Cognitive Robotics 4 (2024) 191–203

Contents lists available at ScienceDirect

Cognitive Robotics

journal homepage: http://www.keaipublishing.com/en/journals/cognitive-robotics/

Research Paper

Autonomous novel class discovery for vision-based recognition in non-interactive environments

Xuelin Zhang [a,*], Feng Liu [b], Xuelian Cheng [a], Siyuan Yan [a], Zhibin Liao [c], Zongyuan Ge [a]

[a] Faculty of Information Technology, Monash University, 20 Exhibition Walk, Melbourne 3168, VIC, Australia
[b] The School of Computing and Information Systems, The University of Melbourne, Parkville, Melbourne 3052, VIC, Australia
[c] Faculty of Sciences, Engineering and Technology, The University of Adelaide, North Terrace, Adelaide, 5001, SA, Australia

ARTICLE INFO

Keywords:
Open set
Image recognition
Image clustering
Deep learning
Deep neural networks

ABSTRACT

Visual recognition with deep learning has recently been shown to be effective in robotic vision. However, these algorithms tend to be build under fixed and structured environment, which is rarely the case in real life. When facing unknown objects, avoidance or human interactions are required, which may miss critical objects or be prohibitively costly to obtain on robots in the real world. We consider a practical problem setting that aims to allow robots to automatically discover novel classes with only labelled known class samples in hand, defined as open-set clustering (OSC). To address the OSC problem, we propose a framework combining three approaches: 1) using selfsupervised vision transformers to mitigate the discard of information needed for clustering unknown classes; 2) adaptive weighting for image patches to prioritize patches with richer textures; and 3) incorporating a temperature scaling strategy to generate more separable feature embeddings for clustering. We demonstrate the efficacy of our approach in six fine-grained image datasets.

1. Introduction

In recent years, the integration of deep learning techniques with robotic vision has revolutionized the capabilities of autonomous systems, enabling robots to perceive and understand their surroundings better. Central to this paradigm shift is the application of deep learning models for object recognition, which empowers robots to identify and localize objects of interest in real-time. Deep learning-based object recognition systems leverage largescale datasets and powerful neural network architectures to extract high-level features from visual data, enabling robots to perform a wide range of tasks.

Despite the tremendous success of deep learning object recognition in controlled environments, deploying these systems in open-world scenarios poses significant challenges. In open-world environments, robots encounter dynamic and unstructured surroundings where the types and configurations of objects may vary widely. Traditional deep learning approaches, trained on static datasets with predefined classes, often struggle to generalize to novel objects and adapt to changing conditions. Moreover, factors such as variations in lighting, object occlusions, and clutter further exacerbate the robustness and reliability of object recognition systems in open-world settings.

To address the inherent risks of this open world challenge, numerous techniques have been introduced to empower models to discern and differentiate between unknown and known classes. Notable among these are open-set recognition (OSR) [1; 9; 33]

* Corresponding author.
E-mail address: xuelin.zhang@monash.edu (X. Zhang).

https://doi.org/10.1016/j.cogr.2024.10.002
Received 5 May 2024; Received in revised form 18 October 2024; Accepted 24 October 2024
Available online 16 November 2024

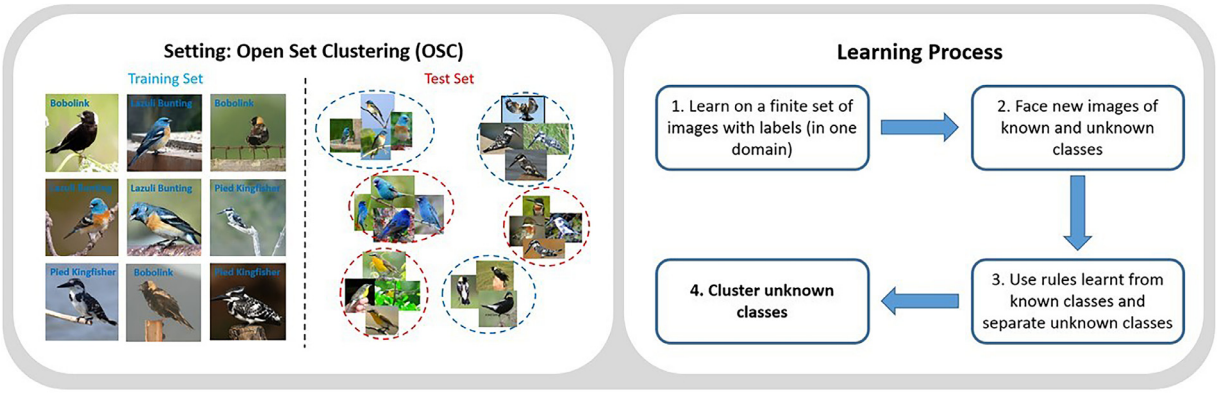

Fig. 1. The OSC problem is exemplified on the left-hand side. In this setting, only labeled samples of known classes are available in the training set. During inference, a model needs to perform clustering on samples coming from both known and unknown classes. The right-hand side flow chart illustrates the learning process of humans. Our setting mimics steps 1 to 4 of the process.

Table 1
Differences between OSC and other settings.

NCD/GNCD aims to detect and uncover novel classes. In the standard			
Task	No Training on Unknown Classes	Known Class Classification	Unknown Class Clustering
OSR	√	√	✗
NCD/GNCD	✗	√	√
OSC	√	√	√

and novel class discovery (NCD) [13]. When samples from unknown classes are detected, subsequent human intervention might be necessary to either eliminate these samples from the unknown classes (in cases where a closed-set classification setup is desired) or to assign labels to the previously unknown samples, thus forming new classes. Despite these efforts, the aforementioned techniques exhibit notable limitations. **OSR** aims to classify samples from known classes while also identifying samples from unknown classes. However, OSR treats all unknown classes as a single category, akin to anomaly detection. In the realm of robotic vision, encountering unknown objects often presents two options: avoidance or the need for additional human intervention through alerts or relearning. However, neither option is optimal. Avoidance may not always be feasible in certain scenarios, while relearning necessitates manual labeling or instructions, introducing a time gap in the process.

NCD/GNCD setup, the model is expected to be trained or finetuned on both known and unknown classes whenever novel classes emerge. In the field of robotic vision, acquiring samples from unknown classes or comprehending their distributions is typically impractical. Retraining models each time they encounter new objects is expensive and can potentially impact the performance of subsequent tasks.

To overcome these prior limitations, we embrace a more formidable challenge labeled as *Open Set Clustering (OSC)*, aiming at *generalized novel class discovery without the need to train on novel classes*. The setting is shown in Fig. 1. This novel setting differentiates OSC from earlier contexts, as highlighted in Table 1. OSC aspires to cluster all classes, thus acquiring aggregating the unknown classes for downstream tasks—a facet that often eludes Open Set Recognition (OSR).

Furthermore, OSC sets itself apart from NCD/GNCD by exclusively utilizing samples from known classes throughout training. This design choice obviates the requirement for model retraining upon the emergence of new classes. Distinct from prior methodologies, OSC unfurls the potential for real-time, on-demand clustering. This empowers OSC with the ability to proactively perform clustering. In doing so, OSC streamlines the path to newfound class revelation for robots and diminishes the intricacies of subsequent human instructions.

To address the challenge of OSC, it is imperative to discern the conditions under which it can be effectively tackled. In the context of NCD, Chi et al. [6] establish that high-level semantic attributes imply inherent connections between known and unknown classes. This fundamental insight is transferable to the realm of OSC, given its shared objective. Capitalizing on this premise, we narrow our focus to fine-grained datasets for OSC. Building upon this foundation, we present a OSC framework characterized by three pivotal concepts: (1) We employ an approach involving the joint training of a vision transformer, encompassing both supervised and self-supervised tasks. This strategy harnesses the limited information inherent in labeled known classes while concurrently mitigating the loss of vital data. (2) Recognizing the pivotal role of image patches rich in textures for effective categorization, we introduce adaptive weighting per image patch. (3) Temperature scaling is harnessed to temper the model's overconfidence, yielding more discernible clusters. Furthermore, our empirical investigation delves into the overlap between the pretrained dataset (ImageNet [24]) and our test dataset. By manipulating the dataset split, we elucidate the negligible influence of class overlap on our model's performance and

underscore the absence of label leakage. Moreover, our findings show the pronounced impact of the proximity between known and unknown classes on OSC performance.

This work contributes in the following ways:

- We formalize the problem setting of generalized novel class discovery without novel data as OSC, which is more practical for robotic vision. • We present a comprehensive OSC pipeline and assess its efficacy across six fine-grained classification datasets.
- Through rigorous experimentation, we demonstrate that a well-crafted pipeline can enable the OSC method to achieve comparable or even superior performance compared to GNCD methods on the same datasets. • We reveal key findings and hands-on experience of developing OSC models through comprehensive analysis, offering useful insights for future research.

2. Related work

Open set recognition. The OSR problem was proposed by Scheirer et al. [25]. OpenMax [1] tackles OSR by using an OpenMax layer and the Extreme Value Theory (EVT). G-OpenMax [9] was the first method to use the generative model GAN to train an OSR and shows the effectiveness of reconstruction loss in OSR. CROSR [33] utilizes the reconstruction of latent space features and discriminative learning. C2AE [23] uses class conditional auto-encoders and EVT to separate unknown classes from known classes. Recently, Chen et al. [4] and Chen et al. [3] use reciprocal points to distinguish unknown classes from known classes, and Zhang et al. [34] use a network architecture search to find an optimal architecture for OSR. Finally, Vaze et al. [30] shows the positive correlation between closed-set accuracy and OSR performance.

Novel category discovery. AutoNovel [11; 12; 14] established a pipeline for NCD. The model was first trained with self-supervised learning on both labeled and unlabeled datasets. The model was then trained on the labeled set to learn higher-level features. Finally, the model was jointly trained on both the labeled and unlabeled sets. Rank statistics was applied during the joint training to transfer the learned knowledge from known classes to unknown classes. The K-means method was modified to leverage the labels in the training set to further improve the clustering accuracy. Vaze et al. [29] formalized generalized novel category discovery (GNCD). Compared to NCD, GNCD removes the unknown class only assumption on the samples of the unlabeled set to include both known classes and unknown classes in the training set. However, GNCD still assumes that the unknown classes at inference time must be available during training. In this paper, we aim to tackle the problem of OSC, where only known classes are available at the training stage but a model will need to cluster on both known and unknown classes during the inference.

Transductive Zero-Shot Learning. Transductive Zero-Shot Learning (ZSL) has emerged as an important variation of zero-shot learning (ZSL) that leverages unlabeled data from unseen classes to improve performance. Traditional ZSL models (inductive ZSL) rely solely on seen class information during training and apply knowledge transfer techniques based on semantic information such as attributes or word embeddings to classify unseen classes. In contrast, transductive ZSL addresses the domain shift problem by exploiting the distribution of unseen test data to enhance model adaptation. One of the early studies that explored the benefits of transductive learning in ZSL was conducted by [8]. They proposed a transductive multi-view embedding approach that aligns the visual features of seen classes with the semantic descriptions of unseen classes while using unlabeled data from unseen classes to better generalize to novel categories. [32] extended the Generalized ZeroShot Learning (GZSL) framework with a transductive setting by incorporating generative models. It synthesized feature representations for unseen classes using both seen class data and the unlabeled test data from unseen classes, significantly improving performance in the ZSL and GZSL settings . [5] introduced a transductive approach that incorporated self-training strategies, using pseudo-labels generated from the unlabeled test data of unseen classes to iteratively refine the model's predictions. This transductive method helped reduce the domain shift between seen and unseen classes . [35] presented a novel graph-based method for transductive ZSL that constructed a graph to propagate information from seen to unseen classes by leveraging the unlabeled test data. The graph convolutional network (GCN) framework exploited both the relationships between classes and the visual feature distribution to improve zero-shot classification accuracy . [20] introduced a Transductive Bi-Directional Mapping framework that improved ZSL by mapping both semantic attributes and visual features between seen and unseen classes using unlabeled test data.

Even though transductive ZSL and OSC are both approaches for handling scenarios with unseen or unknown classes, but they differ in their assumptions, data, and objectives. Unlike transductive ZSL, OSC assumes no prior knowledge of the unseen classes and no access to any semantic information or attributes that describe them. It works solely based on the feature space and the inherent structure of the data to form clusters of novel classes. Furthermore, OSC does not have access to any unknown classes during training, which significantly relax the data limitation. Finally, transductie ZSL is primarily a classification task where the goal is to assign each instance to a known or unseen class while OSC is a class discovery task where the model aims to identify and group previously unseen or novel classes from the data, and doesn't necessarily assign labels to the newly discovered classes but rather creates clusters that correspond to these unknown categories.

3. Problem formulation

We first differentiate the OSC task from the existing settings. The only available knowledge for the model is a set of images with known labels. The task is to cluster arbitrary images that may or may not be seen in the training set. Compared to OSR which only aims at identifying unknown classes, OSC further requires assigning correct class labels to samples of unknown classes. The goal is similar to GNCD. However, the difference from GNCD is that OSC only relies on data from known classes for training, and unknown classes will only appear at the inference stage. Table 2 details the comparisons of closed-set classification, OSR, NCD, GNCD, and OSC.

Table 2
Comparisons of open set problem settings. K and U represent the set of known and unknown classes.

Task	Train		Test Set	# Target Classes
	labeled	Unlabeled		
CS	K		K	N_K
OSR	K		$K+U$	N_K+1
NCD	K	U	$K+U$	$NK+NU$
GNCD	K	$K+U$	$K+U$	$NK+NU$
OSC	K		$K+U$	$NK+NU$

Formally, OSC is defined as follows. Given two datasets $D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ and $D_{\text{test}} = \{(x_j, y_j)\}_{j=1}^{M}$, where N and M represent the number of images in the training and test datasets respectively, and x and y represent an image and the corresponding class label. Individual labels in training and test sets (*i.e.,* y_i and y_j) should satisfy the following conditions: $y_i \in K$ and $y_j \in K \cup U$ while $K \cap U = \emptyset$, where K and U are the known class set and unknown class set. Given only the labeled known classes at training, the model needs to assign class labels to all samples from known and unknown classes at the inference stage. The evaluation criteria of OSC are defined as the clustering accuracy for all classes.

4. Method

Self-supervised pretrained backbone. We use a vision transformer (ViT-B-16) as the backbone network. Since the datasets we use are small-scale finegrained datasets, we initialize the model's backbone with large-scale dataset pretrained weights to improve the generalization of the model. The ideal candidates are the ImageNet pretrained weights. To prevent label leakage, we choose self-supervised ImageNet weights.

More specifically, we use the DINO [2] pretrained weights on ImageNet. The main reason is that DINO is a strong nearest neighbor classifier, which makes the model much more transferable to solve a clustering task. Secondly, since the final goals of OSC and GNCD are same, using DINO pretrained weights allows us to make a fair comparison of our model against the method in GNCD [29] (which uses the same pretrained model) and determine whether having samples from unknown classes would affect the clustering performance.

Adaptive weighted reconstruction. To reduce the bias toward known classes and retain important features for clustering unknown classes, we use a selfsupervised training objective for the model. Self-supervised training has been proven to learn robust low-level features [18] without the requirement of additional annotations.

We use the reconstruction loss and masking patches from Mask AutoEncoder (MAE) [15]. We apply it to reduce the amount of information loss when projecting an image to a feature space. This property makes it ideal for OSC since we do not know what information might be useful to cluster unknown classes.

In a ViT model, an input image x is split into P patches: $\{p_i\}_{i=1}^{P}$. Each patch is then projected to a patch token so that the tokens are $\{t_i\}_{i=1}^{P}$. In MAE, a certain percentage of the patches are randomly masked before entering the encoder (ViT structure). Assume 50 % of the patches will be masked and the number of patch tokens generated is $p/2$. Then for each masked patch, a learned masked token is inserted into the location where the patch belongs. This restores the number of patch tokens to P. The P patch tokens pass through the decoder and for each patch token, a reconstructed patch is generated $\{\widehat{p}_i\}_{i=1}^{P}$. The reconstruction loss is then calculated between the original patches and the reconstructed ones, *i.e.*, $\mathcal{L}_{rec} = \frac{1}{P}\sum_{i}^{P}(p_i - \widehat{p}_i)^2$. In MAE, with 75 % of the patches masked, the model can still reconstruct the original image with good quality. However, we observe that details of objects are missing after the reconstruction. Since we work on fine-grained tasks, the details of objects play a big role in clustering. Thus we first reduce the percentage of patches masked to 50 % to keep more information. We think each patch has a different richness of information, which deserves a different importance rating. For example, given a bird flying in the sky, all patches that intersect the bird should have more texture information, and therefore more important. In contrast, patches of the background sky are mainly blue and white and lack useful information.

Thus, we apply weights $w = \{w_i\}_{i=1}^{P}$ to different patches and modify the masking procedure. The whole process is shown in the middle part of Fig. 2. At the start of the training, due to initialization, the patches are masked randomly. After reconstruction, the softmax normalized $\mathbf{w}$ is multiplied by the reconstruction loss of each patch and then summed as the weighted reconstruction loss for back-propagation. The rationale behind this is that if a patch has a larger reconstruction loss, it means that the model needs to focus more on that patch. To reduce the reconstruction loss, the weight of the patch would be reduced. During the training iterations, patches with higher w_i would be masked as they are easier to reconstruct by the model.

By introducing patch weighting to the model, there is an importance priority among the patches, and patches with richer information would not be masked. We generate the $\mathbf{w}$ by adding a separated fully-connected layer to process the class token. And $\mathbf{w}$ are softmax normalized which prevents the trivial solution of setting all weight values to zero. This allows the class token to focus not only on the most discriminative patches of known classes but also on other less discriminative and still descriptive patches that contain object parts.

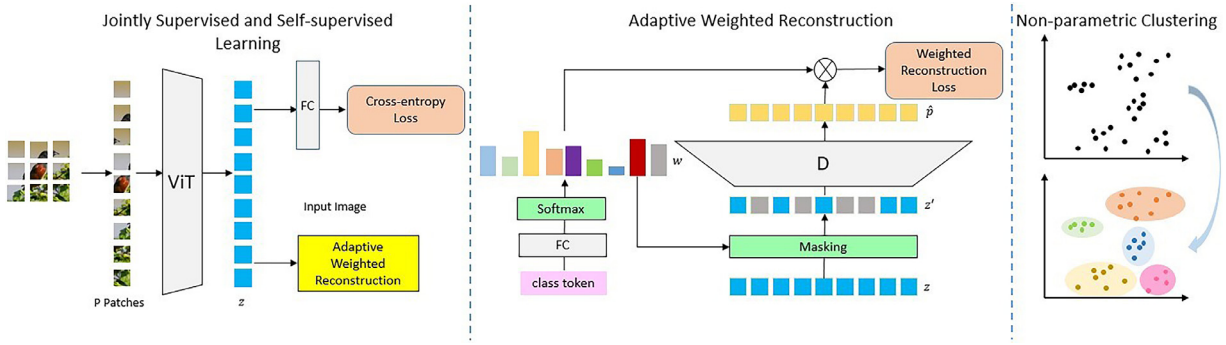

Fig. 2. The proposed method. Our method is jointly trained with supervised and selfsupervised learning. The clustering results are obtained through non-parametric clustering on feature embedding.

Softened model prediction with temperature scaling. Unlike GNCD, where there is a labeled set and an unlabeled set during training, the only information available to the model is the labeled set consisting of samples only from known classes. Thus we follow the common design of object classification models and use a parametric classification head as well. A classification head trained by the cross-entropy loss will encourage the backbone model to discover the most discriminative features among the known classes.

Since the datasets we use are fine-grained classification datasets, the known and unknown classes share many high-level semantic similarities. For example, birds from the same ancestors may share very close beaks or claws. Using cross-entropy loss allows direct supervision from label knowledge to the representation learning. However, training classification heads with crossentropy loss only on known classes make them biased towards the known classes and output overconfident probabilities [10]. To alleviate this issue, we apply a common technique used in OSR: temperature scaling [10]. Temperature scaling is a post-processing technique to soften the predicted probabilities of a model, which divides prediction logits of a model by a scalar parameter.

Given an input image x, denoting the backbone as E which projects x into a feature vector (the class token) z:$z = E(x)$,$z \in R^Z$, where Z is the length of the feature vector. The feature vector z enters the classifier head f to generate the output logits $\hat{}y$. Therefore, the temperature scaling is applied as:

$$\hat{y} = \frac{\exp\left(f(z)/T\right)}{\sum_i^Z \exp\left(f(z)/T\right)} \tag{1}$$

where T denotes the temperature and the cross-entropy loss is calculated as:

When temperature scaling is applied, it operates by dividing these logits by a temperature value T, effectively modifying the scale of the logits before they are passed through the softmax function. When T is at 1, the model behaves as usual. When $T > 1$, the logits are scaled down, which softens the softmax outputs, producing more uniform probabilities across classes. And when $T < 1$, the logits are scaled up, making the softmax output more peaked, thereby increasing confidence.

Temperature scaling can also influence the clustering process in OSC. First of all, by reducing the confidence in predictions for outliers or ambiguous inputs (those potentially from unknown classes), temperature scaling encourages the clustering algorithm to treat these inputs as separate from the known class clusters. This helps in forming distinct clusters for novel classes, improving the model's ability to discover new classes in OSC. Secondly, when logits are overconfident, the differences between logits for different classes may be extreme, leading to tight clusters within known classes but poor separation for outliers. Temperature scaling smooths these differences, allowing a clustering algorithm to find more reasonable distances between points and better balance cluster sizes. This is especially important for clustering algorithms that rely on distance measures. Finally, the scaled logits, when used for clustering, provide a better representation of uncertainty in the feature space. Instead of forcing ambiguous inputs into tight clusters, the softened probabilities encourage the clustering algorithm to consider a wider distribution of possibilities, which can lead to more accurate clustering of novel or unknown data.

The cross-entropy loss with temperature scaling is:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) \tag{2}$$

where N is the number of samples in the dataset and y is the ground truth label.

5. Experiment

5.1. Experimental setup

Dataset. We perform extensive experiments across six fine-grained classification datasets: CUB [31], NABird [28], Stanford Dog [17], Stanford Car [19], FGVC-Aircraft [22], and HERB19 [26]. Specifically, CUB, NABird, and Stanford Dog datasets encompass

Table 3
Results on the evaluated fine-grained datasets. *OffShelf-ImNet* and *OffShelfDINO* utilize the off-the-shelf ImageNet-supervised and DINO self-supervised image representations. Ours* represents our method on non-overlapping split datasets.

	CUB			SCAR			NABird			Aircraft			SDOG		
Method	K	U	All	K	U	All	K	U	All	K	U	All	K	U	All
OffShelf-ImNet	63.6	59.9	53.4	14.9	16.8	12.4	45.9	62.1	42.5	18.3	.17.7	14.6	60.7	73.8	43.2
OffShelf-DINO	38.7	40.4	34.2	13.3	13.1	11.3	26.5	38.7	25.1	17.7	18.0	14.7	51.5	61.3	39.8
Baseline(ViT-B-16)	66.7	61.8	54.6	64.0	50.5	47.5	53.1	61.9	48.1	63.8	54.7	44.8	61.7	72.0	46.4
Ours	**69.5**	63.3	**60.2**	**75.5**	**57.0**	**55.6**	**55.6**	63.0	**50.7**	**73.1**	**59.3**	**51.0**	**63.2**	**74.1**	**46.7**
	Ours*	67.6	**63.8**	59.8			54.0	**67.4**	50.5						

animals, with CUB and NABird being dedicated to bird species. In contrast, Stanford Car and FGVC-Aircraft comprise distinct sub-types of vehicles and aircraft respectively. Lastly, HERB19 serves as a herbarium species dataset. These six datasets collectively span varying degrees of class similarity: sub-classes under the same super-class (CUB and NABird), sub-classes within the same domain (animals and transportation), and classes from diverse domains. Each dataset is partitioned into training, validation, and test sets. Notably, a subset of classes is randomly selected to serve as the known classes. The test set amalgamates D^K_{test} and D^U_{test}. Details of datasets are shown in Table 9.

Evaluation protocol. For each dataset, we train the model on D_{train}. At the inference stage, we evaluate the performance of the model using the clustering accuracy, the same criteria used in NCD [11; 12] and GNCD [30]:

$$clustering\_acc = \max_{u \in U} \frac{1}{N} \sum_{i=1}^{N} \{y_i = \hat{y}_i\} \tag{3}$$

where y_i is the ground truth label of an image, $\hat{y}_i$ is the predicted label of the image, N is the number of possible classes, and U is the set of all permutations of possible classes. We report three clustering accuracy: on known classes in the test set, on unknown classes in the test set, and on all classes in the test set.

Implementation details. The model backbone used across all experiments is a ViT-B-16 backbone. We initialize the backbone with DINO pretrained weights on ImageNet, except for the ablation experiments where we compare supervised pretrained weights with DINO pretrained weights. The learning rate starts at 1×10^{-4} and decays by a cosine annealed schedule. The batch size is 8 and the input image size is $224 \times 224 \times 3$. We use padding, random crop, random flip, rotation, translation, and shearing for data augmentation. The temperature scaling factor T is determined by grid search on CUB dataset and is set to 1.5. The weights of the cross-entropy loss and the reconstruction are 1 and 0.001. After the training, we extract the class tokens as the feature vectors and directly apply k-means clustering to them to obtain the predicted classes. The number of unknown classes is assumed to be known in advance.

5.2. Backbone comparison

We test the clustering accuracy of four backbones (ResNet [16], EfficientNet [27], ViT [7], and Swin transformer [21]) on two datasets (CUB and SCAR). Following the fine-grained classification procedure, we initialize the backbone with trained weights on ImageNet. Then the models are finetuned on the train set. ViT and Swin transformers have much higher clustering accuracy than ResNet and EfficientNet on both datasets. This may suggest that Vision Transformers are comparably better than ConvNets at retaining subtle information which could be used to distinguish between the fine-grained classes. We choose ViT-B-16 as our backbone because it has a performance comparable to Swin transformer but requires significantly fewer resources to train, and also because ViT-B-16 is a widely used backbone with DINO pretrained weights.

5.3. Comparisons with the baseline

As OSC is new, there are no existing methods designed to solve OSC. Methods for OSR do not focus on clustering unknown classes while methods for NCD and GNCD require the usage of unknown sets during training. Therefore, our OSC baseline is a ViT-B-16 finetuned on the target dataset. In Table 3, we report results of two additional baselines, *i.e.*, running K-means directly on 1) the off-the-shelf DINO image representation (OffShelf-ImNet) and 2) off-the-shelf ImageNet-Supervised image representation (OffShelf-DINO).

We can see that OffShelf-ImNet has a biased performance on different datasets. The performance on CUB, NABird, and SDOG (all animal datasets) are numerically higher than that from SCAR and Aircraft (transportation). The same bias is observed on OffShelf-DINO. Due to the lack of supervised labels, the performance of OffShelf-DINO is lower than that of OffShelf-ImNet.

The difference between OffShelf-ImNet and Baseline(ViT-B-16) shows that by simply finetuning on the known classes of the target dataset, the overall clustering accuracy increases significantly. The clustering accuracy of known and unknown sub-sets both increase but the unknown set gain is relatively smaller. This suggests that the unknown classes do share significant high-level semantic features with the known classes. Finally, our methods outperform the baseline on all clustering accuracy across all datasets with a substantial gain varies from 2.5 % to 8 % except a marginal 0.3 % on SDOG. The comparison suggests that our method produces a better feature representation for clustering for both known and unknown classes.

Table 4
Numbers of overlapping classes between target datasets and ImageNet.

Dataset	Total	Overlapping	Non-overlapping
CUB	200	55	145
NABird	400	199	201
SCAR	196	0	196
FGVC	102	0	100
SDOG	120	120	0
HERB19	683	0	683

Table 5
Subclass clustering accuracies of known classes in CUB.

Class	# of classes	# of samples	Clustering accuracy
Cowbird	2	59	1.0
Gull	8	238	0.962
Auklet	4	97	1.0
Blackbird	4	119	0.992
Crow	2	60	1.0
Oriole	4	119	0.983
Grosbeak	4	120	0.992
Cuckoo	3	78	0.987
Merganser	2	60	0.983
Jay	3	90	0.967
Cormorant	3	89	0.989
Jaeger	2	60	1.0
Bunting	3	78	1.0
Finch	2	59	1.0
Hummingbird	3	90	1.0
Goldfinch	2	59	1.0
Kingbird	2	58	1.0
Catbird	2	60	1.0
Albatross	3	89	0.989
Flycatcher	7	187	0.925
Grebe	4	119	0.992
Kingfisher	5	149	1.0

5.4. Effect of class overlap

Since our model uses the DINO pretrained weights, which are trained on the whole ImageNet dataset, there is the potential for class overlap be-tween the test sets and ImageNet. We calculate the numbers of overlapping classes between our target datasets and ImageNet and show them in Table 4. For NABird, since there are hierarchical class labels, we use them to make sure classes with parent classes in ImageNet are also considered overlapping classes. Since SDOG is a subset of ImageNet, there is a 100 % overlap rate. For SCAR, FGVC, and HERB19, there are no exact overlap classes between them and ImageNet. And for CUB and NABird, there are some overlappings. Thus we resplit CUB and NABird to make sure the unknown classes are from the non-overlapping classes. The results are reported in Row *Ours** in Table 3. By comparing Row *Ours* and *Ours**, we see a drop in clustering accuracy in known classes, an increase in unknown classes, and no difference in all classes. The clustering accuracies are still higher than the baselines in all sets. Furthermore, even though SDOG is a subset of ImageNet and has a lower number of classes to cluster than CUB and NABird, the clustering accuracy is higher for CUB and NABird than SDOG. This shows the existence of unlabelled unknown images in the pretrained set does not affect the clustering performance.

Another potential overlapping is due to some classes' parent classes being in ImageNet. For example, even though there are no exact classes from FGVC and SCAR exist in ImageNet, the classes airplane and car are in ImageNet. However, since we only use DINO pretrained weights to have a better initialization, and the pretrained weights are trained self-supervised, there is no class information leakage. Furthermore, to make sure our model performs well in subclasses, we aggregate the 200 classes from CUB to 70 parent classes and calculate the clustering accuracy in each parent class. After training on 100 known classes, we check the subclass clustering accuracies in each parent class. We ignore parent classes with only one subclass. The results are reported in Table 5 an Table 6. In all parent classes, the intra-parent class clustering accuracies are higher than 65 %. This means that even if the parent class appears in ImageNet and gives the model extra information on separating the parent class from other classes, our model is able to learn to separate the subclasses.

5.5. Proximity between known and unknown classes

One assumption in OSC is that the known and unknown classes need to exhibit similar discriminative features learnable by models. In Table 7, we show the transferability of learned image representation as a function of clustering accuracy on a set of datasets using

Table 6
Subclass clustering accuracies of unknown classes in CUB.

Class	# of classes	# of samples	Clustering accuracy
Raven	2	50	1.0
Waxwing	2	60	1.0
Sparrow	21	581	0.907
Thrasher	2	59	1.0
Warbler	25	738	0.794
Woodpecker	6	178	0.944
Wren	7	206	0.951
Shrike	2	60	1.0
Tern	7	208	0.957
Vireo	7	194	0.943
Waterthrush	2	60	1.0
Swallow	4	119	0.681
Tanager	2	56	1.0

Table 7
Effect of proximity between known and unknown classes measure by the surrogate of clustering accuracy.

Training Set	CUB	SCAR	NABird	Aircraft	SDOG
OffShelf-DINO	34.2	11.3	25.1	14.7	39.0
CUB	**60.2**	11.5	**36.2**	15.4	**51.0**
SCAR	19.7	**55.6**	13.7	**16.6**	28.1

Table 8
Ablation studies. *MAE* stands for using the original reconstruction loss with masking proposed in MAE. *WRE* denotes the adaptive weighted reconstruction loss and *TS* denotes the temperature scaling. *CS* stands for closed-set classification accuracy.

MAE	WRE	TS	CUB				SCAR			
			CS	K	U	All	CS	K	U	All
			81.7	66.7	61.8	54.6	86.7	64.0	50.5	47.5
√			80.4	67.6	60.2	54.8	87.4	64.9	51.6	48.5
√	√		81.5	67.7	64.4	56.0	86.9	66.0	52.1	49.4
√		√	**83.5**	68.9	62.6	57.9	87.4	71.3	54.0	54.0
√	√	√	83.0	**69.5**	**63.3**	**60.3**	**89.1**	**75.5**	**57.0**	**55.6**

the model trained on a foreign dataset. We choose two datasets as the training set, CUB and SCAR. Row 1 of Table 7 shows the clustering accuracy on OffShelf-DINO as the baseline. Row 2 and row 3 report the clustering accuracy on five datasets by models trained on CUB and SCAR. When the model was trained on CUB, the performance on CUB, NABird, and SDOG improves compared to Row 1. It is expected since NABird also contains classes of birds. The relatively smaller improvement in the SDOG data may be a result of the shared highlevel semantic features between dogs and birds (*i.e.*, they are both animals and shared features such as fur *vs*. feathers and have common body parts such as the eyes). We see no distinct improvement in SCAR and Aircraft datasets, as birds and transportation are not related to each other. Similar observations can be made when we compare Row 3 and Row 1, *i.e.*, car finetuned image representations do not help cluster airplanes, nor do they for the animal datasets. In fact, the animal dataset clustering performance is lower as the finetuning may have "erased" the model's discriminative power between the animal classes obtained during the DNIO pretraining stage.

Based on these observations in Table 7, we demonstrate the necessity of the fine-grained assumption in our OSC setting.

5.6. Ablation study

In Table 8, we evaluate the effect of two components of our method: adaptive weighted reconstruction and temperature scaling.

Adaptive weighted reconstruction Row 1 and Row 3 show the effect of adding weighted reconstruction loss with masking. Improvements are seen across clustering accuracy on CUB and SCAR. The overall clustering accuracy is increased by 2.5 % on CUB and by 1.8 % on SCAR respectively. This shows that our reconstruction loss allows the learned features to include more descriptive information so that the feature space is more suitable for clustering on both known and unknown classes. By comparing Row 2 and Row 3, we show the modified reconstruction loss improves the model's clustering accuracy on CUB and SCAR.

Temperature scaling Row 1 and Row 3 show the effect of temperature scaling. By using the temperature scaling to reduce the model's overconfidence in known classes, we observe a 3.3 % and 6.5 % increase in overall clustering accuracy on CUB and

Table 9
Datasets used in our experiment. We show the number of known and unknown classes, N_K and N_U in the first two rows. We also show the number of images in the training, validation, and test set (both known&unknown proportions), and test unknown class set as *D*train, *D*val, *D*test*K,* and *D*testU respectively.

	CUB	NABird	SCAR	Aircraft	SDOG
N_K	100	304		50	60
N_U	100	100	98	50	60
Dtrain	2.4K	16.3K	3.3K	2.7K	5.9K
Dval	0.6K	4.1K	0.8K	0.7K	1.5K
$D_{test}K$	2.9K	18.7K	4.1K	1.7K	2.1K
$D_{test}U$	2.9K	5.9K	4.0K	1.6K	4.3K

SCAR respectively. On the CUB dataset, the increase in unknown clustering accuracy is small while the increase in overall clustering accuracy is larger which could imply that the improvement is from separating the known classes and unknown classes. This makes sense as the temperature scaling is used in OSR to induce lower confidence in the class predictions and promote descriptive features to be kept.

When both the weighted reconstruction loss with masking and the temperature scaling are used (Row 4), there is a further increase in clustering accuracy compared to using either. This shows that the two components are complementary and overall, our method boosts the clustering accuracy by

5.7 % on CUB and 8.1 % on SCAR.

5.7. OSR methods

Let ACC_K, ACC_U, and ACC_{All} be the clustering accuracy on known, unknown, and all classes respectively. Assume we have N_K samples in the known classes and N_U samples in the unknown classes, then if samples from known classes and unknown classes are perfectly separated, which is the goal of OSR, we have:

$$ACC_{All} = \frac{N_K}{N_K + N_U} ACC_K + \frac{N_U}{N_K + N_U} ACC_U \tag{4}$$

A more common calculation equation would be:

$$ACC_{All} = \frac{1}{N_K + N_U} \left(ACC_K * C_K + ACC_U * C_U \right) \tag{5}$$

where C_K is the number of samples that are correctly classified as from known classes and C_U is the number of samples that are correctly classified as from unknown classes. If the OSR method perfectly separates known and unknown classes, C_K and C_U would equal N_K and N_U correspondingly, and Eq. (5) would become Eq. (4). Note that in Eq. (5), any misclassified samples during the OSR step are considered to be clustering wrong.

Based on Table 8, we can easily see samples from known and unknown classes are not separated well, as ACC_{All} is always lower than both ACC_K and ACC_U. Even though with more classes, the clustering accuracy is inclined to be lower, it is obvious OSR method may be useful in OSC. [30] shows that a model with a good closed-set classification accuracy also has high OSR performance. We have reproduced experiments on general coarsegrained image datasets and obtained similar results. However, we do not get the same results on fine-grained datasets. In Fig. 3, we plot the max logits of the known classes of CUB, unknown classes of CUB, SDOG, and SCAR. The shift of the max logit distribution is significant between CUB and DOG, and CUB and SCAR. On the other hand, the shift of max logit distribution on unknown classes from CUB is not large enough to generate good OSR results. Assuming the OSR binary classification accuracy is ACC_{OSR}, which is between 0 and 1, if the clustering algorithm is applied on predicted known and predicted unknown sets after the OSR method is applied, ACC_{OSR} would be the ceiling of the clustering accuracy. Due to this accumulation of errors, the OSR performance is crucial if OSR is applied.

We show the clustering accuracy on different datasets when OSR is combined into our pipeline. The process is as follows: 1) we train the model and extract the features of images in the test dataset 2) Following the common OSR process, we use the max logit of True Positive Rate (TPR) 95 % on known classes to get the threshold τ of separating known and unknown classes. Images with max logit lower than τ are considered to be from unknown classes. 3) We use k-means to cluster on the predicted known class sample sets, assuming the number of classes is the number of known classes. Any sample that is classified wrong during the OSR step is automatically categorized as a wrong prediction. The same process is applied to the predicted unknown class sample sets with the number of classes equal to the number of unknown classes. Then we use formula 4 to get the total clustering accuracy. The results are reported in Table 11. We can see that due to the similarity in max logit distribution between known and unknown classes, the recall rate is low on all datasets. This results in a large percentage of unknown classes classified wrong at the OSR step and the final clustering accuracy is lower on all datasets.

Clustering accuracies on CUB when using different %TPR thresholds in OSR are calculated. The highest clustering accuracy is at 55 % TPR. However, in reality, 55 % TPR is too low and is unlikely to be chosen.

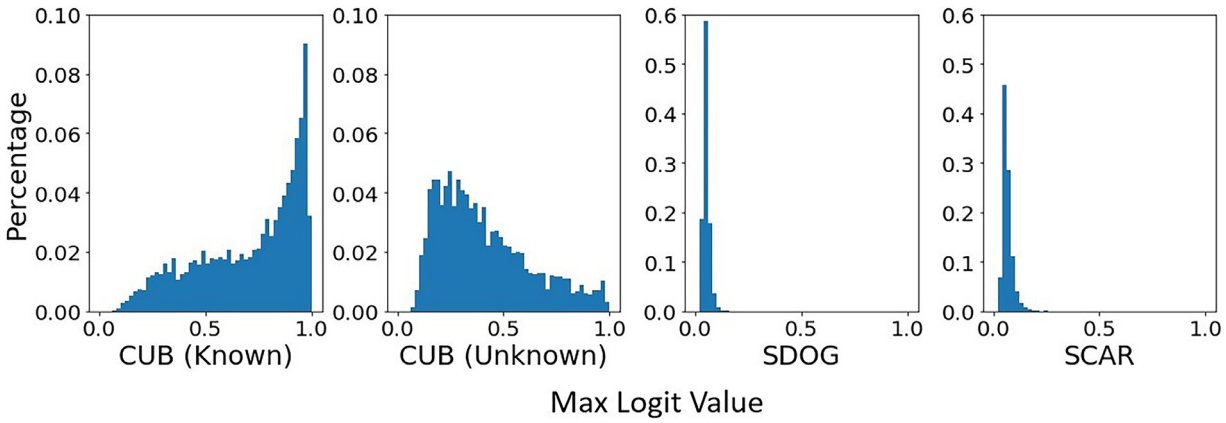

Fig. 3. Max logit outputs of our model on different datasets. There is a large shift in the distribution of max logits between CUB and SDOG or SCAR. The difference between max logits of known classes and unknown classes of CUB is not big enough as there are still some overlaps.

Table 10
Comparison with GNCD.

Method	CUB	SCAR	FGVC	HERB19	K U	All	K U	All	K U	All	K U	All
GNCD	64	50	57	53	29	42	60	37	49	51	27	35
Ours	**70**	**63**	**60**	**76**	**57**	**56**	**73**	**59**	**51**	47	**31**	**36**

We consider another commonly used OSR method, which is the difference between the reconstruction loss of known and unknown classes. Even though our method uses a reconstruction loss, its purpose is to retain information of the original information. Even if an image from an unknown class is given to the model, we still want the model to be able to reconstruct the image well as information loss can be vital in clustering unknown classes. Furthermore, OSR methods that use reconstruction loss, [9; 23; 33; 34] require additional class tokens to reconstruct images conditionally. In these cases, an input image is reconstructed N_K times, where N_K is the number of known classes, and the minimum reconstruction loss is used. In our datasets, we have 56 to 304 classes, which makes it computationally expensive to apply these methods to fine-grained datasets. Due to the reasons mentioned above, our method chooses to directly apply a clustering algorithm on the extracted features, without any OSR classification in advance.

5.8. Comparing to GNCD

OSC shares a similar objective as GNCD. The difference is that OSC does not rely on any knowledge and data of unknown classes during training. In Table 10, we show the comparison of our model under the OSC setting and the method (GNCD) under the GNCD setting. Both methods use the ViTB-16 as the backbone and the initialization weights are the DINO pretrained weights. We can see that for the evaluated datasets, our model performs even better on CUB, SCAR, and FGVC-Aircraft than GNCD despite no knowledge of unknown classes being available in training. In addition, with the usage of a classification head, our method shows a much higher clustering accuracy than GNCD for both known and known classes. This affirms that the similarity between known and unknown classes can serve as a catalyst, enabling a well-honed representation of known classes to enhance the clustering performance of unknown classes.

5.9. Feature-level analysis

Utilizing T-SNE, we visualize the feature-level representations in Fig. 4. To enhance clarity, we randomly select 20 classes from the extensive array of fine-grained classification datasets. Our results clearly demonstrate that following fine-tuning, our method effectively delineates clusters with distinct boundaries for both known and unknown classes. Moreover, distinct boundaries are evident in CUB v.s. SCAR and CUB v.s. SDOG, affirming our method's proficiency in segregating samples from dissimilar classes rather than those originating from the same superclass. The model's ability to cluster dog breeds while facing challenges with car classes in SCAR can be attributed to the shared resemblances between dogs and birds, in contrast to the lack thereof between cars and birds.

We further show the TSNE visualization on the full CUB datasets and full CUB datasets combined with SDOG/SCAR in Fig. 5. On the top row, we compare the known and unknown classes, and on the bottom row, we compare subclasses. Comparing Fig. 5a and Fig. 5b, we see that after finetuning, the known and unknown classes in CUB are more separated while each class forms a tighter and more separated cluster. In Fig. 5c and Fig. 5d, the known classes are from CUB and the unknown classes are from SDOG and SCAR. The top row shows that when the known and unknown classes are from different superclasses, they are perfectly separated. If we look at the bottom row, we see that the model can cluster some classes in SDOG but not in SCAR. The reason can be the closeness

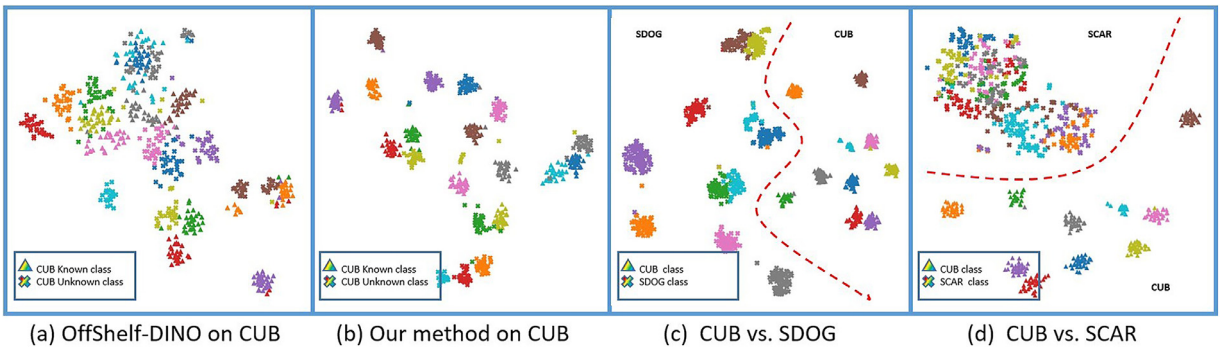

Fig. 4. T-SNE Visualization. Different colors and shapes represent different classes. We show four settings: OffShelf-DINO, our method on CUB (known *vs*. unknown), our method on CUB and SDOG, and our method on CUB and SCAR.

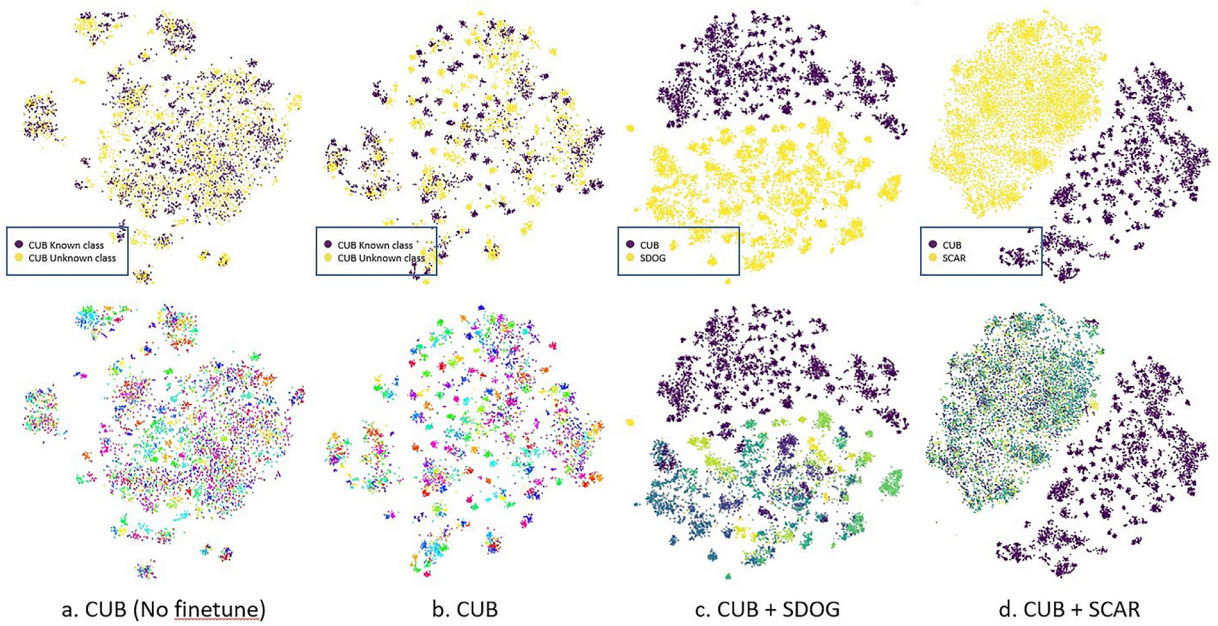

Fig. 5. TSNE visualization on full CUB and full CUB with SDOG/SCAR. The bottom and top row show visualization of the same features but with different colors. The top row compares the known and unknown classes, while the bottom row shows all classes. For *c* and *d*, in the bottom row, we do not show subclasses under CUB as they are shown in *b*.

Table 11
Effect of applying OSR in OSC setting on different datasets.

	CUB	SCAR	NABird	Aircraft	SDOG
AUROC	78.4	83.4	43.1	70.4	62.2
Recall	27.9	35.2	14.8	13.8	24.1
Clu Acc w OSR	38.4	39.5	41.0	34.7	25.7
Clu Acc wo OSR	**60.2**	**55.6**	**50.7**	**51.0**	**46.7**

between CUB and SDOG as birds and dogs are both animals. Cars and birds share fewer high-level semantic features, resulting in no useful feature extracted by our model to cluster classes from SCAR. Another reason can be due to the SDOG being a subset from ImageNet, so the loaded DINO pretrained weights include information about dogs.

5.10. Estimating the number of classes

Following the procedure in GNCD [29], we estimate the number of total classes. We combine the training set with the test set and apply k-means to the combined set with different numbers of classes. We calculate the clustering accuracy on the training set and choose the number of classes with the highest accuracy as the predicted number of total classes. The results are reported inTable 12.

Table 12
Estimation of the total number of classes in different datasets.

	CUB	SCAR	NABird	Aircraft	SDOG
Ground truth	200	198	400	112	120
Predicted	146	153	307	58	98
Error rate	27 %	23 %	24 %	48 %	18 %

On most datasets, there is a 20−30 % error rate, and on FGVC-Aircraft [22], the error rate is 48 %. We speculate the reason behind this is the similarity between the known and unknown classes. This causes the known and unknown classes to be close in feature space.

6. Conclusion

In this study, we address the challenge of autonomous novel class discovery in open world without human interactions, a concept termed as OSC. Unlike GNCD, OSC introduces a more practical scenario, marking a significant step forward in facilitating continuous learning for robotic systems. By autonomously uncovering novel classes, OSC empowers robotic systems to seamlessly execute tasks without experiencing interruptions or slowdowns. Our empirical experiments on fine-grained categorization datasets demonstrate the feasibility of OSC. Moreover, our approach integrates supervised learning with adaptive weighted reconstruction, enabling the learning of distinctive features from known classes while preserving potential information about unknown classes. Notably, our method outperforms GNCD despite lacking access to the novel classes.

Yet OSC has some limitations. One limitation is its reliance on finegrained datasets, where class distinctions are subtle and well-defined. This specificity can hinder performance when applied to more diverse or less finegrained datasets, where class boundaries may be more ambiguous or the variation within classes is higher. In such cases, the methods developed for OSC may struggle to distinguish novel classes or could misclassify dissimilar instances as the same class due to no guidance on what features used to distinguish classes. Addressing these challenges may require more robust feature extraction techniques to better handle broader, more heterogeneous data distribution. Another limitation of Open Set Clustering (OSC) is the challenge of estimating the number of unknown classes. In many cases, the true number of novel or unseen classes is not known in advance, making it difficult for OSC algorithms to effectively determine how many clusters to form. Incorrect estimation can lead to over- or under-clustering, where the model either splits a single class into multiple clusters or merges distinct unknown classes into one. This uncertainty complicates the discovery of new classes and can impact the overall performance of the clustering process.

Future work will focus on integrating open-set recognition and incremental learning into the system to enable robotics to autonomously discern new objects, cluster them into groups, and augment the existing knowledge base with new classes. A primary challenge lies in mitigating error propagation by automatically rectifying misclassified objects in subsequent stages.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Xuelin Zhang: Conceptualization, Data curation, Formal analysis, Methodology, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Feng Liu:** Methodology, Resources, Supervision, Writing – review & editing. **Xuelian Cheng:** Conceptualization, Methodology, Writing – review & editing. **Siyuan Yan:** Methodology, Visualization, Writing – review & editing. **Zhibin Liao:** Methodology, Writing – original draft, Writing – review & editing. **Zongyuan Ge:** Conceptualization, Investigation, Methodology, Software, Supervision, Writing – review & editing.

References

[1] A. Bendale, T. Boult, Towards open set deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[2] M. Caron, H. Touvron, I. Misra, H. J´egou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: Proceedings of the International Conference on Computer Vision (ICCV), 2021.
[3] G. Chen, P. Peng, X. Wang, Y. Tian, Adversarial reciprocal points learning for open set recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[4] 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, 2020.
[5] L. Chen, H. Zhang, J. Xiao, W. Liu, S.-F. Chang, Zero-shot visual recognition using semantics-preserving adversarial embedding networks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017, pp. 1043–1052. pages.
[6] H. Chi, F. Liu, B. Han, W. Yang, L. Lan, T. Liu, G. Niu, M. Zhou, M. Sugiyama, Meta discovery: learning to discover novel classes given very limited data, International Conference on Learning Representations, 2022.
[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: transformers for image recognition at scale. ICLR, 2021.
[8] Y. Fu, T.M. Hospedales, T. Xiang, S. Gong, Transductive multiview zero-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. 37 (11) (Nov. 2015) 2332–2345.
[9] Z. Ge, S. Demyanov, Z. Chen, R. Garnavi, Generative openmax for multi-class open set classification, in: Proceedings of the British Machine Vision Conference Proceedings (BMVC), 2017.

[10] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: Proceedings of the 34th International Conference on Machine Learning, 70, ICML'17, 2017, pp. 1321–1330. pageJMLR.org.
[11] K. Han, S. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman. Automatically discovering and learning new visual categories with ranking statistics. CoRR, abs/2002.05714, 2020.
[12] K. Han, S. Rebuffi, S. Ehrhardt, A. Vedaldi, A. Zisserman, Autonovel: automatically discovering and learning novel visual categories, IEEE Trans. Pattern Anal. Mach. Intell. 44 (10) (2022) 6767–6781.
[13] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, A. Zisserman, Automatically discovering and learning new visual categories with ranking statistics, International Conference on Learning Representations (ICLR), 2020.
[14] K. Han, A. Vedaldi, A. Zisserman, Learning to discover novel visual categories via deep transfer clustering, International Conference on Computer Vision (ICCV), 2019.
[15] K. He, X. Chen, S. Xie, Y. Li, P. Dolla´r, and R. Girshick. Masked autoencoders are scalable vision learners. arXiv:2111.06377, 2021.
[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVchen2020learningPR, 2016.
[17] A. Khosla, N. Jayadevaprakash, B. Yao, L. Fei-Fei, Novel dataset for fine-grained image categorization, First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition, June 2011.
[18] A. Kolesnikov, X. Zhai, L. Beyer, Revisiting self-supervised visual representation learning, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
[19] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3d object representations for fine-grained categorization, 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), 2013.
[20] X. Li, D. Zhang, M. Ye, X. Li, Q. Dou, Q. Lv, Bidirectional generative transductive zero-shot learning, Neural Comput. Appl. 33 (10) (May 2021) 5313–5326.
[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.
[22] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. FineGrained visual classification of aircraft. working paper or preprint, June 2013.
[23] P. Oza, V.M. Patel, C2AE: class conditioned auto-encoder for openset recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[24] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the masses, 2021.
[25] W.J. Scheirer, A. de Rezende Rocha, A. Sapkota, T.E. Boult, Toward open set recognition, TPAMI (2013).
[26] K.C. Tan, Y. Liu, B. Ambrose, M. Tulig, S. Belongie, The Herbarium Challenge 2019dataset, 2019.
[27] M. Tan, Q. Le. EfficientNet, Rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 6105–6114, PMLR, 2019 09–15 Jun.
[28] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, S. Belongie, Building a bird recognition app and large scale dataset with citizen scientists: the fine print in fine-grained dataset collection, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 595–604. pages.
[29] S. Vaze, K. Han, A. Vedaldi, A. Zisserman, Generalized category discovery, IEEE Conference on Computer Vision and Pattern Recognition, 2022.
[30] S. Vaze, K. Han, A. Vedaldi, A. Zisserman, Open-set recognition: a good closed-set classifier is all you need? International Conference on Learning Representations, 2022.
[31] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, Caltechucsd Birds 200, California Institute of Technology, 2010 CNS-TR-2010001.
[32] Y. Xian, T. Lorenz, B. Schiele, Z. Akata, Feature generating networks for zero-shot learning, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5542–5551. pages.
[33] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, T. Naemura, Classification-reconstruction learning for open-set recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[34] X. Zhang, X. Cheng, D. Zhang, P. Bonnington, Z. Ge, Learning network architecture for open-set recognition, Proceedings of the AAAI Conference on Artificial Intelligence 36 (3) (Jun. 2022) 3362–3370.
[35] P. Zhao, H. Xue, X. Ji, H. Liu, L. Han, Zero-shot learning via visual feature enhancement and dual classifier learning for image recognition, Inf. Sci. (Ny) 642 (2023) 119161.