Open-set Recognition and its Applications in Computer Vision

by

Jun Cen

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfilment of the Requirements for
the Degree of Doctor of Philosophy
in Individualized Interdisciplinary Program
Robotics and Autonomous Systems

April 2024, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

Jun Cen

3 April 2024

Open-set Recognition and its Applications in Computer Vision

by

Jun Cen

This is to certify that I have examined the above PhD thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Prof. Qifeng Chen, Thesis Supervisor

Prof. Michael Yu Wang, Thesis Co-supervisor

Prof. Huamin Qu, Head of Department

Division of Emerging Interdisciplinary Areas

3 April 2024

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my deepest gratitude to my advisors, Prof. Qifeng Chen and Prof. Michael Yu Wang. It was their generosity that gave me the invaluable opportunity to pursue my doctoral studies at HKUST. They have provided an exceptionally tolerant, cutting-edge, and harmonious research environment, which has been instrumental in enabling me to achieve the research accomplishments I have attained to date.

I would like to express my gratitude to the members of my thesis defense committee, including Prof. Ping Tan, Prof. Jun Ma, and Prof. Haoang Li. I sincerely appreciate their detailed review of my materials and the constructive suggestions they offered during my defense, which have significantly contributed to enhancing the quality of my research.

I would also like to express my sincere thanks to the mentors with whom I had closely research collaboration, including Ziwei Liu, Jianguo Zhang, Chenfei Wu, Nan Duan, Shiwei Zhang, and Deli Zhao. They have expanded my research horizons and dedicated a considerable amount of time to guide me on how to conduct better research. Their insights and support have been invaluable to my academic journey.

Additionally, I am also grateful to my colleagues, whose assistance went beyond academic support to include daily companionship and encouragement. They include Lei Zhang, Jianghua Duan, Haoran Song, Zicheng Kan, Alexander Yu Tse, Qingping Ma, Shuai Liu, Cho Hei Pang, Yipai Du, Di Luan, Kun Zhang, Qi Wang, Haokun Wang, Junhao Cai, Qicheng Wang, Guanlan Zhang, Muleilan Pei, Mingpei Cang, Xiang Wang, Zhiwu Qing, Hangjie Yuan, Jingkang Yang, Kewei Wang, Yizheng Wu, Xingyi Li, Jinglong Yang, Wenjian Huang, Lu Huo, and Li Liu.

Lastly, I wish to extend my heartfelt appreciation to my parents, Jianhua Cen and Hairong Lyu, as well as to my girlfriend, Yixuan Pei. Their love, patience, and understanding have been the bedrock of my Ph.D. journey, providing me with the courage and motivation to persevere and overcome all obstacles. Their unwavering support has been my constant source of strength throughout this endeavor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Open-set Recognition and its Applications in Computer Vision

by Jun Cen

Division of Emerging Interdisciplinary Areas

The Hong Kong University of Science and Technology

Abstract

Current deep learning models are trained to fit the training set distribution. Despite the remarkable advancements attributable to cutting-edge architectural designs, these models cannot inference for out-of-distribution (OOD) samples—instances that diverge from the training set's scope. Unlike humans, who can naturally recognize something that is unknown for themselves, current deep learning models lack this capability. Since it is hard to include all objects of the open world into the training set, how to design an open-set recognition algorithm to detect the OOD samples and reject them is essential. This thesis focuses on studying the open-set recognition and its application in computer vision. Initially, we introduce an open-set 3D semantic segmentation system for the autonomous driving applications. We aim to detect anomalous objects that are not common on the road and not in the training set, as such outliers are critical for the safety of autonomous driving systems. Subsequently, we analyze the open-set problem from the Information Bottleneck perspective, and propose a prototypical similarity learning algorithm to learn more class-specific and instance-specific information for better open-set performance. Ultimately, we deeply analyze a new setting called unified open-set recognition, in which both OOD samples and in-distribution but wrongly-classified samples are supposed to be detected, since the model's predictions of them are wrong.

CHAPTER 1

INTRODUCTION

1.1 Background

Current deep learning models in the computer vision area could achieve remarkable performance on various of tasks, including image classification, object detection, semantic segmentation, etc. The model is first trained on the training set with a loss function, and the model's ability for the specific task is gradually improved with the decreasing of loss. If the test set shares the similar pattern with the training set, which is called closed-set recognition, the trained model will behave well since it is already fitted on the training set.

However, it is difficult to guarantee that the distribution of the test set is similar to the training set in the real-world applications. For example, the autonomous driving dataset usually contains common classes like the car, bus, truck, person, bicycle, etc. The model trained on such dataset is capable to recognize these common classes. However, there are some rare classes that may not in the training set, such as the cone, construction vehicle, animals on the road, etc. These objects are also significant for the safety of the autonomous cars, but they cannot be recognized since they are not in the training set. The classes that are not included in the training set are called out-of-distribution (OOD) classes, and in-distribution (ID) classes refer to the classes in the training set. Open-set recognition is to detect OOD samples and meanwhile give the correct prediction results for ID samples.

The key of open-set recognition is uncertainty estimation. Samples whose uncertainty scores u are higher than a threshold λ are regarded as OOD samples, and vice versa.

$$Y = \begin{cases} OOD & u > \lambda \\ ID & u \leq \lambda \end{cases}. \tag{1.1}$$

Fig. 1.1 gives an example of open-set semantic segmentation in the autonomous driving scenario, in which we include the results of three uncertainty estimation methods, i.e., baseline MSP [1] and our proposed EDS and EDS+MMSP [2]. Several OOD objects are shown in the synthetic images and real images, like helicopter, construction vehicle, duck and toy car. The

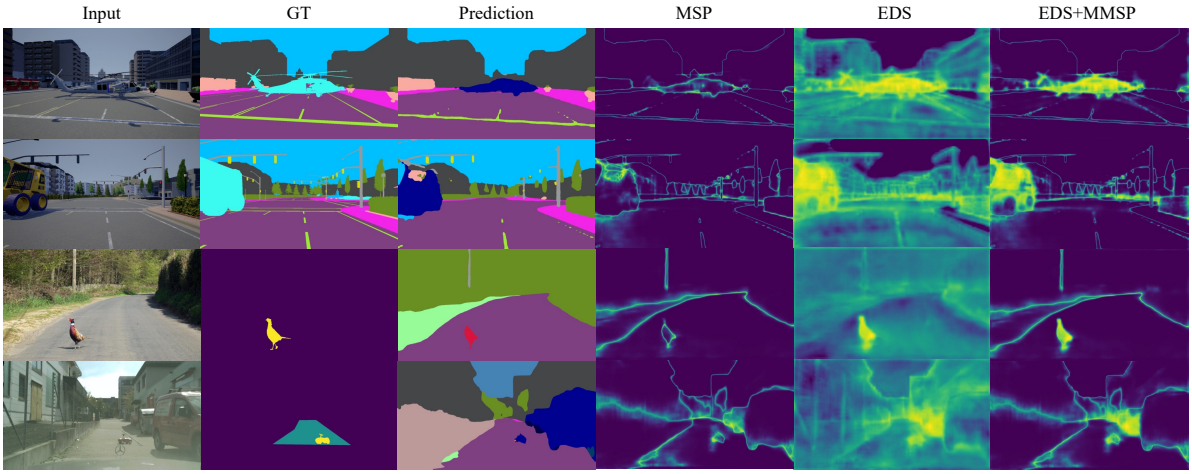


Figure 1.1: Open-set recognition in semantic segmentation.

traditional closed-set segmentation model classifies the OOD classes into one of the ID classes, which is wrong and might cause traffic disasters. In contrast, the open-set recognition model estimates the uncertainty for each pixel to identify the OOD objects.

This thesis focuses on the open-set recognition and its applications. We provide analysis of the open-set recognition problem from different perspectives, including the network design, loss function design, problem setting, dataset setting, and different applications.

1.2 Dissertation Overview

This thesis is organized as follows.

In Chapter 2, we give a comprehensive related work of the open-set recognition. We discuss the classification-based methods, distance-based methods, and reconstruction-based methods. We also include the discussion of open-set recognition with outlier exposure, which contains using real outlier data and generated outlier data.

In Chapter 3, we analyze the open-set 3D semantic segmentation task. We first define the open-world recognition problem in LIDAR semantic segmentation. The open-world segmentation includes the open-set segmentation to identify OOD objects and incremental learning to learn OOD classes without forgetting ID classes. We introduce a redundancy classifier framework for both open-set recognition and incremental learning tasks.

In Chapter 4, we first analyze the open-set recognition problem through the Information Bottleneck theory, and find that the open-set performance is related to the class-specific and instance-specific information. Then we introduce the prototypical similarity learning framework to enhance the class-specific and instance-specific information for better closed-set and open-set

performance.

In Chapter 5, we deeply analyze a new setting called unified open-set recognition. Unlike open-set recognition that only needs to detect OOD samples, unified open-set recognition aims to detect both OOD samples and ID but wrongly-classified samples since the prediction results of them are wrong. We find that OOD samples and ID but wrongly-classified samples have the similar uncertainty distribution. Besides, we explore the effect of outlier exposure and pre-training under the unified open-set recognition setting. Ultimately, we propose a method under the few-shot setting that fully utilizes the given OOD templates during inference.

In Chapter 6, we make a conclusion of this thesis and discuss the future direction of open-set recognition.

CHAPTER 2

RELATED WORK

2.1 Open-set Recognition Method

2.1.1 Classification-based method

The baseline of open-set recognition with the deep learning models was proposed in [1]. They utilized the maximum softmax probability as the confidence score or the additive inverse of the maximum softmax score as the uncertainty score. Two settings are explored in this work. The first one is OOD detection, which is to distinguish OOD samples and ID samples. Another one is mis-classification detection, which is to distinguish ID but wrongly-classified samples and ID and correctly-classified samples. They show that the maximum softmax probability is a simple and effective baseline for OOD detection and mis-classification detection.

However, the maximum softmax probability was found to be over-confident, i.e., some OOD samples also have extremely low uncertainty scores like ID samples [3]. So they proposed to use the maximum logits value as the uncertainty score for better open-set performance. Based on the logits value, [4] proposed to use the energy score for OOD detection, which was theoretically interpreted from a likelihood perspective [5].

ReAct [6] found that a primary factor contributing to the overconfidence problem with OOD data: the application of BatchNorm [7] statistics, which are estimated from ID data, to OOD data during testing. This mismatch can lead to abnormally high unit activation logits and, consequently, inflated model outputs. So they proposed to truncate the logits value which is higher than a threshold, and proven that using the truncated logits for existing uncertainty score functions could further improve the OOD detection performance.

DICE [8] prioritized weights by their level of contribution, selectively employing the most significant ones to compute the output for OOD detection. Filtering out less relevant signals demonstrably narrowed the variance of the output for OOD samples, leading to a more distinct output distribution that is more effectively differentiated from ID samples. Similarly, ASH [9] also focused on the activation space but employed a divergent approach. It discarded a substantial proportion (for instance, 90%) of the feature representations from an input at a late stage

based on a top-K selection process. The method then modified the remaining activations (about 10%) through scaling or assigning fixed values, to achieve results that are surprisingly effective.

All methods mentioned above are post-hoc open-set methods, i.e., there is no need to change the training process and all operations are conducted under the inference stage. Some open-set recognition methods require to modify the training strategy for better OOD detection [10–18]. For example, LC [19] added another small branch to specifically output the uncertainty score. To train the neural network to estimate confidence, they guided it towards the correct output by offering hints when it demonstrates low confidence in its predictions. These hints push the prediction towards the target distribution through interpolation, and the extent was determined by the network's assessed confidence in the accuracy of its prediction.

ODIN [16] introduced a technique of input pre-processing that involves the addition of slight perturbations, which are derived from the gradients of the input. These perturbations are designed to amplify the softmax score of a given input, thereby strengthening the model's confidence in its predicted label. It has been observed that these perturbations effectively widen the disparity between the softmax scores for ID and OOD samples, leading to better OOD detection performance. G-ODIN [20] was built upon ODIN [16] by concentrating on the neural network's intermediate hidden representations to better detect covariate shifts. G-ODIN employed a unique training goal known as decomposing confidence, along with carefully selected hyperparameters such as the degree of perturbation on input data for better OOD detection performance.

Bayesian models utilized Bayes' theorem to encapsulate all forms of uncertainty within the model [21]. A prime example of this approach is the Bayesian neural network [22], which estimated the model's epistemic uncertainty by sampling from its posterior distribution using techniques such as MCMC [23], Laplace approximation [24], and variational inference [25]. Despite their theoretical appeal, limitations such as imprecise predictions and substantial computational demands have hindered their widespread application [26]. To address these issues, recent advancements have explored more pragmatic yet approximate strategies like MC-dropout [27] and deep ensembles [28] to achieve quicker and more accurate uncertainty assessments for OOD detection.

2.1.2 Distance-based method

Distance-based methods are based on the idea that OOD samples are supposed to be significantly distant from the ID class centroids or prototypes. [29] leveraged the minimum Mahalanobis distance from all class centroids for OOD detection. Following this, another study separated

images into foreground and background, computing the Mahalanobis distance ratio between these segments [30]. Diverging from these parametric methods, recent research [31] highlighted the effectiveness of a non-parametric nearest-neighbor distance approach for identifying OOD samples. It calculated the distance between the feature of the test sample with features of all training samples and utilized the minimum one as the uncertainty score. This non-parametric strategy, unlike the Mahalanobis distance, did not rely on assumptions about the feature space's distribution, offering enhanced simplicity, adaptability, and broader applicability.

Various distance functions can be employed for open-set recognition. Certain studies employ cosine similarity between test sample and class features for OOD sample identification [32, 33]. It has been discovered that a one-dimensional subspace, defined by the training features' first singular vector, enhances the efficacy of cosine similarity for OOD detection [34]. Additionally, other research utilized distances based on the radial basis function kernel [35], Euclidean distance [36], and geodesic distance [37] to compare the input's embedding with class centroids. Beyond the distance calculations to class centroids, assessing the feature norm within the orthogonal complement of the principal space has proven to be effective for OOD detection [38]. The recent study CIDER [39] delved into the effectiveness of embeddings within hyperspherical space, aiming to foster between-class dispersion and within-class compactness.

2.1.3 Reconstruction-based method

Reconstruction-based methods operate on the principle that an encoder-decoder framework, when trained on ID data, typically produces different results for ID and OOD samples. Specifically, models trained solely on ID data struggled to accurately reconstruct OOD samples, enabling their identification as OOD [40]. Although pixel-level reconstruction in models is less favored for OOD detection due to high training costs, using hidden features for reconstruction emerges as a viable alternative [41]. Instead of reconstructing the full image, the recent MoodCat model [42] masked parts of the input image and detected OOD samples based on the quality of the reconstruction for classification purposes. The READ method [43] integrated disparities between a classifier's predictions and an autoencoder's outputs, translating pixel reconstruction errors into the classifier's latent space. Additionally, the MOOD approach [44] illustrated the advantage of using masked image modeling over contrastive and traditional classifier training for pretraining in enhancing OOD detection efficiency.

2.2 Open-set Recognition with Outlier Exposure

2.2.1 Real outlier data

Some open-set recognition approaches involve leveraging a collection of known OOD samples, or outliers, during the training phase to enhance the model's ability to distinguish between ID and OOD samples. Such outlier exposure methods could significantly improve the open-set performance because of additional information from the outlier data. Initial methods advocated for inducing models to generate uniform or high-entropy predictions for OOD samples [45] and aimed to reduce the magnitudes of OOD features [46]. Subsequent innovations, such as MCD [47], introduced a dual-branch network design to amplify entropy discrepancies for OOD samples during training. Another method involved an additional class for all recognized OOD samples [48], effectively separating them from ID data.

To manage the large volumes of OOD data during training, strategies such as outlier mining [49, 50] and adversarial resampling [51] have been employed to select the representative subset of outliers. In situations lacking near OOD samples which are similar to ID samples, MixOE [52] suggested creating synthetic outliers through interpolating between ID and distant OOD samples for enhanced regularization. Additionally, considering more realistic scenarios where OOD datasets may inadvertently include ID samples, techniques like pseudo-labeling [53], ID filtering [54] with optimal transport methods [55], are utilized to minimize ID data contamination. Overall, OOD detection methods that incorporate outlier exposure generally achieve superior performance. However, the effectiveness of these approaches can be significantly influenced by the relevance of the chosen OOD samples to actual OOD scenarios [56].

2.2.2 Generated outlier data

In the absence of available OOD samples during the training stage, some methods aim to create synthetic OOD samples to facilitate the distinction between ID and OOD data. Techniques include employing Generative Adversarial Networks (GANs) [57] to produce OOD training examples that encourage uniform model predictions [58], or applying meta-learning to refine sample generation [59]. Nevertheless, generating synthetic images in high-dimensional pixel space poses optimization challenges. To address this, recent methods, such as VOS [60], have introduced the creation of virtual outliers within the feature space's low-likelihood areas, benefiting from the reduced complexity of lower-dimensional spaces. VOS adopted a paramet-

ric method by approximating the feature space with a class-conditional Gaussian distribution, whereas NPOS [61] generated outlier data through a non-parametric approach. Recognizing that synthetic OOD data might be inaccurate or irrelevant, DOE [62] crafted challenging OOD scenarios to refine OOD detection via a min-max learning strategy. Similarly, ATOL [63] employed auxiliary tasks to mitigate errors in OOD generation. [64] proposed a novel technique for crafting unknown objects from real-world videos, leveraging spatial-temporal distillation to enrich training data with previously unrecognized items.

2.3 Open-set Recognition Applications

2.3.1 Open-set semantic segmentation

Open-set semantic segmentation is to detect anomaly objects in an image, which is significant for the autonomous driving system. [65] improved OOD detection using two strategies: First, it leveraged COCO dataset [66] samples as OOD proxies and introduced a secondary goal to increase softmax entropy, enhancing OOD detection across diverse datasets. Second, it applied a post-processing step with linear models on DNN softmax probabilities to reduce false positives, using meta classification. PEBAL [67] combined pixel-wise abstention learning (AL) to adaptively identify anomaly pixels and an energy-based model (EBM) to understand the distribution of normal pixels. It employed joint training of EBM and AL, where EBM flagged anomaly pixels with high energy, and AL applied a reduced penalty for these pixels when categorizing them as anomalies. DenseHybrid [68] introduced a hybrid method that integrated the class posterior, dataset posterior, and an unnormalized data likelihood for anomaly detection. Mask2Anomaly [69] utilized the Mask2former [70] as the backbone which considered the semantic segmentation as the mask-level classification rather than a pixel-level classification problem. It showed that the mask-level uncertainty estimation has better performance than pixel-level uncertainty estimation since they argued that all pixels from an object should have the same uncertainty scores.

Reconstruction-based methods for open-set semantic segmentation [71–74] had the similar pipeline. They first utilized a generative model like GANs to reconstruct the image and then compared the difference between the synthesized image and original image to locate the OOD objects. It is based on the idea that the OOD objects cannot be generated well since they are not in the training set.

In the field of open-set 3D semantic segmentation, we first defined this problem and primar-

ily explored this task in [75]. A method called REAL was proposed in our work to estimate the uncertainty scores using redundancy classifiers. APF [76] followed our work and developed a feature extraction module for extracting point features, a prototypical constraint module, and a feature adversarial module for OOD detection.

2.3.2 Open-set action recognition

DEAR [77] provided a comprehensive analysis of the open-set action recognition challenge, adapting various open-set image recognition techniques to video analysis. These adaptations include methods like SoftMax [1], MC Dropout [27], OpenMax [78], and RPL [79]. According to the benchmarks in [77], only BNN SVI [80] and DEAR [77], their own contribution, are explicitly formulated for video data. BNN SVI applied Bayesian neural networks to open-set action recognition, whereas DEAR employed deep evidential learning [81] for uncertainty estimation and introduces two modules aimed at reducing overconfidence in predictions and mitigating appearance bias. While the focus of existing approaches is on refining uncertainty estimation, our proposed method, PSL [82], emphasized the significance of varied feature representations to enhance the separability of open-set activities.

CHAPTER 3

OPEN-SET 3D SEMANTIC SEGMENTATION VIA REDUNDANCY CLASSIFIERS

3.1 Introduction

3D LIDAR sensors play an important role in the perception system of autonomous vehicles. Semantic segmentation for LIDAR point clouds has grown very fast in recent years [83–86], benefiting from well-annotated datasets including SemanticKITTI [87–89] and nuScenes [90]. However, existing methods for LIDAR semantic segmentation are all *closed-set* and *static*. The closed-set network regards all inputs as categories encountered during training, so it will assign the labels of old classes to OOD classes by mistake, which may have disastrous consequences in safety-sensitive applications, such as autonomous driving [91]. Meanwhile, the static network is constrained to certain scenarios, as it cannot update itself to adapt to new environments. In addition, training from scratch to adapt to new scenes is extremely time-consuming, and the annotations of old classes are sometimes unavailable, due to privacy constrains.

To solve the *closed-set* and *static* problem, we propose the *open-world semantic segmentation* for LIDAR point clouds, which is composed of two tasks: 1) open-set semantic segmentation (OSeg) to assign the *unknown* label to OOD classes as well as to assign the correct labels to old classes, and 2) incremental learning (IL) to gradually incorporate the OOD or novel classes into the knowledge base after labellers provide the labels of novel classes. Fig. 3.1 illustrates an example of open-world semantic segmentation for LIDAR point clouds.

As we are the first to study OSeg task in the 3D LIDAR point cloud domain, we refer to the existing methods in the 2D image domain, which can be divided into two types, generative network-based methods [74, 92, 93] and uncertainty-based methods [1, 27, 94], though none of them can be directly utilized. Generative network-based methods adopt a conditional generative adversarial network (cGAN) [95] to reconstruct the input based on the closed-set prediction results, and assume the OOD regions have a larger difference in appearance between the reconstructed input and original input. However, cGAN is not appropriate for reconstruction of the point cloud as all information is determined by the geometry information, *i.e.*, coordinates

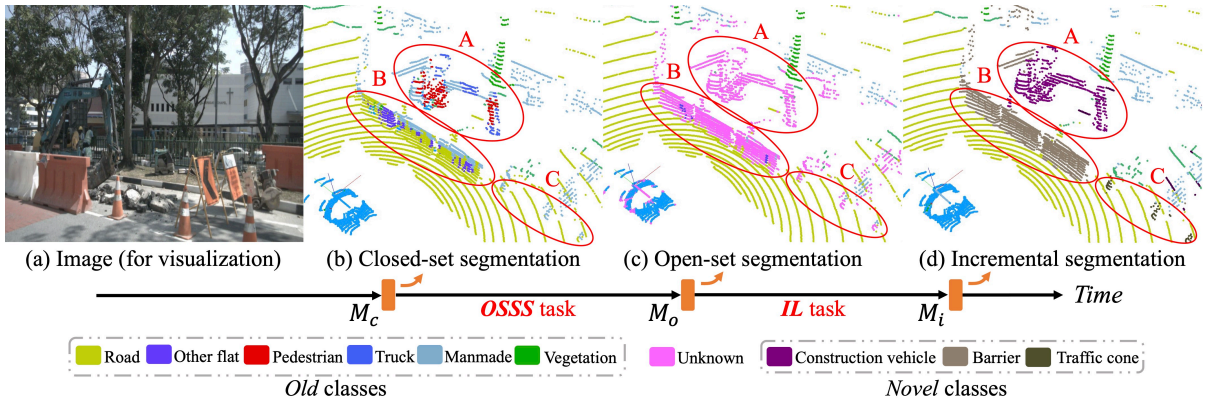


Figure 3.1: Closed-set model $\mathcal{M}_c$ wrongly assigns the labels of old classes to OOD objects (A: construction vehicle is classified as the manmade, truck, and even pedestrian; B: barrier is classified as the road, manmade and other flat; C: traffic cone is classified as the manmade). After open-set semantic segmentation (OSeg) task, the open-set model $\mathcal{M}_o$ can identify the OOD objects and assign the label *unknown* for them. After incremental learning (IL) task, the model $\mathcal{M}_i$ can classify both old and novel classes.

of points, and cGAN can only reconstruct the channel information, *i.e.*, RGB values, while keeping the geometry information, including coordinates of pixels and the shape of an image, unchanged. The uncertainty-based methods also work poorly as we find the network predicts the OOD classes as old classes with high confidence scores.

In addition to the challenges of the OSeg task, the catastrophic forgetting of old classes in incremental learning [96] is another problem to solve. Directly finetuning the network using only the labels of novel classes will make the network classify everything as novel classes. Thus a method is needed to incrementally learn novel classes while keeping the performance of the old classes.

We find that the closed-set and static properties of the traditional closed-set model is due to the fixed classifier architecture, *i.e.*, one classifier corresponds to one old class. Therefore, we propose a **RE**dund**A**ncy c**L**assifier (REAL) framework to provide a dynamic classifier architecture to adapt the model to both the OSeg and IL tasks. For the OSeg task, we add several redundancy classifiers (RCs) on the basis of the original network to predict the probability of the unknown class. Then, during the IL task, several RCs are trained to classify the newly introduced classes, while the remaining RCs are still responsible for the unknown class. We provide the training strategies for the OSeg and IL tasks under REAL, based on the unknown object synthesis, predictive distribution calibration, and pseudo label generation. We show the effectiveness of REAL and corresponding training strategies through our comprehensive experiments. In summary, our contributions are three-folds:

- We are the first to define the open-world semantic segmentation problem for LIDAR point

clouds, which is composed of OSeg and IL tasks;

- We propose a REAL model to provide a general architecture for both the OSeg and IL tasks, as well as training strategies for each task, based on the unknown objects synthesis, predictive distribution calibration, and pseudo labels generation;
- We construct benchmark and evaluation protocols for OSeg and IL in the 3D LIDAR point cloud domain, based on the SemanticKITTI and nuScenes datasets, to measure the effectiveness of our training strategies under REAL.

3.2 Open-world Semantic Segmentation

In this section, we formalise the definition of open-world semantic segmentation for LIDAR point clouds. Let the classes of the training set be called old classes and labeled by positive integers $\mathcal{K}_0 = \{1, 2, ..., C\} \subset \mathbb{N}^+$. Unlike the traditional closed-set semantic segmentation where the classes of the test set are the same as the training set, some novel or OOD classes $\mathcal{U} = \{C+1, ...\}$ are involved in the test set in the open-world semantic segmentation problem. Let one LIDAR point cloud sample be formulated as $\mathcal{D} = \{\mathbf{P}, \mathbf{Y}\}$, where $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_M\}$ is the input LIDAR point cloud composed of M points and every point $\mathbf{p}$ is represented by three coordinates $\mathbf{p} = (x, y, z)$. The label $\mathbf{Y} = \{y_1, y_2, ..., y_M\}$ contains the semantic class for every point, in which $y \in \mathcal{K}_0$ for the training data and $y \in \mathcal{K}_0 \cup \mathcal{U}$ for the test data.

Suppose we already have a model $\mathcal{M}_c$ which is trained under the closed-set condition, so its outputs are within the domain of $\mathcal{K}_0$. As discussed in Sec. 3.1, the open-world semantic segmentation is composed of two tasks: open-set semantic segmentation (OSeg) and incremental learning (IL). For the OSeg task, the model $\mathcal{M}_c$ will be finetuned to $\mathcal{M}_o$ so that it can assign the correct labels for the points of old classes $\mathcal{K}_0$, as well as assign the *unknown* label to the points of OOD classes $\mathcal{U}$. For the IL task, the model $\mathcal{M}_o$ will be further finetuned to $\mathcal{M}_i$ when the labels of OOD or novel classes $\mathcal{K}_n$ are given, so that its knowledge base is enlarged from $\mathcal{K}_0$ to $\mathcal{K}_0 \cup \mathcal{K}_n$, where $\mathcal{K}_n = \{C+1, ..., C+n\}$. So the classes in $\mathcal{K}_n$ change from *unknown* to *known* for the network. We follow the classical task IL setting [97–99] that the new given labels only contain the annotation of the novel class $\mathcal{K}_n$, while the remaining points of old classes $\mathcal{K}_0$ are not annotated. Additionally, the model after IL $\mathcal{M}_i$ still keeps the open-set property, *i.e.*, assigns the *unknown* label to the remaining novel classes $\mathcal{K}_{rn} = \{C+n+1, ...\}$.

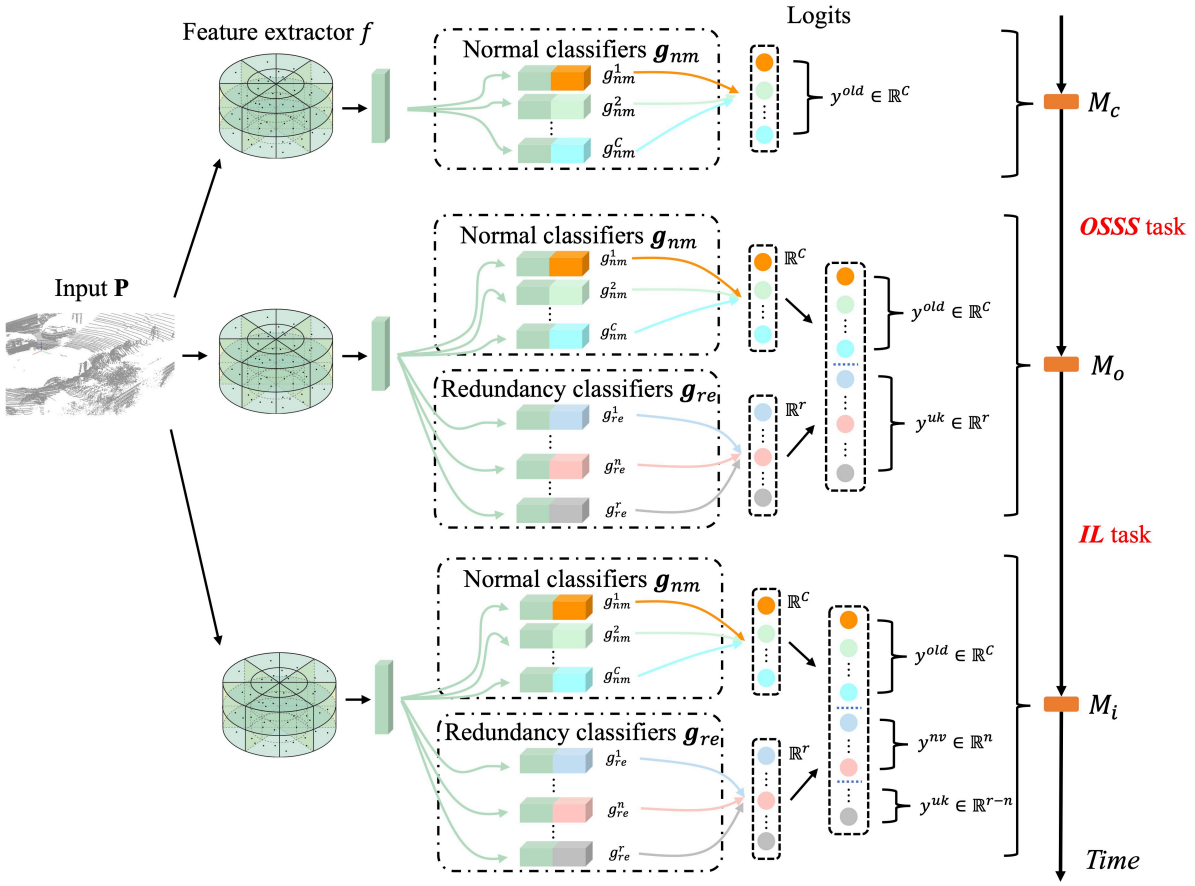


Figure 3.2: Redundancy classifier framework (REAL). Closed-set model $\mathcal{M}_c$ can only output logits for old classes y^{old}. Redundancy Classifiers g_{re} are added on top of the original framework in our REAL. All g_{re} in $\mathcal{M}_o$ are used to output the scores y^{uk} for the unknown class. After the IL task, part of g_{re} are used to output logits for the newly introduced classes y^{nv}, while the remaining are still for the unknown class y^{uk}.

3.3 Redundancy Classifier Framework (REAL)

The overall view of REAL is shown in Fig. 3.2. The trained closed-set model $\mathcal{M}_c$, which can well classify old classes $\mathcal{K}_0$, is composed of a feature extractor f and normal classifiers $g_{nm} = \{g^1_{nm}, g^2_{nm}, ..., g^C_{nm}\}$. For a certain input $\mathbf{P} \in \mathbb{R}^{M\times 3}$, the output of the model $\mathcal{M}_c$ is

$$\mathcal{M}_c(\mathbf{P}) = [y^{old}] = [g_{nm}(f(\mathbf{P}))] \in \mathbb{R}^{M\times C}. \tag{3.1}$$

OSeg task: The OSeg task is to adapt closed-set model $\mathcal{M}_c$ to open-set model $\mathcal{M}_o$ so that $\mathcal{M}_o$ can identify novel or OOD classes $\mathcal{U}$ as *unknown*. To achieve this goal, we add r redundancy classifiers (RCs) $g_{re} = \{g^1_{re}, g^2_{re}, ..., g^r_{re}\}$ on top of the original feature extractor f, as shown in Fig. 3.2 $\mathcal{M}_o$. All RCs in $\mathcal{M}_o$ are used to predict the scores y^{uk} for the unknown class. We let the maximum response of y^{uk} be the score of the unknown class, which is represented by class

0. In this way, the output of the open-set model $\mathcal{M}_o$ is

$$\mathcal{M}_o(\mathbf{P}) = [\max y^{uk}, y^{old}] = [\max\ g_{re}(f(\mathbf{P})), g_{nm}(f(\mathbf{P}))] \in \mathbb{R}^{M\times(1+C)}. \tag{3.2}$$

IL task: The IL task is to train open-set model $\mathcal{M}_o$ to $\mathcal{M}_i$ so that newly introduced classes $\mathcal{K}_n$ change from *unknown* to *known*. $\mathcal{M}_i$ is still open-set, *i.e.*, it can classify remaining novel classes $\mathcal{K}_{rn}$ as *unknown*. In this task, among all RCs g_{re}, some of the RCs $g_{re}^{nv} = \left\{g_{re}^{1}, g_{re}^{2}, ..., g_{re}^{n}\right\}$ are used to classify newly introduced classes $\mathcal{K}_n$, *i.e.*, y^{nv} in Fig. 3.2 $\mathcal{M}_i$, and the remaining RCs $g_{re}^{uk} = \left\{g_{re}^{n+1}, g_{re}^{n+2}, ..., g_{re}^{r}\right\}$ are kept for the unknown class $\mathcal{K}_{rn}$, *i.e.*, y^{uk} in Fig. 3.2 $\mathcal{M}_i$. In this way, the output of $\mathcal{M}_i$ can be represented as

$$\mathcal{M}_i(\mathbf{P}) = [\max y^{uk}, y^{old}, y^{nv}] = [\max\ g_{re}^{uk}(f(\mathbf{P})), g_{nm}(f(\mathbf{P})), g_{re}^{nv}(f(\mathbf{P}))]. \tag{3.3}$$

where $\mathcal{M}_i(\mathbf{P}) \in \mathbb{R}^{M\times(1+C+n)}$.

3.3.1 Open-set Semantic Segmentation (OSeg)

The OSeg task is to train the closed-set model $\mathcal{M}_c$ to the open-set model $\mathcal{M}_o$ which can identify novel classes $\mathcal{U}$ as *unknown*, as shown in Fig. 3.1 (c). The network architecture of $\mathcal{M}_o$ is shown in Fig. 3.2 $\mathcal{M}_o$. We introduce two training methods including *Unknown Object Synthesis* and *Predictive Distribution Calibration* as well as inference procedure in this section.

Unknown Object Synthesis: We synthesize pseudo unknown objects in the LIDAR point cloud to approximate the distribution of real novel objects. The synthesis process should meet two requirements: 1) the synthesized object should share some invariant basic geometry features with existing objects, such as curved and flat surfaces, so that it can be regarded as an *object* rather than noise and possibly have a similar appearance to real unknown objects; 2) the synthesis process should be as quick as possible.

We find that resizing the existing objects with a proper factor is a simple but effective way to conduct the synthesis process, as it keeps the geometric shape of an object, but the different size determines it is a new object. For instance, a car, truck, bus, and construction vehicle have similar local geometric features, such as the shape of the body and tires, but their size can be different. Therefore, we pick up objects of specific old classes $\mathcal{K}_{syn}$ with a probability p_{syn} and resize them from 0.25 to 0.5 times or 1.5 to 3 times as pseudo unknown objects, such as B in Fig. 3.4 (c) and (d). In this way, the input $\mathbf{P}$ is divided into two parts: $\mathbf{P} = \mathbf{P}_{syn} \cup \mathbf{P}_{nm}$, where $\mathbf{P}_{syn}$ and $\mathbf{P}_{nm}$ represent the points of synthesized objects and unchanged normal objects respectively.

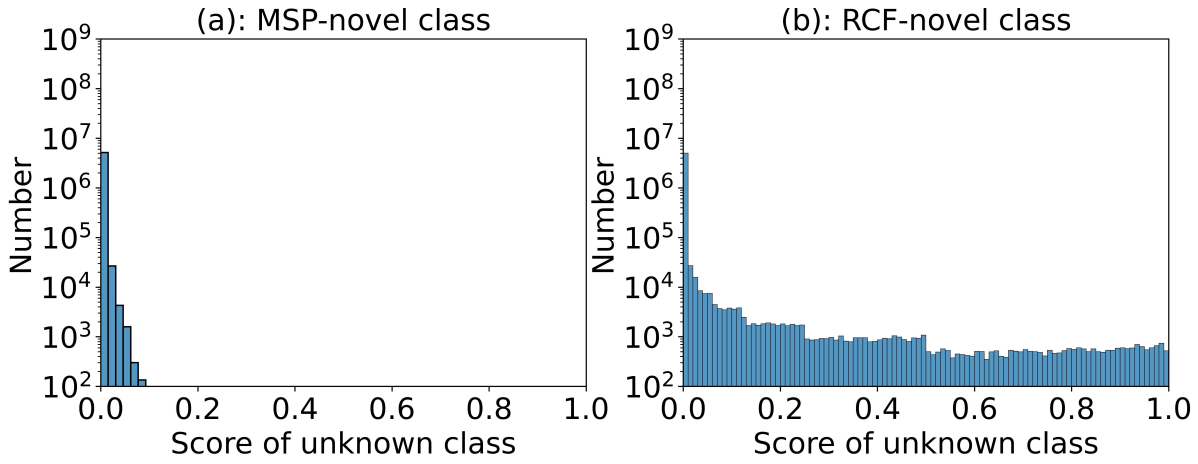


Figure 3.3: Distribution of scores of the unknown class for Maximum Softmax Probability (MSP) and our REAL method. The scores of the unknown class for novel classes are low in MSP (a), meaning the closed-set prediction classifies novel classes as old classes with high confidence.

For the points of synthesized objects $\mathbf{P}_{syn}$, the synthesis loss $\mathcal{L}_{syn}$ is

$$\mathcal{L}_{syn} = \ell(\mathcal{M}(\mathbf{P}_{syn}), \mathbf{0}), \tag{3.4}$$

where ℓ is the cross-entropy loss. The ground truth labels of synthesized objects are set to be the unknown class 0, so the first term in Eq. 3.2 is trained to give high scores to objects never seen before.

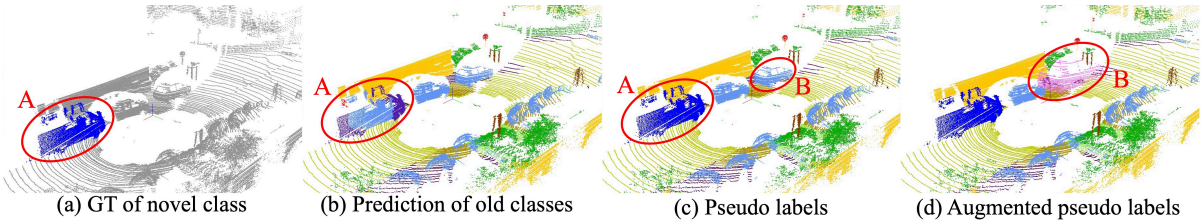
Predictive Distribution Calibration: We find that in the closed-set prediction, the novel objects are classified as old classes with high probability, as shown in Fig. 3.3 (a). We intend to alleviate this problem by probability calibration, and the calibrated scores of the unknown class are shown as Fig. 3.3 (b). We force every point of old classes to have the largest score on its original class, and have the second largest score on the unknown class [100]. By this design, the network is supposed to output high probability scores on the unknown class for the novel objects as they do not belong to any one of the old classes. Therefore, for the points of unchanged normal objects $\mathbf{P}_{nm}$, the calibration loss is designed as

$$\mathcal{L}_{cal} = \mathcal{L}_{cal}^{ori} + \lambda_{cal}\mathcal{L}_{cal}^{uk}, \tag{3.5}$$

where $\mathcal{L}_{cal}^{ori}$ and $\mathcal{L}_{cal}^{uk}$ are defined as

$$\mathcal{L}_{cal}^{ori} = \ell(\mathcal{M}(\mathbf{P}_{nm}), \mathbf{Y}_{nm}), \tag{3.6}$$

$$\mathcal{L}_{cal}^{uk} = \ell(\mathcal{M}(\mathbf{P}_{nm}) \setminus \mathbf{Y}_{nm}, \mathbf{0}), \tag{3.7}$$



(a) GT of novel class (b) Prediction of old classes (c) Pseudo labels (d) Augmented pseudo labels

Figure 3.4: Pseudo labels generating process for incremental learning. Ground truth (a) only contains the label of the novel class (A: other-vehicle). So we combine the prediction results of $\mathcal{M}_o$ (b) to generate the pseudo labels (c). Then we resize objects of old classes as the synthesized objects in (d) (B: resized car).

where $\mathbf{Y}_{nm}$ is the ground truth of $\mathbf{P}_{nm}$. $\mathcal{M}(\mathbf{P}_{nm}) \setminus \mathbf{Y}_{nm}$ means to remove the response of the corresponding ground truth old class. $\mathcal{L}_{cal}^{ori}$ is to ensure the good closed-set prediction, while $\mathcal{L}_{cal}^{uk}$ is to make every point have the second largest probability on the unknown class.

Loss Function: The overall loss function to train the model $\mathcal{M}_c$ to $\mathcal{M}_o$ is

$$\mathcal{L}^{OSeg} = \mathcal{L}_{cal}^{OSeg} + \lambda_{syn}\mathcal{L}_{syn}^{OSeg}, \tag{3.8}$$

where $\mathcal{L}_{cal}^{OSeg}$ is determined by Eq. 3.5, Eq. 3.6, and Eq. 3.7, while $\mathcal{L}_{syn}^{OSeg}$ is determined by Eq. 3.4. All $\mathcal{M}$ in the related terms are $\mathcal{M}_o$ in the OSeg task.

Inference: Both the closed-set and open-set performance of the finetuned model $\mathcal{M}_o$ will be evaluated. For the closed-set prediction, the inference result $\hat{\mathbf{Y}}_{close}$ is defined as

$$\hat{\mathbf{Y}}_{close} = \underset{i=1,2,...,C}{\arg\max}\ g_{nm}(f(\mathbf{P})). \tag{3.9}$$

For the open-set prediction, we have to classify both old classes and the novel class, so the inference result $\hat{\mathbf{Y}}_{open}$ is defined as:

$$\hat{\mathbf{Y}}_{open} = \begin{cases} \underset{i=1,2,...,C}{\arg\max}\ g_{nm}(f(\mathbf{P})) & \lambda_{conf} < \lambda_{th} \\ 0 & otherwise, \end{cases} \tag{3.10}$$

where $\lambda_{conf} = \max\ g_{re}(f(\mathbf{P}))$ is the confidence score of the unknown class, and λ_{th} is the threshold. The unknown class is represented by class 0.

3.3.2 Incremental Learning (IL)

The IL task is to train $\mathcal{M}_o$ to $\mathcal{M}_i$ when the labels of novel classes $\mathcal{K}_n$ are available. $\mathcal{M}_i$ can classify both newly introduced classes $\mathcal{K}_n$ and old classes $\mathcal{K}_0$, as well as identify remaining novel

classes $\mathcal{K}_{rn}$ as *unknown*. The inference example is shown in Fig. 3.1 (d) and the architecture is shown in Fig. 3.2 $\mathcal{M}_i$.

As mentioned in Sec. 3.2, only the labels of introduced novel classes $\mathcal{K}_n$ are given in this task. Therefore, we divide the unchanged normal points $\mathbf{P}_{nm}$ into two parts, $\mathbf{P}_{nm}^{old}$, which belongs to old classes $\mathcal{K}_0$, and $\mathbf{P}_{nm}^{nv}$, which belongs to newly introduced classes $\mathcal{K}_n$, so that $\mathbf{P}_{nm} = \mathbf{P}_{nm}^{old} \cup \mathbf{P}_{nm}^{nv}$. The labels of points $\mathbf{P}_{nm}^{nv}$ are given as $\mathbf{Y}_{nm}^{nv}$, *e.g.*, labels of A in Fig. 3.4 (a), but labels of $\mathbf{P}_{nm}^{old}$ are not given, *e.g.*, gray points in Fig. 3.4 (a). If we only use $\mathbf{Y}_{nm}^{nv}$ to directly finetune the model, it will classify all points as the newly introduced class as there is only one kind of class in the training process. This is called the catastrophic forgetting and we use *Pseudo Label Generation* to solve this problem.

Pseudo Label Generation: We use model $\mathcal{M}_o$ to predict the pseudo labels $\mathbf{pY}_{nm}^{old}$ for $\mathbf{P}_{nm}^{old}$ [2, 99], as shown in Fig. 3.4 (b). In this way, the learned knowledge of old classes is preserved in $\mathbf{pY}_{nm}^{old}$ to alleviate the catastrophic forgetting problem. Then we combine $\mathbf{pY}_{nm}^{old}$ with $\mathbf{Y}_{nm}^{nv}$ to generate the pseudo labels of the whole point cloud $\mathbf{Y}_{nm}$, such as in Fig. 3.4 (c).

Loss Function: Note that we keep the open-set property after IL, so the methods in OSeg task including *Unknown Object Synthesis* and *Predictive Distribution Calibration* are still used in IL task. The overall loss function to train the model $\mathcal{M}_o$ from $\mathcal{M}_i$ is

$$\mathcal{L}^{il} = \mathcal{L}_{cal}^{il} + \lambda_{syn}\mathcal{L}_{syn}^{il}, \tag{3.11}$$

where $\mathcal{L}_{cal}^{il}$ and $\mathcal{L}_{syn}^{il}$ are determined by Eq. 3.5, Eq. 3.6, Eq. 3.7, and Eq. 3.4. All $\mathcal{M}$ in the related terms are $\mathcal{M}_i$. Note that $\mathbf{Y}_{nm}$ in Eq. 3.6 and Eq. 3.7 are generated as

$$\mathbf{Y}_{nm} = \mathbf{pY}_{nm}^{old} \cup \mathbf{Y}_{nm}^{nv}, \tag{3.12}$$

where $\mathbf{Y}_{nm}^{nv}$ is the ground truth label of newly introduced classes $\mathcal{K}_n$ and $\mathbf{pY}_{nm}^{old}$ is the pseudo labels of old classes $\mathcal{K}_0$ generated by $\mathcal{M}_o$,

$$\mathbf{pY}_{nm}^{old} = \mathcal{M}_o(\mathbf{P}_{nm}^{old}). \tag{3.13}$$

The $\mathbf{Y}_{nm}$ in Eq. 3.12 contains both newly introduced classes K_n and old classes K_0, so $\mathcal{M}_i$ can learn new classes without forgetting old classes.

Inference: To evaluate the performance of IL, we only calculate the closed-set prediction results. This is because, for incremental learning we care about how well the catastrophic forgetting problem is alleviated and the new classes are learned, while the ability to classify the

unknown class is already evaluated by Eq. 3.10 in OSeg task, although after IL the model $\mathcal{M}_i$ can still classify the unknown class $\mathcal{K}_{rn}$. The closed-set inference result $\hat{\mathbf{Y}}'_{close}$ is defined as

$$\hat{\mathbf{Y}}'_{close} = \underset{i=1,2,...,C+n}{\arg\max} \; [g_{nm}(f(\mathbf{P}), g^{nv}_{re}(f(\mathbf{P}))]. \tag{3.14}$$

3.4 Experiments

We conduct experiments for both tasks of the open-world semantic segmentation, including OSeg and IL tasks. We evaluate our proposed method on two large-scale datasets, SemanticKITTI and nuScenes.

3.4.1 Open-world Evaluation Protocol

Data Split: We set the novel classes of SemanticKITTI $\mathcal{K}^{sk}_n$ and nuScenes $\mathcal{K}^{ns}_n$ as:

$$\mathcal{K}^{sk}_n = \{other\text{-}vehicle\}$$

$$\mathcal{K}^{ns}_n = \{barrier,\ construction\text{-}vehicle,\ traffic\text{-}cone,\ trailer\}$$

All remaining classes are included in the old class set $\mathcal{K}^{sk}_0$ and $\mathcal{K}^{ns}_0$. During training of the closed-set model $\mathcal{M}_c$ and open-set model $\mathcal{M}_o$, we set the labels of novel classes $\mathcal{K}^{sk}_n$ and $\mathcal{K}^{ns}_n$ to be void and ignore them. During incremental learning, we gradually introduce the labels of novel classes $\mathcal{K}^{sk}_n$ and $\mathcal{K}^{ns}_n$ one by one, and set the labels of old classes $\mathcal{K}^{sk}_0$ and $\mathcal{K}^{ns}_0$ to be void.
Evaluation Metrics: To evaluate the performance of the open-set semantic segmentation model $\mathcal{M}_o$, we consider both the closed-set and open-set segmentation ability. The closed-set ability is measured by $\text{mIoU}_{\text{close}}$, while the open-set evaluation is regarded as a binary classification problem between the known class and unknown class, which is measured by area under the ROC curve (AUROC) and area under the precision-recall curve (AUPR) [3].

To evaluate the performance of the model $\mathcal{M}_i$ after incremental learning, we calculate the performance of the old classes mIoU_{old} and newly introduced classes $\text{mIoU}_{\text{novel}}$ respectively, and also the mIoU of all classes.

3.4.2 Open-set Semantic Segmentation (OSeg)

Implementation: We adopt Cylinder3D [83] as the base network and train the traditional closed-set model $\mathcal{M}_c$ following the training settings in [83] using the labels of old classes $\mathcal{K}^{sk}_0$ and $\mathcal{K}^{ns}_0$.

Table 3.1: Benchmark of open-set semantic segmentation for LIDAR point clouds. Results are evaluated on the validation set.

Dataset	SemanticKITTI			nuScenes		
Methods	AUPR	AUROC	mIoU_{old}	AUPR	AUROC	mIoU_{old}
Closed-set	0	0	58.0	0	0	58.7
Upper bound	73.6	97.1	63.5	86.1	99.3	73.8
MSP	6.7	74.0	58.0	4.3	76.7	58.7
MaxLogit	7.6	70.5	58.0	8.3	79.4	58.7
MC-Dropout	7.4	74.7	58.0	14.9	82.6	58.7
REAL	**20.8**	**84.9**	57.8	**21.2**	**84.5**	56.8

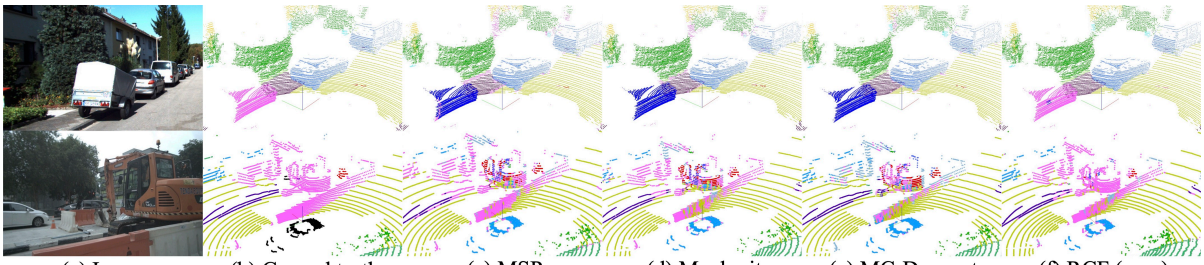


Figure 3.5: Qualitative results of OSeg task. Novel classes are in pink (other-vehicle in SemanticKITTI (top), and construction-vehicle and barrier in nuScenes (bottom)). The results show that our method has a better performance in distinguishing the novel class from old classes than all the baselines. Best viewed in zoom.

Then we add several redundancy classifiers on top of the $\mathcal{M}_0$ and finetune the model $\mathcal{M}_c$ to $\mathcal{M}_o$ following Sec. 3.3.1. The old classes used to synthesize novel objects $\mathcal{K}_{syn}$ are *car* for SemanticKITTI and *car, bus,* and *truck* for nuScenes. The probability of resizing these objects p_{syn} is set to 0.5. The unknown object synthesis time is 0.5-4 *ms* based on our experiments, which is sufficiently quick.

Baselines and Upper Bound: We refer to several methods from the open-set 2D semantic segmentation domain and implement them in our 3D LIDAR points domain as our baselines, including MSP [1], Maxlogit [3], and MC-Dropout [27]. The upper bound is to use labels of all classes $\mathcal{K}_0 \cup \mathcal{K}_n$ to train the network and regard the softmax probability of the classes $\mathcal{K}_n$ as the confidence score.

Quantitative results: The quantitative results of OSeg are shown in Tab. 3.1. The closed-set method does not consider the unknown class at all, so the open-set evaluation metrics are 0. Among all open-set semantic segmentation baselines, our REAL achieves remarkably better results on the open-set evaluation metrics. The closed-set mIoU_{old} shows that our method does

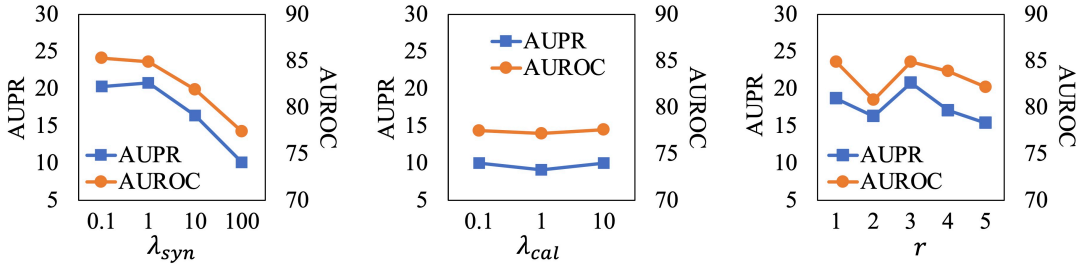


Figure 3.6: Ablation experiments of coefficient λ_{syn}, λ_{cal} and number of redundancy classifiers r for OSeg task on SemanticKITTI.

Table 3.2: Ablation study results of $\mathcal{L}_{cal}$ and $\mathcal{L}_{syn}$ for OSeg task on SemanticKITTI.

Row ID	$\mathcal{L}_{cal}$	$\mathcal{L}_{syn}$	AUPR	AUROC	mIoU_{old}
1	×	×	0	0	58.0
2	✓	×	10.0	77.5	**58.1**
3	✓	✓	**20.8**	**84.9**	57.8

not sacrifice the ability to classify old classes. The upper bound naturally achieves the best performance as it is conducted in a supervised manner, while the information of the unknown class is not provided for other open-set methods.

Qualitative results: Fig. 3.5 contains the qualitative results from SemanticKITTI and nuScenes respectively. Fig. 3.5 top row shows that our method can identify the other-vehicle as the novel class, while all baselines consider it as the truck. In Fig. 3.5 bottom row, the baselines classify the construction-vehicle as the truck, pedestrian, and manmade, while our method distinguishes it as the novel object.

Ablation experiments: We carefully conduct ablation experiments on the SemanticKITTI dataset to verify the effectiveness of our we proposed components. According to the results of Row ID 2 in Tab. 3.2, using the calibration loss alone can already outperforms all baselines in Tab. 3.1. Furthermore, the result of Row ID 3 illustrates that resizing the objects of existing classes with a proper factor is a simple but useful way to imitate novel objects. λ_{syn} and r are set to be 1 and 3 according to Fig. 3.6. λ_{cal} is 0.1, and it does not influence the result with a large margin based on Fig. 3.6.

3.4.3 Incremental Learning (IL)

Implementation: We adopt the training strategies described in Sec. 3.3.2 to finetune the model $\mathcal{M}_o$ to $\mathcal{M}_i$. The old classes used for synthesis $\mathcal{K}_{syn}$ are the same as the set during training from

Table 3.3: Incremental learning results on SemanticKITTI 18 + 1 (other-vehicle) setting.

SemanticKITTI 18+1	Validation set			Test set		
Method	mIoU	$\text{mIoU}_{\text{novel}}$	mIoU_{old}	mIoU	$\text{mIoU}_{\text{novel}}$	mIoU_{old}
Closed-set	58.0	0	61.2	61.8	0	65.3
Train from scratch	63.5	44.1	64.6	62.2	40.1	63.5
Finetune	0	0.5	0	0	0	0
Feature extraction	6.8	0.6	7.1	6.9	0.4	7.3
LwF	21.6	1.7	22.7	20.2	0.9	21.3
REAL	**64.3**	**51.5**	**65.0**	**61.1**	**25.3**	**63.1**

Table 3.4: Incremental learning results on nuScenes for 12 + 4 (barrier, construction-vehicle, traffic-cone, and trailer) setting.

nuScenes 12+4	Validation set			Test set		
Method	mIoU	$\text{mIoU}_{\text{novel}}$	mIoU_{old}	mIoU	$\text{mIoU}_{\text{novel}}$	mIoU_{old}
Closed-set	58.7	0	78.3	55.8	0	74.4
Train from scratch	73.8	62.5	77.6	73.8	70.4	74.8
Finetune	0	0	0	0	0	0
Feature extraction	5.5	2.1	6.6	5.3	1.9	6.4
LwF	6.1	2.4	7.3	5.6	2.5	6.6
REAL	**74.9**	**62.2**	**79.1**	**74.2**	**71.9**	**75.0**

$\mathcal{M}_c$ to $\mathcal{M}_o$.

Baselines and upper bound: We adopt direct finetuning of $\mathcal{M}_o$ to $\mathcal{M}_i$ using only the labels of novel classes $\mathcal{K}_n^{sk}$ and $\mathcal{K}_n^{ns}$ to illustrate the catastrophic forgetting problem. Two methods including Feature Extraction and Learning without Forgetting (LwF) [101] using $\mathcal{K}_n^{sk}$ and $\mathcal{K}_n^{ns}$ are regarded as the baselines. The upper bound is the same as the upper bound in the open-set semantic segmentation task, which uses all labels $\mathcal{K}_0 \cup \mathcal{K}_n$ to train the network.

Quantitative results: Tab. 3.3 and Tab. 3.4 show the IL performance of SemanticKITTI and nuScenes dataset respectively. Directly finetuning the model $\mathcal{M}_o$ to $\mathcal{M}_i$ only using labels of the novel class incurs the catastrophic forgetting problem, *i.e.*, the network classifies all points as the new class. mIoU_{old} becomes 0 as there is no prediction results in old classes. $\text{mIoU}_{\text{novel}}$ is also close to 0 as newly introduced class only counts a little portion in the whole point cloud. In contrast, mIoU_{old} in our method is similar with the closed-set, meaning our method can learn the new classes one by one without forgetting the old classes. Our methods has better perfor-

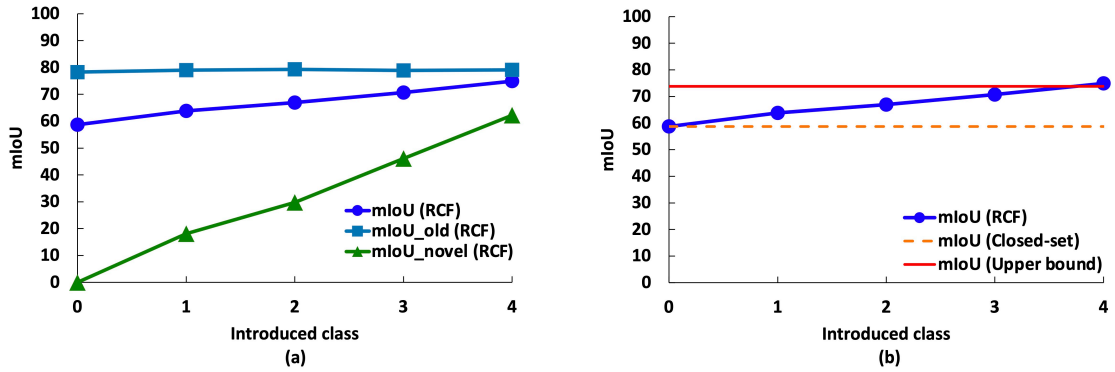


Figure 3.7: Incremental learning results for nuScenes validation set. Introduced class: 1: barrier; 2: construction-vehicle; 3: traffic-cone; 4: trailer.

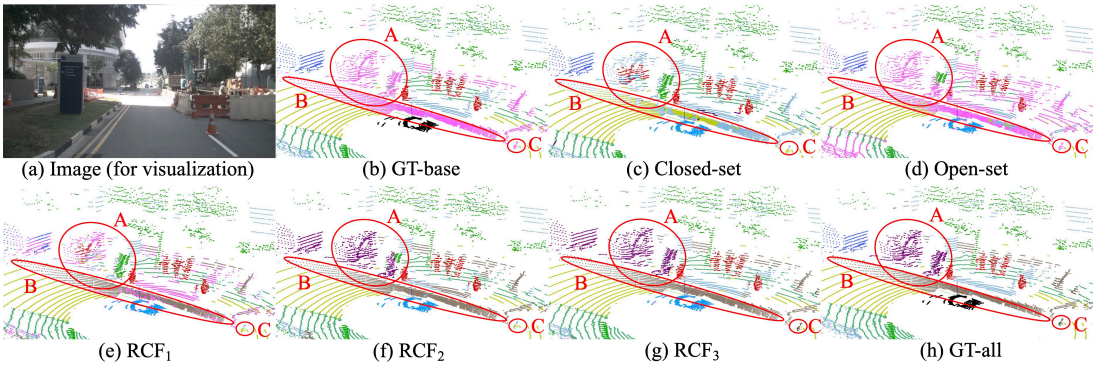


Figure 3.8: Qualitative results of open-world semantic segmentation. GT: ground truth. In (b) GT-base we set the novel classes $\mathcal{K}_n$ in pink (**A**: construction-vehicle; **B**: barrier; **C**: traffic-cone). (c) Closed-set prediction classifies novel objects as old classes. (d) Open-set prediction can identify these novel objects as *unknown*. We gradually introduce the labels of barrier, construction-vehicle, and traffic-cone in (e) REAL$_1$, (f) REAL$_2$, and (g) REAL$_3$, so they can classify these novel classes one by one. (h) GT-all contains ground truth of all classes.

mance compared to two baselines, showing that using the unlabeled background points $\mathbf{Y}_{nm}^{old}$ is extremely helpful to preserve the old knowledge. Compared to training from scratch using all data of old classes and novel classes, our method only needs the ground truth of newly introduced classes $\mathcal{K}_n$ and consumes much less time in training (5 epochs v.s. 35 epochs), while keeping the similar performance.

We show the performance of the model on the nuScenes dataset during IL in Fig. 3.7. Fig. 3.7 (a) shows during IL the model are gradually learning novel classes while keeping the performance of old classes. Fig. 3.7 (b) illustrates the model starts from the closed-set model and finally achieves the comparable performance with the upper bound.

3.4.4 Open-world Semantic Segmentation

We illustrate the whole open-world semantic segmentation system in Fig. 3.8. Traditional closed-set model $\mathcal{M}_c$ classifies objects of novel classes $\mathcal{K}_n$ as old classes $\mathcal{K}_0$. In Fig. 3.8 (c), A (construction vehicle) is classified as manmade, pedestrian, and truck; B (barrier) is classified as road and manmade; C (traffic-cone) is classified as road. Such misclassification may cause serious problems in autonomous driving. Thus we conduct the methods in Eq. 3.8 to finetune $\mathcal{M}_c$ to $\mathcal{M}_o$ so that this open-set model can identify these novel objects as *unknown*, as shown in pink area of Fig. 3.8 (d). Then, after incremental learning using the methods described in Eq. 3.11, the model can gradually classify new classes, *e.g.*, A (barrier), B (construction-vehicle), and C (traffic-cone) in Fig. 3.8 (e), (f), and (g). Note that after incremental learning the model can still identify unknown classes, as shown in the pink areas of Fig. 3.8 (e).

3.5 Conclusion

Traditional closed-set semantic segmentation cannot handle objects of novel classes. In this paper, we propose the open-world semantic segmentation for LIDAR point clouds, where the model can identify novel objects (open-set semantic segmentation) and then gradually learn them when labels are available (incremental learning). We propose the redundancy classifier framework (REAL) and corresponding training and inference strategies to fulfill the open-world semantic segmentation system. We hope this work can draw the attention of researchers toward this meaningful and open problem.

CHAPTER 4

OPEN-SET ACTION RECOGNITION VIA PROTOTYPICAL SIMILARITY LEARNING

4.1 Introduction

Deep learning methods for video action recognition have developed very fast and achieved remarkable performance in recent years [102–105]. However, these methods operate under the *closed-set* condition, *i.e.*, to classify all videos into one of the classes encountered during training. This closed-set condition is not practical in the real-world scenario, as videos whose classes are beyond the range of the training set will be misclassified as one of the known classes. Therefore, *open-set action recognition* (OSAR) is proposed to require the network to correctly classify in-distribution (ID) samples and identify out-of-distribution (OOD) samples. ID and OOD classes refer to classes involved and not involved in the training set, respectively.

Open-set video action recognition is systematically studied in the recent work [77], in which they transfer the existing methods for open-set image recognition into the video domain [1, 27, 78, 106] as the baselines, and propose their own method to introduce deep evidential learning [81] to calculate the uncertainty and propose a contrastive evidential debiasing module to alleviate the appearance bias issue in the video domain. All of these methods tend to improve the OSAR performance by calculating a better uncertainty score, based on the feature representations extracted by the neural network (NN). However, the main purpose of training in these methods is still to classify ID samples, which determines the learned feature representations are merely sufficient for ID classification. We find that almost all methods have a significantly better open-set performance when the NN is pretrained with a large dataset (Fig. 4.1 (a)), so we argue that the diversity of feature representation is extremely important for the OSAR task. Therefore, we propose to boost the open-set ability from the feature representation perspective rather than finding a better uncertainty score.

We first analyze the feature representation behavior in the open-set problem based on the information bottleneck (IB) theory [107, 108]. We divide the information of the feature into *Instance-Specific (IS)* and *Class-Specific (CS) information*. CS information is used for inter-

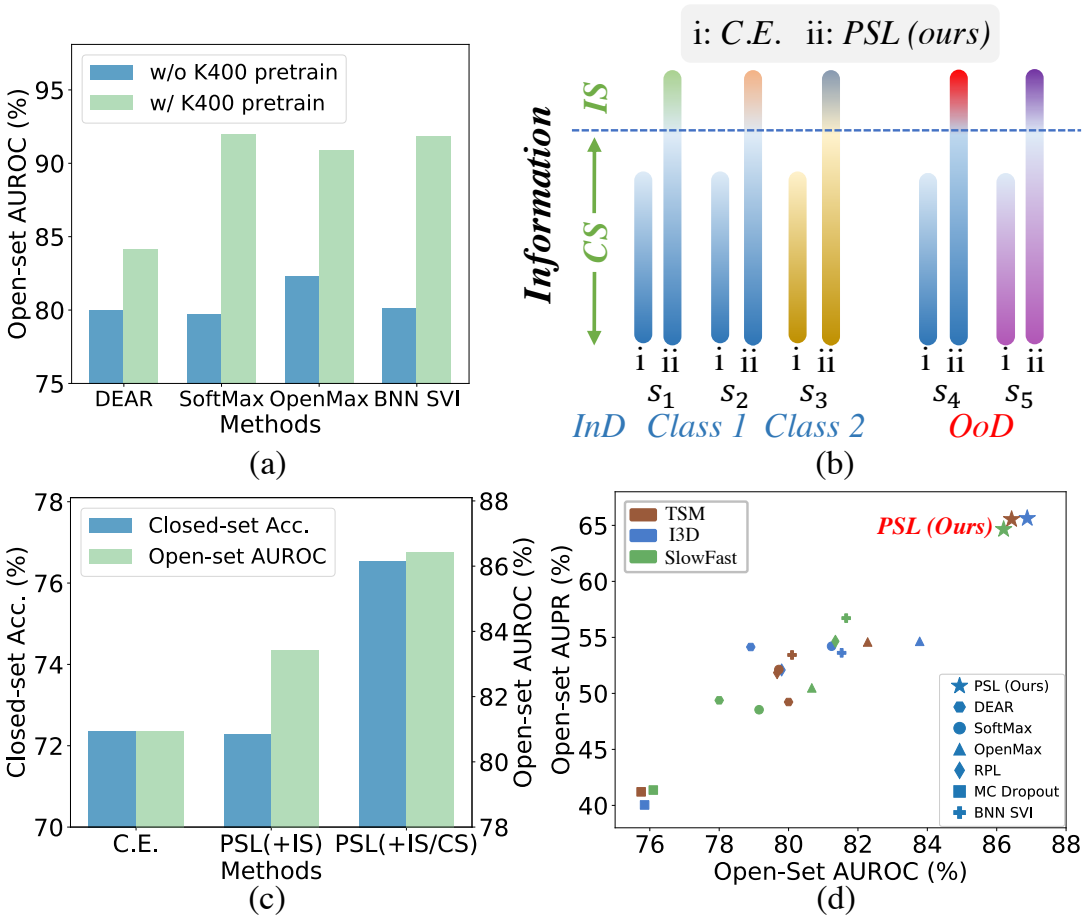


Figure 4.1: (a) Richer semantic features brought by the pretraining can significantly improve the open-set performance. (b) Information in the feature is divided into IS and CS information. s_4 can be identified as OOD since it has distinct IS information (IS bars in different colors) with s_1 and s_2, while s_5 has distinct CS information (CS bars in different colors) with all ID samples so it may be OOD. Our PSL aims to learn more IS and CS information (bars in longer lengths) than Cross-Entropy (C.E.). (c) Both enlarged IS and CS information boosts the open-set performance. (d) Our PSL achieves the best OSAR performance.

class recognition, so it is similar for samples within the same class but different for samples from other classes. IS information is the special information of each sample within the same class, as two samples cannot be exactly the same even if they belong to the same class. Both CS and IS information are crucial for the open-set task, as illustrated in Fig. 4.1 (b), where s_4 and s_5 can be identified as OOD samples based on the IS and CS information, respectively. We find that the closed-set classification setting tends to eliminate IS information during training, and cannot fully extract the minimum sufficient CS information for the classification task, so we aim to enlarge IS and CS information in learned feature representations for better OSAR performance.

To enlarge the IS information, we propose the *Prototypical Similarity Learning* (PSL) framework, in which the representation of an instance is encouraged to have less than 1 similarity with

the corresponding prototype. In this way, we encourage the IS information to be retained and not eliminated. In addition, [77] finds that OOD videos can be easily classified as ID videos in a similar appearance. To alleviate this issue, we introduce the shuffled video into PSL and make it have less than 1 similarity with the original sample. As the shuffled video almost shares the same appearance information with the original one, we encourage the similarity to be less than 1 so that the network can extract the distinct temporal information among them. We find this technique actually enlarges the CS information in the feature representation. Fig. 4.1 (c) shows that enlarging the IS information is helpful for the open-set performance, and more CS information can further benefit the open-set and closed-set performance. To summarize, our contributions include:

- We provide a novel perspective to analyze the open-set recognition task based on the information bottleneck theory, and find that the classical closed-set cross-entropy tends to eliminate the IS information which is helpful to identify OOD samples.
- We propose to enlarge the IS and CS information for better OSAR performance. Specifically, PSL is designed to retain the IS information in the features, and we involve video shuffling in PSL to learn more CS information.
- Experiments on multiple datasets and backbones show our PSL's superiority over a large margin compared to other state-of-the-art counterparts, as shown in Fig. 4.1 (d).

4.2 Information Analysis in OSAR

4.2.1 Prototypical Learning

Let f be the encoder to extract the information for an input video sample x and output the feature representation $z = f(x), z \in \mathbb{R}^d$. We first define a prototypical learning (PL) loss [109], which is a general version of the cross-entropy (C.E.) loss:

$$\mathcal{L}_{PL} = -\log \frac{\exp\left(\frac{z^T k_i}{\tau}\right)}{\exp\left(\frac{z^T k_i}{\tau}\right) + \sum_{n \in K_i^-} \exp\left(\frac{z^T n}{\tau}\right)}, \tag{4.1}$$

where i is the ground truth label of x, $k_i \in \mathbb{R}^d$ is the prototype for class i, τ is a temperature parameter, $K_i^- = \left\{k_j | j \in \{1, 2, ..., N\}, j \neq i\right\}$ is the negative prototype set, and N is the number of ID classes. Note that z and k_i are normalized by L2 norm, so that $z^T k_i$ is the cosine similarity. If we regard prototypes as the row vector of the linear classifier $W \in \mathbb{R}^{N \times d}$, and do

not normalize z and k as well as remove τ, $\mathcal{L}_{PL}$ will degenerate to the C.E. loss. We introduce the $\mathcal{L}_{PL}$ so that we can directly manipulate the feature representation z.

4.2.2 Information Analysis of OSAR

Let x_{ID}, z_{ID}, and Y be the random variables of ID sample, extracted representation of ID sample, and the task to predict the label of x_{ID}, where $z_{ID} = f(x_{ID})$. Given the joint distribution of $p(x_{ID}, Y)$, the relevant information between x_{ID} and Y is defined as $I(x_{ID}, Y)$, where I denotes the mutual information [110]. The learned representation z_{ID} satisfies:

$$I(x_{ID}; z_{ID}) = \underbrace{I(x_{ID}; z_{ID}|Y)}_{IS} + \underbrace{I(z_{ID}; Y)}_{CS}, \tag{4.2}$$

in which $I(x_{ID}; z_{ID}|Y)$ and $I(z_{ID}; Y)$ denote the *Instance-Specific (IS)* and *Class-Specific (CS) information* respectively. In Fig. 4.2, IS information is blue and orange areas, and CS information is yellow and green areas. CS information is for the closed-set label prediction task Y, while IS information is the special information of each sample that is not related to Y.

To analyze the information about OSAR, we let T be a random variable that represents the task to distinguish OOD samples from ID samples, then we divide the information contained in z_{ID} about T into two parts [108]:

$$I(z_{ID}; T) = \underbrace{I(z_{ID}|Y; T)}_{IS \text{ about } T} + \underbrace{I(z_{ID}; Y; T)}_{CS \text{ about } T}, \tag{4.3}$$

where $I(z_{ID}|Y; T)$ and $I(z_{ID}; Y; T)$ are the information about the OOD detection task T in IS and CS information (orange and green areas in Fig. 4.2 respectively). We can see that larger IS and CS information are helpful for OSAR.

In this work, we aim to enlarge the information about T contained in CS and IS information for better OSAR performance, as illustrated in Fig. 4.1 (b) and the enlarged green and orange areas in Fig. 4.2. We first analyze the CS and IS information behaviors under the classical C.E. loss, and find that CS information is encouraged to be maximized but IS information tends to be eliminated in Sec. 4.2.3. Then we explain this conclusion from the IB theory view in Sec. 4.2.4.

4.2.3 CS and IS Information Behavior under C.E.

CS information is for closed-set classification task Y, so it is similar for the same class sample, but distinct for the different class sample (s_1, s_2/s_3 in Fig. 4.1). In contrast, IS information is

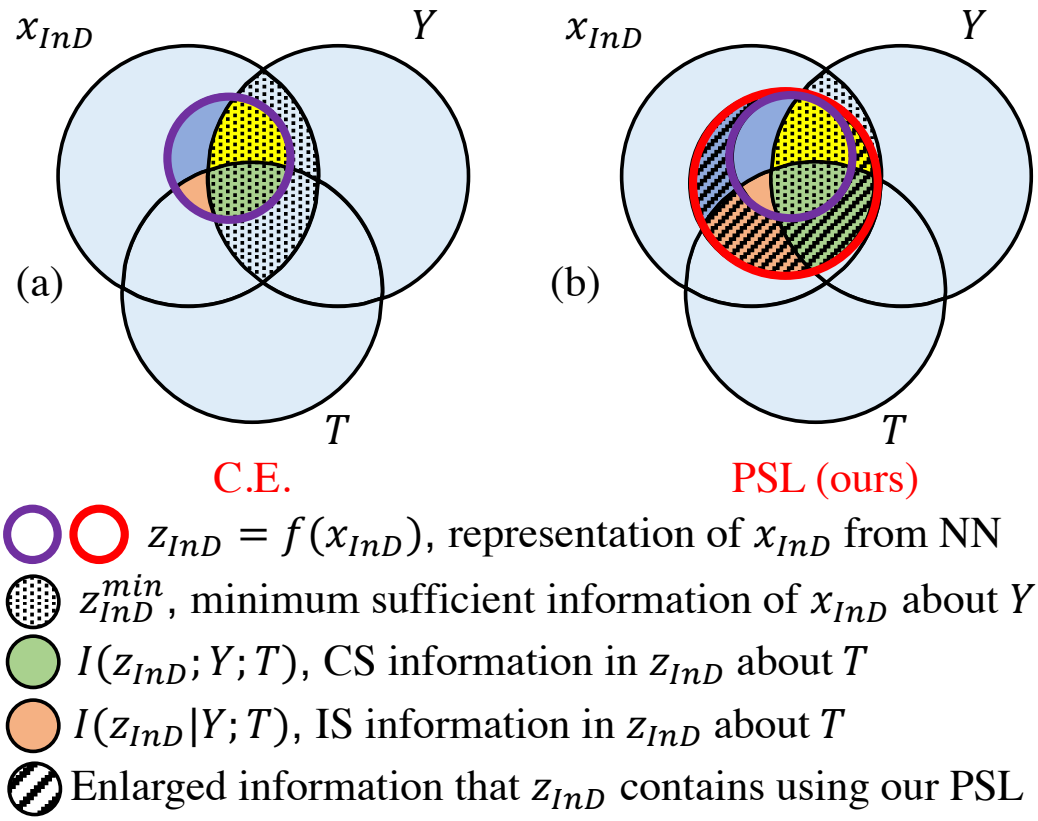


Figure 4.2: The neural network (NN) can only extract limited representations z_{ID} of the ID sample x_{ID} for the current task Y (predict the closed-set label), which is not diverse enough for the task T (distinguish OOD samples), as green and orange areas are small in (a). In our PSL, we encourage the NN to learn a more diverse representation so that more IS and CS information about T are contained.

not related to Y and it is distinct for samples in the same class (s_1, s_2 in Fig. 4.1). Therefore, we have the following proposition which describe the relation between CS/IS information and feature representation similarity.

Proposition 1 For two feature representations of samples in the same class, more CS information means these two feature representations are more similar, and more IS information decreases their feature similarity.

CS information is for the closed-set label prediction task Y, which is fully supervised by C.E. loss, so it is maximized during training. In contrast, Eq. 4.1 shows that C.E. encourages representations of the same class to be exactly same with the corresponding prototype, and such high similarity eliminates the IS information according to Proposition 1. Therefore, C.E. loss tends to maximize the CS information and eliminate the IS information in the feature representation. We analyze this conclusion based on Information Bottleneck (IB) theory in next Sec. 4.2.4.

4.2.4 IB Theory Analysis for CS and IS Information

Applying the Data Processing Inequality [111] to the Markov chain $Y \rightarrow x_{ID} \rightarrow z_{ID}$, we have

$$I(z_{ID}; Y) \leq I(x_{ID}; Y). \tag{4.4}$$

It means that the compressed representation z_{ID} cannot contain more information of Y compared to the original data x_{ID}.

According to the IB theory [107, 110], the NN is to find the optimal solution of z_{ID} with minimizing the following Lagrange:

$$\mathcal{L}[p(z_{ID}|x_{ID})] = I(z_{ID}; x_{ID}) - \beta I(z_{ID}; Y), \tag{4.5}$$

where β is the Lagrange multiplier attached to the constrained meaningful condition. Eq. 4.5 demonstrates the NN is solving a trade-off problem, as the first term tends to keep the information of x_{ID} as less as possible while the second term tends to maximize the information of Y.

Inspired by [108, 112], the sufficient and minimum sufficient representation of x_{ID} about Y can be defined as:

Definition 1 (Sufficient Representation) A feature representation z_{ID}^{suf} of x_{ID} is sufficient for Y if and only if $I(z_{ID}^{suf}; Y) = I(x_{ID}; Y)$.

Definition 2 (Minimum Sufficient Representation) A sufficient representation z_{ID}^{min} of x_{ID} is minimum if and only if $I(z_{ID}^{min}; x_{ID}) \leq I(z_{ID}^{suf}; x_{ID})$, $\forall z_{ID}^{suf}$ that is sufficient for Y.

CS Information Maximization. The goal of training is to optimize f so that $I(z_{ID}; Y)$ (CS information) can approximate $I(x_{ID}; Y)$, which stays unchanged as data distribution is fixed during training. Therefore, CS information is supposed to be maximized to the upper bound $I(x_{ID}; Y)$ because of Eq. 4.4. In this way, the closed-set classification task pushes the NN to learn the sufficient representation z_{ID}^{suf} according to definition 1 [113].

IS Information Elimination. When z_{ID} is close to the sufficient representation z_{ID}^{suf}, the second term in Eq. 4.5 will be the fix value $I(x_{ID}; Y)$ based on the definition 1. So the key to minimize Eq. 4.5 is to minimize the first term $I(z_{ID}^{suf}; x_{ID})$. Based on the definition 2, the lower bound of $I(z_{ID}^{suf}; x_{ID})$ is $I(z_{ID}^{min}; x_{ID})$, so we can conclude that the learned representation is supposed to be the minimum sufficient representation z_{ID}^{min} [108]. We substitute $I(z_{ID}^{suf}; x_{ID})$ and $I(z_{ID}^{min}; x_{ID})$

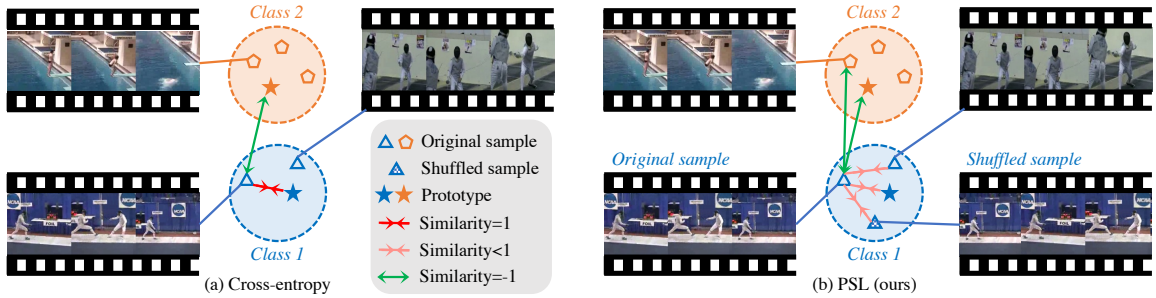


Figure 4.3: (a) C.E. encourages the sample feature z to be exactly same with the corresponding prototype k_i. (b) Our PSL encourages the similarity between z and k_i, features of shuffled sample Q_{shuf} and other samples in the same class Q_{sc} to have a similarity less than 1.

in definition 2 with Eq. 4.2 and we have

$$\begin{aligned} &I(x_{ID}; z_{ID}^{min}|Y) + I(z_{ID}^{min}; Y) \\ \leq &I(x_{ID}; z_{ID}^{suf}|Y) + I(z_{ID}^{suf}; Y). \end{aligned} \tag{4.6}$$

As both z_{ID}^{min} and z_{ID}^{suf} are sufficient, the second term of both sides in Eq. 4.6 is $I(x_{ID}; Y)$, so we have

$$0 \leq I(x_{ID}; z_{ID}^{min}|Y) \leq I(x_{ID}; z_{ID}^{suf}|Y). \tag{4.7}$$

Therefore, the learned IS information in z_{ID}^{min} is smaller than any IS information in z_{ID}^{suf}, which could be eliminated to 0 [108] (no blue and orange areas in z_{ID}^{min} in Fig. 4.2).

4.2.5 Enlarge CS and IS Information for OSAR

Based on the analysis in Sec. 4.2.3 and Sec. 4.2.4, we show that C.E. tends to maximize the CS information and eliminate the IS information in the feature representation. Both larger IS and CS information are crucial for OSAR according to Eq. 4.3, but C.E. does not bring the optimal information. On the one hand, IS information is eliminated so we lose a part of information which is beneficial for the OSAR. On the other hand, the learned representation is not sufficient and does not contain enough CS information in practice due to the model capacity and data distribution shift between training and test sets, which can be supported by the fact that test accuracy cannot reach 100%. Therefore, we propose our method to enlarge the CS and IS information for better OSAR performance in next Sec. 4.3.

4.3 Methods

4.3.1 Prototypical Similarity Learning

According to Sec. 4.2.3, we notice that IS information is suppressed by the C.E. loss and a key reason is C.E. encourages feature representations of the same class to be exactly same. Therefore, we argue that the feature representation of the same class samples should have a similarity $s < 1$. In other words, we aim to keep the intra-class variance which prevents intra-class collapse to retain IS information. Based on the classical PL loss Eq. 4.1, we develop prototypical similarity learning (PSL):

$$\mathcal{L}_{PSL} = -\log \frac{\exp\left(\frac{1-\left|z^T k_i - s\right|}{\tau}\right)}{\exp\left(\frac{1-\left|z^T k_i - s\right|}{\tau}\right) + \sum_{n \in K_i^-} \exp\left(\frac{z^T n}{\tau}\right)}, \tag{4.8}$$

where s and τ are fixed hyperparameters. In this way, we expect the prototype k_i to act as the CS information for the ID class i, which is used to predict the label, and the dissimilarity between the z and k_i represents the IS information. Traditional PL loss (or C.E. loss) encourages the features of samples in the same classes to be as tight as possible, while our PSL aims to keep the variance within the same class.

However, we find Eq. 4.8 will converge to the trivial solution, where the z converges to the training result of Eq. 4.1 and only k_i shifts. To solve this problem, we introduce the similarity between different samples within a mini-batch into the denominator of Eq. 4.8. In this way, we directly constrain the relationship between sample features instead of only supervising the similarity between the sample feature and its prototype. We name the modified loss as PSL with contrastive terms (CT):

$$\mathcal{L}_{PSL}^{CT} = \frac{\exp\left(\frac{1-\left|z^T k_i - s\right|}{\tau}\right)}{\exp\left(\frac{1-\left|z^T k_i - s\right|}{\tau}\right) + \sum_{n \in Q_n} \exp\left(\frac{z^T n}{\tau}\right) + \sum_{p \in Q_{sp}} \exp\left(\frac{\left|z^T p - s\right|}{\tau}\right)}, \tag{4.9}$$

where $Q_n = K_i^- \cup Q_{ns}$. Q_{ns} refers to the negative samples, *i.e.*, samples in other classes, and Q_{sp} refers to the soft positive samples which contains samples in the same class Q_{sc} here. The reason we call soft positive samples is that we think samples in the same class share CS information

but have distinct IS information.

4.3.2 Video Shuffling for PSL

PSL aims to keep IS information during training, and in this section we introduce how to enlarge CS information through video shuffling. The appearance bias is a significant problem in the OSAR. For instance, the OOD classes *Smile* and *Chew* are easily classified as ID classes *ApplyEyeMakeup* and *ApplyLipstick*, as the majority area of all these classes are occupied by a face, as shown in Fig. 4.7. The NN is confused by the extremely similar spatial information and neglects the minor different temporal information. This phenomenon encourages us to strengthen the temporal information extraction ability of the NN to distinguish classes with very similar appearances but different actions. We find that introducing a simple yet effective way, *i.e.*, to regard the shuffled video Q_{shuf} as the soft positive sample in Eq. 4.9, is extremely suitable and useful in our PSL framework. In this case, $Q_{sp} = Q_{sc} \cup Q_{shuf}$. Shuffled video means shuffling the frames within a single video. As the appearance information of the shuffled video is almost the same as the original video, a smaller than 1 similarity forces the NN to learn the distinct temporal information between them. Unlike existing works which predict the sequence or the type of the shuffled video [114–117], we regard the shuffle video as a whole sample and directly compare its feature representation with the original video in our PSL. We find this technique can improve the closed-set accuracy which indicates more CS information is learned. We summarize the difference between our PSL and classical C.E. in Fig. 4.3.

4.3.3 Uncertainty Score

Uncertainty score is to determine whether the current sample is OOD or not based on Eq. 1.1. As our PSL aims to learn richer CS and IS information in the feature representation, we use the Mahalanobis distance to measure the uncertainty as it can be calculated from the feature representation perspective [29, 118]:

$$u = (z - \mu_m)^T \sum_m^{-1} (z - \mu_m), \tag{4.10}$$

where μ_m and $\sum_m$ denote the mean and covariance of the whole training set features, and z is the test sample feature.

Table 4.1: Overlapping classes in HMDB51 and UCF101.

HMDB51	35, Shoot bow	29, Push up	15, Golf	26, Pull up
UCF101	2, Archery	71, PushUps	32, GolfSwing	69, PullUps
HMDB51	30, Ride bike	34, Shoot ball	43, Swing baseball	31, Ride horse
UCF101	10, Biking	7, Basketball	6, BaseballPitch	41, HorseRiding

4.4 Experiments

Datasets. We follow the datasets setting in [77]. The training ID dataset is UCF101, which contains 101 classes with 9537 training samples and 3783 test samples. The OOD datasets for open-set evaluation are HMDB51 and MiT-v2. We use the test sets of them which contain 1530 samples and 30500 samples respectively. For UCF101 and HMDB51, we follow the MMAction [119] to use the split 1 for training and evaluation, which is the same with [77]. Note that in [77], they find some classes in HMDB51 overlap with those in UCF101 but they do not clean them. We remove the overlapping classes in UCF101 and HMDB51 so that OOD data does not contain any samples of ID classes. The classes we remove in HMDB51 and the corresponding same classes in UCF101 are in Tab. 4.1.

Evaluation protocols. For closed-set performance, we evaluate like the traditional way to calculate the top-1 accuracy Acc. (%). For open-set performance, we follow the classical open-set recognition protocol [1, 45] to use the obtained uncertainty score Eq. 4.10 to calculate AUROC (%), AUPR (%) and FPR95(%).[1]

Implementation details. For Kinetics400 (K400) [104] pretrained model, our implementation setting is the same with [77]. The base learning rate is 0.001 and step-wisely decayed every 20 epochs with total of 50 epochs. We argue that as K400 is extremely large, the K400 pretrained model may already have seen the OOD data used in inference, so we conduct experiments from scratch (no ImageNet pretrained) to ensure that OOD data is absolutely unavailable during training. We use the LARS optimizer [120] and set the base learning rate and momentum as 0.6 and 0.9 with total of 400 epochs. The experiments are conducted on TSM [102], I3D [104] and SlowFast [103]. The batch size for all methods is 256. More details are in Appendix C.

[1] We find AUROC in [77] only considers one specific threshold based on their code, and after discussion and agreement they provide the modified correct score in our Tab. 4.2.

Table 4.2: Comparison with state-of-the-art methods on HMDB51 and MiTv2 (OOD) using TSM backbone. Acc. refers to closed-set accuracy. AUROC, AUPR and FPR95 are open-set metrics. Best results are in **bold** and second best results in *italic*. DEAR and our methods contain video-specific operation.

		w/o K400 Pretrain				w/ K400 Pretrain			
Datasets	**Methods**	**AUROC↑**	**AUPR↑**	**FPR95↓**	**Acc.↑**	**AUROC↑**	**AUPR↑**	**FPR95↓**	**Acc.↑**
UCF101 (ID) HMDB51 (OOD)	OpenMax [78]	*82.28*	*54.59*	*50.69*	*73.92*	90.89	73.16	38.77	95.32
	MC Dropout [27]	75.75	41.21	54.78	73.63	88.23	67.62	38.12	95.06
	BNN SVI [80]	80.10	53.43	52.33	71.51	*91.81*	*79.65*	31.43	94.71
	SoftMax [1]	79.72	52.13	53.22	*73.92*	91.75	77.69	*28.60*	95.03
	RPL [79]	79.67	51.85	56.40	71.46	90.53	77.86	37.09	*95.59*
	DEAR [77]	80.00	49.23	53.28	71.33	84.16	75.54	89.40	94.48
	PSL(ours)	**86.43**	**65.54**	**41.67**	**76.53**	**94.05**	**86.55**	**23.18**	**95.62**
UCF101 (ID) MiTv2 (OOD)	OpenMax [78]	*84.43*	*76.69*	*47.74*	*73.92*	*93.34*	88.14	*28.95*	95.32
	MC Dropout [27]	75.66	62.20	51.57	73.63	88.71	83.36	39.46	95.06
	BNN SVI [80]	79.48	71.73	52.52	71.51	91.86	*90.12*	36.21	94.71
	SoftMax [1]	80.55	73.17	50.49	*73.92*	91.95	89.16	32.00	95.03
	RPL [79]	80.21	72.04	52.83	71.46	90.64	88.79	38.43	*95.59*
	DEAR [77]	79.00	67.10	52.44	71.33	86.04	87.38	87.40	94.48
	PSL(ours)	**86.53**	**79.95**	**40.99**	**76.53**	**95.75**	**94.96**	**18.96**	**95.62**

4.4.1 Evaluation Results

Comparison with state-of-the-art. We report the results on HMDB51 (OOD) and MiT-v2 (OOD) using TSM [102], I3D [104] and SlowFast [103] backbones in Tab. 4.2, Tab. 4.3 and Tab. 4.4. We can see that for w/ or w/o K400 pretrain, our PSL method has significantly better open-set and closed-set performance than all baselines. The uncertainty distribution of ID and OOD samples are depicted in Fig. 4.4 for MiT-v2 (OOD) with K400 pretrained. Three baseline methods have a clear over confidence problem, *i.e.*, the far left column is extremely high (red circles in Fig. 4.4), which means a large number of OOD samples have almost 0 uncertainty, while our method significantly alleviates this problem through the distinct representation of OOD samples, illustrated in Fig. 4.5. Besides, we can find that the open-set performance w/ K400 pretrain is higher than w/o pretrain for almost all methods in Tab. 4.2 and Fig. 4.1 (a), which can testify the importance of richer semantic representation for OSAR.

Comparison with metric learning methods. Our method concentrates on the feature representation aspect for the OSAR problem, so we also implement several well-known metric learning

Table 4.3: OSAR performance under I3D backbone.

		w/o K400 Pretrain				w/ K400 Pretrain			
Datasets	**Methods**	**AUROC↑**	**AUPR↑**	**FPR95↓**	**Acc.↑**	**AUROC↑**	**AUPR↑**	**FPR95↓**	**Acc.↑**
UCF101 HMDB51	OpenMax	83.78	54.65	47.60	74.42	92.03	77.72	41.02	95.01
	MC Dropout	75.85	40.04	50.34	74.39	91.66	78.87	33.60	94.11
	BNN SVI	81.53	53.62	49.18	73.15	91.57	78.65	34.60	93.89
	SoftMax	81.24	54.21	48.20	74.42	91.28	79.73	34.18	94.11
	RPL	79.80	52.09	54.07	71.62	92.49	81.72	28.89	94.26
	DEAR	78.91	54.14	81.96	74.42	89.80	80.86	75.63	93.89
	PSL(ours)	**86.88**	**65.63**	**39.85**	**78.85**	**93.62**	**85.54**	**28.38**	**95.46**
UCF101 MiTv2	OpenMax	86.33	77.49	44.40	74.63	93.29	90.17	29.84	94.90
	MC Dropout	76.61	62.32	48.43	74.24	93.53	90.97	25.21	94.11
	BNN SVI	83.13	76.20	48.63	73.15	93.52	91.24	25.34	93.89
	SoftMax	82.58	74.91	46.39	74.63	92.62	90.87	30.55	94.11
	RPL	81.47	73.98	49.62	71.89	93.69	92.04	25.97	94.26
	DEAR	81.48	77.03	77.58	74.42	90.88	90.55	60.28	93.89
	PSL(ours)	**88.88**	**83.30**	**34.91**	**78.69**	**95.70**	**95.06**	**20.03**	**95.51**

methods and show the result in Tab. 4.5. The evaluation is conducted using TSM model and OOD dataset is HMDB51. We do not use video shuffling in our method for fair comparison. We can see that our method still achieves the best open-set performance. The most important difference between our method and all other metric learning methods is that they aim to push the features of one class as tight as possible like C.E., while our method aims to keep the feature variance within a class to retain IS information. We calculate the mean similarity between the sample feature and the corresponding class center. The mean similarity ranges from 0.77 to 0.82 for other metric learning methods, while mean similarity is 0.71 ($s = 0.8$) and 0.6 ($s = 0.6$) for our PSL. So our method has looser feature distribution within a class, as shown in Fig. 4.5.

4.4.2 Ablation Study

Contrastive terms in $\mathcal{L}_{PSL}^{CT}$ for IS information. The intuition of PSL is to keep the intra-class variance to retain the IS information which is helpful for OSAR. We expect that the representation z within a class has a similarity $s < 1$ with the prototype k_i, so each sample can keep its own IS information. However, we find that the loss $\mathcal{L}_{PSL}$ may lead the network to find the trivial representation of samples z which is similar to using loss $\mathcal{L}_{PL}$, where only k_i shifts and z does not. We calculate the mean of similarity $sim(z, \bar{z}_i)$, where $\bar{z}_i$ denotes the mean representation

Table 4.4: OSAR performance under SlowFast backbone.

		w/o K400 Pretrain				w/ K400 Pretrain			
Datasets	**Methods**	**AUROC↑**	**AUPR↑**	**FPR95↓**	**Acc.↑**	**AUROC↑**	**AUPR↑**	**FPR95↓**	**Acc.↑**
UCF101 HMDB51	OpenMax	80.67	50.49	52.46	75.40	92.49	78.27	35.65	96.30
	MC Dropout	76.10	41.37	50.82	75.16	91.83	77.71	29.82	96.70
	BNN SVI	81.66	56.72	49.66	76.58	93.34	85.57	27.89	96.56
	SoftMax	79.15	48.54	48.79	75.63	93.82	85.56	24.74	96.70
	RPL	81.35	54.65	51.64	78.36	93.81	85.41	24.06	**96.93**
	DEAR	78.00	49.38	68.49	76.21	92.28	87.09	62.99	96.48
	PSL(ours)	**86.20**	**64.65**	**42.48**	**79.40**	**95.24**	**89.76**	**18.72**	96.52
UCF101 MiTv2	OpenMax	79.60	70.05	51.08	75.63	94.34	89.90	25.42	96.30
	MC Dropout	75.88	63.12	51.40	75.63	93.43	90.43	24.52	96.70
	BNN SVI	82.89	76.13	46.88	76.58	93.53	92.34	28.81	96.56
	SoftMax	51.08	75.63	79.60	70.05	94.67	93.34	22.14	96.70
	RPL	81.42	73.07	49.13	78.36	94.76	93.39	21.99	**96.93**
	DEAR	78.21	69.30	62.02	76.21	92.60	93.09	59.98	96.48
	PSL(ours)	**85.00**	**77.08**	**43.16**	**79.40**	**96.81**	**96.22**	**14.52**	96.52

of all samples in the same class i, and the mean of similarity with the corresponding prototype $sim(z, k_i)$, as well as the feature variance in all dimensions. Fig. 4.6 (a) and (b) show that with the hyper-parameter s decreasing, the $sim(z, k_i)$ decreases as expected by $\mathcal{L}_{PSL}$ (green curves), but the $sim(z, \bar{z}_i)$ and variance stay unchanged (blue curves), meaning that the representation of samples are still similar with using $\mathcal{L}_{PL}$, and only the prototypes are pushed away by the sample representations. In contrast, with CT in $\mathcal{L}_{PSL}^{CT}$, the $sim(z, \bar{z}_i)$ decreases and variance increases with s decreases (red curves), indicating that CT is significantly effective to keep the intra-class variance.

To individually study the effectiveness of Q_{ns} and Q_{sc} in $\mathcal{L}_{PSL}^{CT}$, we provide the ablation results in Tab. 4.6. For OOD samples, we calculate the similarity with the mean representation of its predicted class. Tab. 4.6 shows that using Q_{ns} alone can significantly increase the intra-class variance for both ID and OOD samples, meaning the pushing effect of representations in other classes can implicitly help retain the IS information. On top of that, Q_{sc} can further learn more IS information that is helpful to distinguish OOD samples, as the mean similarity of ID samples stay unchanged, but OOD samples are smaller which means OOD samples are far away from ID samples.

Shuffled videos for CS information. Tab. 4.6 shows that Q_{shuf} can improve both closed-set

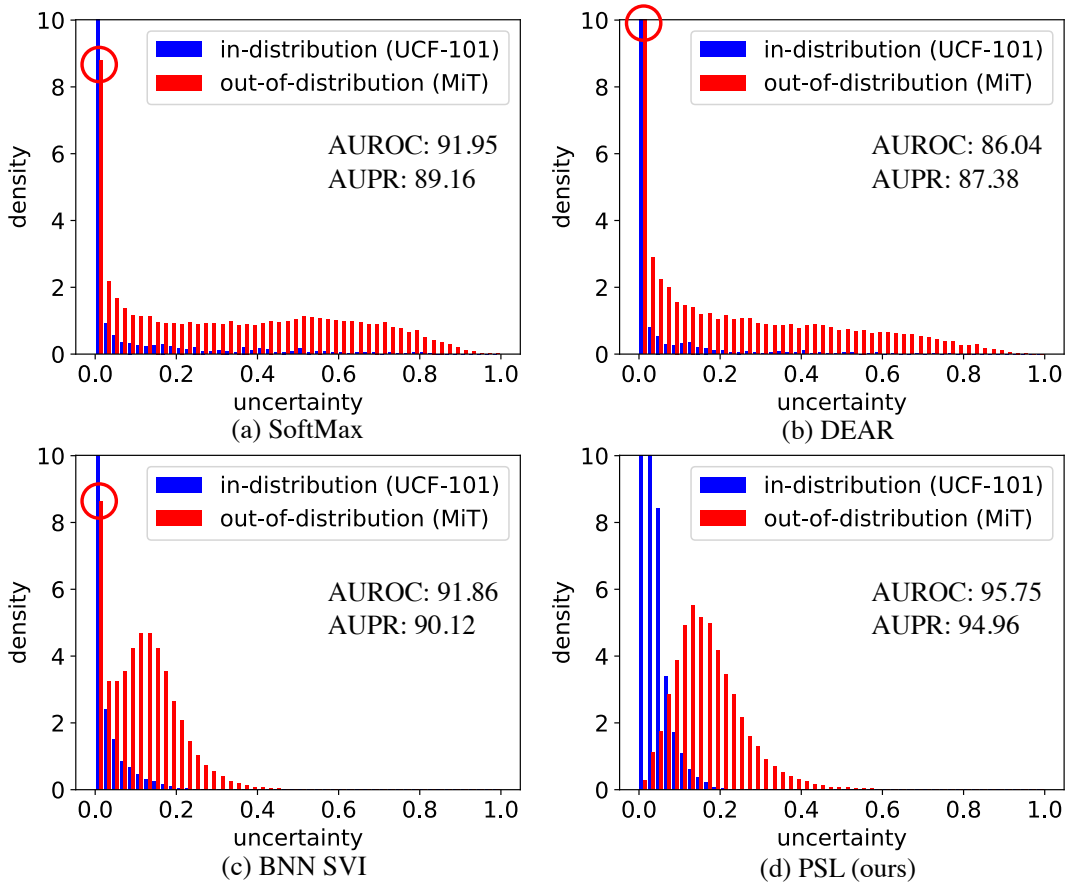
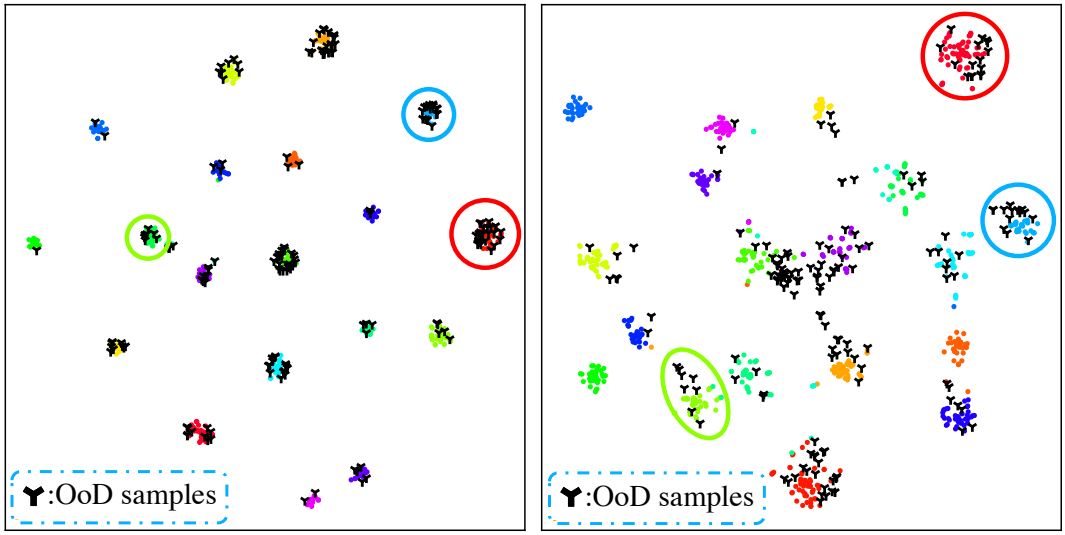


Figure 4.4: The uncertainty distribution of ID and OOD samples of (a) Softmax, (b) DEAR, (c) BNN SVI and (d) our PSL method.

and open-set performance, which proves introducing shuffled videos in PSL can enlarge CS information. Smaller intra-class variance brought by Q_{shuf} testify Proposition 1 that more CS information means more similar features within the same class.

We draw the uncertainty of all classes in HMDB51, as shown in Fig. 4.7. Note that some classes in HMDB51 are actually ID as they appear in the UCF101, like the class 3 *golf* and 4 *shoot bow* in Fig. 4.7. We find that in C.E. some OOD classes have extremely low uncertainty, such as class 1 *chew* and 2 *smile*, because they are spatially similar to some ID classes like *ApplyEyeMakeup* and *ApplyLipstick* in Fig. 4.7 (a). Comparing (b) and (c) shows that our PSL can increase the average uncertainty of OOD classes (higher yellow points), and some OOD classes which are similar to ID classes like 1 and 2 have much higher uncertainty in our PSL method. After shuffled samples are involved, some ID classes whose uncertainty are increased in (c) like 3 and 4 have lower uncertainty in (d), and the uncertainty of some OOD classes sharing similar appearance with ID classes like class 1 is further improved.

Q_{sp} in Eq. 4.9 contains Q_{shuf} and Q_{sc}, so we analyze whether should we assign the same s for the shuffled video Q_{shuf} and other videos in the same class Q_{sc}. Tab. 4.7 shows that the



(a) Cross-entropy (b) PSL (ours)

Figure 4.5: Feature representation visualization of cross-entropy and our PSL method. OOD samples are in black and ID samples are in other colors. In the red, blue and green circles, it is clear that OOD samples distribute at the edge of ID samples in our PSL, while greatly overlap with each other in the cross-entropy method.

same s have good enough performance. So we set the same s for Q_{shuf} and Q_{sc} in the default setting to reduce the number of hyper-parameters.

4.4.3 Discussion

Both CS and IS information are useful. We provide the closed-set and open-set performance under different hyper-parameter s and feature dimension d in Fig. 4.8. (a) shows that $s = 0.8$ has better open-set performance than $s = 1$ and has comparable closed-set accuracy, which illustrates that retaining the IS information which is eliminated by C.E. ($s = 1$) is beneficial. When $s < 0.8$, the NN cannot learn enough CS information, so both closed-set and open-set performance drops. Therefore, a proper mixture of CS and IS information is ideal. (b) shows that when d grows from 4 to 16, more CS information is contained so that both closed-set and open-set performance improves. When d grows from 16 to 128, the feature does not include more CS information as closed-set accuracy is comparable. However, open-set performance keeps increasing which means more IS information is contained based on more feature dimensions. This interesting experiment shows that enough information for closed-set recognition is not enough for open-set recognition because IS information is not related to the closed-set task but useful for the open-set task.

Feature variance and open-set performance analysis. Fig. 4.8 (a) shows that when features

Table 4.5: Comparison with different metric learning methods.

	AUROC↑	AUPR↑	FPR95↓	Acc.↑
SoftMax	80.95	52.79	52.51	72.36
Triplet [121]	81.02	54.75	53.88	75.50
Normface [122]	80.99	54.90	53.19	73.34
Circle [123]	78.76	51.65	55.27	72.15
Arcface [124]	81.23	55.03	53.67	**75.95**
LSoftMax [125]	80.87	54.01	52.29	73.05
PSL($s = 0.8$)	**83.42**	**59.05**	**51.32**	72.28
PSL($s = 0.6$)	82.75	58.57	52.27	73.26

Table 4.6: Abaltion results of different components in $\mathcal{L}_{PSL}^{CT}$.

					ID		OOD					
	s	Q_{ns}	Q_{sc}	Q_{shuf}	Mean	Variance	Mean	Variance	AUROC↑	AUPR↑	FPR95↓	Acc.↑
$\mathcal{L}_{PL}$	✗	✗	✗	✗	0.81	0.0015	0.63	0.0029	80.95	52.79	52.51	72.36
$\mathcal{L}_{PSL}$	✓	✗	✗	✗	0.79	0.0016	0.62	0.0028	81.79	54.16	52.33	72.33
	✓	✓	✗	✗	0.71	0.0022	0.61	0.0036	82.60	57.36	50.03	72.17
$\mathcal{L}_{PSL}^{CT}$	✓	✓	✓	✗	0.71	0.0023	0.49	0.0035	83.42	59.05	51.32	72.28
	✓	✓	✓	✓	0.74	0.0016	0.63	0.0029	86.43	65.58	41.75	77.19

get looser ($s = 1 - 0.8$), the open-set performance is improved, but if features get continually looser ($s = 0.8 - 0.1$), the open-set performance drops. So there is no strict relation between the feature variance and open-set performance. One may argue that continual training can benefit the open-set performance [126], which is alongside with smaller feature variance [127]. We show that the benefit of continual training comes from better closed-set performance, not tighter features. Tab. 4.8 shows that when we train the model from 200 to 400 epochs, the closed-set accuracy is higher, and feature is tighter (larger mean similarity and smaller variance), and the open-set performance is better. But from epoch 400 to 800 we find the model is already overfitted to the training set, as the accuracy of test set remains unchanged. So although the features get tighter in the 800 epoch, both the closed-set and open-set performance remain same.

Representation analysis through singular value spectrum To deeply understand the feature representations learned by our method, we analyze the representation through singular value

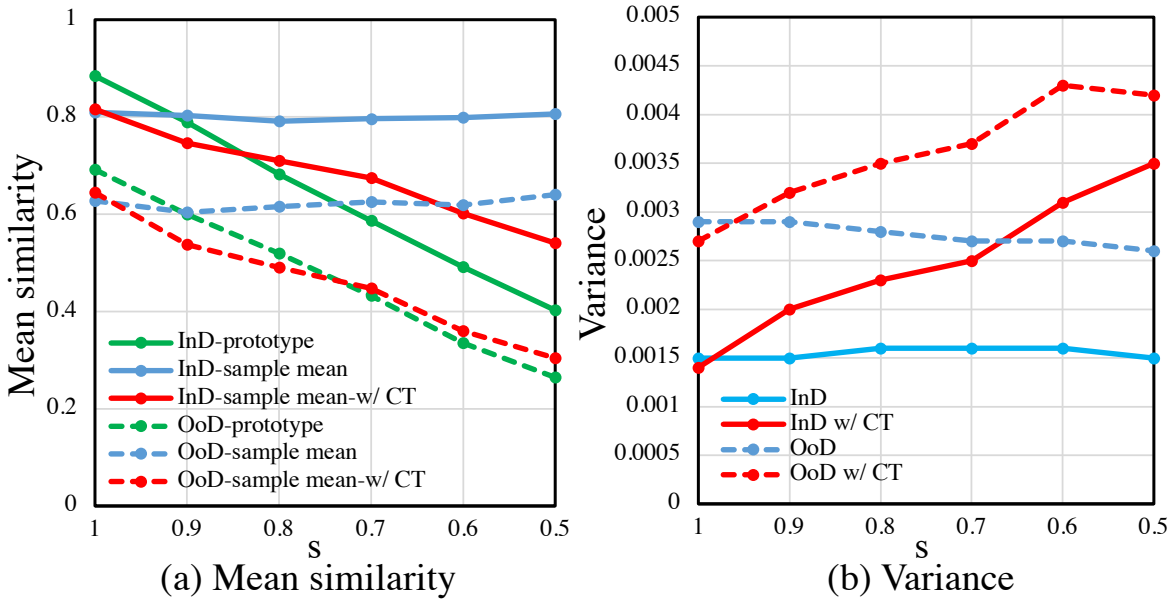


Figure 4.6: Mean similarity and variance analysis for CT terms.

Table 4.7: Ablation study of similarity s for Q_{shuf} and Q_{sc}.

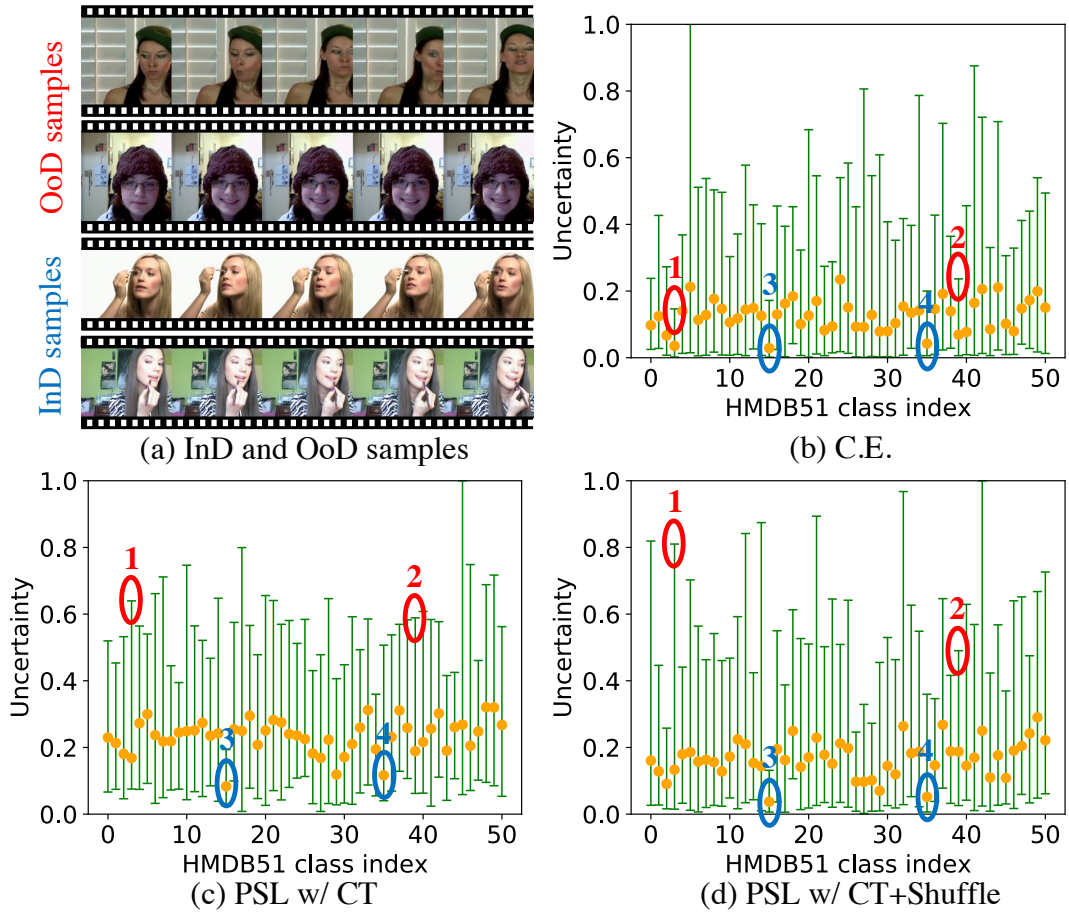
$s(Q_{shuf})$	$s(Q_{sc})$	AUROC↑	AUPR↑	FPR95↓	Acc.↑
0.7	0.7	85.25	63.91	48.34	76.98
0.5		86.03	64.36	43.70	76.53
0.3		83.80	60.42	48.76	75.50
0		79.54	50.59	54.43	72.59
0.8	0.8	86.43	65.58	41.75	76.53
0.9	0.9	83.12	57.04	46.84	73.31
1	1	82.04	53.82	51.82	72.89

spectrum. We first compute the covariance matrix $C \in \mathbb{R}^{d\times d}$ of the embedding matrix:

$$C = \frac{1}{M}\sum_{i=1}^{M}(z_i - \bar{z})(z_i - \bar{z})^T, \tag{4.11}$$

where z_i and $\bar{z}_i$ denote the feature representation of a sample and mean representation of all samples respectively. M is the total number of samples. Then we conduct singular value decomposition on the matrix $C = USV^T, S = diag(\sigma^k)$, and plot the singular values in sorted order and logarithmic scale $log(\sigma^k)$. We provide the singular value spectrum in Fig. 4.9.

PSL has larger singular values than the PL in the larger rank index, illustrating that more information is contained in the not significant dimensions, which is reasonable as PSL keeps the IS information with no direct supervision signal, but these IS information does help for better



(a) InD and OoD samples (b) C.E. (c) PSL w/ CT (d) PSL w/ CT+Shuffle

Figure 4.7: (a) *chew* and *smile* are OOD samples from HMDB51, and *ApplyEyeMakeup* and *ApplyLipstick* are ID samples from UCF101. (b-d) Uncertainty distribution of each class in HMDB51. Class 1: *chew*, 2: *smile*, 3: *golf*, 4: *shoot bow*. Classes 1 and 2 are OOD while 3 and 4 are ID.

OSAR performance according to Tab. 4.6. PSL with shuffled samples Q_{shuf} has larger singular values than PSL in the small rank index, indicating more diverse information is learned in the important dimensions, which are supposed to refer to CS information as CS information is learned by the explicit supervision signal. The closed-set accuracy with Q_{shuf} is higher than without Q_{shuf} in Tab. 4.6 further testifies our conclusion. In Tab. 4.6 we see that the representations of the same class are tighter with more CS information. Therefore, learning the distinct temporal information from shuffled videos can enlarge the open-set task related CS information while PSL can enlarge the IS information, which fulfills the goal to enlarge Eq. 4.3 for better OSAR performance.

t-SNE visulization We provide the t-SNE visualization for straight understanding. All results are based on HMDB (OOD) from scratch. We provide the visualization results of PSL, PSL with Q_{ns}, PSL with Q_{ns}, Q_{sc}, and PSL with Q_{ns}, Q_{sc}, Q_{shuf} in Fig. 4.10, Fig. 4.11, Fig. 4.12, Fig. 4.13 respectively. From Fig. 4.10 we can see PSL alone cannot keep the intra-class variance when

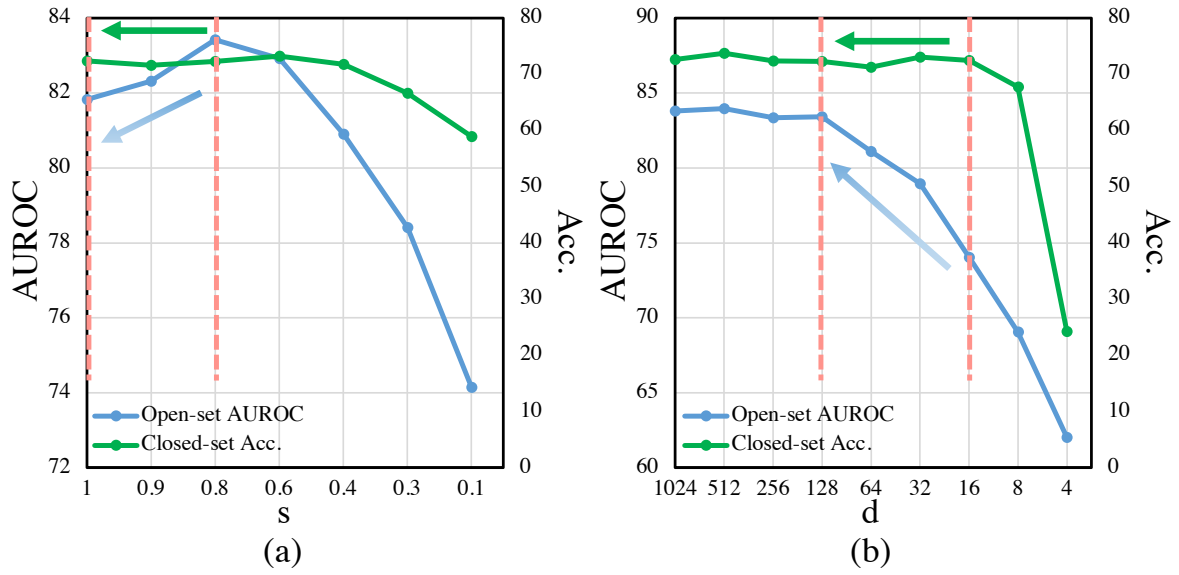


Figure 4.8: Ablation study of similarity s and feature dimension d.

Table 4.8: Training process analysis when $s = 0.6$ w/o Q_{shuf}.

Epoch	Mean	Variance	AUROC↑	Acc-Test.↑	Acc-Train.↑
200	0.577	3.3e-3	75.08	68.39	99.85
400	0.602	3.1e-3	82.92	73.26	100
800	0.613	3.0e-3	82.54	73.29	100

s decreases. Fig. 4.11 and Fig. 4.12 tell us that Q_{ns} and Q_{sc} are important for PSL to keep the intra-class variance. Furthermore, Q_{shuf} makes the feature representation tighter if we compare Fig. 4.12 and Fig. 4.13, which shows the model learns more CS information with Q_{shuf}.

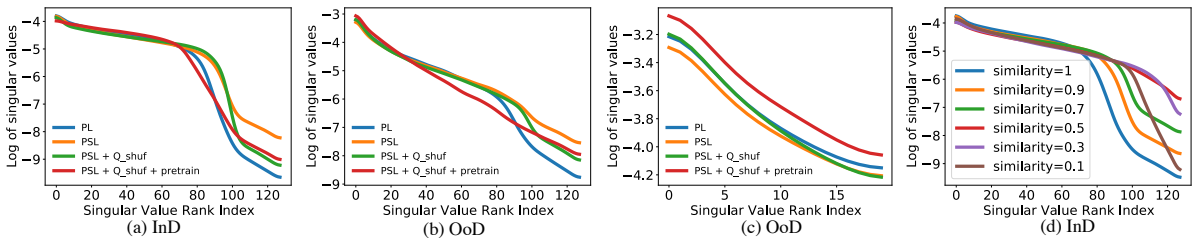
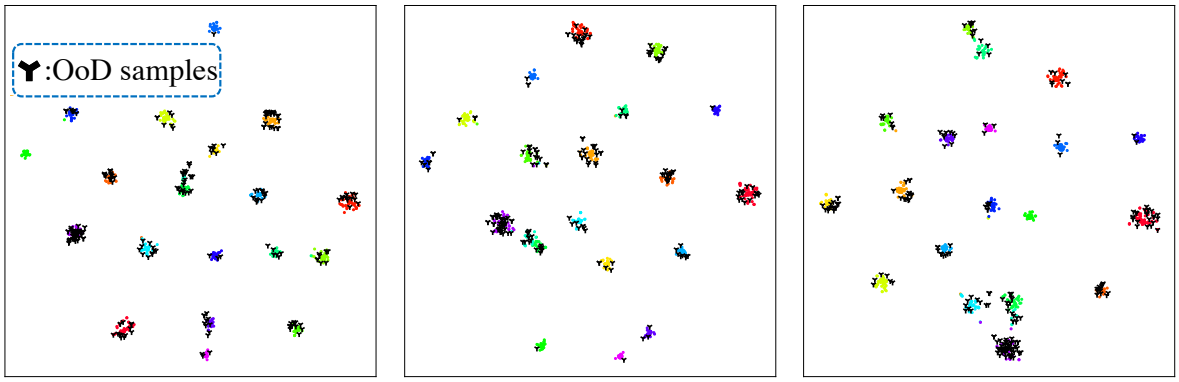
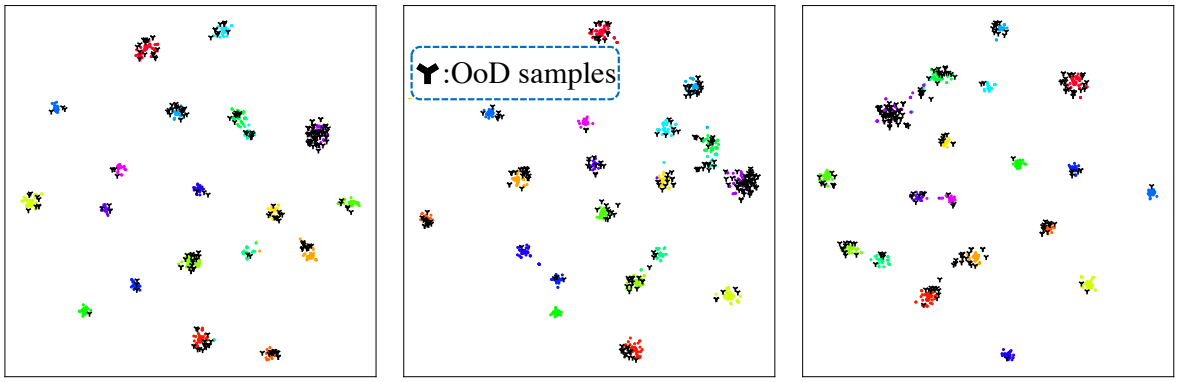


Figure 4.9: Singular value spectrum on HMDB51 (OOD) under different training conditions (a)-(c) and hyper-parameter s (d). (c) contains the top 20 singular values in (b).



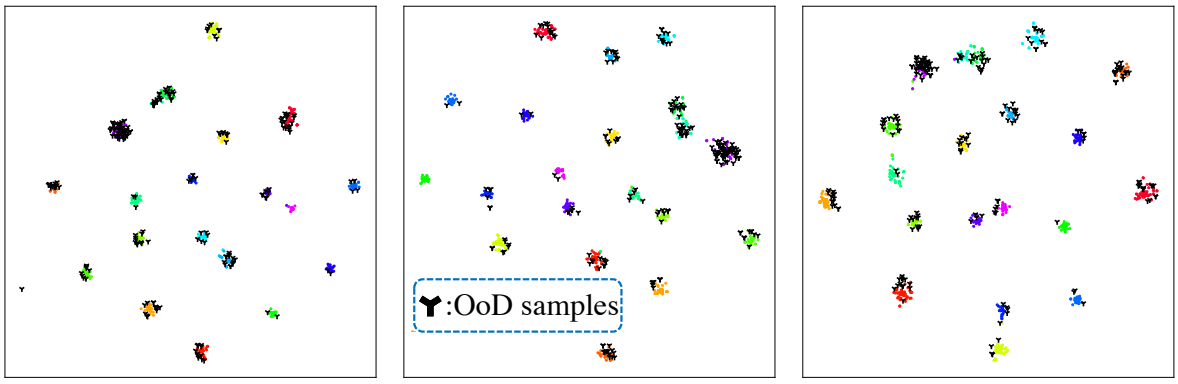
(a) similarity=1 (b) similarity=0.7 (c) similarity=0.5

Figure 4.10: t-SNE visualization of PSL.



(a) similarity=0.9 (b) similarity=0.8 (c) similarity=0.7

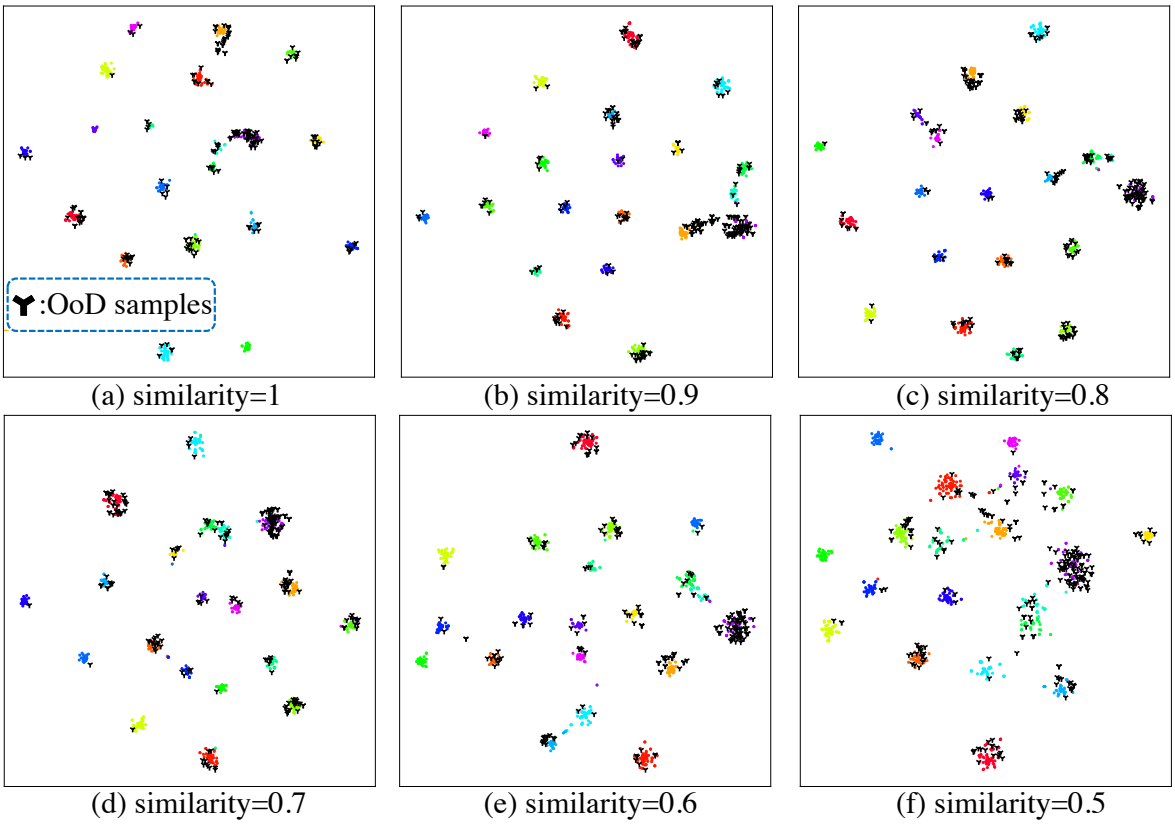
Figure 4.11: t-SNE visualization of PSL with Q_{ns}.



(a) similarity=1 (b) similarity=0.9 (c) similarity=0.8

(d) similarity=0.7 (e) similarity=0.6 (f) similarity=0.5

Figure 4.12: t-SNE visualization of PSL with Q_{ns}, Q_{sc}.



(a) similarity=1 (b) similarity=0.9 (c) similarity=0.8

(d) similarity=0.7 (e) similarity=0.6 (f) similarity=0.5

Figure 4.13: t-SNE visualization of PSL with Q_{ns}, Q_{sc}, Q_{shuf}.

4.5 ID and OOD uncertainty distribution

We provide the ID and OOD distribution on HMDB51 (OOD) and MiT-v2 (OOD) with K400 pretrain and without K400 pretrain. All results are based on TSM backbone for illustration. The results are shown in Fig. 4.14, Fig. 4.15, Fig. 4.16, and Fig. 4.17.

From Fig. 4.14 and Fig. 4.16 we can see that if there is no K400 pretrain, all methods have the overlapping uncertainty between ID and OOD distribution except OpenMax and our PSL. For instance, Fig. 4.14 (f) DEAR [77] shows the uncertainty of ID and OOD samples both cover the range from 0 to 1. In contrast, Fig. 4.14 (g) PSL shows that in our method, the ID distribution covers from 0 to 0.3, while the OOD distribution covers from 0 to 0.8. It means our method tends to assign higher uncertainty to OOD samples. For OpenMax, Fig. 4.14 (a) shows that ID uncertainy distribution is extremely close to 0, which is a good phenomenon, but the OOD uncertainty distribution only covers from 0 to 0.3, and the OOD samples whose uncertainty is larger than 0.3 is too sparse, which means OpenMax tends to assign low uncertainty to both ID and OOD samples, but assigner lower uncertainty to ID samples.

If we compare Fig. 4.14 to Fig. 4.15 or compare Fig. 4.16 to Fig. 4.17, we can find that the ID distribution of all methods are closer to 0 with K400 pretrain. But all methods except our

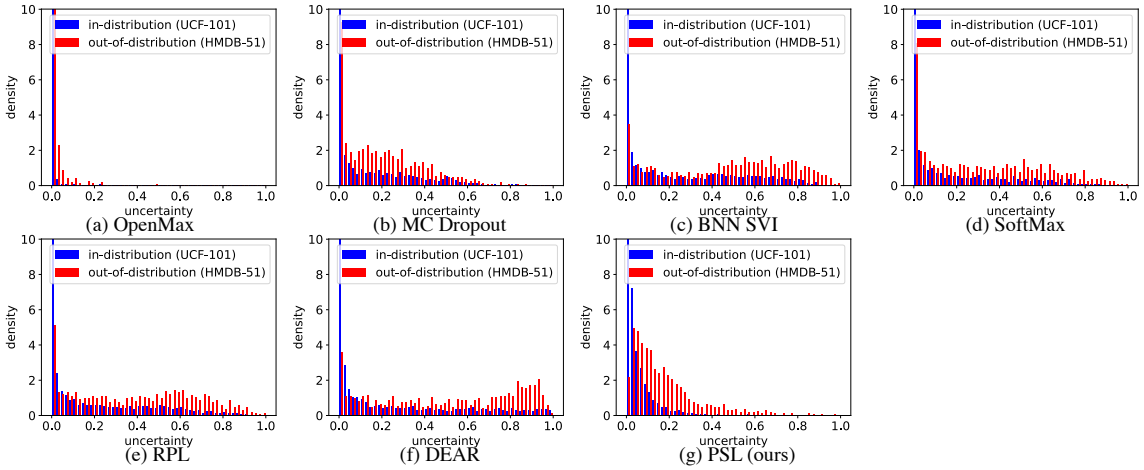


Figure 4.14: Uncertainty distribution on HMDB51 (OOD) w/o K400 pretrain.

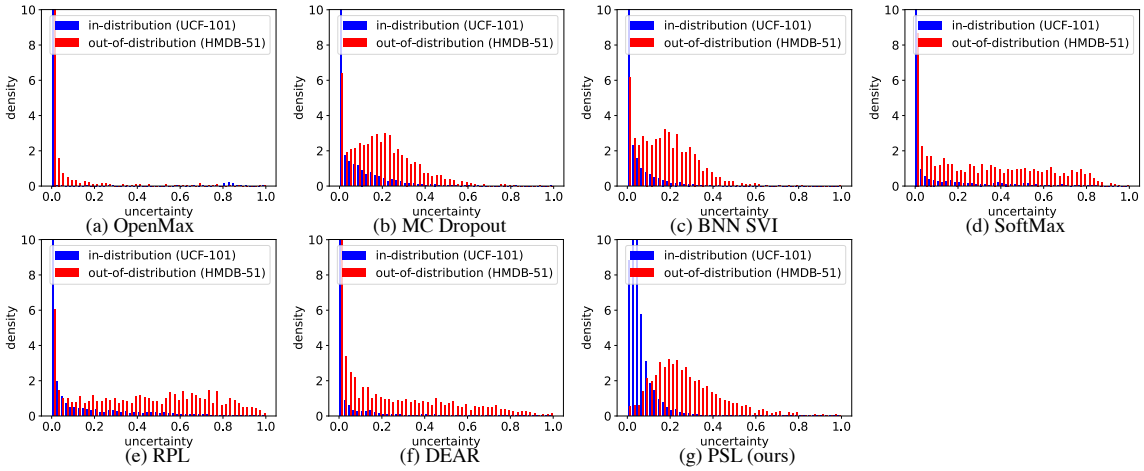


Figure 4.15: Uncertainty distribution on HMDB51 (OOD) w/ K400 pretrain.

PSL have a serious over confidence problem, which is illustrated by the fact that the far left column of OOD samples is extremely high, which is also emphasized through the red circles in Fig. 4.4. In contrast, the density of OOD distribution is highest at 0.2 uncertainty in our PSL method, and the density of OOD distribution is almost 0 at 0 uncertainty. Besides, it is very clear that the OOD distribution and ID distribution in our PSL is most distinguishable among all methods.

4.6 Conclusion

We analyze the OSAR problem from the information perspective, and show that cross-entropy tends to eliminate IS information and cannot fully learns CS information which are both useful for the open-set task. So we propose PSL to retain IS information and introduce shuffle videos into PSL to enlarge CS information. Comprehensive experiments demonstrate the effectiveness

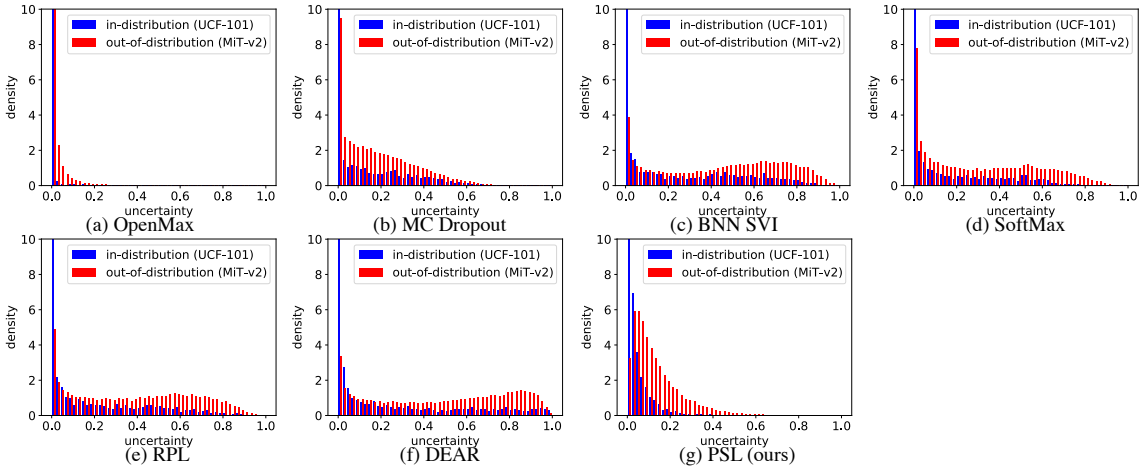


Figure 4.16: Uncertainty distribution on MiT-v2 (OOD) w/o K400 pretrain.

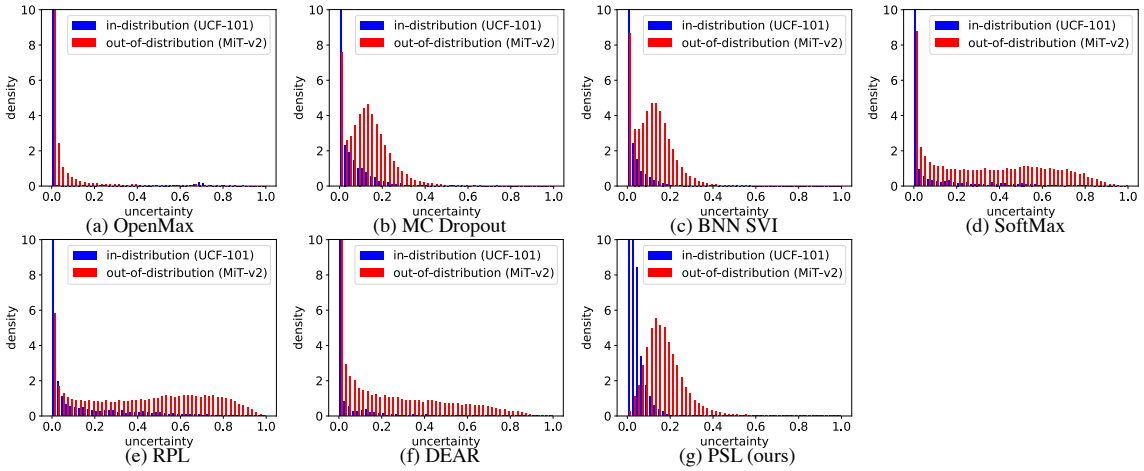


Figure 4.17: Uncertainty distribution on MiT-v2 (OOD) w/ K400 pretrain.

of our PSL and the importance of IS and CS information in the OSAR task.

CHAPTER 5

TOWARDS UNIFIED OPEN-SET RECOGNITION

5.1 Introduction

Neural networks have achieved tremendous success in the closed-set classification [128], where the test samples share the same In-Distribution (ID) class set with training samples. Open-Set Recognition (OSR) [129] is proposed to tackle the challenge that some samples whose classes are not seen during training, which are Out-of-Distribution (OOD) data, may occur in the real world applications and should be rejected. However, some researchers have argued that the model should not only reject OOD samples but also ID samples that are Wrongly classified (InW), as the model gives the wrong answers for both of them. So Unified Open-set Recognition (UOSR) is proposed to only accept ID samples that are correctly classified (InC) and reject OOD and InW samples [130] simultaneously. The difference between the UOSR and OSR lies in the InW samples, where OSR is supposed to accept them while UOSR has the opposite purpose. Actually, UOSR is more useful in most real-world applications, but it receives little attention from the research community as it has been proposed very recently and lacks comprehensive systematic research. Therefore, we deeply analyze the UOSR problem in this work to fill this gap.

We first apply existing OSR methods for UOSR in Sec. 5.3, and then analyze UOSR under different *training settings* and *evaluation settings* in Sec. 5.4 and Sec. 5.5 respectively. In Sec. 5.3, several existing OSR methods are applied for UOSR, and we find that the UOSR performance is consistently and significantly better than the OSR performance for the same method, as shown in Fig. 5.1 (a). We show that this phenomenon holds for different network architectures, datasets, and domains (image and video recognition). We find *the devil is in the InW samples that have similar uncertainty distribution with OOD samples rather than InC samples*. Therefore, the false positive predictions in OSR tend to be InW samples, which is extremely important but dismissed by all existing OSR works.

In Sec. 5.4, we introduce two *training settings* into UOSR, including pre-training [131] and outlier exposure [45, 132, 133], as they are both helpers of the OSR that introduce extra information beyond the training set. Pre-training is to use the weights that are trained on a large-scale

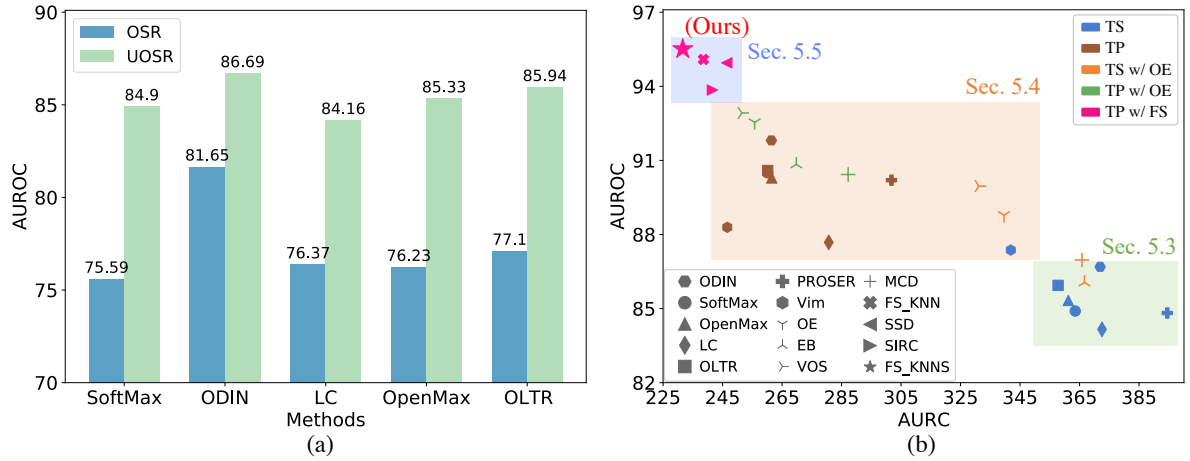


Figure 5.1: (a) shows that the UOSR performance is significantly better than OSR performance for the same method, which illustrates the uncertainty distribution of these OSR methods is actually closer to the expectation of UOSR than OSR. (b) shows the UOSR performance under different settings and the skeleton of this paper. Results are based on the ResNet50 backbone. CIFAR100 and TinyImageNet are ID and OOD datasets, respectively. (TS: Train from Scratch. TP: Train from Pre-training. OE: Outlier Exposure. FS: Few-shot.)

dataset for better down-task performance, and outlier exposure is to introduce some background data without labels into training to help the model classify ID and OOD samples. We find both of them have better performance for InC/OOD discrimination, which explains why they are beneficial for OSR. However, pre-training is also helpful for InC/InW discrimination, while outlier exposure has a comparable or even worse performance to distinguish InC and InW samples. The performance of UOSR can be regarded as the comprehensive results of InC/OOD and InC/InW discrimination so that both techniques can boost the performance of UOSR. We build up a comprehensive UOSR benchmark that involves both pre-training and outlier exposure settings, as shown in Fig. 5.1 (b).

In addition to the two aforementioned *training settings*, we introduce a new *evaluation setting* into UOSR in Sec. 5.5. We formulate the few-shot UOSR, similar to SSD [118] that proposes few-shot OSR, where 1 or 5 samples per OOD class are introduced for reference to better identify OOD samples. We first develop a KNN-based baseline [31] FS-KNN for the few-shot UOSR. Although InC/OOD discrimination is improved due to the introduced OOD reference samples, the InC/InW discrimination is severely harmed compared to SoftMax baseline [1]. To alleviate this problem, we propose FS-KNNS that dynamically fuses the FS-KNN with SoftMax uncertainty scores to keep high InC/InW and InC/OOD performance simultaneously. Our FS-KNNS achieves state-of-the-art performance under all settings in the UOSR benchmark, as shown in Fig. 5.1 (b), even without outlier exposure during training. Note that InC/OOD performances are comparable between FS-KNNS and FS-KNN, but their distinct InC/InW per-

Table 5.1: Comparison of uncertainty-related task settings. Cls: Classification. 0 and 1 refer to the corresponding ground truth uncertainty u, and u is not fixed in MC.

	InC	InW	OOD	Ordinal Rank	ID Cls
SP	0	1	✗	✓	✓
AD/OD	0	0	1	✓	✗
OSR	0	0	1	✓	✓
UOSR	0	1	1	✓	✓
MC	-	-	✗	✗	✓

formances makes FS-KNN better at OSR and FS-KNNS better at UOSR, which illustrates the difference between few-shot OSR and UOSR and the importance of InW samples during evaluation.

5.2 Towards Unified Open-set Recognition

In this section, we first formalize the UOSR problem and then discuss the relation between UOSR and other uncertainty-related tasks.

Unified Open-set Recognition. Suppose the training dataset is $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{C}$, where $\mathcal{X}$ refers to the input space, *e.g.*, images or videos, and $\mathcal{C}$ refers to the ID sets. In closed-set recognition, all test samples come from the ID sets, *i.e.*, $\mathcal{D}_{\text{test}}^{\text{closed}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M} \subset \mathcal{X} \times \mathcal{C}$. In OSR and UOSR, the test samples may come from OOD sets $\mathcal{U}$ which are not overlap with ID sets $\mathcal{C}$, so we have $\mathcal{D}_{\text{test}}^{\text{open}} = \mathcal{D}_{\text{test}}^{\text{closed}} \cup \mathcal{D}_{\text{test}}^{\text{unknown}}$, where $\mathcal{D}_{\text{test}}^{\text{unknown}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M'} \subset \mathcal{X} \times \mathcal{U}$. The ID test samples $\mathcal{D}_{\text{test}}^{\text{closed}}$ can be divided into two splits based on whether the sample is correctly classified or wrongly classified, *i.e.*, $\mathcal{D}_{\text{test}}^{\text{closed}} = \mathcal{D}_{\text{test}}^{\text{closed-c}} \cup \mathcal{D}_{\text{test}}^{\text{closed-w}}$, where $\mathcal{D}_{\text{test}}^{\text{closed-c}} = \{(\mathbf{x}_i, y_i) | \hat{y}_i = y_i\}_{i=1}^{M}$, $\mathcal{D}_{\text{test}}^{\text{closed-w}} = \{(\mathbf{x}_i, y_i) | \hat{y}_i \neq y_i\}_{i=1}^{M}$, and $\hat{y}_i$ refers to the model classification results of sample $\mathbf{x}_i$. The goal of UOSR is to reject InW and OOD samples and accept InC samples, so the ground truth uncertainty u of $\mathcal{D}_{\text{test}}^{\text{closed-w}}$ and $\mathcal{D}_{\text{test}}^{\text{unknown}}$ is 1 while for $\mathcal{D}_{\text{test}}^{\text{closed-c}}$ it is 0, as shown in Tab. 5.1. The key of UOSR is how to estimate the uncertainty $\hat{u}$ to be close to the ground truth uncertainty u.

The UOSR is proposed by [130] very recently, so it has not attracted many researchers to this problem yet. SIRC [134] augments SoftMax [1] baseline for the better UOSR performance. Build upon these existing methods, we make a deep analysis of UOSR in this work. In Tab. 5.1, we compare different settings of the related uncertainty estimation tasks, and the detailed discussions are as follows.

Selective Prediction (SP). Apart from the classical classification, SP also tries to estimate which

sample is wrongly classified [135–137], so the ground truth uncertainty of InW is 1. SP is constrained under the closed-set scenario and does not consider OOD samples during evaluation, as shown in Tab. 5.1, which is the key difference with UOSR. We involve some SP methods in the UOSR benchmark in Sec. 5.4.

Anomaly/Outlier Detection (AD/OD). AD is to detect anomaly patches within an image or anomaly events within a video [138, 139]. OD regards a whole dataset as ID and samples from other datasets as OOD [140, 141]. Both AD and OD do not require ID classification, so there is no InC/InW discrimination problem.

Open-set Recognition (OSR) and Out-of-distribution Detection (OODD). The task settings of OSR and OOD detection are same, but their datasets might be different. Both of them aim to accept all ID samples no matter they are InC or InW samples, and reject OOD samples. OSR divides one dataset into two splits and uses one of them as ID data to train the model, while another split is regarded as OOD samples [129]. In contrast, OODD uses a whole dataset as ID data and regards another dataset as OOD data [1]. However, in the recent works about OSR in the video domain [77], it utilizes the OODD setting rather than the OSR setting. In this work, we use OSR to represent the task setting and use one dataset as ID data and another dataset as OOD data. As mentioned before, the distinction between UOSR and OSR is InW samples, where OSR aims to accept them, and UOSR aims to reject them, so the ground truth uncertainty of InW samples is 0 in OSR and 1 in UOSR, as shown in Tab. 5.1. *Better InC/OOD performance is beneficial for both UOSR and OSR, but higher InC/InW discrimination is preferred by UOSR but not wanted by OSR.*

Model Calibration (MC). All tasks mentioned above solve the uncertainty ordinal ranking problem, *i.e.*, the ground truth uncertainty of each type of data is fixed, and the performance will be better if the estimated uncertainty is closer to the ground truth uncertainty [142]. In contrast, MC uses uncertainty to measure the probability of correctness [143].

A perfect calibrated model should meet [143]:

$$P(\hat{y} = y | f(x) = p) = p, \quad \forall p \in [0, 1], \tag{5.1}$$

where p is the confidence or the negative version of uncertainty $p = 1 - u$. For example, given 100 test samples whose confidence scores are all 0.8, then the model is perfectly calibrated if 80% samples are correctly classified. In this case, 20% samples with confidence 0.8 is consistent with the requirement of a perfect calibrated model. Different from Selective Prediction (SP), Anomaly Detection (AD), OSR, and UOSR, Model Calibration (MC) is not an uncertainty

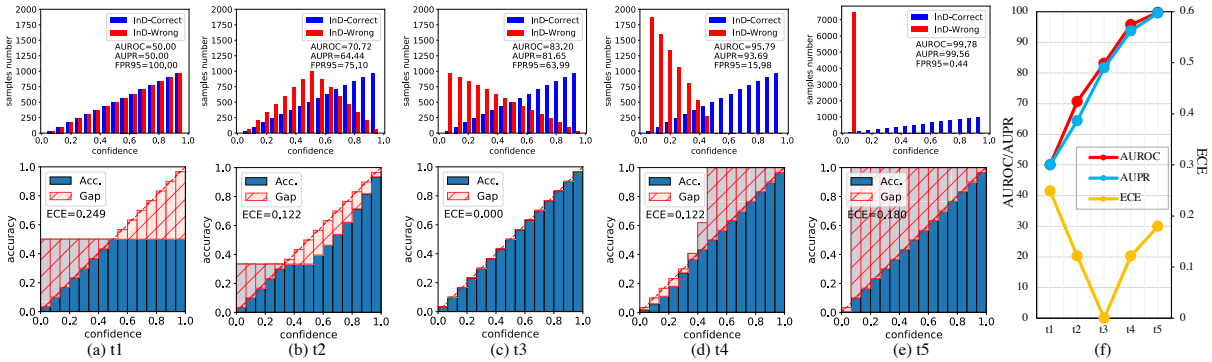


Figure 5.2: We provide 5 samples in (a)-(e), where we keep the confidence distribution of InC and change the confidence distribution of InW samples. The evaluation metrics of UOSR are AUROC and AUPR, and ECE is for the MC.

ordinal ranking problem. In other words, all other settings expect the uncertainty of a test sample to be either 0 or 1, while MC is not. The performance of MC is evaluated by ECE, and the readers may refer to [143] for the formal definition. Smaller ECE means better calibrated model. We provide 5 examples to illustrate the relation between the performance of UOSR and MC in Fig. 5.2. Note that MC does not consider OOD samples during evaluation, so we also do not involve OOD samples in Fig. 5.2. Therefore, UOSR in Fig. 5.2 is also equal to SP. From (f) we can see that the performance of UOSR and model calibration is not perfectly positively correlated, *i.e.*, the best case for model calibration (c) is not the best case for UOSR and vice versa (e).

5.3 OSR Approaches for UOSR

In this section, we evaluate the existing OSR approaches for the UOSR problem, and show that the InW samples play a crucial role when evaluating the uncertainty quality. Specifically, simply changing the ground truth uncertainty u of InW samples from 0 to 1 can bring a large performance boost, as shown in Fig. 5.3. Then we provide the comprehensive experiment results and discussion of this phenomenon.

Applied Methods. We reproduce several classical OSR methods and evaluate their UOSR and OSR performance in our experiments, including SoftMax [1], ODIN [144], LC [145], OpenMax [146], OLTR [147] and PROSER [148] in the image domain, as well as DEAR [77], RPL [149], MC Dropout [150] and BNN SVI [151] in the video domain.

Datasets. In the image domain, we follow the datasets setting in [144]. The training ID dataset is CIFAR-100 [152], which contains 100 classes with 50000 training images and 10000 test images. The OOD datasets for open-set evaluation are TinyImageNet [153] and LSUN [154].

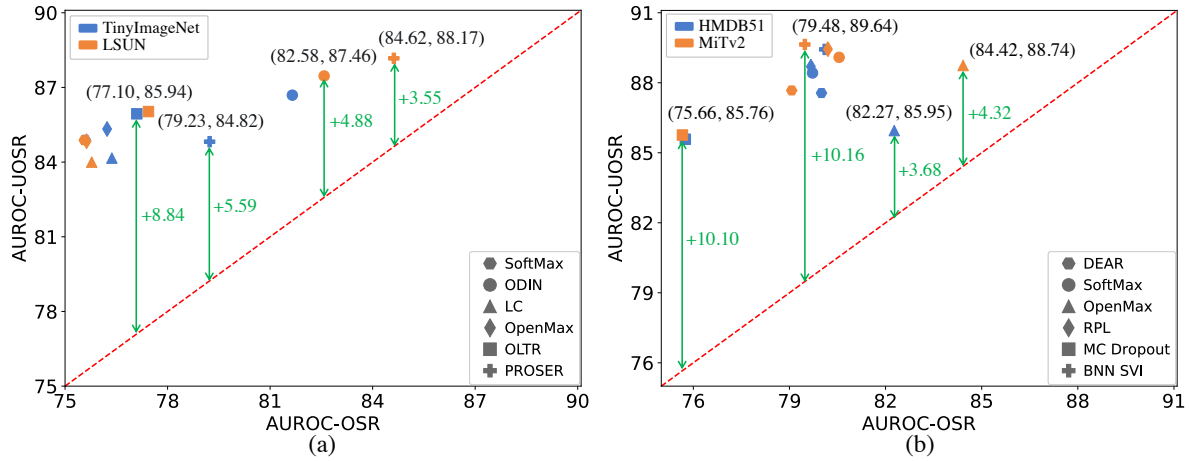


Figure 5.3: (a) and (b) show the relation between UOSR and OSR performance in the image and video domain under ResNet50 and TSM backbones. Different color indicates different OOD datasets. The red-dotted diagonal is where UOSR has the same AUROC as OSR. Green arrows show the performance gap between UOSR and OSR for the same method.

The TinyImageNet dataset contains 10000 test images from 200 different classes, we use the TinyImageNet (resize) in [144]. The Large-scale Scene UNderstanding (LSUN) dataset consists of 10000 images of 10 different scenes categories like classroom, conference room, dining room, etc. We use LSUN (resize) from [144]. The size of images in both ID and OOD datasets is 32 × 32. We use the same strategy in image preprocessing stage as [155], if the resolution of the training image is lower than 96 × 96, we use 128 × 128 image cropping technique and random horizontal mirroring followed by 160 × 160 image resize. Test images and OOD images are directly resized to 128 × 128. In the video domain, we follow the datasets setting in [77]. The training ID dataset is UCF101 [156], which contains 101 classes with 9537 training samples and 3783 test samples. The OOD datasets for open-set evaluation are HMDB51 [157] and MiTv2 [158]. We use the test sets of them which contain 1530 samples and 30500 samples respectively. For UCF101 and HMDB51, we follow the MMAction [119] to use the split 1 for training and evaluation, which is the same with [77].

Experiments settings. We train the network using the ID dataset and evaluate the UOSR and OSR performance based on the ground truth in Tab. 5.1. The evaluation metric is AUROC [1] which is a threshold-free value. The AUROC reflects the distinction quality of two uncertainty distributions. We adopt VGG13 [159] and ResNet50 [160] as the network backbone in the image domain and TSM [161] and I3D [104] in the video domain.

Results. We provide the UOSR and OSR performance of different methods in Fig. 5.3 and Fig. 5.4. They show that all data points are above the red dotted diagonal, illustrating that the UOSR performance is significantly better than the OSR performance of the same method, and

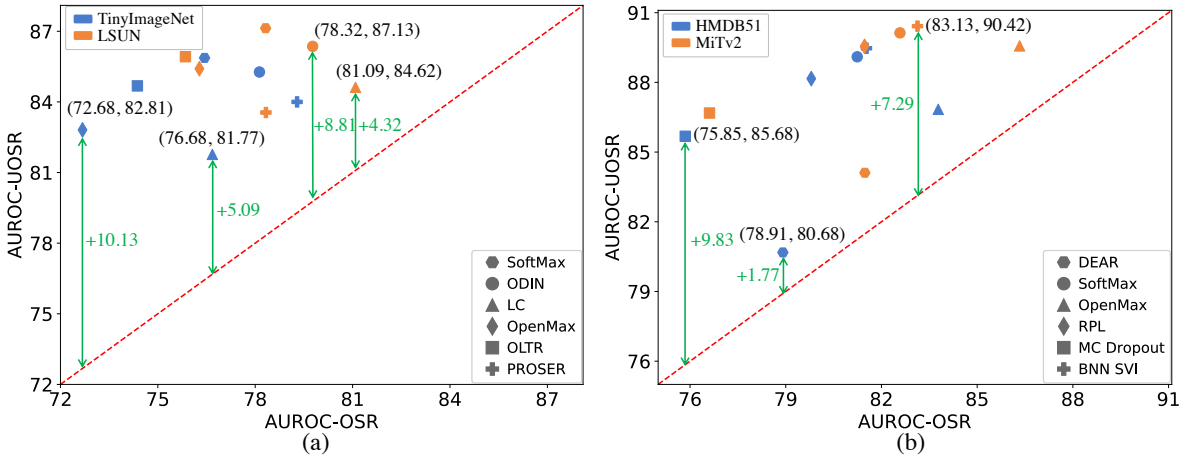


Figure 5.4: (a) and (b) are conducted using the VGG13 and I3D backbone in the image and video domain respectively. ID datasets are CIFAR100 and UCF101 for (a) and (b), and OOD datasets are shown with different colors.

Table 5.2: Uncertainty distribution analysis in image domain with ResNet50. OOD dataset: TinyImageNet. AUROC (%) is reported.

Methods	**InC/OOD**	**InC/InW**	**InW/OOD**	**OSR**	**UOSR**
SoftMax	84.69	85.68	50.64	75.59	84.90
ODIN	88.35	80.76	64.36	81.65	86.69
LC	84.60	82.58	55.14	76.37	84.16
OpenMax	85.16	85.96	51.27	76.23	85.33
OLTR	85.99	85.74	52.22	77.10	85.94
PROSER	87.04	77.84	62.57	79.23	84.82

this relationship holds across different datasets, domains (image and video), and network architectures. The performance gap between UOSR and OSR of the same method can be very large, such as 8.84% for OLTR when TinyImageNet is the OOD dataset, and 10.16% for BNN SVI when the OOD dataset is MiTv2. Therefore, existing OSR methods have uncertainty distributions that are actually closer to the expectation of UOSR than OSR.

Analysis. To better understand our findings, we provide a detailed analysis of the uncertainty distribution relationships between InC, InW, and OOD samples in Tab. 5.2 and Tab. 5.3. Note that higher AUROC means a better distinction between two uncertainty distributions, and AUROC=50% means two distributions overlap with each other. From Tab. 5.2 and Tab. 5.3 we can clearly see that AUROC of InC/OOD and InC/InW are significantly higher than InW/OOD, and AUROC of InW/OOD is very close to 50%. Therefore, the uncertainty distribution of InC samples is distinguishable from OOD and InW samples, and there is a lot of overlap between the uncertainty distributions of InW and OOD samples. Several uncertainty distribution visual-

Table 5.3: Uncertainty distribution analysis in video domain with TSM backbone. OOD dataset is HMDB51. AUROC (%) is reported.

Methods	InC/OOD	InC/InW	InW/OOD	OSR	UOSR
OpenMax	88.56	82.53	64.47	82.27	85.95
Dropout	85.36	85.86	48.92	75.75	85.58
BNN SVI	89.93	88.85	55.44	80.10	89.44
SoftMax	88.73	88.01	54.19	79.72	88.42
RPL	89.22	88.06	55.74	79.67	88.70
DEAR	89.33	85.47	56.78	80.00	87.56

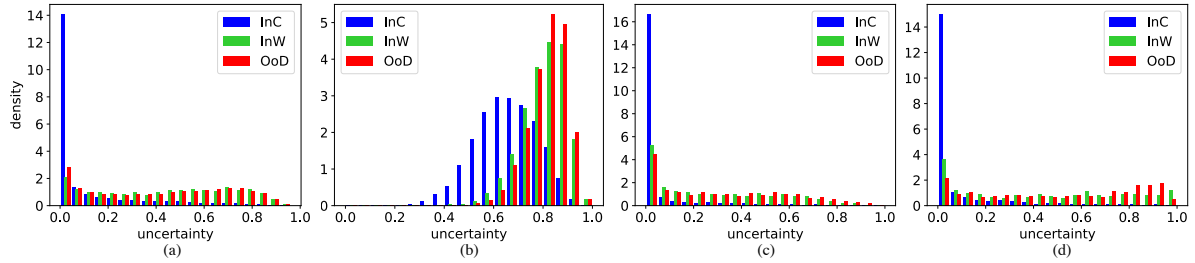


Figure 5.5: (a) and (b) are the SoftMax and ODIN methods in the image domain, while (c) and (d) are the SoftMax and DEAR methods in the video domain. OOD datasets are TinyImageNet for the image domain and HMDB51 for the video domain.

izations are in Fig. 5.5, where we can see that InW and OOD samples share a similar uncertainty distribution, while InC samples have smaller uncertainty.

Importance. Based on the above analysis, we conclude that the false positive predictions in OSR tend to be InW samples, since they share similar uncertainty distributions with OOD samples. This conclusion is extremely important as InW samples significantly deteriorate the OSR performance. Without InW samples, the OSR performance increases by a large margin (75.59 to 84.69 for SoftMax method in Tab. 5.2). However, no existing OSR works mentioned and considered this phenomenon. We explicitly point out this conclusion and hope the following researchers take it into account when they design new OSR or UOSR methods.

Why? To deeply understand why InW samples share similar uncertainty distribution with OOD samples instead of InC samples, we begin by analyzing the feature distributions of InC/InW/OOD samples. We first find that the features of InC/InW/OOD follow the hierarchy structure where InC/InW/OOD samples are gradually far away from the training samples, so their features are separable. Please see Fig. 5.6 and Fig. 5.7 for reference. Then we calculate the feature similarity and surprisingly find that InW features are more similar to InC features rather than OOD features in Tab. 5.4, which contradicts the uncertainty score phenomenon. Therefore, the reason that InW samples have similar uncertainty scores with OOD samples is not they have

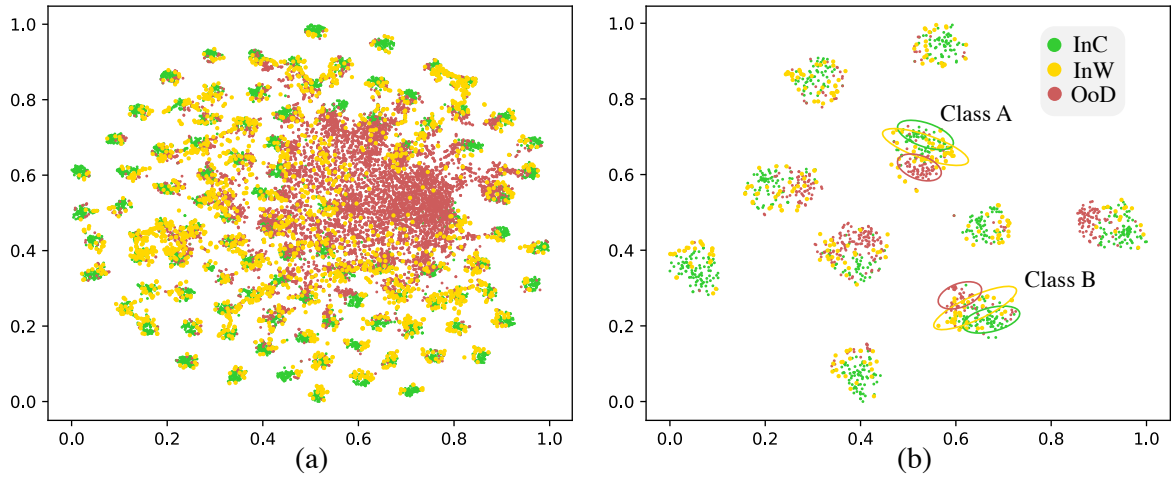


Figure 5.6: (a) and (b) are t-SNE visualization results of the whole test dataset and 10 classes.

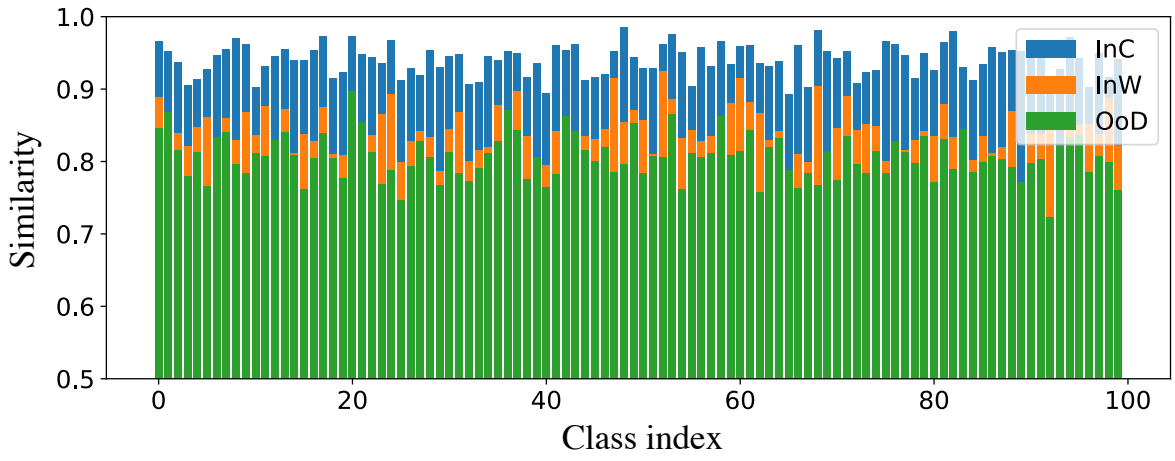


Figure 5.7: Similarity between with training samples of each class.

similar features, but lies in the uncertainty estimation methods. The details are as follows.

Features of InC/InW/OOD samples are separable. We visualize the feature representations in Fig 5.6. We find the feature distribution of InC, InW and OOD samples follow a hierarchy structure. The InW samples surround the InC samples, and OOD samples are further far away and located at the outer edge of InW samples, such as three distributions of class A and B in Fig 5.6 (b). Therefore, the features of InC, InW, and OOD samples are separable from the feature representation perspective. To further testify this idea, we calculate the similarity between InC/InW/OOD samples and training samples of each class, as shown in Fig. 5.7. We can see that the InC samples are the most similar samples with training samples, and InW samples have smaller similarity, and OOD samples have the smallest similarity. In conclusion, features of InC/InW/OOD samples are in a hierarchy structure and distinguishable.

Features of InW samples are more similar with InC samples than OOD samples. Our

Table 5.4: We provide the feature similarity of InW/InC and InW/OOD, the mean of uncertainty score, and the AUROC of InW/InC and InW/OOD in this table.

	Similarity		**Uncertainty**			**Discrimination (AUROC)**	
	InW/InC	**InW/OOD**	**InC**	**InW**	**OOD**	**InW/InC**	**InW/OOD**
Feature space	0.797	0.733	0.057	0.158	0.179	84.91	57.25
Logit space	0.718	0.571	0.314	0.537	0.583	83.36	57.95

finding in Sec. 5.3 is that InW samples share similar uncertainty scores with OOD samples rather than InC samples, so we analyze whether the feature representation also follows the same behavior. We calculate the similarity between InW/InC samples and InW/OOD samples in the feature space and logit space. Then we provide the mean of uncertainty scores based on the KNN method and MaxLogit method, as well as the AUROC of InW/InC and InW/OOD to illustrate the uncertainty discrimination performance. In Tab. 5.4 we can see that the similarity of InW/InC is larger than InW/OOD (0.797-0.733), which means InW samples have more similar features with InC samples than OOD samples. However, the uncertainty scores of InW samples are more similar with OOD scores than InC samples (0.158/0.179-0.057), so that InW/OOD can not be distinguished very well like InW/InC (57.25-84.91). Therefore, we draw a very interesting conclusion that the feature behavior and uncertainty score behavior of InW/InC and InW/OOD are contradictory. InW samples are more similar to InC samples in the feature/logit space but more similar to OOD samples from the uncertainty score perspective.

Let us formulate this phenomenon mathematically. Suppose we have x_c, x_w, x_o which represents the feature of an InC, InW, and OOD sample, respectively. From Tab. 5.4 we know that

$$sim(x_w, x_c) > sim(x_w, x_o), \tag{5.2}$$

where *sim* refers to the similarity. Then, we have an uncertainty estimation function f to measure the uncertainty u of a sample based on the features, so $u_c = f(x_c), u_w = f(x_w), u_o = f(x_o)$. Based on our finding in Fig. 5.5 that InW samples share similar uncertainty distribution with OOD samples rather than InC samples, we have

$$sim(u_w, u_c) < sim(u_w, u_o), \text{or } sim(f(x_w), f(x_c)) < sim(f(x_w), f(x_o)). \tag{5.3}$$

Comparing Eq. 5.3 and Eq. 5.2 we find that the uncertainty estimation function f changes the similarity relationship between InW/InC and InW/OOD.

Let us give a toy example of how f changes the similarity relationship. Suppose the logit

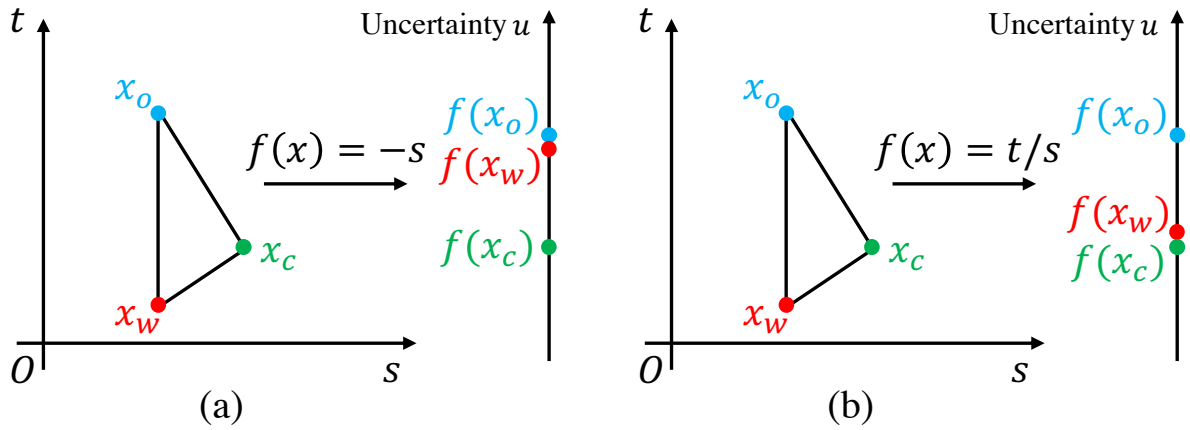


Figure 5.8: (a): x_w is close to x_c in (s,t) space, but $f(x_w)$ is close to $f(x_o)$ in uncertainty space; (b) x_w is close to x_c in (s,t) space, and $f(x_w)$ is also close to $f(x_c)$ in uncertainty space.

space is under 2 dimensions (s,t), and $x_c = (2,2), x_w = (1,1), x_o = (1,3)$. The similarity is measured with Euclidean distance, and in this case x_w is closer to x_c, so Eq. 5.2 holds for this example. If $f(x) = -s$ like Fig. 5.8 (a), then $u_c = -2, u_w = -1, u_o = -1$. In this case, $u_w = u_o > u_c$. So the uncertainty score of InW sample is similar with OOD sample instead of InC sample. This example illustrates why an InW sample has a similar feature to an InC sample, but has a similar uncertainty score with OOD sample. This is how existing uncertainty estimation methods work, as the results in Fig. 5.5 show that InW samples have similar uncertainty distribution with OOD samples. This kind of method is suitable for UOSR problem where InW and OOD samples are supposed to be rejected at the same time.

We provide another uncertainty estimation function f in Fig. 5.8 (b), where $f(x) = t/s$. In this case, $u_w = u_c = 1 < u_o = 3$, so the uncertainty score of the InW sample is similar to the InC sample instead of the OOD sample. This is the ideal case for the traditional OSR problem to reject OOD samples and accept InC and InW samples.

UOSR benchmark under different training set. We provide the UOSR evaluation results in Tab. 5.5 when ID and OOD datasets are TinyImageNet and CIFAR100 respectively. Pre-training weights are used. TinyImageNet has 200 classes in the training set which is more diverse than CIFAR100. We can see that training the model on TinyImageNet is more challenging than CIFAR100, as the Acc. of CIFAR100 is 86.44 and Acc. of TinyImageNet only reaches 77.02. In this way, the impact of InW samples becomes further huge. For example, the performance gap between UOSR and OSR for the SoftMax method is 8.95 when ID dataset is TinyImageNet, and this value is only 0.34 when ID dataset is CIFAR100. So the InW samples are more important for the performance when ID dataset is difficult, as lower closed-set Acc. means more InW samples.

Table 5.5: Unified open-set recognition benchmark in the image domain. All methods are conducted under the R50 model. ID and OOD Dataset are TinyImageNet and CIFAR100 respectively. $\dagger, \ddagger, \diamondsuit$ refer to OSR-based, SP-based, UOSR-based methods. Pre-training weights are used.

	UOSR			OSR	InC/InW	InC/OOD	InW/OOD
Methods	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**
SoftMax†	77.02	340.70	86.22	77.27	88.83	85.62	49.27
ODIN†	77.23	359.57	84.01	75.32	86.53	83.43	47.81
LC†	77.23	385.15	79.76	71.85	83.21	78.98	47.69
OpenMax†	76.90	340.21	86.53	77.09	89.63	85.81	48.04
OLTR†	77.00	341.60	86.04	76.75	89.02	85.36	47.92
PROSER†	75.93	392.91	80.50	74.65	79.20	80.50	55.20
BCE‡	76.91	339.43	86.40	76.83	89.64	85.66	47.44
TCP‡	77.82	336.61	86.48	77.47	89.72	85.76	48.38
DOCTOR‡	77.23	339.02	86.69	77.62	89.86	85.96	49.30
SIRC(MSP, $\|z\|_1$)$^\diamondsuit$	77.03	337.28	86.82	78.86	88.78	86.37	53.65
SIRC(MSP, Res.)$^\diamondsuit$	77.03	316.14	90.66	87.00	88.82	91.08	73.31
SIRC($-\mathcal{H}$, $\|z\|_1$)$^\diamondsuit$	77.03	333.73	87.60	80.13	89.00	87.28	56.19
SIRC($-\mathcal{H}$, Res.)$^\diamondsuit$	77.03	311.98	91.39	88.20	89.02	91.94	75.67

Table 5.6: Relation between closed-set accuracy *Acc.* (%) and open-set performance. *Aug:* Augmentation; *Ep:* Epoch. AUROC (%) is reported.

Aug.	Ep.	Acc.	InC/OOD	InC/InW	OSR	UOSR
✗	100	59.41	81.65	82.47	68.64	81.89
✓	100	68.83	84.18	85.13	73.71	84.40
✓	300	73.28	84.69	85.68	75.59	84.90

A better closed-set model is better for UOSR. Recently, [126] found that better closed-set performance means better OSR performance. We provide a deeper explanation of this finding and show that this conclusion also holds for UOSR. Tab. 5.6 shows that we improve the closed-set accuracy through data augmentation and longer training [126]. The open-set method is the SoftMax baseline with ResNet50 backbone, and the OOD dataset is TinyImageNet. For OSR, we can see that AUROC of InC/OOD is significantly better than OSR, which indicates the uncertainty distribution of InW samples are contradictory with the expectation of OSR. So less InW samples and better InC/OOD performance are two reasons for better OSR performance when closed-set accuracy is higher. For UOSR, both the InC/InW and InC/OOD performance are improving with the growth of closed-set accuracy, which brings better UOSR performance.

Table 5.7: UOSR and MC performance under different temperatures T.

	w/o pre-training						w/ pre-training					
	ECE↓	AURC↓	AUROC↑				ECE↓	AURC↓	AUROC↑			
T	MC	UOSR	UOSR	OSR	InC/InW	InC/OOD	MC	UOSR	UOSR	OSR	InC/InW	InC/OOD
0.1	0.247	355.69	66.58	62.06	66.96	66.48	0.128	257.20	69.07	66.67	68.49	69.15
0.5	0.207	351.39	84.10	74.87	85.01	83.86	0.106	257.26	88.66	83.71	88.61	88.67
1	0.146	358.31	85.57	76.90	**85.18**	85.67	0.081	260.14	90.51	85.93	**89.58**	90.64
2	**0.119**	352.60	86.79	79.19	84.62	87.35	**0.018**	250.99	92.40	89.09	89.08	92.85
5	0.344	351.45	87.05	79.94	84.05	87.84	0.523	**249.00**	**92.96**	90.86	86.00	93.90
10	0.256	351.39	87.08	80.04	83.94	87.90	0.509	249.55	92.93	91.02	85.22	93.97
20	0.197	**351.38**	**87.09**	**80.08**	83.89	**87.92**	0.414	249.73	92.92	**91.07**	84.99	**93.99**

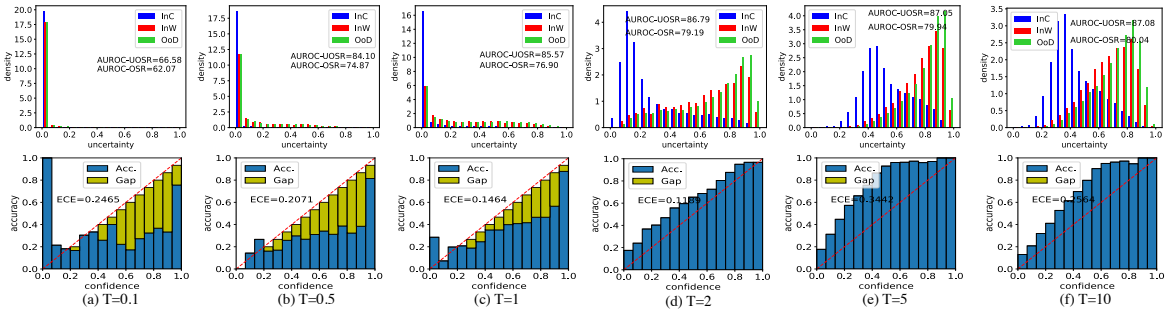
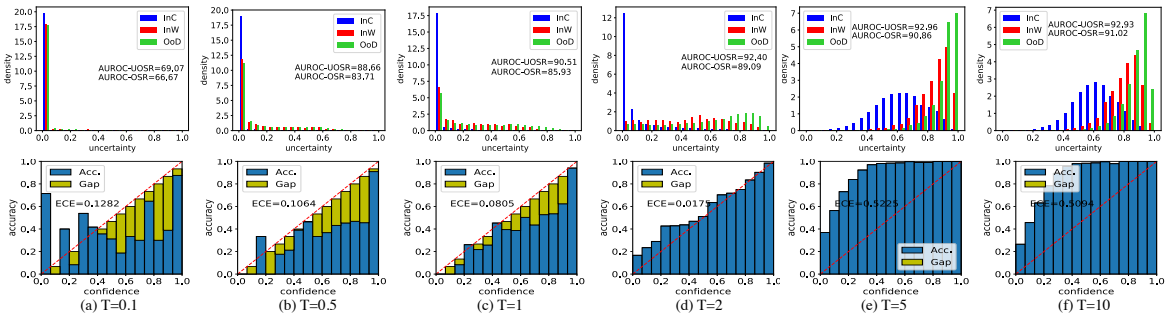


Figure 5.9: Uncertainty distribution under different temperatures T without pre-training.

Temperature scaling for UOSR. Temperature scaling is a convenient and effective method for model calibration [143]. We study how this method influences the UOSR performance. The experiments are conducted under R50 backbone while ID and OOD datasets are CIFAR100 and TinyImageNet, respectively. The quantitative results are in Tab. 5.7. The uncertainty distribution under different temperatures T are in Fig. 5.9 and Fig. 5.10.

From Tab. 5.7 we can see that the optimal T for MC ($T = 2$) is not the best case for UOSR. When T grows, the InC/OOD discrimination increases, but the InC/InW discrimination drops. Therefore, the OSR performance keeps improving with larger T, but the UOSR may not benefit from larger T because of lower InC/InW discrimination. For example, the best T for UOSR with pre-training is 5 rather than 20. But in general, temperature scaling is a simple and useful technique for both MC and UOSR, as $T = 2$ has better MC and UOSR performance than $T = 1$ (without temperature scaling). The only drawback of temperature scaling is it needs the validation set to determine the optimal T.

UOSR benchmark under traditional OSR setting. We evaluate the UOSR and OSR performance under the traditional OSR dataset setting [126], where a part of data within one dataset



(a) T=0.1 (b) T=0.5 (c) T=1 (d) T=2 (e) T=5 (f) T=10

Figure 5.10: Uncertainty distribution under different temperature T with pre-training.

is regarded as ID and the remaining data is regarded as OOD. The experiments are conducted under the most challenging TinyImageNet dataset and the results are in Tab. 5.8. We can see that the UOSR performance is still higher than OSR performance for most methods, which means InW samples share more similar uncertainty distribution with InC samples than OOD samples. Surprisingly, we find the InW/OOD AUROC of ODIN method achieves 84.32, which means InW and OOD samples can be well distinguishable. This proves our claim that the features of InC/InW/OOD samples are separable and it is possible to find a proper uncertainty estimation method to distinguish these three groups of data.

UOSR benchmark under SSB datasets. [126] proposed the Semantic Shift Benchmark (SSB) which compose several fine-grained datasets, including CUB, Standford Cars, FGCV-Aircraft, and a part of ImageNet. OSR in SSB is more challenging as OOD samples share the same coarse labels with ID samples, and only have some minor differences in the fine-grained properties. We evaluate the UOSR performance on CUB and FGCV-Aircraft and provide the results of EASY and HARD modes. Pre-training weights are used for better performance. From Tab. 5.9 and Tab. 5.10 we can see the UOSR and OSR performance are higher under the EASY mode compared to HARD mode as expected, since the OOD samples are more similar with ID samples in the HARD mode. Our conclusion that InW samples share similar uncertainty with OOD samples still holds, as the AUROC of InW/OOD is close to 50 and much lower than InC/OOD and InC/InW.

5.4 Pre-training and Outlier Exposure

After directly applying existing OSR methods for UOSR in Sec. 5.3, we explore two additional training settings in this section, including pre-training [131] and outlier exposure [45], which are effective methods to improve the OSR performance because of introduced extra information beyond the training set. Pre-training is to use large-scale pre-trained weights for initialization

Table 5.8: Unified open-set recognition benchmark in the image domain under the traditional OSR dataset setting. All methods are conducted under the R50 model. Dataset is TinyImageNet. †, ‡, ◊ refer to OSR-based, SP-based, UOSR-based methods. Pre-training weights are used.

	UOSR			OSR	InC/InW	InC/OOD	InW/OOD
Methods	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**
SoftMax†	87.40	730.10	94.06	90.59	91.02	94.11	66.21
ODIN†	87.40	724.90	96.62	95.21	85.50	96.78	84.32
LC†	87.40	756.49	88.02	84.32	81.27	88.11	58.05
OpenMax†	86.60	731.73	94.49	90.25	91.77	94.53	62.59
OLTR†	87.60	731.93	93.69	90.09	89.96	93.74	64.34
PROSER†	86.90	750.26	92.29	90.21	77.01	92.51	74.91
BCE‡	87.60	732.79	93.74	90.46	89.63	93.80	66.92
TCP‡	87.70	731.66	94.20	90.54	90.20	94.25	64.06
DOCTOR‡	87.40	731.69	93.97	90.54	90.25	94.02	66.40
SIRC(MSP, $\|z\|_1$)◊	87.40	731.48	94.02	90.53	91.04	94.06	66.06
SIRC(MSP, Res.)◊	87.40	729.96	94.77	91.71	91.19	94.82	70.13
SIRC($-\mathcal{H}$, $\|z\|_1$)◊	87.40	730.12	94.74	91.65	91.24	94.79	69.89
SIRC($-\mathcal{H}$, Res.)◊	87.40	731.34	94.06	90.59	91.02	94.10	66.22

for better down-task performance. Outlier exposure is to introduce some unlabeled outlier data (OD) during training and regard these outlier data as the proxy as OOD data to improve the open-set performance. We find both of them also have a positive effect on UOSR, but for different reasons.

Pre-training settings. In the image domain, we use BiT pre-training weights [155] for ResNet50 and ImageNet [128] pre-training weights for VGG13. In the video domain, Kinetics400 [104] pre-trained weights are used for initialization.

Outlier exposure settings. In the image domain, we use 300K Random Images dataset from [45] as outlier dataset for those outlier exposure methods. The 300K Random Images dataset is a debiased dataset with real images scraped from online. According to [45], all images that belong to CIFAR classes and images with divisive metadata have been removed. In the video domain, we use Kinetics400 as our outlier datasets for those outlier exposure methods. To ensure that the classes of outlier data is not overlapping with ID data and OOD data, we remove corresponding classes in Kinetics400. The overlapping classes between Kinetics400 and UCF101/HMDB51 are too many to be listed here (129 overlapping classes). The available training sample ID list of Kinetics400 and all codes will be public. We pick up 271 classes from Kinetics400 and 25

Table 5.9: Unified open-set recognition benchmark of CUB-200-2011 dataset. All methods are conducted under the R50 model. †, ‡, ◊ refer to OSR-based, SP-based, UOSR-based methods. Pre-training weights are used. EASY/HARD

	UOSR			OSR	InC/InW	InC/OOD	InW/OOD
Methods	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**
SoftMax†	91.78	77.69/120.46	92.79/84.31	90.33/78.78	90.56	93.37/82.81	56.34/33.73
ODIN†	91.61	86.89/157.09	91.20/77.37	91.45/73.11	82.42	93.52/76.14	68.90/40.09
LC†	91.61	78.34/121.66	92.66/84.35	89.86/78.63	91.15	93.06/82.69	54.95/34.38
OpenMax†	91.30	78.07/119.61	92.87/85.43	90.01/79.78	91.14	93.35/83.98	54.99/35.67
OLTR†	91.33	80.38/118.50	92.43/85.30	89.42/79.72	90.66	92.91/83.95	52.65/35.19
PROSER†	91.33	79.39/128.14	92.50/83.81	90.32/78.53	89.31	93.37/82.42	58.21/37.60
BCE‡	91.50	79.75/122.19	92.24/84.76	89.34/79.27	90.64	92.67/83.31	53.43/35.80
TCP‡	92.06	77.31/116.93	92.55/84.85	90.28/79.92	90.07	93.17/83.64	56.71/36.78
DOCTOR‡	91.61	78.09/121.61	92.76/84.37	90.09/78.68	91.13	93.18/82.71	56.26/34.70
SIRC(MSP, $\|z\|_1$)◊	91.78	78.04/119.42	92.72/84.46	90.19/78.95	90.59	93.27/83.00	55.78/33.70
SIRC(MSP, Res.)◊	91.78	77.69/120.46	92.79/84.31	90.33/78.78	90.56	93.37/82.81	56.33/33.74
SIRC($-\mathcal{H}$, $\|z\|_1$)◊	91.78	76.97/119.40	93.11/84.52	91.06/83.08	90.52	93.78/83.08	60.65/34.78
SIRC($-\mathcal{H}$, Res.)◊	91.78	76.65/120.65	93.18/84.33	91.19/78.91	90.47	93.88/82.86	61.20/34.81

samples in each class as outlier data. We implement several outlier exposure based methods, including OE [45], EB [133], VOS [162] and MCD [132].

Implementation details. In the image domain, when we train the model from scratch, we find that setting the base learning rate as 0.1 and step-wisely decayed by 10 every 24000 steps with totally 120000 steps can achieve good enough closed-set performance. We use a linear warmup strategy to warmup the training in the first 500 steps. We use SGD with momentum, batch size 128 for all models. When we fine tune the model with ImageNet pretrained model from [155], we set the base learning rate as 0.003 and step-wisely decayed every 3000 steps with totally 10000 steps. We use a linear warmup strategy to warmup the training in the first 500 steps. We use SGD with momentum, batch size 512 for all models. For outlier exposure methods, the batch size of outlier data is set to 128. In the video domain, when we train the model from scratch, we find that setting the base learning rate as 0.05 and step-wisely decayed every 160 epochs with totally 400 epochs can achieve good enough closed-set performance. The batch size is 256 for all methods. We follow [77] to set the base learning rate as 0.001 and step-wisely decayed every 20 epochs with totally 50 epochs. For those methods without outlier exposure, we fix the parameters of all Batch Normalization layers except the first one, and set the learning

Table 5.10: Unified open-set recognition benchmark of Fine-Grained Visual Classification of Aircraft (FGVC-Aircraft) dataset. All methods are conducted under the R50 model. †, ‡, ◊ refer to OSR-based, SP-based, UOSR-based methods. Pre-training weights are used. EASY/HARD

	UOSR			OSR	InC/InW	InC/OOD	InW/OOD
Methods	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**
SoftMax†	85.61	129.71/127.18	89.46/82.46	84.24/72.85	88.54	89.80/79.08	51.18/35.75
ODIN†	85.25	152.17/173.15	87.11/75.75	84.92/68.36	80.04	89.72/73.32	57.17/39.70
LC†	85.25	134.44/144.29	88.98/80.15	83.55/69.74	87.58	89.50/75.93	49.13/33.95
OpenMax†	86.69	123.31/123.86	90.13/82.10	85.63/73.20	88.12	90.80/79.01	51.99/35.36
OLTR†	85.97	128.01/124.92	89.86/82.79	85.11/73.45	88.57	90.31/79.66	53.26/35.35
PROSER†	85.97	124.15/137.70	90.73/80.40	86.92/70.44	87.31	91.93/76.67	56.24/32.22
BCE‡	85.37	134.56/127.05	88.75/82.37	84.05/73.93	86.49	89.58/80.06	51.75/38.17
TCP‡	85.25	132.09/131.80	89.30/82.19	83.60/72.12	88.41	89.63/78.66	48.73/34.30
DOCTOR‡	85.25	133.93/144.07	89.16/80.24	83.79/69.80	87.68	89.71/76.00	49.58/33.96
SIRC(MSP, $\|z\|_1$)◊	85.61	130.50/126.36	89.24/82.57	83.86/73.04	88.52	89.49/79.27	50.31/35.98
SIRC(MSP, Res.)◊	85.61	129.71/127.18	89.46/82.46	84.24/72.85	88.54	89.80/79.08	51.17/35.76
SIRC($-\mathcal{H}$, $\|z\|_1$)◊	85.61	128.97/125.59	89.71/82.85	84.70/73.60	88.55	90.13/79.70	52.40/37.35
SIRC($-\mathcal{H}$, Res.)◊	85.61	128.04/126.58	89.97/82.69	85.16/73.35	88.52	90.49/79.46	53.46/37.01

rate of the fully connected layer to be 10 times of the base learning rate. For those methods with outlier exposure, all parameters are updated with the same learning rate.

UOSR benchmark settings. We include OSR-based, SP-based, and UOSR-based methods to build a comprehensive UOSR benchmark. We implement several Selective Prediction (SP) based methods, including BCE, TCP [135], CRL [136], and DOCTOR [137]. Although these SP methods are originally designed to differentiate InC and InW samples, we adopt them in the UOSR setting to build a more comprehensive benchmark. The only UOSR-based method is SIRC [134], which combines two uncertainty scores for better UOSR performance. The evaluation metrics are AURC and AUROC [130].

Results. The UOSR benchmarks for image and video domains are Tab. 5.11, Tab. 5.12, and Tab. 5.13 respectively. First, we can see the AUROC of UOSR is higher than OSR for almost all the methods under both settings, *i.e.*, pre-training and outlier exposure, which further strengthens the conclusion that the uncertainty distribution of OSR methods is closer to the ground truth of UOSR. For instance, AUROC is 89.59 for UOSR and 83.93 for OSR under the Ensemble method w/o pre-training in Tab. 5.11. Second, pre-training and outlier exposure can effectively boost the UOSR and OSR performance , *e.g.*, the pre-training boost the AUROC

Table 5.11: UOSR benchmark in the image domain under the ResNet50 model. ID dataset is CIFAR100 while the OOD dataset is TinyImageNet. †, ‡, ◊ refer to OSR-based, SP-based, UOSR-based methods. OD: Outlier Data. N/G/R means No/Generated/Real OD. AUROC (%), AURC ($\times 10^3$) and Acc. (%) are reported.

		w/o pre-training				**w/ pre-training**			
		UOSR			**OSR**	**UOSR**			**OSR**
Methods	**OD**	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**
ODIN†	N	72.08	371.92	86.69	81.65	86.48	261.42	91.81	91.47
LC†	N	72.08	372.55	84.16	76.37	86.48	280.69	87.68	85.16
OpenMax†	N	73.64	361.22	85.33	76.23	86.43	261.53	90.29	85.86
MaxLogits†	N	73.89	351.39	87.09	80.11	86.42	249.86	92.91	91.11
Entropy†	N	73.89	355.54	86.25	78.14	86.43	257.36	91.21	87.22
OLTR†	N	73.69	357.79	85.94	77.10	86.22	260.17	90.59	86.05
Ensemble†	N	76.78	314.56	89.59	83.93	87.41	240.34	93.67	92.02
Vim†	N	73.89	341.88	87.37	80.92	86.43	246.59	93.01	92.33
BCE‡	N	73.29	369.91	84.29	74.59	86.39	257.74	90.79	86.50
TCP‡	N	71.80	369.17	85.17	75.46	86.83	261.97	89.95	85.67
DOCTOR‡	N	72.08	378.45	84.48	75.06	86.48	262.05	90.24	85.75
SIRC(MSP, $\|z\|_1$)◊	N	73.13	358.93	85.77	76.67	86.44	260.08	90.53	85.98
SIRC(MSP, Res.)◊	N	73.13	348.40	87.26	80.29	86.44	247.75	92.99	90.39
SIRC($-\mathcal{H}$, $\|z\|_1$)◊	N	73.13	355.62	86.57	78.16	86.44	257.28	91.23	87.27
SIRC($-\mathcal{H}$, Res.)◊	N	73.13	346.21	87.71	81.43	86.44	244.55	93.68	91.62
SoftMax†	N	73.28	363.55	84.90	75.59	86.44	260.14	90.50	85.93
SoftMax†(OE†)	R	73.54	339.59	88.78	84.35	85.43	255.77	92.54	90.65
ARPL†	N	73.03	345.84	88.13	80.49	84.67	301.27	87.01	83.49
ARPL+CS†	R	72.78	349.50	87.65	82.81	83.60	268.00	92.00	91.17
MCD†(Dropout†)	N	76.49	375.01	82.25	79.21	87.21	301.71	83.57	81.69
MCD†	R	70.88	365.82	86.97	79.88	81.96	287.21	90.43	86.59
PROSER†	G	68.08	394.48	84.82	79.23	81.32	301.78	90.20	89.29
PROSER†(EB†)	R	71.82	366.65	86.06	81.95	85.06	269.79	90.84	90.38
VOS †	G	73.44	356.68	86.65	79.72	86.62	249.24	92.94	91.37
VOS †	R	73.18	331.12	89.96	85.78	85.93	251.44	92.92	91.34
OpenGAN †	R	73.61	334.04	88.92	85.48	86.25	260.19	91.21	90.94

of UOSR/OSR from 84.90/75.59 to 90.50/85.93 for the SoftMax, and outlier exposure method OE has 90.19/86.29 AUROC of UOSR/OSR compared to 84.90/75.59 of SoftMax in Tab. 5.11. Note that to ensure outlier data is useful, we keep methods w/ and w/o outlier data as similar as possible, like SoftMax/OE, ARPL/ARPL+CS, and Dropout/MCD. Third, real outlier data is more beneficial than generated outlier data in the UOSR and OSR tasks. The UOSR/OSR AUROC of VOS method is 89.96/85.78 for real outlier data and 86.65/79.72 for generated outlier data, provided in Tab. 5.11.

Analysis. To better understand why pre-training and outlier exposure are helpful for UOSR and OSR, we provide the InC/InW and InC/OOD discrimination performance of each method in Fig. 5.11. We can see that almost all outlier exposure methods are in the Q2, which means that

Table 5.12: UOSR benchmark in the video domain under the TSM model. ID dataset is UCF101 while the OOD dataset is HMDB51. †, ‡, ◊ refer to OSR-based, SP-based, UOSR-based methods. OD: Outlier Data. AUROC (%), AURC ($\times 10^3$) and Acc. (%) are reported.

		w/o pre-training				**w/ pre-training**			
		UOSR			**OSR**	**UOSR**			**OSR**
Methods	**OD**	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**
OpenMax†	N	73.92	185.81	85.95	82.27	95.32	75.75	91.22	90.89
BNN SVI†	N	71.51	181.45	89.44	80.10	94.71	69.89	93.58	91.81
RPL†	N	71.46	186.18	88.70	79.67	95.59	72.88	92.44	90.53
DEAR†	N	71.33	215.80	87.56	80.00	94.41	102.01	91.50	91.49
BCE‡	N	69.90	223.57	83.27	78.96	93.66	110.42	83.83	81.64
CRL‡	N	71.80	183.76	88.75	78.57	95.22	67.61	93.36	91.38
DOCTOR‡	N	72.01	182.04	88.73	79.76	95.06	65.61	93.89	91.80
SIRC(MSP, $\|z\|_1$)◊	N	73.59	173.32	88.71	80.33	95.00	65.97	93.74	91.42
SIRC(MSP, Res.)◊	N	73.59	174.76	88.17	78.74	95.00	65.43	93.83	91.73
SIRC($-\mathcal{H}$, $\|z\|_1$)◊	N	73.59	172.30	89.27	81.61	95.00	66.11	94.06	91.95
SIRC($-\mathcal{H}$, Res.)◊	N	73.59	175.79	88.50	79.62	95.00	69.06	94.07	92.18
SoftMax†	N	73.92	173.14	88.42	79.72	95.03	68.08	93.94	91.75
SoftMax†(OE†)	R	74.42	162.36	90.19	86.29	94.71	67.93	94.33	93.40
MCD†(Dropout†)	N	73.63	184.66	85.58	75.75	95.06	79.53	90.30	88.23
MCD†	R	72.49	168.83	91.26	85.57	93.47	71.19	95.34	93.68
VOS†	G	74.00	187.82	86.10	84.51	95.27	65.68	94.44	93.62
VOS†	R	74.68	172.71	87.98	87.09	94.79	64.99	94.97	93.72
EB†	R	70.90	212.01	85.32	86.47	94.66	67.83	94.40	93.06

outlier exposure methods have lower InC/InW AUROC and higher InC/OOD AUROC than the SoftMax baseline. For OSR, both lower InC/InW and higher InC/OOD AUROC are beneficial, but only higher InC/OOD AUROC is wanted for UOSR, while lower InC/InW AUROC is not. This can explain why some of the UOSR performances with outlier exposure are comparable or even worse than the baseline, such as EB and VOS in the video domain Tab. 5.12, but all of the OSR performances are increased. In contrast, pre-training is helpful for both InC/InW and InC/OOD AUROC, as methods with pre-training are in the Q1. Outlier exposure methods also benefit from pre-training, as green marks are at the upper right location compared to orange marks. Therefore, outlier exposure and pre-training bring the UOSR performance gain for different reasons. InC/OOD performance is improved by both of the techniques, but pre-training is also helpful for InC/InW, while outlier exposure is not. This can be explained by the fact that the model may see additional ID samples during pre-training so that the closed-set accuracy and InC/InW discrimination are improved. In contrast, outlier exposure only provides OOD

Table 5.13: Unified open-set recognition benchmark in the image domain. All methods are conducted under the VGG13 model. ID dataset is CIFAR100 while OOD dataset is TinyImageNet. †, ‡, ◊ refer to OSR-based, SP-based, UOSR-based methods. OD: use Outlier Data in training.

		w/o pre-training				**w/ pre-training**			
		UOSR			**OSR**	**UOSR**			**OSR**
Methods	**OD**	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**
SoftMax†	✗	75.07	341.06	85.87	76.44	74.69	311.70	91.01	85.13
ODIN†	✗	75.07	346.70	85.27	78.13	74.69	314.80	91.24	88.90
LC†	✗	75.07	365.96	81.77	76.68	74.69	339.96	88.27	88.95
OpenMax†	✗	74.52	368.58	82.81	72.68	75.05	312.70	90.47	84.25
OLTR†	✗	73.80	365.61	84.68	74.37	74.52	306.73	92.12	87.08
PROSER†	✗	70.95	376.63	84.00	79.29	71.11	367.16	86.45	85.97
BCE‡	✗	74.74	339.25	86.54	76.99	74.45	325.29	88.85	81.14
TCP‡	✗	75.09	340.41	85.94	76.18	74.83	313.80	90.57	84.35
DOCTOR‡	✗	75.07	340.69	85.95	76.72	74.69	310.29	91.37	85.97
SIRC(MSP, $\|z\|_1$)◊	✗	75.07	343.44	85.33	75.63	74.69	313.22	90.67	84.59
SIRC(MSP, Res.)◊	✗	75.07	336.87	86.77	78.23	74.69	309.67	91.46	86.14
SIRC($-\mathcal{H}$, $\|z\|_1$)◊	✗	75.07	343.53	85.25	76.37	74.69	309.19	91.56	86.82
SIRC($-\mathcal{H}$, Res.)◊	✗	75.07	335.71	86.93	79.13	74.69	305.34	92.39	88.34
OE	✓	71.71	312.44	93.71	89.76	73.38	306.72	93.34	92.77
EB	✓	74.19	340.66	87.67	85.73	72.76	340.81	89.40	89.79
VOS	✓	71.73	312.75	93.68	90.76	73.08	306.37	93.65	92.97
MCD	✓	70.45	316.57	94.21	91.54	72.08	300.87	95.40	94.26

data but no more ID data, so only InC/OOD performance is improved. Tab. 5.14 is the complementary analysis of Tab. 5.11, which further illustrates that outlier data can improve InC/OOD discrimination but may not be helpful for InC/InW discrimination.

UOSR under Noisy Outlier Exposure. When we introduce outlier data into the training process, the labels of some outlier data may be corrupted and become the labels of ID class. We call this kind of outlier data as noisy outlier data. We study how noisy outlier data influence the UOSR and OSR performance in this section. In addition, we use NGC [163] to find the noisy outlier data and correct them. The results are shown in Tab. 5.15. We can see that the closed-set Acc. gradually decreases with the growth of noise level when NGC is not used. This is natural as some noisy outlier data corrupt the ID data distribution. In contrast, the closed-set Acc. does not drop with the growth of noise level when NGC is used. So NGC is very effective in finding

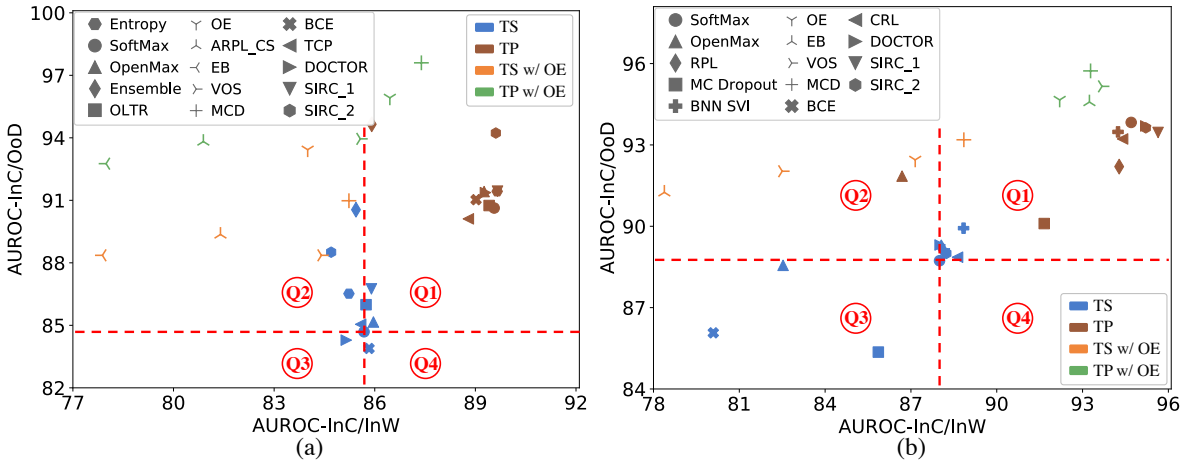


Figure 5.11: (a) and (b) plot the InC/InW and InC/OOD discrimination in the image and video domain. We set the SoftMax method training from scratch as the original point and divide the coordinate system into 4 quadrants (Q1 to Q4). (TS: Train from Scratch. TP: Train from Pre-training. OE: Outlier Exposure.)

corrupted samples. When the noise level is 0%, the UOSR performance with NGC is better than the performance without NGC (lower AURC value), meaning that other parts in NGC that are not related to label corruption, such as contrastive learning, are helpful for UOSR. Surprisingly, the performance of UOSR and OSR are relatively stable under different noise levels no matter we use NGC or not, compared to the clear performance drop of closed-set Acc. when NGC is not used. So the model is robust in open-set related performance when noisy outlier data is introduced. Our finding that ID samples share similar uncertainty scores with OOD samples still holds as AUROC of InW/OOD is close to 50.

5.5 Few-shot Unified Open-set Recognition

In addition to the analysis of two useful training settings in Sec. 5.4, we introduce a new evaluation setting into UOSR in this section. Inspired by the recent work SSD [118] which proposes the few-shot OSR, we introduce the few-shot UOSR, where 1 or 5 samples per OOD class can be introduced for reference to better distinguish OOD samples. The introduced reference OOD datasets are marked as $\mathcal{D}_{test}^{ref} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N'} \subset \mathcal{X} \times \mathcal{U}$. We show that few-shot UOSR has distinct challenges from few-shot OSR because of InW samples, and we propose our FS-KNNS method to achieves the state-of-the-art UOSR performance even without outlier exposure during training.

Baselines and dataset settings. SSD is the only existing few-shot OSR method that utilizes the Mahalanobis distance with ID training samples $\mathcal{D}_{train}$ and OOD reference samples $\mathcal{D}_{test}^{ref}$.

Table 5.14: Uncertainty distribution analysis in image domain with ResNet50. Pre-training is not used. OOD dataset: TinyImageNet. AUROC (%) is reported.

Methods	OD	InC/OOD	InC/InW	InW/OOD	OSR	UOSR
ODIN	N	88.35	80.76	64.36	81.65	86.69
LC	N	84.60	82.58	55.14	76.37	84.16
OpenMax	N	85.16	85.96	51.27	76.23	85.33
OLTR	N	85.99	85.74	52.22	77.10	85.94
SoftMax	N	84.69	85.68	50.64	75.59	84.90
SoftMax(OE)	R	90.04	84.00	68.54	84.35	88.78
ARPL	N	88.76	85.77	58.09	80.49	88.13
ARPL+CS	R	89.36	81.40	65.32	82.81	87.65
MCD(Dropout)	N	84.15	74.17	63.15	79.21	82.25
MCD	R	87.47	85.23	61.39	79.88	86.97
PROSER	G	87.04	77.84	62.57	79.23	84.82
PROSER (EB)	R	88.36	77.88	65.60	81.95	86.06
VOS	G	87.51	83.38	58.17	79.72	86.65
VOS	R	91.44	84.41	70.32	85.78	89.96

KNN [31] utilizes the feature distance with samples in $\mathcal{D}_{train}$ as uncertainty scores and shows it is better than Mahalanobis distance. So we slightly modify KNN to adapt to the few-shot UOSR and find the modified FS-KNN can already beat SSD. $\mathcal{D}_{test}^{ref}$ comes from the validation set of TinyImageNet and training set of UCF101 in the image and video domain. We repeat the evaluation process until all OOD reference samples are used and calculate the mean as the final few-shot result.

FS-KNN. Given a test sample $\mathbf{x}^*$ and a feature extractor f, the feature of $\mathbf{x}^*$ is $\mathbf{z}^* = f(\mathbf{x}^*)$. The cosine similarity set between $\mathbf{z}^*$ and the feature set of $\mathcal{D}_{train}$ is $\mathcal{S}_{train} = \{d_i\}_{i=1}^{N}, d_i = (\mathbf{z}^* \cdot \mathbf{z}_i)/(\|\mathbf{z}^*\| \cdot \|\mathbf{z}_i\|), \mathbf{z}_i = f(\mathbf{x}_i), \mathbf{x}_i \in \mathcal{D}_{train}$. $\mathcal{S}_{test}^{ref}$ is similar with $\mathcal{S}_{train}$ except $\mathbf{x}_i$ comes from $\mathcal{D}_{test}^{ref}$. We believe the uncertainty should be higher if the test sample is more similar to reference OOD samples and not similar with ID samples. Therefore, the uncertainty score is

$$\hat{u}_{fs-knn} = 1 - topK(S_{train}) + topK(S_{test}^{ref}), \tag{5.4}$$

where $topK$ means the K^{th} largest value. The performances of SoftMax, KNN, FS-KNN, and SSD are in Tab. 5.16 and Tab. 5.18. Pre-training is used for better performance. FS-KNN has better InC/OOD performance than KNN as reference OOD samples are introduced. Although the overall UOSR performance of FS-KNN is better than the SoftMax baseline, *the InC/InW per-*

Table 5.15: UOSR and OSR performance under noisy outlier data. ID dataset is CIFAR100 and outlier dataset is 300K Random Images. OOD dataset is TinyImageNet. Experiments are conducted with ResNet18 backbone.

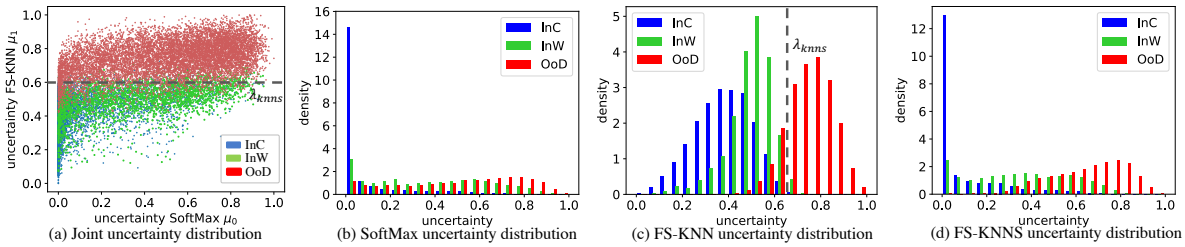
		UOSR			OSR	InC/InW	InC/OOD	InW/OOD
Noise level	**NGC**	**Acc.↑**	**AURC↓**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**	**AUROC↑**
0%	✗	76.19	334.13	85.50	77.29	85.05	85.60	50.70
20%	✗	72.15	343.46	87.84	77.53	87.34	87.98	50.45
40%	✗	70.06	343.60	89.91	79.86	88.36	90.37	55.25
60%	✗	69.46	346.62	90.07	79.79	88.76	90.47	55.52
80%	✗	69.20	348.18	89.95	79.05	88.84	90.29	53.77
100%	✗	68.05	356.35	89.97	78.31	89.63	90.08	53.23
0%	✓	77.52	315.32	87.75	81.12	85.61	88.23	56.61
20%	✓	77.50	316.86	87.81	80.46	86.88	88.02	54.40
40%	✓	76.87	321.66	87.58	79.48	87.15	87.68	52.23
60%	✓	77.07	319.58	87.91	79.88	87.43	88.02	52.53
80%	✓	77.12	317.90	88.17	80.14	87.41	88.34	52.49
100%	✓	77.38	328.85	86.24	78.00	86.46	86.20	49.94

formance is significantly sacrificed, which is also an important aspect of UOSR. For example, InC/InW performance drops from 89.58 to 79.58 in the 5-shot results of Tab. 5.16. So we naturally ask a question: *Can we improve the InC/OOD performance based on the introduced OOD reference samples while keeping similar InC/InW performance with SoftMax baseline?* This is the key difference between few-shot OSR and few-shot UOSR, as low InC/InW performance is wanted in OSR but not preferred in UOSR.

FS-KNNS. Inspired by SIRC [134], we aim to find a way to fuse SoftMax and FS-KNN scores so that the mixed score can keep the high InC/OOD performance of FS-KNN and meanwhile has the comparable InC/InW performance with SoftMax. The uncertainty distributions of SoftMax and FS-KNNS are depicted in Fig. 5.12. We find that OOD samples have larger uncertainty than InW and InC samples in the FS-KNN method, but the uncertainty of InC samples overlaps a lot with InW samples, which explains the reason that InC/OOD performance is high but InC/InW performance is low. In contrast, InW and OOD samples share similar uncertainty in the SoftMax method, which brings higher InC/InW performance. Therefore, we want to keep the uncertainty of InW samples in the SoftMax method, as well as the uncertainty of OOD samples in the FS-KNN method. In this way, the mixed scores obtain the high InC/OOD performance from FS-KNN while keeping the comparable InC/InW performance of SoftMax. We call this method

Table 5.16: Results of few-shot UOSR in the image domain. Model is ResNet50 with pre-training. ID and OOD datasets are CIFAR100 and TinyImageNet. AUROC (%) and AURC ($\times 10^3$) are reported.

	5-shot					1-shot				
	AURC↓	AUROC↑				AURC↓	AUROC↑			
Methods	**UOSR**	**UOSR**	**OSR**	**InC/OOD**	**InC/InW**	**UOSR**	**UOSR**	**OSR**	**InC/OOD**	**InC/InW**
SoftMax	260.14	90.51	85.93	90.64	89.58	260.14	90.51	85.93	90.64	89.58
KNN	245.48	93.45	92.34	94.86	83.08	245.48	93.45	92.34	94.86	83.08
FS-KNN	238.54	95.09	95.91	97.20	79.58	239.71	94.67	94.91	96.52	81.04
SSD	246.42	94.95	**97.89**	**98.32**	70.14	245.99	94.16	**96.51**	**97.16**	72.09
FS-KNN+S	239.49	94.14	91.64	95.27	85.85	240.30	93.97	91.36	94.99	86.45
FS-KNN*S	255.82	91.26	87.10	91.47	**89.67**	255.69	91.30	87.16	91.51	**89.69**
SIRC	241.58	93.85	91.42	94.44	89.58	239.64	94.20	92.03	94.87	89.26
FS-KNNS	**231.61**	**95.51**	94.16	96.54	87.98	**234.84**	**94.91**	93.10	95.84	88.08



(a) Joint uncertainty distribution (b) SoftMax uncertainty distribution (c) FS-KNN uncertainty distribution (d) FS-KNNS uncertainty distribution

Figure 5.12: Uncertainty scores of each test sample (a) and uncertainty distribution of SoftMax (b), FS-KNN (c), and FS-KNNS (d).

FS-KNN with SoftMax (FS-KNNS), and the uncertainty is

$$\hat{u}_{fs-knns} = u_0 + \frac{1}{1 + e^{-\alpha(u_1 - \lambda_{knns})}} u_1, \tag{5.5}$$

where u_0 and u_1 refer to the uncertainty score of SoftMax and FS-KNN, respectively. λ_{knns} is a threshold to determine when the weight of u_1 becomes large, and α is a coefficient to control the change rate of the weight. $\hat{u}_{fs-knns}$ will be largely influenced by u_1 when $u_1 > \lambda_{knns}$. A proper λ_{knns} should be located between the InW and OOD samples, as shown in Fig. 5.12 (a) and (c). In this way, the uncertainty of InW samples is mainly controlled by SoftMax, and OOD samples are strengthened by FS-KNN.

Results. The uncertainty distribution of FS-KNNS is shown in Fig. 5.12 (d), which shows the uncertainty of InC and InW samples are similar to SoftMax (b), but the uncertainty of OOD

Table 5.17: Results of few-shot UOSR in the image domain. Model is VGG13 with pre-training. ID and OOD datasets are CIFAR100 and TinyImageNet respectively.

	5-shot					1-shot				
	AURC↓	AUROC↑				AURC↓	AUROC↑			
Methods	**UOSR**	**UOSR**	**OSR**	**InC/OOD**	**InC/InW**	**UOSR**	**UOSR**	**OSR**	**InC/OOD**	**InC/InW**
SoftMax	311.68	91.00	85.13	92.10	86.68	311.68	91.00	85.13	92.10	86.68
KNN	306.04	93.04	91.61	95.75	82.32	306.04	93.04	91.61	95.75	82.32
FS-KNN	295.80	95.44	98.87	**99.39**	79.83	296.75	95.27	98.48	99.16	79.87
SSD	321.31	92.57	**99.26**	99.36	65.72	315.46	93.11	**99.10**	**99.27**	68.77
FS-KNN+S	296.42	94.22	92.15	96.50	85.20	296.67	94.20	92.15	96.48	85.16
FS-KNN*S	303.01	92.85	89.42	94.38	**86.83**	302.84	92.88	89.45	94.41	**86.83**
SIRC	287.91	95.71	96.33	98.00	86.66	288.12	95.68	96.30	98.00	86.53
FS-KNNS	**285.85**	**95.95**	96.52	98.30	86.66	**287.02**	**95.75**	95.96	98.04	86.67

samples is larger. From Tab. 5.16, Tab. 5.18, and Tab. 5.17, we can see our FS-KNNS has significantly better InC/InW performance than FS-KNN (87.98 and 79.58 under ResNet50), and meanwhile keeps the high InC/OOD performance, so the overall UOSR performance is better than both SoftMax and FS-KNN. We also try three score fusion methods, including FS-KNN+S, FS-KNN*S, and SIRC, but these methods are general score fusion methods, while our method is specifically designed for UOSR, so our FS-KNNS surpass them. Our FS-KNNS is totally based on the existing model trained by the classical cross-entropy loss, so there is no extra effort and no outlier data during training, and our FS-KNNS still has better performance than all outlier exposure methods and achieves state-of-the-art UOSR performance, as shown in Fig. 5.1 (b) and Fig. 5.13. Note that the best choice for OSR (FS-KNN or SSD) may not be the best choice for UOSR (FS-KNNS), as their expectation of InC/InW is contradictory, which shows the necessity of our proposed few-shot UOSR.

Analyze of K. Similar with [31], we find the value of K influences the performance a lot. The ablation study of K is shown in Fig. 5.14. It shows that the best K for InC/OOD discrimination is between 3 and 5, and drops quickly after 7. In contrast, the InC/InW performance keeps increasing until 20 and then drops. The overall UOSR performance achieves the best when $K = 5$.

Analyze of α and λ_{knns}. Two hyper parameters of FS-KNNS including α and λ_{knns} are significant for the performance. λ_{knns} is a threshold to determine when the weight of u_1 becomes large, *i.e.*, $\hat{u}_{fs-knns}$ is important when $u_1 > \lambda_{knns}$. α is to control the change rate of the weight. The

Table 5.18: Results of few-shot UOSR in the video domain. Model is TSM with pre-training. ID and OOD datasets are UCF101 and HMDB51. AUROC (%) and AURC ($\times 10^3$) are reported.

	5-shot					1-shot				
	AURC↓	AUROC↑				AURC↓	AUROC↑			
Methods	**UOSR**	**UOSR**	**OSR**	**InC/OOD**	**InC/InW**	**UOSR**	**UOSR**	**OSR**	**InC/OOD**	**InC/InW**
SoftMax	66.55	93.66	91.44	93.46	**94.99**	66.55	93.66	91.44	93.46	**94.99**
KNN	73.73	93.38	92.99	94.11	88.44	73.73	93.38	92.99	94.11	88.44
FS-KNN	68.44	95.04	**96.14**	**96.71**	83.74	73.19	93.60	**94.32**	95.09	83.53
SSD	70.36	93.91	95.41	95.97	79.97	75.61	92.48	93.34	94.07	81.75
FS-KNN+S	65.64	95.44	94.69	96.13	90.78	68.98	94.54	93.66	95.14	90.48
FS-KNN*S	65.29	94.09	92.13	93.96	**94.99**	65.42	94.04	92.08	93.91	94.97
SIRC	62.41	95.11	93.79	95.13	94.94	63.11	94.88	93.49	94.88	94.84
FS-KNNS	**60.00**	**96.19**	95.37	96.40	94.82	**62.65**	**95.35**	94.26	**95.48**	94.49

ideal λ_{knns} should be located between the uncertainty of InW and OOD samples as shown in Fig. 5.12 (a) and (c), so that the uncertainty of InW samples is still determined by SoftMax and the uncertainty of OOD samples is enlarged because of FS-KNN. However, which sample is InW or OOD is unknown during test, so we cannot determine the λ_{knns} based on the uncertainty distribution of test samples. [134] proposed to determine the hyper parameters of SIRC based on the training set, but we find that the uncertainty distribution $\hat{u}_{fs-knns}$ of training samples significantly different from the test InC samples, as shown in Fig. 5.15. Therefore, we seek help from the OOD reference samples, as we find their uncertainty distribution is extremely similar with OOD test samples. We calculate the mean $\overline{u}$ and standard deviation σ of the $\hat{u}_{fs-knns}$ of OOD reference samples, and we aim to find a proper λ_{knns} through:

$$\lambda_{knns} = \overline{u} - \beta \cdot \sigma \tag{5.6}$$

We draw several λ_{knns} with different β in Fig. 5.15 (a to e). A smaller λ_{knns} or a larger β means more samples will be influenced by the FS-KNN uncertainty, so that the $\hat{u}_{fs-knns}$ of more OOD test samples will be strengthened by the FS-KNN uncertainty which brings better InC/OOD performance, but meanwhile more InW samples will also be influenced by the FS-KNN uncertainty which brings worse InC/InW performance, as shown in Fig. 5.16. In other words, λ_{knns} controls the trade-off between InC/OOD and InC/InW performance. Overall, $\beta = 1$ achieves the best UOSR performance which is a balanced result of InC/OOD and InC/InW

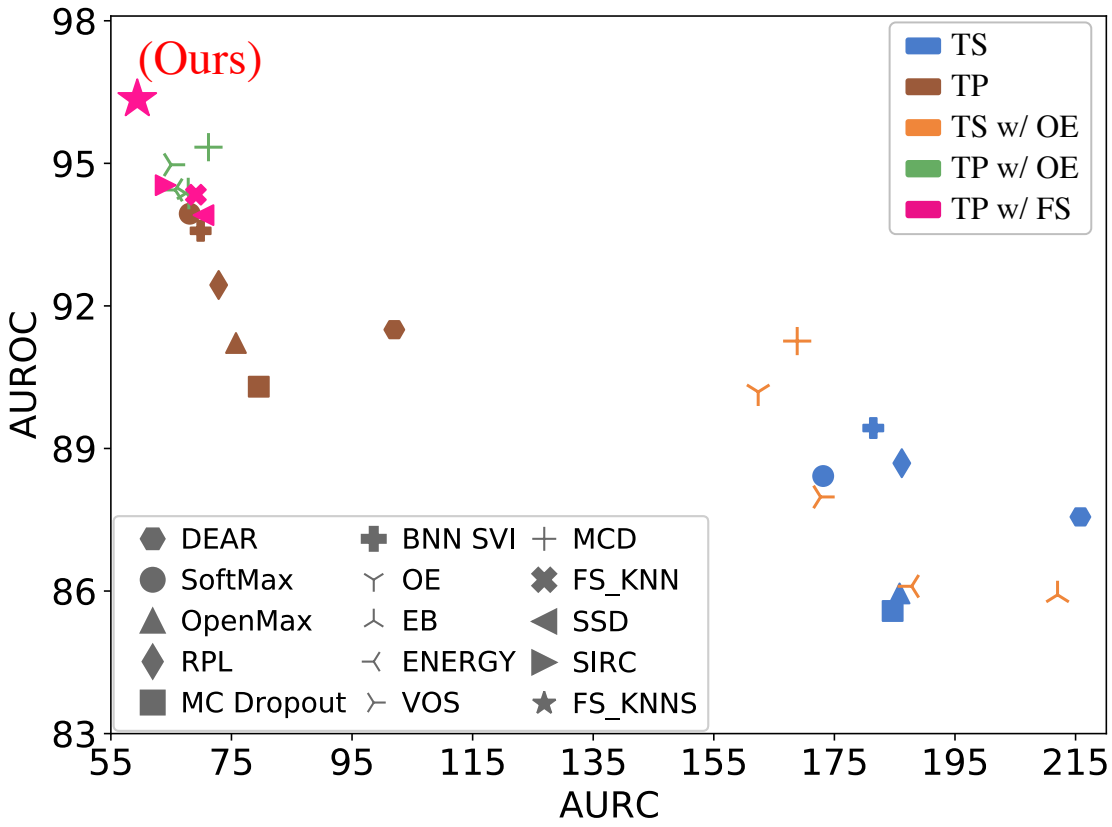


Figure 5.13: UOSR performance under all settings of TSM backbone in the video domain. OOD dataset is HMDB51.

trade-off. Larger α means the weight of FS-KNN uncertainty grows quickly when FS-KNN uncertainty is closed to λ_{knns}. We find that a smaller α makes the performance insensitive of λ_{knns}, and vice versa. So finally we pick α and β as 50 and 1 as the optimal hyper parameters.

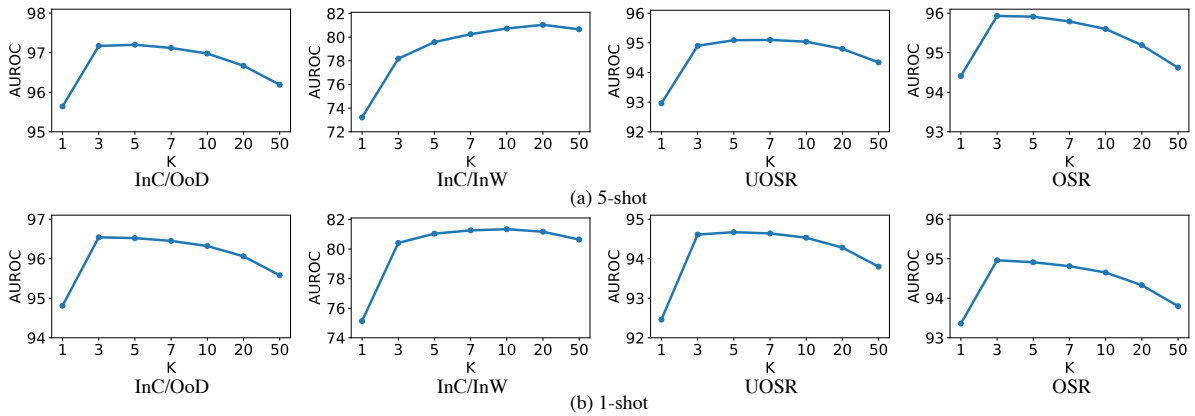


Figure 5.14: Ablation study of K used in FS-KNN. The backbone is ResNet50.

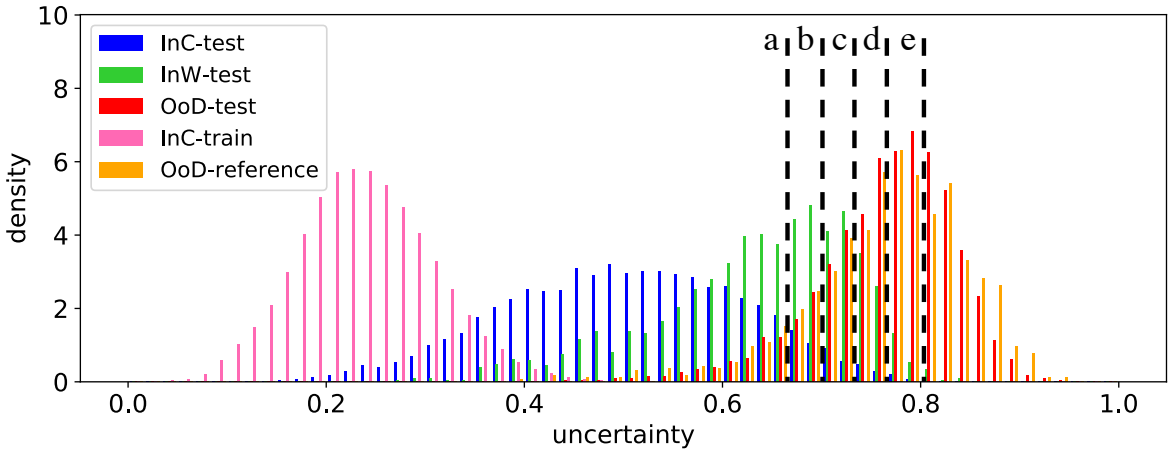


Figure 5.15: Uncertainty distribution of $\hat{u}_{fs-knns}$. InC-train samples have distinct uncertainty distribution with InC-test samples, but OOD reference samples share similar uncertainty distribution with OOD test samples. a to e correspond to the λ_{knns} when $\beta = 1.5, 1, 0.5, 0, -0.5$ in Eq. 5.6.

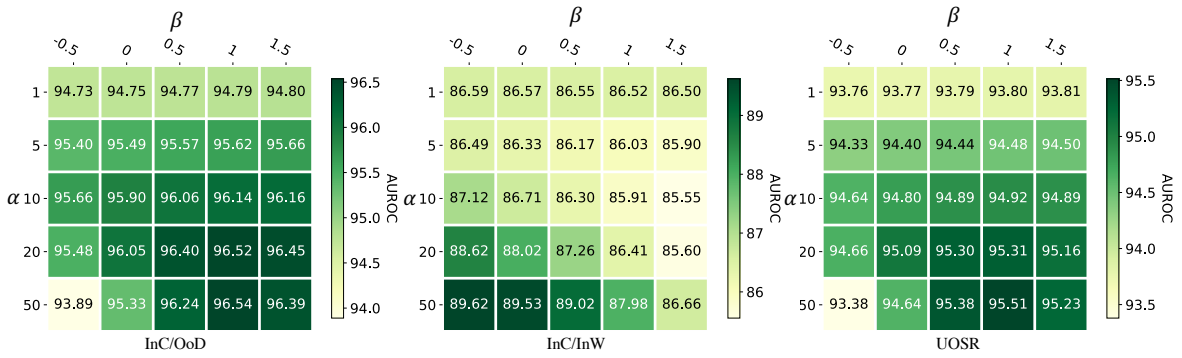


Figure 5.16: Ablation study of β and α in Eq. 5.5 and Eq. 5.6.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this chapter, we make a conclusion of the whole thesis and provide several potential directions of future work.

6.1 Conclusion

In Chapter 3, we delve into the task of open-set 3D semantic segmentation. Initially, we define the open-world recognition problem within LIDAR semantic segmentation. This realm encompasses open-set segmentation for identifying OOD objects and incremental learning to gradually learn OOD classes without compromising the knowledge of ID classes. We propose a redundancy classifier framework tailored for both open-set recognition and incremental learning tasks. Unknown Object Synthesis, Predictive Distribution Calibration, and Pseudo Label Generation are designed for OOD detection and preserving the old knowledge.

In Chapter 4, we commence by analyzing the open-set recognition problem through the lens of the Information Bottleneck theory. We find that the performance in open-set scenarios is closely tied to the availability of class-specific and instance-specific information. To enhance these aspects and thereby improve both closed-set and open-set performance, we introduce the prototypical similarity learning framework. We encourage the features of samples in the same class to be not exactly same to keep the intro-class variance which helps keep the instance-specific information. Besides, the shuffled samples are pushed away by the original samples to learn temporal information which is an important part of class-specific information.

In Chapter 5, we conduct an in-depth analysis of a novel paradigm termed unified open-set recognition. Unlike traditional open-set recognition, which solely focuses on detecting OOD samples, unified open-set recognition aims to identify both OOD samples and ID samples that are misclassified. This is due to the incorrect prediction results for these samples. We discover that OOD samples and misclassified ID samples share similar uncertainty distributions. Additionally, we investigate the impact of outlier exposure and pre-training in the unified open-set recognition setting. Finally, we propose a method FS-KNNS under the few-shot learning paradigm that fully leverages the provided OOD templates during inference.

6.2 Future Work

Open-set recognition in pure computer vision modality reaches saturated performance. One future work can be the exploration of open-set recognition in the multi-modality models like CLIP [164]. The introducing of language branch brings more opportunities for such multi-modality open-set recognition problem. The CLIP model has strong zero-shot performance which makes the definition of open-set recognition problem might be changed in such context, since it is hard to define what is OOD for the CLIP model. Another future direction is the hallucination problem of large language models (LLM) and multi-modality large language models (MLLM). The hallucination problem means LLM gives wrong answer for the user query. We show that the ID but wrongly-classified samples actually behaves similar to the OOD samples in Chapter 5. Therefore, how to detect the cases that LLM is making mistakes is similar to OOD detection. Determining when a large language model is likely to make mistakes can help it self-correct or output the correct answer under human guidance.

REFERENCES

[1] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *ICLR*, 2017 (cit. on pp. 1, 4, 9, 10, 19, 24, 33, 34, 49–53).

[2] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu, "Deep metric learning for open world semantic segmentation," in *ICCV*, 2021 (cit. on pp. 1, 17).

[3] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song, "Scaling out-of-distribution detection for real-world settings," *arXiv preprint arXiv:1911.11132*, 2019 (cit. on pp. 4, 18, 19).

[4] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *NeurIPS*, 2020 (cit. on p. 4).

[5] P. Morteza and Y. Li, "Provable guarantees for understanding out-of-distribution detection," in *AAAI*, 2022 (cit. on p. 4).

[6] Y. Sun, C. Guo, and Y. Li, "React: Out-of-distribution detection with rectified activations," in *NeurIPS*, 2021 (cit. on p. 4).

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015 (cit. on p. 4).

[8] Y. Sun and Y. Li, "Dice: Leveraging sparsification for out-of-distribution detection," in *ECCV*, 2022 (cit. on p. 4).

[9] A. Djurisic, N. Bozanic, A. Ashok, and R. Liu, "Extremely simple activation shaping for out-of-distribution detection," *arXiv preprint arXiv:2209.09858*, 2022 (cit. on p. 4).

[10] Y. Wang, B. Li, T. Che, K. Zhou, D. Li, and Z. Liu, "Energy-based open-world uncertainty modeling for confidence calibration," in *ICCV*, 2021 (cit. on p. 5).

[11] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke, "Out-of-distribution detection using an ensemble of self supervised leave-out classifiers," in *ECCV*, 2018 (cit. on p. 5).

[12] J. Bitterwolf, A. Meinke, and M. Hein, "Certifiably adversarially robust detection of out-of-distribution data," in *NeurIPS*, 2020 (cit. on p. 5).

[13] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, "Robust out-of-distribution detection for neural networks," *arXiv preprint arXiv:2003.09711*, 2020 (cit. on p. 5).

[14] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," in *NeurIPS*, 2019 (cit. on p. 5).

[15] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, 2019 (cit. on p. 5).

[16] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017 (cit. on p. 5).

[17] X. Dong, J. Guo, A. Li, W.-T. Ting, C. Liu, and H. Kung, "Neural mean discrepancy for efficient out-of-distribution detection," in *CVPR*, 2022 (cit. on p. 5).

[18] Z. Lin, S. D. Roy, and Y. Li, "Mood: Multi-level out-of-distribution detection," in *CVPR*, 2021 (cit. on p. 5).

[19] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018 (cit. on p. 5).

[20] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *CVPR*, 2020 (cit. on p. 5).

[21] E. T. Jaynes, "Bayesian methods: General background," *Maximum entropy and Bayesian methods in applied statistics*, pp. 1–25, 1985 (cit. on p. 5).

[22] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118 (cit. on p. 5).

[23] D. Gamerman and H. F. Lopes, *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman and Hall/CRC, 2006 (cit. on p. 5).

[24] A. Y. Foong, Y. Li, J. M. Hernández-Lobato, and R. E. Turner, "'in-between'uncertainty in bayesian neural networks," *arXiv preprint arXiv:1906.11537*, 2019 (cit. on p. 5).

[25] C. Peterson and E. Hartman, "Explorations of the mean field theory learning algorithm," *Neural Networks*, 1989 (cit. on p. 5).

[26] F. Wenzel, K. Roth, B. S. Veeling, J. Świątkowski, L. Tran, S. Mandt, J. Snoek, T. Salimans, R. Jenatton, and S. Nowozin, "How good is the bayes posterior in deep neural networks really?" *arXiv preprint arXiv:2002.02405*, 2020 (cit. on p. 5).

[27] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *ICML*, 2016 (cit. on pp. 5, 9, 10, 19, 24, 34).

[28] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *NeurIPS*, 2017 (cit. on p. 5).

[29] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in *NeurIPS*, 2018 (cit. on pp. 5, 32).

[30] J. Ren, S. Fort, J. Liu, A. G. Roy, S. Padhy, and B. Lakshminarayanan, “A simple fix to mahalanobis distance for improving near-ood detection,” *arXiv preprint arXiv:2106.09022*, 2021 (cit. on p. 6).

[31] Y. Sun, Y. Ming, X. Zhu, and Y. Li, “Out-of-distribution detection with deep nearest neighbors,” in *ICML*, 2022 (cit. on pp. 6, 49, 69, 72).

[32] E. Techapanurak, M. Suganuma, and T. Okatani, “Hyperparameter-free out-of-distribution detection using cosine similarity,” in *ACCV*, 2020 (cit. on p. 6).

[33] X. Chen, X. Lan, F. Sun, and N. Zheng, “A boundary based out-of-distribution classifier for generalized zero-shot learning,” in *ECCV*, 2020 (cit. on p. 6).

[34] A. Zaeemzadeh, N. Bisagno, Z. Sambugaro, N. Conci, N. Rahnavard, and M. Shah, “Out-of-distribution detection using union of 1-dimensional subspaces,” in *CVPR*, 2021 (cit. on p. 6).

[35] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty estimation using a single deep deterministic neural network,” in *ICML*, 2020 (cit. on p. 6).

[36] H. Huang, Z. Li, L. Wang, S. Chen, B. Dong, and X. Zhou, “Feature space singularity for out-of-distribution detection,” *arXiv preprint arXiv:2011.14654*, 2020 (cit. on p. 6).

[37] E. D. C. Gomes, F. Alberge, P. Duhamel, and P. Piantanida, “Igeood: An information geometry approach to out-of-distribution detection,” *arXiv preprint arXiv:2203.07798*, 2022 (cit. on p. 6).

[38] H. Wang, Z. Li, L. Feng, and W. Zhang, “Vim: Out-of-distribution with virtual-logit matching,” in *CVPR*, 2022 (cit. on p. 6).

[39] Y. Ming, Y. Sun, O. Dia, and Y. Li, “How to exploit hyperspherical embeddings for out-of-distribution detection?” *arXiv preprint arXiv:2203.04450*, 2022 (cit. on p. 6).

[40] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance," *arXiv preprint arXiv:1812.02765*, 2018 (cit. on p. 6).

[41] Y. Zhou, "Rethinking reconstruction autoencoder-based out-of-distribution detection," in *CVPR*, 2022 (cit. on p. 6).

[42] Y. Yang, R. Gao, and Q. Xu, "Out-of-distribution detection with semantic mismatch under masking," in *ECCV*, 2022 (cit. on p. 6).

[43] W. Jiang, Y. Ge, H. Cheng, M. Chen, S. Feng, and C. Wang, "Read: Aggregating reconstruction error into out-of-distribution detection," in *AAAI*, 2023 (cit. on p. 6).

[44] J. Li, P. Chen, Z. He, S. Yu, S. Liu, and J. Jia, "Rethinking out-of-distribution (ood) detection: Masked image modeling is all you need," in *CVPR*, 2023 (cit. on p. 6).

[45] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *ICLR*, 2019 (cit. on pp. 7, 33, 48, 61–63).

[46] A. R. Dhamija, M. Günther, and T. Boult, "Reducing network agnostophobia," in *NeurIPS*, 2018 (cit. on p. 7).

[47] Q. Yu and K. Aizawa, "Unsupervised out-of-distribution detection by maximum classifier discrepancy," in *ICCV*, 2019 (cit. on p. 7).

[48] S. Thulasidasan, S. Thapa, S. Dhaubhadel, G. Chennupati, T. Bhattacharya, and J. Bilmes, "An effective baseline for robustness to distributional shift," in *ICMLA*, 2021 (cit. on p. 7).

[49] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, "Atom: Robustifying out-of-distribution detection using outlier mining," in *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*, 2021 (cit. on p. 7).

[50] Y. Ming, Y. Fan, and Y. Li, "Poem: Out-of-distribution detection with posterior sampling," in *ICML*, 2022 (cit. on p. 7).

[51] Y. Li and N. Vasconcelos, "Background data resampling for outlier-aware classification," in *CVPR*, 2020 (cit. on p. 7).

[52] J. Zhang, N. Inkawhich, R. Linderman, Y. Chen, and H. Li, "Mixture outlier exposure: Towards out-of-distribution detection in fine-grained environments," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023 (cit. on p. 7).

[53] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, "Self-supervised learning for generalizable out-of-distribution detection," in *AAAI*, 2020 (cit. on p. 7).

[54] J. Yang, H. Wang, L. Feng, X. Yan, H. Zheng, W. Zhang, and Z. Liu, "Semantically coherent out-of-distribution detection," in *ICCV*, 2021 (cit. on p. 7).

[55] F. Lu, K. Zhu, W. Zhai, K. Zheng, and Y. Cao, "Uncertainty-aware optimal transport for semantically coherent out-of-distribution detection," in *CVPR*, 2023 (cit. on p. 7).

[56] A. Shafaei, M. Schmidt, and J. J. Little, "A less biased evaluation of out-of-distribution sample detectors," *arXiv preprint arXiv:1809.04729*, 2018 (cit. on p. 7).

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, 2020 (cit. on p. 7).

[58] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," *arXiv preprint arXiv:1711.09325*, 2017 (cit. on p. 7).

[59] T. Jeong and H. Kim, "Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification," in *NeurIPS*, 2020 (cit. on p. 7).

[60] X. Du, Z. Wang, M. Cai, and Y. Li, "Vos: Learning what you don't know by virtual outlier synthesis," *arXiv preprint arXiv:2202.01197*, 2022 (cit. on p. 7).

[61] L. Tao, X. Du, X. Zhu, and Y. Li, "Non-parametric outlier synthesis," *arXiv preprint arXiv:2303.02966*, 2023 (cit. on p. 8).

[62] Q. Wang, J. Ye, F. Liu, Q. Dai, M. Kalander, T. Liu, J. Hao, and B. Han, "Out-of-distribution detection with implicit outlier transformation," *arXiv preprint arXiv:2303.05033*, 2023 (cit. on p. 8).

[63] H. Zheng, Q. Wang, Z. Fang, X. Xia, F. Liu, T. Liu, and B. Han, "Out-of-distribution detection learning with unreliable out-of-distribution sources," in *NeurIPS*, 2023 (cit. on p. 8).

[64] X. Du, X. Wang, G. Gozum, and Y. Li, "Unknown-aware object detection: Learning what you don't know from videos in the wild," in *CVPR*, 2022 (cit. on p. 8).

[65] R. Chan, M. Rottmann, and H. Gottschalk, "Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation," in *ICCV*, 2021 (cit. on p. 8).

[66] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014 (cit. on p. 8).

[67] Y. Tian, Y. Liu, G. Pang, F. Liu, Y. Chen, and G. Carneiro, "Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes," in *ECCV*, 2022 (cit. on p. 8).

[68] M. Grcić, P. Bevandić, and S. Šegvić, "Densehybrid: Hybrid anomaly detection for dense open-set recognition," in *ECCV*, 2022 (cit. on p. 8).

[69] S. N. Rai, F. Cermelli, D. Fontanel, C. Masone, and B. Caputo, "Unmasking anomalies in road-scene segmentation," in *ICCV*, 2023 (cit. on p. 8).

[70] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *CVPR*, 2022 (cit. on p. 8).

[71] K. Lis, K. Nakka, P. Fua, and M. Salzmann, "Detecting the unexpected via image resynthesis," in *ICCV*, 2019 (cit. on p. 8).

[72] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena, "Pixel-wise anomaly detection in complex driving scenes," in *CVPR*, 2021 (cit. on p. 8).

[73] T. Vojir, T. Šipka, R. Aljundi, N. Chumerin, D. O. Reino, and J. Matas, "Road anomaly detection by partial image reconstruction with segmentation coupling," in *ICCV*, 2021 (cit. on p. 8).

[74] Y. Xia, Y. Zhang, F. Liu, W. Shen, and A. L. Yuille, "Synthesize then compare: Detecting failures and anomalies for semantic segmentation," in *ECCV*, 2020 (cit. on pp. 8, 10).

[75] J. Cen, P. Yun, S. Zhang, J. Cai, D. Luan, M. Tang, M. Liu, and M. Yu Wang, "Open-world semantic segmentation for lidar point clouds," in *ECCV*, 2022 (cit. on p. 9).

[76] J. Li and Q. Dong, "Open-set semantic segmentation for point clouds via adversarial prototype framework," in *CVPR*, 2023 (cit. on p. 9).

[77] W. Bao, Q. Yu, and Y. Kong, "Evidential deep learning for open set action recognition," in *ICCV*, 2021 (cit. on pp. 9, 24, 26, 33, 34, 45, 51–53, 63).

[78] A. Bendale and T. E. Boult, "Towards open set deep networks," in *CVPR* (cit. on pp. 9, 24, 34).

[79] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian, "Learning open set network with discriminative reciprocal points," in *ECCV*, 2020 (cit. on pp. 9, 34).

[80] R. Krishnan, M. Subedar, and O. Tickoo, “BAR: Bayesian activity recognition using variational inference,” in *NeurIPS Workshops*, 2018 (cit. on pp. 9, 34).

[81] A. Amini, W. Schwarting, A. Soleimany, and D. Rus, “Deep evidential regression,” in *NeurIPS*, 2020 (cit. on pp. 9, 24).

[82] J. Cen, S. Zhang, X. Wang, Y. Pei, Z. Qing, Y. Zhang, and Q. Chen, “Enlarging instance-specific and class-specific information for open-set action recognition,” in *CVPR*, 2023 (cit. on p. 9).

[83] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, “Cylindrical and asymmetrical 3d convolution networks for lidar segmentation,” in *CVPR*, 2021 (cit. on pp. 10, 18).

[84] R. Cheng, R. Razani, E. Taghavi, E. Li, and B. Liu, “2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network,” in *CVPR*, 2021 (cit. on p. 10).

[85] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, “Polarnet: An improved grid representation for online lidar point clouds semantic segmentation,” in *CVPR*, 2020 (cit. on p. 10).

[86] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, “Rangenet++: Fast and accurate lidar semantic segmentation,” in *IROS*, 2019 (cit. on p. 10).

[87] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences,” in *ICCV*, 2019 (cit. on p. 10).

[88] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, “Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset,” *The International Journal on Robotics Research*, 2021 (cit. on p. 10).

[89] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite,” in *CVPR*, 2012 (cit. on p. 10).

[90] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “Nuscenes: A multimodal dataset for autonomous driving,” in *CVPR*, 2020 (cit. on p. 10).

[91] D. Bozhinoski, D. Di Ruscio, I. Malavolta, P. Pelliccione, and I. Crnkovic, "Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective," *Journal of Systems and Software*, 2019 (cit. on p. 10).

[92] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain mr images," in *International MICCAI Brainlesion Workshop*, 2018 (cit. on p. 10).

[93] K. Lis, K. Nakka, P. Fua, and M. Salzmann, "Detecting the unexpected via image resynthesis," in *ICCV*, 2019 (cit. on p. 10).

[94] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NeurIPS*, 2017 (cit. on p. 10).

[95] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019 (cit. on p. 10).

[96] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, 1989 (cit. on p. 11).

[97] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021 (cit. on p. 12).

[98] P. Yun, J. Cen, and M. Liu, "Conflicts between likelihood and knowledge distillation in task incremental learning for 3d object detection," in *3DV*, 2021 (cit. on p. 12).

[99] F. Cermelli, M. Mancini, S. R. Bulo, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *CVPR*, 2020 (cit. on pp. 12, 17).

[100] Y. Wang, B. Li, T. Che, K. Zhou, Z. Liu, and D. Li, "Energy-based open-world uncertainty modeling for confidence calibration," in *ICCV*, 2021 (cit. on p. 15).

[101] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018 (cit. on p. 21).

[102] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *ICCV*, 2019 (cit. on pp. 24, 33, 34).

[103] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *ICCV*, 2019 (cit. on pp. 24, 33, 34).

[104] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *CVPR*, 2017 (cit. on pp. 24, 33, 34, 53, 62).

[105] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, "Temporal pyramid network for action recognition," in *CVPR*, 2020 (cit. on p. 24).

[106] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" In *NeurIPS*, 2017 (cit. on p. 24).

[107] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop*, 2015 (cit. on pp. 24, 29).

[108] H. Wang, X. Guo, Z.-H. Deng, and Y. Lu, "Rethinking minimal sufficient representation in contrastive learning," in *CVPR*, 2022 (cit. on pp. 24, 27, 29, 30).

[109] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *CVPR*, 2018 (cit. on p. 26).

[110] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000 (cit. on pp. 27, 29).

[111] T. M. Cover, *Elements of information theory*. 1999 (cit. on p. 29).

[112] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *The Journal of Machine Learning Research*, 2018 (cit. on p. 29).

[113] M. Federici, A. Dutta, P. Forré, N. Kushman, and Z. Akata, "Learning robust representations via multi-view information bottleneck," in *ICLR*, 2021 (cit. on p. 29).

[114] Q. Shi, H.-B. Zhang, Z. Li, J.-X. Du, Q. Lei, and J.-H. Liu, "Shuffle-invariant network for action recognition in videos," *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2022 (cit. on p. 32).

[115] S. Jenni, G. Meishvili, and P. Favaro, "Video representation learning by recognizing temporal transformations," in *ECCV*, 2020 (cit. on p. 32).

[116] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *CVPR*, 2017 (cit. on p. 32).

[117] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang, "Unsupervised representation learning by sorting sequences," in *ICCV*, 2017 (cit. on p. 32).

[118] V. Sehwag, M. Chiang, and P. Mittal, "SSD: A unified framework for self-supervised outlier detection," in *ICLR*, 2021 (cit. on pp. 32, 49, 68).

[119] D. L. Yue Zhao Yuanjun Xiong, *Mmaction*, https://github.com/open-mmlab/mmaction, 2019 (cit. on pp. 33, 53).

[120] Y. You, I. Gitman, and B. Ginsburg, "Large batch training of convolutional networks," *arXiv preprint arXiv:1708.03888*, 2017 (cit. on p. 33).

[121] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *CVPR*, 2015 (cit. on p. 39).

[122] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "NormFace: L2 hypersphere embedding for face verification," in *ACM MM*, 2017 (cit. on p. 39).

[123] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle Loss: A unified perspective of pair similarity optimization," in *CVPR*, 2020 (cit. on p. 39).

[124] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *CVPR*, 2019 (cit. on p. 39).

[125] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *ICML*, 2016 (cit. on p. 39).

[126] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, "Open-set recognition: A good closed-set classifier is all you need," in *ICLR*, 2022 (cit. on pp. 39, 59–61).

[127] X. Han, V. Papyan, and D. L. Donoho, "Neural collapse under mse loss: Proximity to and dynamics on the central path," in *ICLR*, 2022 (cit. on p. 39).

[128] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009 (cit. on pp. 48, 62).

[129] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013 (cit. on pp. 48, 51).

[130] J. Kim, J. Koo, and S. Hwang, "A unified benchmark for the unknown detection capability of deep neural networks," *arXiv preprint arXiv:2112.00337*, 2021 (cit. on pp. 48, 50, 64).

[131] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *ICML*, 2019 (cit. on pp. 48, 61).

[132] Q. Yu and K. Aizawa, "Unsupervised out-of-distribution detection by maximum classifier discrepancy," in *ICCV*, 2019 (cit. on pp. 48, 63).

[133] S. Thulasidasan, S. Thapa, S. Dhaubhadel, G. Chennupati, T. Bhattacharya, and J. Bilmes, "An effective baseline for robustness to distributional shift," in *ICMLA*, 2021 (cit. on pp. 48, 63).

[134] G. Xia and C.-S. Bouganis, "Augmenting softmax information for selective classification with out-of-distribution data," *arXiv preprint arXiv:2207.07506*, 2022 (cit. on pp. 50, 64, 70, 73).

[135] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *NeurIPS*, 2019 (cit. on pp. 51, 64).

[136] J. Moon, J. Kim, Y. Shin, and S. Hwang, "Confidence-aware learning for deep neural networks," in *ICML*, 2020 (cit. on pp. 51, 64).

[137] F. Granese, M. Romanelli, D. Gorla, C. Palamidessi, and P. Piantanida, "Doctor: A simple method for detecting misclassification errors," in *NeurIPS*, 2021 (cit. on pp. 51, 64).

[138] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in *CVPR*, 2022 (cit. on p. 51).

[139] M. Z. Zaheer, A. Mahmood, M. H. Khan, M. Segu, F. Yu, and S.-I. Lee, "Generative cooperative learning for unsupervised video anomaly detection," in *CVPR*, 2022 (cit. on p. 51).

[140] K. Chauhan, P. Shenoy, M. Gupta, D. Sridharan, *et al.*, "Robust outlier detection by de-biasing vae likelihoods," in *CVPR*, 2022 (cit. on p. 51).

[141] A. Goodge, B. Hooi, S.-K. Ng, and W. S. Ng, "Lunar: Unifying local outlier detection methods via graph neural networks," in *AAAI*, 2022 (cit. on p. 51).

[142] Y. Geifman, G. Uziel, and R. El-Yaniv, "Bias-reduced uncertainty estimation for deep neural classifiers," in *ICLR*, 2019 (cit. on p. 51).

[143] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017 (cit. on pp. 51, 52, 60).

[144] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *ICLR*, 2018 (cit. on pp. 52, 53).

[145] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018 (cit. on p. 52).

[146] A. Bendale and T. E. Boult, "Towards open set deep networks," in *CVPR*, 2016 (cit. on p. 52).

[147] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *CVPR*, 2019 (cit. on p. 52).

[148] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Learning placeholders for open-set recognition," in *CVPR*, 2021 (cit. on p. 52).

[149] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian, "Learning open set network with discriminative reciprocal points," in *ECCV*, 2020 (cit. on p. 52).

[150] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016 (cit. on p. 52).

[151] R. Krishnan, M. Subedar, and O. Tickoo, "Bar: Bayesian activity recognition using variational inference," in *NeurIPS*, 2018 (cit. on p. 52).

[152] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009 (cit. on p. 52).

[153] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, 2015 (cit. on p. 52).

[154] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015 (cit. on p. 52).

[155] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *ECCV*, 2020 (cit. on pp. 53, 62, 63).

[156] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012 (cit. on p. 53).

[157] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in *ICCV*, 2011 (cit. on p. 53).

[158] M. Monfort, B. Pan, K. Ramakrishnan, A. Andonian, B. A. McNamara, A. Lascelles, Q. Fan, D. Gutfreund, R. Feris, and A. Oliva, "Multi-moments in time: Learning and interpreting models for multi-action video understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021 (cit. on p. 53).

[159] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015 (cit. on p. 53).

[160] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016 (cit. on p. 53).

[161] J. Lin, C. Gan, and S. Han, “Tsm: Temporal shift module for efficient video understanding,” in *ICCV*, 2019 (cit. on p. 53).

[162] X. Du, Z. Wang, M. Cai, and Y. Li, “Vos: Learning what you don’t know by virtual outlier synthesis,” in *ICLR*, 2022 (cit. on p. 63).

[163] Z.-F. Wu, T. Wei, J. Jiang, C. Mao, M. Tang, and Y.-F. Li, “Ngc: A unified framework for learning with open-world noisy data,” in *ICCV*, 2021 (cit. on p. 67).

[164] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *ICML*, 2021 (cit. on p. 77).

APPENDIX A

LIST OF PUBLICATIONS

Journal Publications

[1] J. Cen, S. Zhang, Y. Pei, K. Li, H. Zheng, M. Luo, Y. Zhang, and Q. Chen, "Cmdfusion: Bidirectional fusion network with cross-modality knowledge distillation for lidar semantic segmentation," *IEEE Robotics and Automation Letters*, 2023.

[2] J. Cai, J. Cen, H. Wang, and M. Y. Wang, "Real-time collision-free grasp pose detection with geometry-aware refinement using high-resolution volume," *IEEE Robotics and Automation Letters*, 2022.

Conference Publications

[1] J. Cen, D. Luan, S. Zhang, Y. Pei, Y. Zhang, D. Zhao, S. Shen, and Q. Chen, "The devil is in the wrongly-classified samples: Towards unified open-set recognition," in *ICLR*, 2023.

[2] J. Cen, S. Zhang, X. Wang, Y. Pei, Z. Qing, Y. Zhang, and Q. Chen, "Enlarging instance-specific and class-specific information for open-set action recognition," in *CVPR*, 2023.

[3] J. Cen, Y. Wu, K. Wang, X. Li, J. Yang, Y. Pei, L. Kong, Z. Liu, and Q. Chen, "Sad: Segment any rgbd," in *NeurIPS workshop*, 2023.

[4] J. Yang, J. Cen, W. Peng, S. Liu, F. Hong, X. Li, K. Zhou, Q. Chen, and Z. Liu, "4d panoptic scene graph generation," in *NeurIPS*, 2023.

[5] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu, "Segment any point cloud sequences by distilling vision foundation models," in *NeurIPS*, 2023.

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

[7] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu, "Deep metric learning for open world semantic segmentation," in *ICCV*, 2021.

[8] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu, “Open-set 3d object detection,” in *3DV*, 2021.

[9] P. Yun, J. Cen, and M. Liu, “Conflicts between likelihood and knowledge distillation in task incremental learning for 3d object detection,” in *3DV*, 2021.