.51*	28.35	27.10	*23.81*	90.64	73.79	61.93	***22.31***	*22.11*	46.06	83.76	***17.60***	47.85	39.84
KITTI-4^3	EUMS†(Zhao et al., 2022)	*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 (Riz et al., 2023)	*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
	SNOPS (Ours)	***4.65***	***31.51***	84.55	88.65	22.81	23.28	*8.23*	2.62	17.89	28.69	*15.05*	38.26	89.71	72.48	60.76	16.14	43.34	45.70	82.87	***14.86***	47.85	40.91
																Avg	EUMS†(Zhao et al., 2022)				*17.05*	35.94	32.76
																	NOPS (Riz et al., 2023)				*22.84*	42.39	37.73
																	SNOPS (Ours)				***26.39***	41.69	38.10

SNOPS outperforms EUMS† and NOPS on all four splits. Full supervision: model trained with labels for base and novel classes. OpenScene*: reference described in Sect. 4.2 ("*n* Syn" indicates the number *n* of synonyms used to build the ensembles). EUMS†: baseline described in Sect. 4.1. Highlighted in *italic* and ***bold italic*** values are the novel classes in each split

Table 6 Novel class discovery results on S3DIS

Split	Model	beam	board	book.	ceil.	chair	clutt.	col.	door	floor	sofa	table	wall	wind.	mIoU Novel	Base	All
	Full supervision	0.05	13.46	58.45	75.88	74.72	35.08	22.55	39.62	91.29	21.31	67.55	68.27	9.33	–	–	44.43
	OpenScene* 1 Syn	0.00	0.00	42.36	72.78	56.25	10.81	0.00	47.17	85.53	45.48	42.31	59.24	15.95	–	–	36.76
S3DIS-4^0	EUMS†(Zhao et al., 2022)	0.02	12.15	42.27	*41.65*	59.30	*10.08*	19.85	24.29	*0.23*	26.99	43.50	*3.56*	6.03	*13.88*	26.04	22.30
	NOPS (Riz et al., 2023)	0.04	8.05	52.79	*48.00*	67.39	***20.03***	25.98	36.83	*0.00*	38.12	63.05	*36.28*	6.48	*26.08*	33.19	31.00
	SNOPS (Ours)	0.55	0.12	49.86	***81.01***	72.82	*9.99*	28.48	35.48	***94.39***	43.50	64.37	***38.37***	2.91	***55.94***	33.12	40.14
S3DIS-3^1	EUMS†(Zhao et al., 2022)	0.17	14.06	34.90	72.63	*2.67*	20.11	19.70	*6.74*	87.56	26.05	*19.66*	63.58	3.66	*9.69*	34.24	28.58
	NOPS (Riz et al., 2023)	0.00	8.14	54.44	78.49	*22.49*	37.03	27.81	*19.07*	94.28	55.17	*49.69*	64.34	11.00	*30.41*	43.07	40.15
	SNOPS (Ours)	0.00	8.81	53.78	81.77	***58.14***	36.61	27.62	***42.43***	94.29	57.45	***59.90***	63.52	10.98	***53.49***	43.49	45.79
S3DIS-3^2	EUMS†(Zhao et al., 2022)	*0.03*	3.90	*27.36*	76.43	68.21	23.99	*2.18*	25.07	91.55	34.68	64.51	63.55	0.99	*9.86*	45.29	37.11
	NOPS (Riz et al., 2023)	*0.52*	12.06	***33.89***	73.28	75.53	33.35	*6.56*	30.24	92.99	50.51	68.36	64.99	6.94	*13.66*	50.83	42.25
	SNOPS (Ours)	***0.93***	6.29	*33.34*	79.59	76.69	36.74	***12.26***	32.57	95.91	46.14	67.65	62.92	5.37	***15.51***	50.99	42.80
S3DIS-3^3	EUMS†(Zhao et al., 2022)	0.02	*5.32*	56.68	77.26	72.70	36.71	28.52	45.24	93.64	*4.41*	69.21	59.09	*2.52*	*4.08*	59.91	42.41
	NOPS (Riz et al., 2023)	0.12	*0.29*	54.54	79.44	78.01	38.07	27.68	39.04	95.50	***30.25***	67.77	68.18	***8.44***	***12.99***	54.83	45.18
	SNOPS (Ours)	0.00	***7.26***	56.55	82.90	76.56	36.82	25.87	44.71	96.38	*20.23*	65.61	66.91	*6.24*	*11.24*	55.23	45.08
										Avg	EUMS†(Zhao et al., 2022)				*9.38*	39.87	32.60
											NOPS (Riz et al., 2023)				*20.79*	45.48	39.65
											SNOPS (Ours)				***34.05***	45.71	43.45

SNOPS outperforms EUMS† on all the four splits and NOPS on three out of the four splits. Full supervision: model trained with annotations for base and novel classes. OpenScene*: reference described in Sect. 4.2 ("*n* Syn" indicates the number *n* of synonyms used to build the ensembles). EUMS†: baseline described in Sect. 4.1. Highlighted in *italic* and ***bold italic*** values are the novel classes in each split

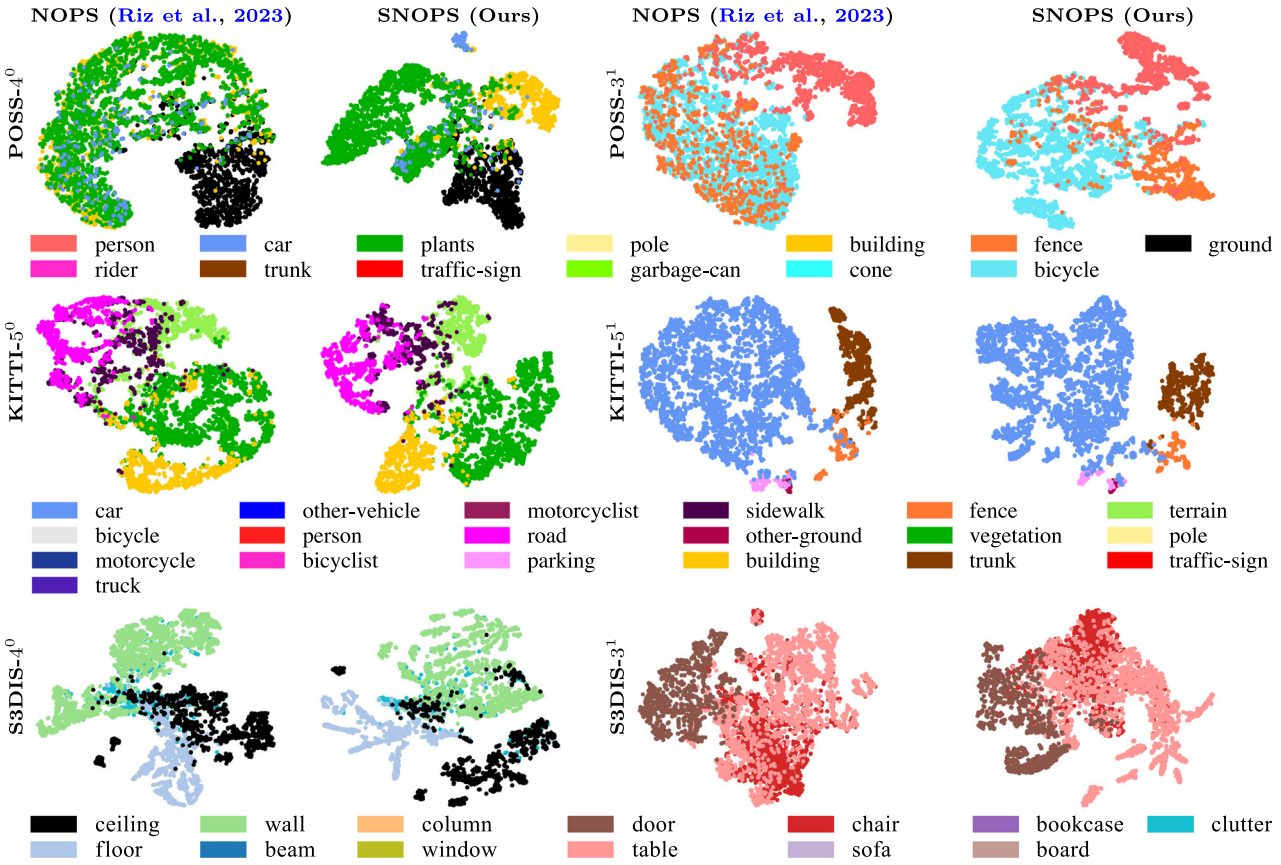

Fig. 7 t-SNE dimensionality reduction of the embedding space output by the feature extractors f_ξ of NOPS and SNOPS for the novel points in different splits of our datasets. Compared to NOPS, SNOPS is able to better organize its embedding space, better grouping features of the novel classes in more compact and separated clusters

zero-shot capabilities of the auxiliary network is not enough for performance. In contrast, SNOPS adeptly integrates the advantages of online pseudo-labelling and semantic alignment, showcasing its ability in making these components work in synergy for superior performance in point cloud semantic segmentation.

In our opinion, the performance of the auxiliary zero-shot network fluctuates significantly based on what we deem as class representation disparities between the distillation and testing datasets; it performs relatively well for well-represented classes but it encounters substantial challenges in accurately identifying rare classes during testing.

Computational time. SNOPS shows a drastic reduction in the computational time when compared to EUMS†. 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, SNOPS 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

Figure 8 depicts segmentation results obtained with SNOPS, NOPS and EUMS† across SemanticPOSS, SemaniticKITTI, and S3DIS datasets. EUMS† faces substantial challenges when it comes to identifying novel objects within scenes, resulting in noisy and mixed labels for these categories. In KITTI-5^1, EUMS† inaccurately labels portions of *car* objects as *trunk*, and in S3DIS-4^1, parts of the *wall* are mislabeled as *table* and *clutter*. NOPS demonstrates improved semantic segmentation capabilities, giving in output more coherent labels and less noisy predictions. However, there are cases in which NOPS exhibits a limited understanding of the scenes. For instance, in POSS-3^2, *trunk* is mixed with *traffic-sign* and *pole*. Moreover, in S3DIS-4^0, NOPS fails in differentiating between the *ceiling* and the *floor* class, likely due to their similar geometric structure. In contrast, SNOPS shows enhanced segmentation and scene understanding capabili-

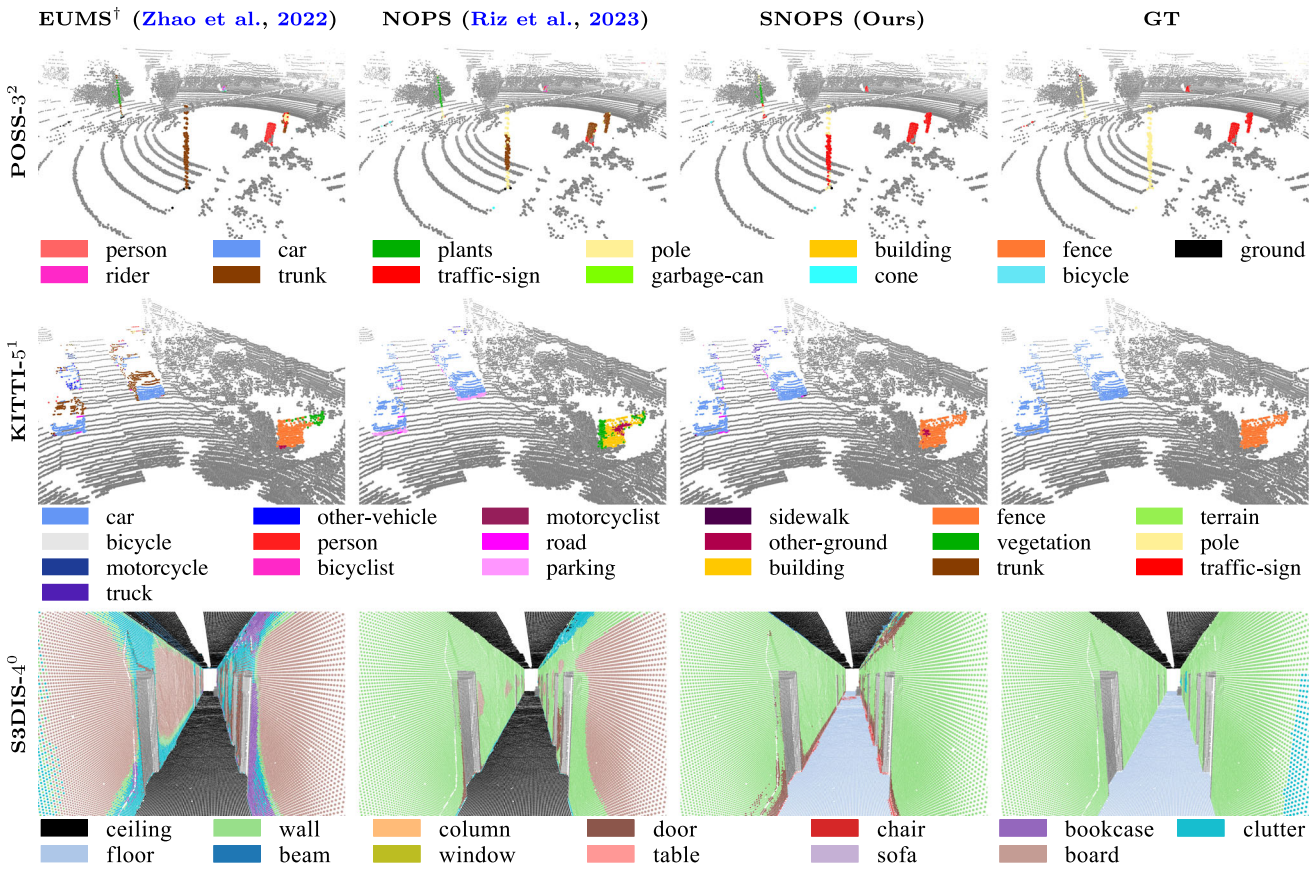

Fig. 8 Qualitative comparisons on SemanticPOSS (top), SemanticKITTI (centre) and S3DIS (bottom). We report results on novel classes. EUMS† fails in recognising novel objects with mixed and noisy predictions, e.g. the *car* class in KITTI-5^1 or the *wall* class in S3DIS-4^0. NOPS shows better segmentation performance on the novel classes, but it still misses a complete knowledge over the meaning of classes, e.g. it mixes *trunk* and *pole* classes in POSS-3^2 or *ceiling* and *floor* in S3DIS-4^0. SNOPS demonstrates superior performances on all three datasets, proving a better understanding of the scene

ties. For example, it proficiently identifies *traffic-sign* and *pole* classes in POSS-3^2 and accurately segments *fence* and *car* in KITTI-5^1. Notably, SNOPS properly distinguishes the *ceiling* from the *floor* class in S3DIS-4^0, thanks to the semantical knowledge acquired through the semantic alignment procedure detailed in Sect. 3.6.

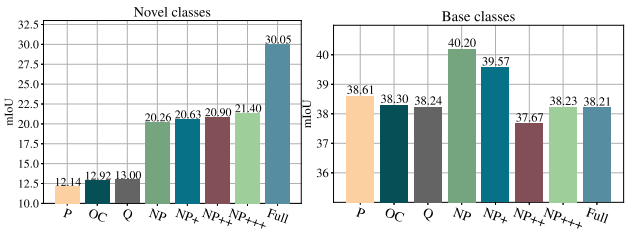

Fig. 9 Ablation study with different components and initialisation strategies on SemanticPOSS. In P, OC and Q, we initialise the model after base pre-training, and use different configurations of the over-clustering heads and of our queue balancing. In NP NP+, NP++ and NP+++, we begin with Q, we avoid pre-training, and we use ϕ and τ_c incrementally. In Full, we add the semantic alignment head. See Sect. 6 for definition of methods

6 Ablation Studies

We thoroughly evaluate SNOPS on SemanticPOSS, conducting an analysis of its core components and exploring how variations in its training parameters affect its performance. Namely, we analyse SNOPS behaviour when changing the value of the percentile p and the semantic alignment loss weighting factor γ, to provide a comprehensive understanding of the efficacy of different part of our architecture.

We also evaluate SNOPS on S3DIS, analysing its stability across different runs, by conducting the same experiments multiple rounds and checking the mean and standard deviation of the obtained results. Lastly, we assess the importance of acquisition sensor and distillation dataset in the comparison between *OpenScene** and SNOPS.

Table 7 Ablation study showing how different values of p affect the performance on SemanticPOSS

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

The lower p is, the less severe the selection of the features, resulting in better performance 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 performance with $p = 0.5$

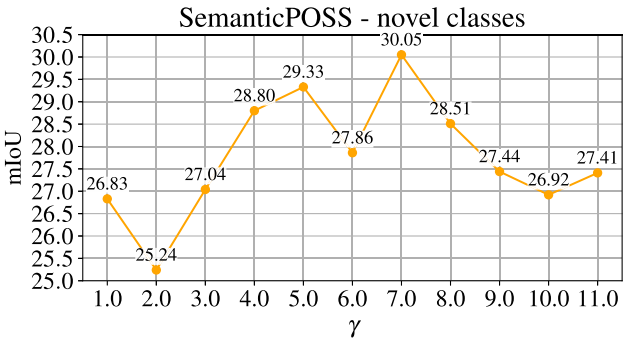

Fig. 10 Ablation study analysing the difference in performance when changing the value of γ, the weighting factor of ℓ_A. Reported performance are in terms of average mIoU of novel classes in the four splits of SemanticPOSS

Method components. Figure 9 shows the performance on novel and base classes of eight versions of SNOPS. The first three versions use the pre-trained model on the base classes, while the last five 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 uncertainty-aware filtering.
- 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.
- NP+++: NP with τ_c and ϕ, without the semantic alignment branch.
- Full: SNOPS 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. With NP+++ we observe a further increase in performance, reaching 21.40 mIoU. The best performance is achieved with Full, with an mIoU of 30.05. Although we can observe variations on the performance of the base classes, their information is retained by the network when we discover the novel categories.

Percentile 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. Table 7 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 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 Table 4. 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.

Loss weighting analysis.

We examine the behavior of SNOPS when adjusting the value of γ, the weighting factor assigned to the alignment loss ℓ_A. The results shown in Fig. 10 depict the average performance across the four SemanticPOSS splits for novel classes

Table 8 Ablation study reporting mean and standard deviation obtained by running the same experiment $N = 10$ times

	S3DIS-4^0			S3DIS-3^1			S3DIS-3^2			S3DIS-3^3		
	Novel	Base	All	Novel	Base	All	Novel	Base	All	Novel	Base	All
μ	55.71	30.82	38.48	52.28	41.48	43.97	18.27	50.92	43.38	10.90	54.80	44.67
σ	0.53	0.76	0.53	0.95	0.42	0.51	1.29	0.48	0.57	1.08	0.39	0.45
$\min_\sigma$	0.14	0.03	0.03	0.95	0.04	0.04	0.20	0.29	0.20	0.62	0.12	0.12
$\max_\sigma$	2.24	2.47	2.47	1.82	3.89	3.89	3.65	3.10	3.65	3.96	2.37	3.96

We report results on novel and base classes, together with the performance on all classes. $\min_\sigma$ and $\max_\sigma$ represent the lowest and the highest standard deviation showcased for the different groups of classes, respectively

as we vary γ. We observe that increasing the value of γ corresponds to higher mIoU values for novel classes. The optimal performance is achieved when γ is set to 7.0. However, further increasing the value of γ leads to a decrease in mIoU values. We attribute this trend to an imbalance between the alignment loss ℓ_A and the segmentation loss ℓ_S. By assigning excessive weight to ℓ_A, there is an indirect reduction in the significance of ℓ_S, which in turn decreases the network's ability to converge to a good solution.

SNOPS's stability. We assess the stability in SNOPS optimisation by running the same experiment $N = 10$ times. Table 8 presents mean μ and standard deviation σ for all the splits in S3DIS. The results show that SNOPS is generally stable across different runs of the same experiment, with an average standard deviation (σ) of 0.52 across the four splits. In three out of four splits, the novel classes exhibit higher standard deviations compared to the base classes. This difference is likely attributed to the fact that the first group of (novel) classes is learned solely with the supervision of pseudo-labels and distillation, whereas the base classes benefit from the availability of labelled data.

Distillation data. In Sect. 5, the *OpenScene** baseline is obtained by testing the OpenScene model on data that differs from the one seen during distillation: we use the ScanNet-OpenSeg model for S3DIS and nuScenes-OpenSeg for SemanticKITTI and SemanticPOSS. In Tables 9 & 10 we report results obtained by testing the *OpenScene** baseline on data which is similar to the one seen during training.

Table 9 reports the results obtained on S3DIS using the Matterport-OpenSeg OpenScene model. The testing dataset (S3DIS) shares the same acquisition sensor as the distillation data (Matterport (Chang et al., 2017)) for the Matterport-OpenSeg model, i.e. the Matterport360 camera. This should reduce the domain gap between training and testing data, resulting in better results for *OpenScene**. Interestingly, on *OpenScene**, the use of the Matterport-OpenSeg model produces very similar results as ScanNet-OpenSeg, with 36.67 average IoU in the first case and 36.76 in the second case. However, when applied to NCD, the use of Matterport-OpenSeg results in worse average results on novel classes, dropping from 34.05 IoU to 32.91.

Table 10 presents a comparative analysis between the baseline method *OpenScene** and SNOPS when applied to the nuScenes dataset (Caesar et al., 2020; Fong et al., 2022). As detailed in Sect. 5.1, we outline in Table 11 the dataset division into splits for the NCD setting. Notably, *OpenScene** achieves 31.81 IoU in its optimal configuration, surpassing the average IoU performance of SNOPS across the four splits on novel classes (20.61). SNOPS only exhibits a superior performance in three specific classes: *barrier*, *bicycle*, and *other-ground*. This experiment underscores that when both distillation and testing are conducted on the same dataset, *OpenScene** demonstrates a very good performance. How-

Table 9 Novel Class discovery results on S3DIS, with two different OpenScene models

Split	Model	beam	board	book.	ceil.	chair	clutt.	col.	door	floor	sofa	table	wall	wind.	mIoU Novel	mIoU Base	mIoU All
	OpenScene* 1 Syn. (ScanNet)	0.00	0.00	42.36	72.78	56.25	10.81	0.00	47.17	85.53	45.48	42.31	59.24	15.95	-	-	36.76
	OpenScene* 1 Syn. (Matterport)	0.00	0.00	41.60	72.78	56.45	10.82	0.00	47.27	85.38	45.18	42.38	59.08	15.76	-	-	36.67
S3DIS-4^0	SNOPS (ScanNet) (Ours)	0.55	0.12	49.86	*81.01*	72.82	***9.99***	28.48	35.48	***94.39***	43.50	64.37	***38.37***	2.91	***55.94***	33.12	40.14
	SNOPS (Matterport) (Ours)	0.68	11.51	52.29	***82.69***	73.16	*4.56*	23.36	24.62	*92.73*	48.70	60.19	*36.55*	7.59	*54.05*	33.57	39.87
S3DIS-3^1	SNOPS (ScanNet) (Ours)	0.00	8.81	53.78	81.77	*58.14*	36.61	27.62	***42.43***	94.29	57.45	***59.90***	63.52	10.98	***53.49***	43.49	45.79
	SNOPS (Matterport) (Ours)	0.00	10.85	54.99	83.95	***60.51***	35.29	29.50	*39.00*	93.36	43.83	*56.77*	62.06	13.93	*52.09*	42.78	44.93
S3DIS-3^2	SNOPS (ScanNet) (Ours)	***0.93***	6.29	*33.34*	79.59	76.69	36.74	***12.26***	32.57	95.91	46.14	67.65	62.92	5.37	*15.51*	50.99	42.80
	SNOPS (Matterport) (Ours)	*0.00*	7.32	***47.34***	71.68	79.25	32.68	*8.63*	31.69	94.91	46.96	67.45	61.77	10.68	***18.65***	50.44	43.10
S3DIS-3^3	SNOPS (ScanNet) (Ours)	0.00	***7.26***	56.55	82.90	76.56	36.82	25.87	44.71	96.38	***20.23***	65.61	66.91	*6.24*	***11.24***	55.23	45.08
	SNOPS (Matterport) (Ours)	0.00	*1.95*	56.07	81.21	75.92	36.25	22.36	44.08	95.30	*11.76*	68.30	66.40	***6.80***	*6.84*	54.59	43.57
										Avg	SNOPS (ScanNet) (Ours)				***34.05***	45.71	43.45
											SNOPS (Matterport) (Ours)				*32.91*	45.34	42.87

OpenScene*: reference described in Sect. 4.2 ("*n* Syn" indicates the number *n* of synonyms used to build the ensembles, (`dataset`) indicates the distillation dataset). SNOPS (`dataset`) is the model with distillation from the OpenScene model trained on `dataset`. Highlighted in *italic* and ***bold italic*** values are the novel classes in each split

Table 10 Novel class discovery results on nuScenes

Split	Model	barr.	bi.cle	bus	car	const.	drivs.	manm.	mt.cle	oth-g.	pede.	sidew.	terr.	tr. c.	trail.	truck	veget.	mIoU Novel	mIoU Base	mIoU All
	OpenScene* 1 Syn	9.92	0.00	41.98	68.61	17.11	79.29	31.92	20.48	0.03	55.71	22.28	0.00	11.62	7.73	46.25	44.92	–	–	28.68
	OpenScene* 3 Syn	9.92	0.00	41.90	62.20	16.85	74.97	55.00	20.27	0.06	42.34	32.06	11.96	9.86	13.09	41.16	77.31	–	–	31.81
	OpenScene* 5 Syn	11.50	0.22	40.98	55.98	17.49	76.58	54.84	21.21	0.13	36.57	29.55	29.27	5.13	14.06	32.10	74.41	–	–	31.25
nuScenes-4^0	SNOPS (Ours)	31.87	5.84	36.51	73.58	8.81	*30.08*	*47.82*	13.86	28.30	21.24	46.18	*14.47*	7.74	17.50	49.94	*69.11*	*40.37*	28.45	31.43
nuScenes-4^1	SNOPS (Ours)	22.62	9.15	43.80	*45.85*	10.25	84.68	82.40	14.61	32.53	29.50	*38.88*	53.81	10.26	22.29	*0.09*	82.77	*26.86*	39.67	36.47
nuScenes-4^2	SNOPS (Ours)	40.42	7.33	*15.71*	79.03	10.80	84.43	80.72	15.28	*16.42*	*13.62*	52.27	51.68	9.40	*0.11*	49.60	82.30	*11.46*	46.94	38.07
nuScenes-4^3	SNOPS (Ours)	39.00	*1.31*	41.76	78.93	*4.24*	87.74	80.03	*4.65*	34.94	29.51	53.31	53.98	*4.86*	21.65	48.20	81.42	*3.76*	54.21	41.60
														Avg	SNOPS (Ours)			*20.61*	42.32	36.89

OpenScene*: reference described in Sect. 4.2 ("*n* Syn" indicates the number *n* of synonyms used to build the ensembles). Highlighted in *italic* values are the novel classes in each split

Table 11 nuScenes splits, defined as nuScenes-n^i, where n is the number of novel classes and i is the split index

Split	Novel Classes
nuScenes-4^0	*driveable s.*, *manmade*, *terrain*, *veget.*
nuScenes-4^1	*barrier*, *car*, *sidewalk*, *truck*
nuScenes-4^2	*bus*, *other g.*, *pedestrian*, *trailer*
nuScenes-4^3	*bicycle*, *constr. v.*, *motorc.*, *traffic c.*

ever, employing distillation on one dataset and evaluating it on another dataset with certain domain gap (as in all our previous experiments) results in a poorer performance. This shows that the distilled open-vocabulary knowledge has limited generalisation capability when tested cross-dataset.

7 Conclusions

We explored the new problem of novel class discovery for 3D point cloud segmentation. Firstly, we adapted the only NCD method for 2D image 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 SNOPS, an extension of our original NOPS method, that tackles NCD for point cloud segmentation by using online clustering, uncertainty quantification and semantic distillation through a foundation model. We showed that the zero-shot accuracy of such foundation model alone is not satisfactory and we proved that by using it in combination with our SNOPS we can achieve higher performance. Lastly, we introduced a novel evaluation protocol to asses the performance of NCD in point cloud segmentation. Experiments on three different segmentation dataset showed that SNOPS outperforms the compared baselines by a large margin.

Limitations The first limitation of SNOPS is the prior knowledge on the number of novel classes C_n to discover. This could be a limitation when C_n is not a known prior and novel classes appear in an incremental manner. We believe that a solution may be to learn novel classes incrementally, as for example proposed by Roy et al. (2022) in the 2D Novel Class Discovery literature. Finally, SNOPS lacks a mechanism to prevent drift when the auxiliary network outputs inaccurate features for novel classes. SNOPS may benefit the introduction of a filtering mechanism to avoid point features when the auxiliary network exhibits high uncertainty, as for example proposed by Saltori et al. (2022).

References

Achlioptas, P., Diamanti, O., Mitliagkas, I. & Guibas, L. (2018). Learning representations and generative models for 3D point clouds. *International conference on machine learning* (pp. 40–49).

Alonso, I., Riazuelo, L., Montesano, L., & Murillo, A. C. (2020). 3d-mininet: Learning a 2D representation from point clouds for fast and efficient 3D LiDAR semantic segmentation. *IEEE Robotics and Automation Letters, 5*(4), 5432–5439.

Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M. & Savarese, S. (2016). 3D semantic parsing of large-scale indoor spaces. *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1534–1543).

Asano, Y.M., Rupprecht, C. & Vedaldi, A. (2020). Self-labelling via simultaneous clustering and representation learning. *8th International conference on learning representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Gall, J., & Stachniss, C. (2021). Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. *The International Journal of Robotics Research, 40*(8–9), 959–967.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C. & Gall, J. (2019). SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9297–9307).

Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q &. Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11621–11631).

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P. & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems* (Vol. 33, pp. 9912–9924).

Cen, J., Yun, P., Zhang, S., Cai, J., Luan, D., Tang, M. & Yu Wang, M. (2022). Open-world semantic segmentation for LiDAR point clouds. *Proceedings of the european conference on computer vision* (pp. 318–334).

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M. & Zhang, Y. (2017). Matterport3d: Learning from RGB-D data in indoor environments. *International conference on 3D vision.*

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(4), 834–848.

Chen, L-C., Zhu, Y., Papandreou, G., Schroff, F. & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European conference on computer vision* (pp. 801–818).

Cheng, R., Razani, R., Taghavi, E., Li, E. & Liu, B. (2021). (AF)2-S3Net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12547–12556).

Choy, C., Gwak, J. & Savarese, S. (2019). 4d spatio-temporal convnets: Minkowski convolutional neural networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3075–3084).

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* (Vol. 26).

Deng, H., Birdal, T. & Ilic, S. (2018). Ppf-foldnet: Unsupervised learning of rotation invariant 3D local descriptors. *Proceedings of the European conference on computer vision* (pp. 602–618).

Dong, X., Bao, J., Zheng, Y., Zhang, T., Chen, D., Yang, H. & Yu, N. (2023). Maskclip: Masked self-distillation advances contrastive language-image pretraining. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10995–11005).

Fini, E., Sangineto, E., Lathuilière, S., Zhong, Z., Nabi, M. & Ricci, E. (2021). A unified objective for novel class discovery. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9284–9292).

Fong, W. K., Mohan, R., Hurtado, J. V., Zhou, L., Caesar, H., Beijbom, O., & Valada, A. (2022). Panoptic nuscenes: A large-scale benchmark for LiDAR panoptic segmentation and tracking. *IEEE Robotics and Automation Letters, 7*(2), 3795–3802.

Gadelha, M., RoyChowdhury, A., Sharma, G., Kalogerakis, E., Cao, L., Learned-Miller, E. & Maji, S. (2020). Label-efficient learning on point clouds using approximate convex decompositions. *Proceedings of the european conference on computer vision* (pp. 473–491).

Geiger, A., Lenz, P. & Urtasun, R. (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. *2012 ieee conference on computer vision and pattern recognition* (pp. 3354–3361).

Giuliari, F., Skenderi, G., Cristani, M., Wang, Y. & Del Bue, A. (2022). Spatial commonsense graph for object localisation in partial scenes. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 19518–19527).

Graham, B., Engelcke, M. & Van Der Maaten, L. (2018). 3D semantic segmentation with submanifold sparse convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9224–9232).

Graham, B., & van der Maaten, L. (2017). *Submanifold sparse convolutional networks.*

Guzhov, A., Raue, F., Hees, J. & Dengel, A. (2022). Audioclip: Extending clip to image, text and audio. *ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 976–980).

Han, K., Vedaldi, A. & Zisserman, A. (2019). Learning to discover novel visual categories via deep transfer clustering. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 8401–8409).

Hinton, G.E., & Salakhutdinov, R.R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507

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

Huang, S., Xie, Y., Zhu, S-C. & Zhu, Y. (2021). Spatio-temporal self-supervised representation learning for 3D point clouds. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6535–6545).

Jatavallabhula, K., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Li, S. & Torralba, A. (2023). Conceptfusion: Open-set multimodal 3D mapping. *Proceedings of robotics: Science and systems.*

Ji, X., Henriques, J.F. & Vedaldi, A. (2019). Invariant information clustering for unsupervised image classification and segmentation. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9865–9874).

Jia, X., Han, K., Zhu, Y. & Green, B. (2021). Joint representation learning and novel category discovery on single-and multi-modal data. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 610–619).

Jiang, H., Shen, Y., Xie, J., Li, J., Qian, J. & Yang, J. (2021). Sampling network guided cross-entropy method for unsupervised point cloud registration. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6128–6137).

Joseph, K., Paul, S., Aggarwal, G., Biswas, S., Rai, P., Han, K. & Balasubramanian, V.N. (2022). Novel class discovery without forgetting. *Proceedings of the European conference on computer vision* (pp. 570–586).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L. & Girshick, R. (2023). *Segment anything.*

Li, R., Li, X., Fu, C-W., Cohen-Or, D. & Heng, P-A. (2019). Pugan: a point cloud upsampling adversarial network. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7203–7212).

Mei, G., Poiesi, F., Saltori, C., Zhang, J., Ricci, E. & Sebe, N. (2023). Overlap-guided gaussian mixture models for point cloud registration. *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 4511–4520).

Mei, G., Saltori, C., Poiesi, F., Zhang, J., Ricci, E., Sebe, N. & Wu, Q. (2022). Data augmentation-free unsupervised learning for 3D point cloud understanding. *British machine vision conference.*

Milioto, A., Vizzo, I., Behley, J. & Stachniss, C. (2019). Rangenet++: Fast and accurate LiDAR semantic segmentation. *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 4213–4220).

Miller, G. A. (1995). Wordnet: A lexical database for English. *Communications of the ACM, 38*(11), 39–41.

Pan, Y., Gao, B., Mei, J., Geng, S., Li, C. & Zhao, H. (2020). SemanticPOSS: A point cloud dataset with large quantity of dynamic instances. *2020 IEEE intelligent vehicles symposium (iv)* (pp. 687–693).

Pang, Y., Wang, W., Tay, F.E., Liu, W., Tian, Y. & Yuan, L. (2022). Masked autoencoders for point cloud self-supervised learning. *Proceedings of the European conference on computer vision* (pp. 604–621).

Peng, S., Genova, K., Jiang, C.M., Tagliasacchi, A., Pollefeys, M. & Funkhouser, T. (2023). Openscene: 3D scene understanding with open vocabularies. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 815–824).

Poiesi, F., & Boscaini, D. (2022). Learning general and distinctive 3D local deep descriptors for point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(3), 3979–3985.

Poursaeed, O., Jiang, T., Qiao, H., Xu, N. & Kim, V.G. (2020). Self-supervised learning of point clouds via orientation estimation. *2020 international conference on 3D vision (3dv)* (pp. 1018–1028).

Qi, C.R., Su, H., Mo, K. & Guibas, L.J. (2017). Pointnet: Deep learning on point sets for 3D classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 652–660).

Qi, C.R., Yi, L., Su, H. & Guibas, L.J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* (Vol. 30).

Qian, G., Abualshour, A., Li, G., Thabet, A. & Ghanem, B. (2021). Pugcn: Point cloud upsampling using graph convolutional networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11683–11692).

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S. & Sutskever, I. (2021a). Learning transferable visual models from natural language supervision. *International conference on machine learning* (pp. 8748–8763).

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S. & Sutskever, I. (2021b). Learning transferable visual models from natural language supervision. *International conference on machine learning* (pp. 8748–8763).

Riz, L., Saltori, C., Ricci, E. & Poiesi, F. (2023). Novel class discovery for 3D point cloud semantic segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9393–9402).

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted intervention–miccai 2015: 18th international conference, Munich, Germany, October 5-9, 2015, Proceedings, Part iii 18* (pp. 234–241).

Roy, S., Liu, M., Zhong, Z., Sebe, N. & Ricci, E. (2022). Class-incremental novel class discovery. *Proceedings of the european conference on computer vision* (pp. 317–333).

Rozenberszki, D., Litany, O. & Dai, A. (2022). Language-grounded indoor 3D semantic segmentation in the wild. *Proceedings of the European conference on computer vision* (pp. 125–141).

Saltori, C., Galasso, F., Fiameni, G., Sebe, N., Ricci, E. & Poiesi, F. (2022). Cosmix: Compositional semantic mix for domain adaptation in 3D LiDAR segmentation. *Proceedings of the European conference on computer vision* (pp. 586–602).

Saltori, C., Krivosheev, E., Lathuiliére, S., Sebe, N., Galasso, F., Fiameni, G. & Poiesi, F. (2022). Gipso: Geometrically informed propagation for online adaptation in 3D LiDAR segmentation. *Proceedings of the european conference on computer vision* (pp. 567–585).

Sauder, J., & Sievers, B. (2019). Self-supervised deep learning on point clouds by reconstructing space. *Advances in neural information processing systems* (Vol. 32).

Shu, D.W., Park, S.W. & Kwon, J. (2019). 3D point cloud generative adversarial network based on tree structured graph convolutions. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3859–3868).

Song, R., Zhang, W., Zhao, Y., Liu, Y. & Rosin, P.L. (2021). Mesh saliency: An independent perceptual measure or a derivative of image saliency? *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8853–8862).

Souly, N., Spampinato, C. & Shah, M. (2017). Semi supervised semantic segmentation using generative adversarial network. *Proceedings of the IEEE international conference on computer vision* (pp. 5688–5696).

Tang, Y., Wang, J., Gao, B., Dellandréa, E., Gaizauskas, R. & Chen, L. (2016). Large scale semi-supervised object detection using visual and semantic knowledge transfer. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2119–2128).

Thomas, H., Qi, C.R., Deschaud, J-E., Marcotegui, B., Goulette, F. & Guibas, L.J. (2019). Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6411–6420).

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using T-SNE. *Journal of Machine Learning Research, 9*(86), 2579–2605.

Vaze, S., Han, K., Vedaldi, A. & Zisserman, A. (2022). Generalized category discovery. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7492–7501).

Wen, X., Li, T., Han, Z. & Liu, Y-S. (2020). Point cloud completion by skip-attention network with hierarchical folding. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1939–1948).

Wu, B., Wan, A., Yue, X. & Keutzer, K. (2018). Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. *2018 IEEE international conference on robotics and automation (ICRA)* (pp. 1887–1893).

Wu, B., Zhou, X., Zhao, S., Yue, X. & Keutzer, K. (2019). Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. *2019 international conference on robotics and automation (ICRA)* (pp. 4376–4382).

Wu, J., Zhang, C., Xue, T., Freeman, B. & Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *Advances in neural information processing systems* (Vol. 29).

Xiao, A., Huang, J., Guan, D., Zhang, X., Lu, S., & Shao, L. (2023). Unsupervised point cloud representation learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(9), 11321–11339.

Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L. & Litany, O. (2020). Pointcontrast: Unsupervised pre-training for 3D point cloud understanding. *Proceedings of the European conference on computer vision* (pp. 574–591).

Yang, G., Huang, X., Hao, Z., Liu, M-Y., Belongie, S. & Hariharan, B. (2019). Pointflow: 3D point cloud generation with continuous normalizing flows. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 4541–4550).

Yang, J., Ahn, P., Kim, D., Lee, H. & Kim, J. (2021). Progressive seed generation auto-encoder for unsupervised point cloud learning. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6413–6422).

Yang, M., Zhu, Y., Yu, J., Wu, A. & Deng, C. (2022). Divide and conquer: Compositional experts for generalized novel class discovery. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14268–14277).

Yang, Y., Feng, C., Shen, Y. & Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 206–215).

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J. & Lu, J. (2022). Point-bert: Pre-training 3D point cloud transformers with masked point modeling. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 19313–19322).

Yuan, W., Khot, T., Held, D., Mertz, C. & Hebert, M. (2018). PCN: Point completion network. *2018 international conference on 3d vision (3dv)* (pp. 728–737).

Zhang, L., & Qi, G-J. (2020). Wcp: Worst-case perturbations for semi-supervised deep learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3912–3921).

Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B. & Foroosh, H. (2020). Polarnet: An improved grid representation for online LiDAR point clouds semantic segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9601–9610).

Zhang, Z., Girdhar, R., Joulin, A. & Misra, I. (2021). Self-supervised pretraining of 3D features on any point-cloud. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10252–10263).

Zhao, Y., Zhong, Z., Sebe, N. & Lee, G.H. (2022). Novel class discovery in semantic segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4340–4349).

Zhong, Z., Fini, E., Roy, S., Luo, Z., Ricci, E. & Sebe, N. (2021). Neighborhood contrastive learning for novel class discovery. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10867–10875).

Zhong, Z., Zhu, L., Luo, Z., Li, S., Yang, Y. & Sebe, N. (2021). Openmix: Reviving known knowledge for discovering novel visual categories in an open world. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9462–9470).

Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3D object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4490–4499).

Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W. & Lin, D. (2021). Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9939–9948).