Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-024-01025-y

A practical approach to novel class discovery in tabular data

Troisemaine Colin[1,2] · Reiffers-Masson Alexandre[1] · Gosselin Stéphane[2] · Lemaire Vincent[2] · Vaton Sandrine[1]

Received: 9 November 2023 / Accepted: 14 April 2024
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2024

Abstract
The problem of novel class discovery (NCD) consists in extracting knowledge from a labeled set of known classes to accurately partition an unlabeled set of novel classes. While NCD has recently received a lot of attention from the community, it is often solved on computer vision problems and under unrealistic conditions. In particular, the number of novel classes is usually assumed to be known in advance, and their labels are sometimes used to tune hyperparameters. Methods that rely on these assumptions are not applicable in real-world scenarios. In this work, we focus on solving NCD in tabular data when no prior knowledge of the novel classes is available. To this end, we propose to tune the hyperparameters of NCD methods by adapting the k-fold cross-validation process and hiding some of the known classes in each fold. Since we have found that methods with too many hyperparameters are likely to overfit these hidden classes, we define a simple deep NCD model. This method is composed of only the essential elements necessary for the NCD problem and shows robust performance under realistic conditions. Furthermore, we find that the latent space of this method can be used to reliably estimate the number of novel classes. Additionally, we adapt two unsupervised clustering algorithms (k-means and Spectral Clustering) to leverage the knowledge of the known classes. Extensive experiments are conducted on 7 tabular datasets and demonstrate the effectiveness of the proposed method and hyperparameter tuning process, and show that the NCD problem can be solved without relying on knowledge from the novel classes.

Keywords Novel class discovery · Clustering · Tabular data · Open world learning · Transfer learning

Responsible editor: Rita P. Ribeiro.

✉ Troisemaine Colin
colin.troisemaine@gmail.com

[1] Department of Computer Science, IMT Atlantique, Brest, France

[2] Orange Labs, Lannion, France

Published online: 31 May 2024

1 Introduction

Recently, remarkable progress has been achieved in supervised tasks, in part with the help of large and fully labeled sets such as ImageNet (Deng et al. 2009). These advancements have predominantly focused on closed-world scenarios, where, during training, it is presumed that all classes are known in advance and have some labeled examples. However, in practical applications, obtaining labeled instances for all classes of interest can be a difficult task due to factors such as budget constraints or lack of comprehensive information. Furthermore, for models to be able to transfer learned concepts to new classes, they need to be designed with this in mind from the start, which is rarely the case. Yet this is an important skill that humans can use effortlessly. For example, having learnt to distinguish a few animals, a person will easily be able to recognise and "cluster" new species they have never seen before. The transposition of this human capacity to the field of machine learning could be a model capable of categorizing new products in novel categories.

This observation has led researchers to formulate a new problem called Novel Class Discovery (NCD) (Troisemaine et al. 2023; Hsu et al. 2018). Here, we are given a labeled set of known classes and an unlabeled set of different but related classes that must be discovered. Lately, this task has received a lot of attention from the community, with many new methods such as AutoNovel (Han et al. 2021), OpenMix (Zhong et al. 2021) or NCL (Zhong et al. 2021) and theoretical studies (Sun et al. 2023; Li et al. 2022). However, most of these works tackle the NCD problem under the unrealistic assumption that the number of novel classes is known in advance, or that the target labels of the novel classes are available for hyperparameter optimization (Troisemaine et al. 2022). These assumptions render these methods impractical for real-world NCD scenarios. To address these challenges, we propose a general framework for optimizing the hyperparameters of NCD methods where the ground-truth labels of novel classes are never used, as they are not available in real-world NCD scenarios. Furthermore, we show that the latent spaces obtained by such methods can be used to accurately estimate the number of novel classes.

We also introduce three new NCD methods. Two of them are unsupervised clustering algorithms modified to leverage the additional information available in the NCD setting. The first one improves the centroid initialization step of k-means, resulting in a fast and easy to use algorithm that can still give good results in many scenarios. The second method focuses on optimizing the parameters of the Spectral Clustering (SC) algorithm. This approach has a potentially higher learning capacity as the representation itself (i.e. the spectral embedding) is tuned to easily cluster the novel data. Finally, the last approach is a deep NCD method composed of only the essential components necessary for the NCD problem. Compared to SC, this method is more flexible in the definition of its latent space and effectively integrates the knowledge of the known classes.

While these contributions can be applied to any type of data, our work focuses on tabular data. The NCD community has focused almost exclusively on computer

vision problems and, to the best of our knowledge, only one paper (Troisemaine et al. 2022) has tackled the problem of NCD in the tabular context. This is despite the many applications of NCD in tabular data. Some examples include *cybersecurity*, where network traffic could be used to infer new types of cyber-attacks that deviate from known attack patterns; *sensor data analysis*, where new categories of equipment failures could be diagnosed and lead to faster repairs; or *fraud detection*, where new types of fraudulent transactions that deviate from known patterns could be discovered. However, this previous work (Troisemaine et al. 2022) required the meticulous tuning of a large number of hyperparameters to achieve optimal results. Methods designed for tabular data cannot take advantage of powerful techniques commonly employed in computer vision. Examples include convolutions, data augmentation or Self-Supervised Learning methods such as DINO (Caron et al. 2021), which have been used with great success in NCD works (Vaze et al. 2022; Fei et al. 2022; Zhang et al. 2022), thanks to their strong ability to obtain representative latent spaces without any supervision. On the other hand, tabular data methods have to rely on finely tuned hyperparameters to achieve optimal results. For this reason, we believe that the field of tabular data will benefit the most from our contributions.

By making the following contributions, we demonstrate the feasibility of solving the NCD problem with tabular data and under realistic conditions:

- We develop a hyperparameter optimization procedure tailored to transfer the results from the known classes to the novel classes with good generalization.
- We show that it is possible to accurately estimate the number of novel classes in the context of NCD, by applying simple clustering quality metrics in the latent space of NCD methods.
- We modify two classical unsupervised clustering algorithms to effectively utilize the data available in the NCD setting.
- We propose a simple and robust method, called PBN (for *Projection-Based NCD*), that learns a latent representation that incorporates the important features of the known classes, without overfitting on them.

The code is available at https://github.com/Orange-OpenSource/PracticalNCD.

2 Related work

The setup of NCD (Troisemaine et al. 2023), which involves both labeled and unlabeled data, can make it difficult to distinguish from the many other domains that revolve around similar concepts. In this section, we review some of the most closely related domains and try to highlight their key differences in order to provide the reader with a clear and comprehensive understanding of the NCD domain.

Semi-supervised learning is another domain that is at the frontier between supervised and unsupervised learning. Specifically, a labeled set is given alongside an unlabeled set containing instances that are assumed to be from the same classes. Semi-supervised Learning can be particularly useful when labeled data is scarce or

annotation is expensive. As unlabeled data is generally available in large quantities, the goal is to exploit it to obtain the best possible generalization performance given limited labeled data.

The main difference with NCD is that the all the classes are known in advance. Some works have shown that the presence of novel classes in the unlabeled set negatively impacts the performance of Semi-Supervised Learning models (Chen et al. 2020; Guo et al. 2020). So as these works do not attempt to discover the novel classes, they are not applicable to NCD.

Transfer learning aims at solving a problem faster or with better performance by leveraging knowledge from a different problem. It is commonly expressed in computer vision by pre-training models on ImageNet (Deng et al. 2009). Transfer Learning can be either *cross-domain*, when a model trained on a given dataset is fine-tuned to perform the same task on a different (but related) dataset. Or it can be *cross-task*, where a model that can distinguish some classes is re-trained for other classes of the same domain.

NCD can be viewed as a cross-task Transfer Learning problem where the knowledge from a classification task on a source dataset is transferred to a clustering task on a target dataset. But unlike NCD, Transfer Learning typically requires the target spaces of both sets to be known in advance. Initially, NCD was characterized as a Transfer Learning problem (e.g. in DTC Han et al. 2019 and MCL Hsu et al. 2019) and the training was done in two stages: first on the labeled set and then on the unlabeled set. This methodology seemed natural, as with Transfer Learning, both sets are not available at the same time.

The field of *open set recognition* (OSR) (Scheirer et al. 2013) stems from the observation that neural networks have a tendency to make predictions with high confidence, even for instances of classes they have never seen before. It is based on the Open-World assumption in which instances of new classes may appear during inference, so standard models cannot be used. Therefore, in OSR, samples from novel classes must be correctly identified as unknown. An example would be a fingerprint recognition model that needs to reject images of fingerprints (or anything else!) from people who aren't registered. However, as in many domains under the Open-World assumption, the goal of OSR is only to identify samples from novel classes. In NCD, we assume that this identification has already been made, and we try to actually discover the novel classes.

Generalized category discovery (GCD) was first introduced by Vaze et al. (2022) and has also attracted attention from the community (Zheng et al. 2022; Yang et al. 2022; Fei et al. 2022). It can be seen as a less constrained alternative to NCD, since it does not rely on the assumption that samples belong exclusively to the novel classes during inference. However, this is a more difficult problem, as the models must not only cluster the novel classes, but also accurately differentiate between known and novel classes while correctly classifying samples from the known classes.

Some notable works in this area include ORCA (Cao et al. 2022) and OpenCon (Sun and Li 2023). Namely, ORCA trains a discriminative representation by

balancing a supervised loss on the known classes and unsupervised pairwise loss on the unlabeled data. And OpenCon proposes a contrastive learning framework which employs Out-Of-Distribution strategies to separate known vs. novel classes. Its clustering strategy is based on moving prototypes that enable the definition of positive and negative pairs of instances.

Novel class discovery has a rich body of papers in the domain of computer vision. Early works approached this problem in a *two-stage* manner. Some define a latent space using only the known classes, and project the unlabeled data into it (DTC Han et al. 2019 and MM Chi et al. 2022). Others train a pairwise labeling model on the known classes and use it to label and then cluster the novel classes (CCN Hsu et al. 2018 and MCL Hsu et al. 2019). But both of these approaches suffered from overfitting on the known data when the high-level features were not fully shared by the known and novel classes.

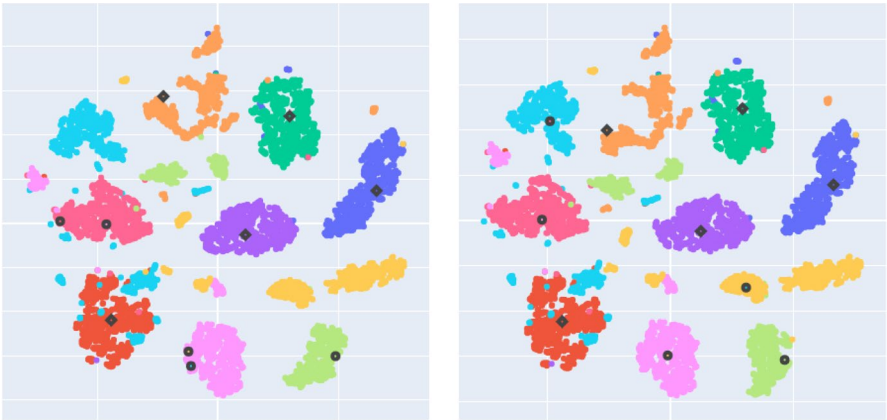
Today, to alleviate this overfitting, the majority of approaches are *one-stage* and try to transfer knowledge from labeled to unlabeled data by learning a shared representation. In this category, AutoNovel (Han et al. 2021) is one of the most highly influential works. After pre-training their latent representation with SSL (Gidaris et al. 2018), two classification networks are jointly trained. The first simply learns to distinguish the known classes with the ground-truth labels. And the other learns to separate unlabeled data from pseudo-labels defined for each epoch based on pairwise similarity. NCL (Zhong et al. 2021) adopts the same architecture as AutoNovel, and extends the loss by adding a contrastive learning term to encourage the separation of novel classes. OpenMix (Zhong et al. 2021) utilizes the MixUp strategy to generate more robust pseudo-labels.

As expressed before, although these methods have achieved some success, they are not applicable to tabular data. To date, and to the best of our knowledge, only TabularNCD (Troisemaine et al. 2022) tackles this problem. Also inspired by AutoNovel, it pre-trains a dense-layer autoencoder with SSL and adopts the same loss terms and dual classifier architecture. Pseudo-labels are defined between pairs of unlabeled instances by checking if they are among the most similar pairs.

For a more complete overview of the state-of-the-art of NCD, we refer the reader to the survey (Troisemaine et al. 2023).

3 Approaches

In this section, after introducing the notations, we define two simple but potentially effective models derived from classical clustering algorithms (Sects. 3.2 and 3.3). The idea is to use the labeled data to improve the unsupervised clustering process, and make the comparison to NCD methods more challenging. Then, we present a new method, PBN (for Projection-Based NCD, Sect. 3.4), characterized by its low number of hyperparameters needed to be tuned.



(a) Before convergence. (b) After convergence.

Fig. 1 t-SNE plots of the *Pendigits* dataset depicting the centroids before and after convergence. Note how the centroids of the known classes (the squares) don't move, as they stay the mean class point

3.1 Problem setting

We start by describing the Novel Class Discovery setup and the necessary notations. Here, data is provided in two distinct sets: a labeled set of known classes $D^l = \left\{\left(x_i^l, y_i^l\right)\right\}_{i=1}^N$ with $x_i^l \in \mathcal{X} = \mathbb{R}^d$ and $y_i^l \in \mathcal{Y}^l = \left\{1, \ldots, C^l\right\}$ the ground-truth labels of x_i^l. And an unlabeled set $D^u = \left\{\left(x_i^u\right)\right\}_{i=1}^M$ where the data samples $x_i^u \in \mathcal{X}$ are only from novel classes $\mathcal{Y}^u = \{1, \ldots, C^u\}$, which are different but related to the known classes. In other words, there is no overlap between the known and novel classes, so $\mathcal{Y}^l \cap \mathcal{Y}^u = \emptyset$. The objective is to exploit the knowledge from D^l to accurately partition D^u into the C^u clusters of the novel classes.

Following previous research, we first assume that the number of novel classes C^u is known in advance, and later propose an approach to estimate this number in a later section.

3.2 NCD k-means

This is a straightforward method that takes inspiration from *k*-means++ (Arthur and Vassilvitskii 2007), which is an algorithm for choosing the initial positions of the centroids (or cluster centers). In *k*-means++, the first centroid is chosen at random from the data points. Then, each new centroid is chosen iteratively from one of the data points with a probability proportional to the squared distance from the point's closest centroid. The resulting initial positions of the centroids are generally spread more evenly, which yields appreciable improvement in the final error of k-means and convergence time.

As shown in Fig. 1a, we naively adapt k-means++ to the NCD setting by defining C^l initial centroids. They are set as the mean class points of the known classes using the ground-truth labels. Then, we follow k-means++ and randomly select C^u new centroids in the unlabeled set, with similarly decreasing probability when closer to existing centroids. We found experimentally (see " Appendix G") that, after the initialization is complete, the best accuracy is achieved when only the centroids of the novel classes are updated, and using the unlabeled data only. In other words, the data of the known classes is only used during the initialization of the new centroids, but not during the convergence phase. Intuitively, if the centroids of the known and novel data are updated together, they have a higher risk of drifting and capturing data of the other set. The pseudocode of the proposed method is summarized in Algorithm 2 of " Appendix A".

Similarly to k-means++, we repeat the initialization process a few times and keep the centroids that achieved the smallest inertia. Note that, to stay consistent with the k-means algorithm, we use also the L_2 norm (i.e. the Euclidean distance) for NCD k-means.

3.3 NCD Spectral Clustering

Spectral Clustering (SC) is an alternative to distance-based clustering methods (such as k-means). It makes no assumptions about the structure of the data and considers the clustering problem as a graph partitioning problem and seeks to decompose the graph into connected components (Luxburg 2007). The input of the Spectral Clustering algorithm is an adjacency matrix (sometimes called similarity graph) which must accurately represent the neighborhood relationships between data points. There are multiple ways to construct such a graph, however there is no theoretical result on the relation between the similarity graph construction method and the Spectral Clustering results. Here, we employ a popular approach to construct the adjacency matrix, which is through a Gaussian kernel: $A_{i,j} = \exp\left(-\|x_i - x_j\|_2^2/(2\sigma^2)\right), \forall x_i, x_j \in \mathcal{X}$ where the parameter σ controls the width of the neighborhood.

Following the Ng-Jordan-Weiss algorithm (Ng et al. 2001), we use the symmetric normalized Laplacian $L_{sym} = D^{-1/2}LD^{-1/2}$, where D is the degree matrix defined as: $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$, in which n is the number of data samples. The next step consists in finding the first u eigenvectors of L_{sym} to form the spectral embedding $U \in \mathbb{R}^{n \times u}$, where row i corresponds to point x_i. Finally, the points in U are partitioned with k-means into clusters.

The optimal value of σ in the Gaussian kernel can vary widely depending on the distribution of inter-point distances. For this reason, we take inspiration from the rules of thumb given in Luxburg (2007) and employ a minimum spanning tree (MST) to choose σ. In the past few years, several graph-based clustering methods that use the MST have been proposed (Khan and Mohanty 2022), as it reliably represents the layout of the data and is inexpensive to compute. In the approach proposed here, we denote d_{max} the length of the longest edge in the MST of inter-point

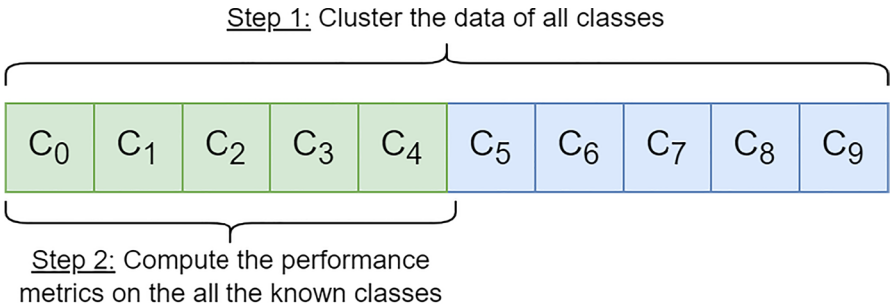


Fig. 2 NCD Spectral Clustering parameter optimization process

distances. The longest edge of the MST is a much studied object (Stuetzle 2003) and is representative of the scale of the dataset. The scaling factor σ is then calculated such that, after applying the Gaussian kernel, d_{max} is transformed into a chosen similarity s_{min}. This ensures that the resulting graph is safely connected. By optimizing this similarity s_{min} we can accurately represent the neighborhood relationships.

Therefore, for a given value of s_{min}, we derive σ from the length of the longest edge d_{max} in the MST:

$$\begin{aligned} \exp\left(-d_{max}^2/(2\sigma^2)\right) &= s_{min} \\ \Leftrightarrow \sigma &= d_{max}/\sqrt{-2\ln\left(s_{min}\right)} \end{aligned} \tag{1}$$

Optimizing s_{min} instead of σ should be more robust to variations in the distribution of inter-point distances and give better results across different datasets or parts of a dataset.

To incorporate the knowledge from the known classes in the Spectral Clustering process, our initial approach was to utilize NCD k-means within the spectral embedding. In short, we would initially compute the full spectral embedding for all the data and determine the mean points of the known classes with the help of the ground truth labels. These mean points would then serve as the initial centroids. However, the observed performance improvement over the fully unsupervised SC was quite marginal.

Instead, the idea that we will use throughout this article (and that we refer to as "NCD Spectral Clustering") stems from the observation that SC can obtain very different results according to the parameters that are used. Among these parameters, the temperature parameter σ of the kernel holds particular importance, as it directly impacts the adjacency matrix's accuracy in representing the neighborhood relationships of the data points. The rule of thumb of Eq. 1 still requires to choose a value, but significantly reduces the space of possible values. Additionally, while the literature often sets the number of components u of the spectral embedding equal to the number of clusters, we have observed that optimizing it can also improve performance.

Therefore, rather than a specific method, we propose the parameter optimization scheme illustrated in Fig. 2.

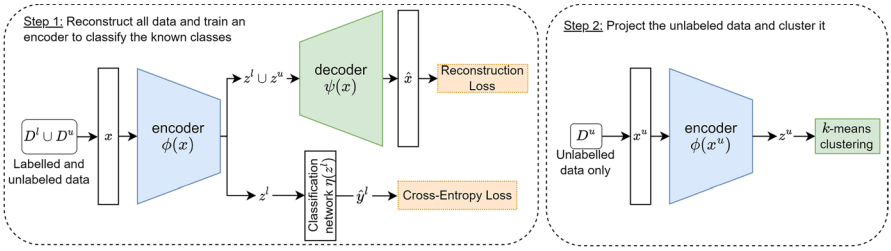


Fig. 3 Architecture of the PBN model

For a given combination of parameters $\{s_{min}, u\}$, the corresponding spectral embedding of all the data is computed and then partitioned with k-means. The quality of the parameters is evaluated from the clustering performance on the known classes, as the ground-truth labels are only available for the known classes. Indeed an important hypothesis behind the NCD setup is that the known and novel classes are related and share similarities, so they should have similar feature scales and distributions. Consequently, if the Spectral Clustering performs well on the known classes, the parameters are likely suitable to represent the novel classes. The pseudocode of this approach can be found in Algorithm 3 of "Appendix A".

Discussion. This idea can be applied to optimize the parameters of any unsupervised clustering algorithm in the NCD context. For example, the *Eps* and *MinPts* parameters of DBSCAN (Ester et al. 1996) can be selected in the same manner. It is also possible to use a different adjacency matrix in the SC algorithm. One option could be to substitute the Gaussian kernel with the k-nearest neighbor graph, and therefore optimise k instead of σ. However, for the sake of simplicity, we will only investigate SC using the Gaussian kernel.

3.4 Projection-based NCD

Projection-based NCD (PBN) can be seen as an extension of the baseline method used in TabularNCD (Troisemaine et al. 2022). PBN is illustrated in Fig. 3 and consists of 3 key components: (1) an encoder that learns a shared representation between the known and novel classes; (2) a classification network trained to distinguish the known classes of D^l in order to incorporate their relevant features into the representation; and (3) a decoder that reconstructs the data for both known and novel classes of $D^l \cup D^u$, ensuring that the latent space contains the information necessary to represent all classes. The decoder serves a dual purpose: it provides regularization and mitigates overfitting on the known classes, thus improving generalization, as shown in Le et al. (2018).

The training loss is defined as:

$$\mathcal{L}_{PBN} = w \times \mathcal{L}_{CE} + (1 - w) \times \mathcal{L}_{MSE} \tag{2}$$

where $w \in (0, 1)$ is a trade-off parameter that allows to balance the strength of the cross-entropy loss and the reconstruction loss.

The cross-entropy loss on the known classes is defined as:

$$\mathcal{L}_{CE} = -\sum_{c=1}^{C^l} y_c \log\left(\eta_c(z)\right) \tag{3}$$

where $\eta(z) = \left(\eta_c(z)\right)_{c=1}^{C^l}$ is the output of a classification network composed of a single dense layer of neurons, $z = \phi(x)$ is the projection of instance x through the encoder ϕ and $\left(y_c\right)_{c=1}^{C^l}$ is the one-hot encoded ground-truth label of instance x.

The reconstruction loss of the instances from all classes is written as:

$$\mathcal{L}_{MSE} = \frac{1}{d}\sum_{j=1}^{d}\left(x_j - \hat{x}_j\right)^2 \tag{4}$$

where $\hat{x} = \psi(z)$ is the reconstruction of instance $x \in \mathbb{R}^d$.

Once the encoder, decoder and classification network have been trained (step 1), unlabeled data D^u is projected by the trained encoder into the latent space and then clustered with k-means to discover novel classes (step 2).

Projection-based NCD requires tuning of four hyperparameters. The trade-off parameter w is inherent to the method and the other three come from the choice of architecture: the learning rate, the dropout rate and the size of the latent space.

Note that this method doesn't employ complex schemes to define pseudo-labels unlike many NCD works. They have been proven to be accurate with image data (notably thanks to data augmentation techniques) (Han et al. 2021; Zhao and Han 2021), but we found in preliminary results not detailed here, that for tabular data, they introduce variability in the results and new hyperparameters that need to be tuned.

Discussion. Similarly to PBN, the baseline method of TabularNCD (Troisemaine et al. 2022) relies on the assumption that known and novel classes share similar high-level features, and defines a latent space that highlights these features. This baseline first trains a deep classifier to distinguish only the known classes of D^l. After training, the output and softmax layers are discarded, and the last hidden layer is now considered as the output of an encoder. It then projects the novel data of D^u into this latent space and partitions it using k-means. This is the basic workflow of two-stage latent space-based NCD methods identified in Troisemaine et al. (2023). It is also similar to DTC (Han et al. 2019), which uses the more refined DEC (Xie et al. 2016) clustering model instead of k-means in the baseline. The problem with such two-stage methods is that the resulting representations are at risk of being heavily biased towards the known classes. Thus, if some concepts or high-level features are not shared between the known and novel classes, the novel classes will not be well represented and these approaches will fail.

3.5 Summary of proposed approaches

In this section, we have proposed 3 distinct methods for solving the NCD problem, all of which leverage knowledge from the known classes in different ways. Firstly,

NCD k-means uses the labeled data to improve the initialization of its centroids. Secondly, instead of using the labels of the known classes during the clustering process itself, NCD Spectral Clustering uses them to find parameters that are likely to be suitable for the whole domain. More precisely, by clustering $D^l \cup D^u$ together, the adequacy of the parameters s_{min} and u can be evaluated on the known classes. Finally, PBN is a straightforward method that includes only the essential components to define a latent representation suitable for clustering the novel classes. In this case, an encoder is trained with a classification loss on the known classes and a reconstruction loss on all the data to ensure that the novel classes are not misrepresented. The novel data is then projected into this representation and clustered with k-means.

In the next section, we present an approach to finding hyperparameters without using the labels of the novel classes, which are not available in realistic scenarios. Indeed in the experiments (see Sect. 7), it should become clear why the simplicity of the proposed approach is a desirable feature for hyperparameter optimization in the NCD context.

4 Hyperparameter optimization

The success of machine learning algorithms (including NCD) can be attributed in part to the high flexibility induced by their hyperparameters. In most cases, a target is available and approaches such as the k-fold Cross-Validation (CV) can be employed to tune the hyperparameters and achieve optimal results. However, in a realistic scenario of Novel Class Discovery, the labels of the novel classes are never available. We must therefore find a way to optimize hyperparameters without ever relying on the labels of the novel classes. In this section, we present a method that leverages the known classes to find hyperparameters applicable to the novel classes. This tuning method is designed specifically for NCD algorithms that require both labeled data (known classes) and unlabeled data (novel classes) during training[1]. This is the case for Projection-based NCD, as described in Sect. 3.4.

The process that we devised is represented in Fig. 4. For each of the splits, the instances of around half of the known classes are selected to form the set D^{hid} and their labels are hidden. The labeled set now becomes the instances of $D^l \setminus D^{hid}$ and the unlabeled set becomes the instances of $D^u \cup D^{hid}$. After training the model with this new data split, it is evaluated for its performance for partitioning the instances of D^{hid} only since their labels are available.

To illustrate, in the split 1 of Fig. 4, the model will be trained with the subsets of classes $\{C_2, C_3, C_4\}$ as known classes and $\{C_0, C_1, C_5, \ldots, C_9\}$ as novel classes. It will be evaluated for its performance on the hidden classes $\{C_0, C_1\}$ only.

To evaluate a given combination of hyperparameters, this approach is applied to all the splits, and the performance on the hidden classes is averaged. After repeating

[1] To optimize purely unsupervised clustering methods for NCD, we refer the reader to the optimization process of Sect. 3.3.

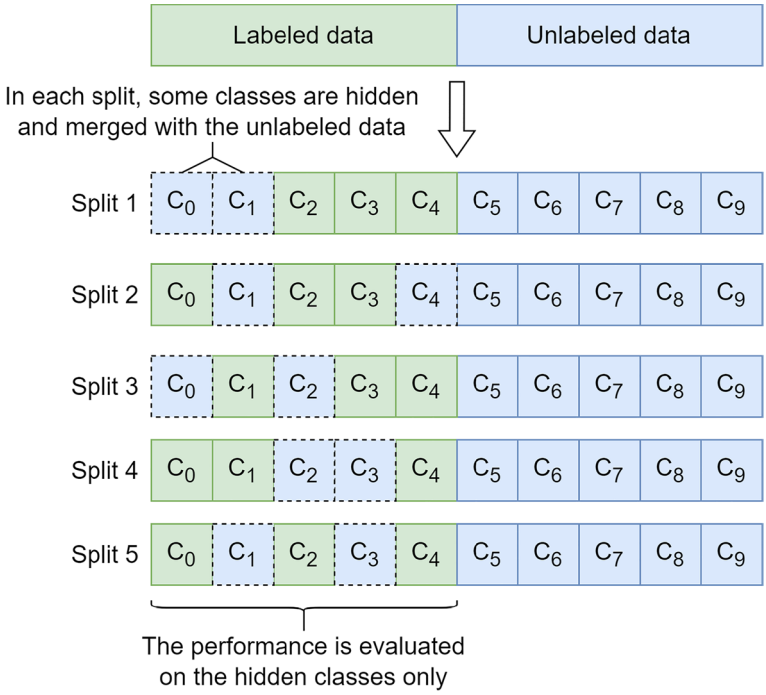


Fig. 4 The *k*-fold cross-validation approach for hyperparameter optimisation of NCD methods

Table 1 Classes splits of the *k*-fold cross-validation

Dataset	Known classes	Novel classes	Hidden classes	Splits
Human	3	3	2	3
Letter	19	7	7	5
Pendigits	5	5	2	5
Census	12	6	6	5
m feat	5	5	2	5
Optdigits	5	5	2	5
CNAE-9	4	5	2	5

this process for many combinations, the combination that achieved the best performance is selected. For the final evaluation on the novel classes, in a realistic scenario of NCD their labels are never available. However, in the datasets employed in this article, the novel classes are comprised of pre-defined classes. Therefore, even though these labels are not employed during training, they can still be used to assess the final performance on the novel classes of different models and compare them against each other.

This tuning method stems from the same idea behind the NCD Spectral Clustering parameterization process. Namely, if the clustering in the learned representation successfully partitions the hidden classes in D^{hid}, it is also likely suitable for the novel classes in D^u. Furthermore, keeping the unlabeled data during training even though the model is not evaluated on the novel classes is important, as it increases the chances of the representation being adapted for all the classes. For the same reason, the k-means of PBN (see Sect. 3.4) is fitted on $D^u \cup D^{hid}$ together (instead of just D^{hid}) and the performance is then computed on D^{hid} only. So cases where the classes in D^u and D^{hid} are tangled will be penalized.

In Table 1, we report for all datasets used in our experiments the number of known classes that are hidden in each split, as well as the number of splits. Note that when the number of known classes is small (e.g. 3 for *Human*), this approach may be difficult to apply.

Discussion. Similarly to NCD, there are no labels available in unsupervised clustering problems, which makes the task of hyperparameter selection very difficult. To address this issue, clustering algorithms are sometimes tuned using *internal* metrics that do not rely on labeled data for computation. These metrics offer a means of comparing the results obtained from different clustering approaches. Examples of such metrics include the Silhouette coefficient, Davies-Bouldin index, or Calinski-Harabasz index (Arbelaitz et al. 2013). However, it is important to note that these metrics make assumptions about the structure of the data and can be biased towards algorithms which make a similar assumption. But unlike unsupervised clustering, the NCD setting provides known classes that are related to the novel classes we are trying to cluster.

5 Estimating the number of novel classes

Cluster Validity Indices (CVIs) are commonly used in unsupervised data analysis to estimate the number of clusters and are also applicable to the NCD problem. CVIs are scores that compare the compactness and separation of clusters without the help of external information such as ground truth labels. However, the knowledge from the known classes isn't used if the CVIs are directly applied to estimate the number of novel classes. Therefore, we propose to apply the CVIs in the latent representation learned by PBN. Projection-based NCD methods such as PBN are designed to create a latent space that emphasizes the relevant features of the known classes. Since these features are shared to some extent with the novel classes, this representation should be better at revealing the clusters we are trying to discover than the original feature space. Consequently, it makes sense that applying the different estimation techniques in the learned latent space should yield better results.

Note that this is only applicable to NCD methods such as PBN that don't require the number of novel classes C^u to train their latent space (unlike TabularNCD). For the others, the estimation can be done once in the original feature space, but should have higher error.

Some NCD works have also previously attempted to estimate the number of novel classes. For instance, Hsu et al. (2018) defines a large number of output neurons in their clustering network (e.g. 100). In this case, the clustering network is expected to use only the necessary number of clusters while leaving the remaining output neurons unused. Clusters were counted if they contained more instances than a certain threshold. However, since, with the exception of TabularNCD, the models studied in this paper do not use a clustering network, we will not evaluate this method.

Another technique, proposed by Vaze et al. (2022), consists in training a k-means on the combined dataset $D^l \cup D^u$ and selecting the k that yielded the highest accuracy on D^l. While this approach worked well for balanced datasets (Fei et al. 2022), it has been shown to underperform in the case of unbalanced class distributions (Yang et al. 2023). For the sake of simplicity, we will call this method KM-ACC (for k-means ACC) in the remainder of this paper.

To select the CVI that we will use for our application, we rely on the results of Arbelaitz et al. (2013). Here, the authors conducted an extensive performance evaluation of 30 CVIs. They concluded that the Silhouette, Davies-Bouldin, Calinski-Harabasz and Dunn indices behaved better than other indices in almost all cases. In the experiments, the performance of these 4 indices will be compared, with the addition of the elbow method and the NCD-specific method KM-ACC.

6 Full training procedure

In the previous sections, we presented the models, the hyperparameter optimization and the estimation procedure of the number of novel classes independently. In this section, these components are brought together to form a complete training procedure. To ensure that no prior knowledge about the novel classes is ever used in this process, the number of novel classes is naturally estimated during the k-fold CV introduced in Sect. 4. As the whole process is quite complex, we try to summarize it in clear terms in this section and in Algorithm 1.

To gauge a given set of hyperparameters, we evaluate the performance of the model over n_{folds}, where in each fold, a random combination of known classes is "hidden" and merged with the unlabeled data of D^u. In a fold, the encoder of the NCD model is first trained on this new data split. The number of novel classes is then estimated with a CVI in the projection of the unlabeled data. At this point, the novel and hidden classes are partitioned by the model in the latent space using the previous estimate of the number of clusters. And the accuracy for this fold is calculated on the hidden classes. This process is repeated for all folds and for many combinations of hyperparameters, and the combination that achieved the best performance on average is selected for the final evaluation of the model.

Table 2 Details of the datasets

Dataset	Human	Letter	Pendigits	Census	m feat	Optdigits	CNAE-9
Features	562	16	16	67	515	62	856
Known classes	3	19	5	12	5	5	4
Known data	3733	10,229	3777	12,000	802	1918	377
Novel classes	3	7	5	6	5	5	5
Novel data	3619	3770	3717	6000	798	1905	487
Test data	1453	1704	1734	6000	202	905	113

Algorithm 1 Agnostic NCD model evaluation

Require: Training data $\{D^l, D^u\}$, hyperparameters θ, number of classes to hide n_{hid}, number of folds n_{folds}

Initialize: $folds \leftarrow$ set of n_{folds} random combinations of n_{hid} known classes
for each $fold$ **in** $folds$ **do**
  $D^{hid} \leftarrow$ the data from D^l of the classes in $fold$
  $D^{l'} \leftarrow D^l \setminus D^{hid}$
  $D^{u'} \leftarrow D^u \cup D^{hid}$
  Train model on $\{D^{l'}, D^{u'}\}$ with hyperparameters θ
  $Z^u \leftarrow \phi_\theta(X^u)$ the projection of the novel data
  $k' \leftarrow$ the estimation of C^u in Z^u with a CVI
  Get the clustering prediction of the model for $D^{u'}$ using $k = n_{hid} + k'$
  $ACC_{hid} \leftarrow$ clustering performance on D^{hid}
end for
Return: Average of all the ACC_{hid}

7 Experiments

7.1 Experimental setup

Datasets. To evaluate the performance of the methods compared in this paper, 7 tabular classification datasets were selected: Human Activity Recognition (Anguita et al. 2013), Letter Recognition (Frey and Slate 2005), Pen-Based Handwritten Digits (Dua and Graff 2017), 1990 US Census Data (Dua and Graff 2017), Multiple Features (Dua and Graff 2017), Handwritten Digits (Dua and Graff 2017) and CNAE-9 (Dua and Graff 2017).

Following the previous NCD works (Zhong et al. 2021; Han et al. 2019, 2021), the instances of about 50% of the classes are hidden a priori to form the unlabeled set of novel classes D^u, while the rest form the labeled set D^l. We use a 70/30% train/test split if it was not already provided. Statistical information on the datasets is shown in Table 2, and the number of known/novel classes (along with the

number of classes hidden during the k-fold CV) can be found in Table 1. The numerical features of all the datasets are pre-processed to have zero mean and unit variance, while the categorical features are one-hot encoded.

Metrics. We report the *clustering accuracy* (ACC) on the unlabeled data. It is defined as:

$$ACC = \max_{perm \in P} \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}\left\{y_i = perm(\hat{y}_i\right\} \tag{5}$$

where y_i and $\hat{y}_i$ are the ground truth labels and predicted labels for instance x_i respectively. Here, $M = |D^u|$. P is the set of all possible permutations between ground truth and predicted labels. It can be easily computed using the Hungarian algorithm (Kuhn and Yaw 1955). The Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) are also reported in "Appendix B".

Competitors. We report the performance of k-means and Spectral Clustering, along with their NCD-adapted versions introduced in Sects. 3.2 and 3.3. We also include PBN and the pioneering NCD work for tabular data, TabularNCD (Troisemaine et al. 2022). Finally, we implement the same baseline used in Troisemaine et al. (2022). It is a simple deep classifier that is first trained to distinguish the known classes. Then, the penultimate layer is used to project the data of the novel classes before clustering it with k-means. See the discussion of Sect. 3.4 for more details. We will call this approach the "baseline" for the remainder of this article.

Implementation details. The neural-network based methods (i.e. PBN, TabularNCD and the baseline) are trained with the same architecture: an encoder of 2 hidden layers of decreasing size, and a final layer for the latent space whose dimension is optimized as an hyperparameter. The dropout probability and learning rate are also hyperparameters to be optimized. The classification networks are all a single linear layer followed by a softmax layer. All methods are trained for 200 epochs and with a fixed batch size of 512. For a fair comparison, the hyperparameters of these neural-network based methods are optimized following the process described in Sect. 4 (whereas the parameters of NCD SC are simply optimized following Sect. 3.3). Thus, the labels of the novel classes are never used except for the computation of the evaluation metrics reported in the result tables. The hyperparameters of the deep-based methods are tuned by a random search of the hyperparameter space and selecting the combination which obtained the best ARI on the hidden classes. The search space and selected values can be found in "Appendix D".

Objectives of the experiments. In the following section, an extensive evaluation of the clustering accuracy of the 7 competitors is performed. This evaluation is carried out in two steps. First, when the number of novel classes is known in advance, we seek to determine whether (1) the NCD-adapted versions of k-means and Spectral Clustering outperform their purely unsupervised versions when labeled data is available, and (2) which of the deep-based NCD methods performs best when the labels of the novel classes are not available for hyperparameter tuning. And second, in the most realistic scenario where the number of classes is not given in advance,

Table 3 Test ACC of the clustering algorithms averaged over 10 runs

Dataset	*k*-means	NCD *k*-means	SC	NCD SC
Human	75.7 ± 0.2	75.9 ± 0.0	76.3 ± 0.3	**93.1 ± 9.7**
Letter	50.7 ± 0.2	51.9 ± 2.3	55.9 ± 0.0	**57.4 ± 5.8**
Pendigits	81.7 ± 0.0	81.7 ± 0.0	**83.0 ± 0.0**	81.7 ± 2.7
Census	49.9 ± 4.0	**50.4 ± 1.1**	48.5 ± 0.3	48.0 ± 1.8
m feat	89.1 ± 0.3	**89.7 ± 0.4**	89.6 ± 0.3	89.2 ± 2.3
Optdigits	79.1 ± 4.5	94.2 ± 0.0	89.7 ± 0.0	**95.4 ± 5.3**
CNAE-9	60.6 ± 5.9	61.2 ± 4.5	53.8 ± 4.8	**69.0 ± 6.7**
Average	69.5	72.1	71.0	**76.3**

Best scores are in bold

Table 4 Test ACC of the NCD methods averaged over 10 runs

Dataset	Baseline	NCD SC	TabularNCD	PBN
Human	71.6 ± 1.7	**93.1 ± 9.7**	72.2 ± 2.6	76.7 ± 1.8
Letter	**64.9 ± 2.6**	57.4 ± 5.8	62.1 ± 3.0	62.4 ± 2.0
Pendigits	53.4 ± 6.6	81.7 ± 2.3	57.0 ± 6.0	**82.8 ± 0.6**
Census	59.1 ± 0.8	48.0 ± 1.8	45.2 ± 4.8	**62.4 ± 0.9**
m feat	66.7 ± 4.1	89.2 ± 2.3	90.2 ± 2.7	**91.7 ± 0.8**
Optdigits	40.7 ± 5.1	**95.4 ± 5.3**	73.0 ± 8.4	92.6 ± 2.3
CNAE-9	40.2 ± 3.2	69.0 ± 6.7	51.3 ± 5.2	**72.6 ± 4.6**
Average	55.7	76.3	64.4	**77.3**

Best scores are in bold

we compare the ability of different CVIs to accurately estimate this number and the impact on the NCD methods of using this estimation.

7.2 Results analysis

7.2.1 Results when the number of novel classes is known in advance

Clustering. In Table 3, we first examine the performance of the unsupervised clustering methods when the number of novel classes C^u is known in advance. The aim is to determine which of the clustering algorithms performs best and should be compared with the NCD methods. We observe that NCD *k*-means is never worse than *k*-means, and NCD SC is only once worse than SC. This result confirms the efficacy of both NCD approaches and demonstrates that even simple clustering techniques can benefit from the known classes, although the improvements are sometimes only marginal.

The comparison between NCD *k*-means and NCD SC confirms the idea that no single clustering algorithm is universally better than the others in all scenarios, as noted by Von Luxburg et al. (2012). However, NCD SC outperforms its competitors

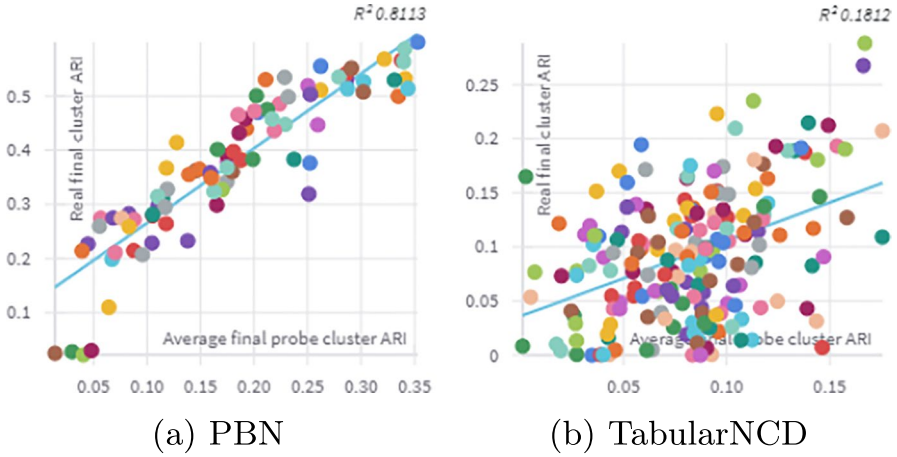


Fig. 5 Comparison between the ARI on the hidden and novel classes. Each point is a different hyperparameter combination

on 4 occasions and has a the highest average accuracy. Therefore, this algorithm is selected for the next step of comparisons.

NCD. As shown in Table 4, PBN outperforms both the baseline and TabularNCD by an average of 21.6% and 12.9%, respectively. It is only outperformed by the baseline on the *Letter Recognition* (Frey and Slate 2005) dataset. This dataset consists of primitive numeric attributes describing the 26 capital letters in the English alphabet, which suggests a high feature correlation between the features used to distinguish the known and novel classes. Since the baseline learns a latent space that is strongly discriminative for the known classes, this gives the baseline model a distinct advantage in this specific context. On the other hand, we observe that it is at a disadvantage when the datasets do not share as many high-level features between the known and novel classes.

Table 4 also demonstrates the remarkable competitiveness of the NCD Spectral Clustering method, despite its low complexity. On average, it trails behind PBN by only 1.0% in ACC and manages to outperform PBN twice over 7 datasets.

To investigate the reasons behind the subpar performance of TabularNCD, we look at the correlation between the ARI on the hidden classes and the final ARI of the model on the novel classes. A strong correlation would imply that if a combination of hyperparameters performed well on the hidden classes, it would also perform well on the novel classes. To examine this, we plot the average ARI on the hidden classes against the ARI on the novel classes. Figure 5 is an example of such a plot. It shows that, in the case of the *Letter Recognition* dataset, PBN has a much stronger correlation than TabularNCD. We attribute this difference to the large number of hyperparameters of TabularNCD (7, against 4 for PBN), which causes the method to overfit on the hidden classes, resulting in a lack of effective transfer of hyperparameters to the novel classes.

In conclusion, this section has shown that when the number C^u of novel classes is known, NCD SC performs almost as well as PBN. Therefore, in this specific

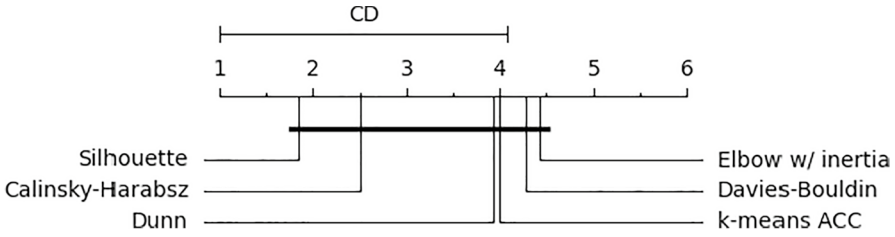


Fig. 6 Comparison of the CVIs in the latent space of PBN using the Nemenyi test with a 95% confidence interval

Table 5 Test ACC averaged over 10 runs. With C^u estimated with the Silhouette coefficient

Dataset	Baseline	NCD SC	TabularNCD	PBN
Human	70.8 ± 2.9	30.2 ± 4.2	**71.1 ± 0.0**	**71.1 ± 0.0**
Letter	**64.0 ± 6.1**	34.8 ± 2.3	41.8 ± 4.9	61.3 ± 4.7
Pendigits	46.7 ± 3.6	74.1 ± 2.1	57.0 ± 6.0	**83.0 ± 0.3**
Census	**56.6 ± 3.6**	29.0 ± 3.5	35.7 ± 1.6	49.8 ± 0.1
m feat	59.5 ± 7.7	73.2 ± 3.9	41.1 ± 0.2	**90.6 ± 2.2**
Optdigits	42.1 ± 4.6	79.5 ± 4.0	**96.9 ± 1.3**	90.5 ± 4.8
CNAE-9	33.8 ± 3.8	44.6 ± 3.9	39.3 ± 0.2	**50.8 ± 1.5**
Average	53.4	52.2	54.7	**71.0**

Best scores are in bold

scenario, NCD SC is a viable candidate for addressing the NCD problem due to its lower complexity and shorter training time. Conversely, despite its strong learning capacity, TabularNCD is penalized by its high number of hyperparameters. Finally, the performance of the methods as the number of novel classes increases is studied in "Appendix C". It shows that PBN is robust over a wide range of values, and that deep-based methods tend to fail when the number of known classes is too small.

7.2.2 Results when the number of novel classes is estimated

As expressed in Sect. 5, we leverage the representation learned by PBN during the k-fold CV to estimate the number of clusters. The result is a method that never relies on any kind of knowledge from the novel classes. This approach is also applicable to the baseline and the spectral embedding of SC, but not to TabularNCD as it requires a number of clusters to be defined during the training of its representation. For a fair comparison, TabularNCD is trained here with a number of clusters that was estimated beforehand with a CVI.

To determine which CVI will perform the best in this application, we estimate the number of classes in the latent spaces learned by PBN when C^u was known in advance. Figure 6 displays the average ranks of the CVIs in the latent spaces, and

the details of the results can be found in "Appendix F". The Nemenyi post-hoc test was used to compare the methods against all others. However, given the relatively small number of datasets, the Critical Difference (CD) is large and the CVIs are not statistically different from each another according to the Nemenyi test.

Nevertheless, we find that the Silhouette coefficient performed the best in the latent space of PBN, closely followed by the Calinski-Harabasz index. The elbow method is ranked last, which could be explained by the difficulty in defining an elbow[2].

To summarize, in the following results, the Silhouette coefficient will be used for all estimations of C^u. The full training procedure described in Sect. 6 will be used to train the baseline and PBN, with C^u estimated in their latent spaces as detailed in Algorithm 1. For TabularNCD, C^u is estimated once in the original feature space (see "Appendix F" Table 10 for the values used). And NCD SC is trained as it was described in Sect. 3.3, but with C^u estimated in its spectral embedding.

As emphasised earlier, Table 5 reports the results of the different NCD algorithms in the most realistic scenario possible, where both the labels and the number of novel classes are not known in advance. This is the setting where PBN exhibits the greatest improvement in performance compared to the other competitors, achieving an ACC that is 17.6%, 18.8% and 16.3% higher than the baseline, NCD SC, and TabularNCD, respectively.

Remarkably, TabularNCD outperforms PBN on the Optdigits datasets where the number of clusters was overestimated by the Silhouette coefficient in the original feature space. This suggests that TabularNCD probably only utilized the output neurons necessary for clustering, leaving the others unused, which was the method proposed in Hsu et al. (2018) for estimating C^u. This, however, is not true for the Letter dataset where C^u was significantly overestimated, indicating that accurate estimations will likely result in improved performance.

Compared to the case where C^u is known in advance, the ACC of the baseline falls from 55.7% to 53.4% and NCD SC falls from 74.3% to 52.2%. This shows that they are both unable to find a latent space suitable for the estimation of C^u.

However, the ACC of PBN remains an impressive 71.0%, demonstrating that this simple method, comprising of only two loss terms, is the most appropriate for tackling the NCD problem in a realistic scenario. In contrast to the baseline and NCD SC methods, its reconstruction term enables it find a latent space where the unlabeled data are correctly represented. And unlike TabularNCD, it has a low number of hyperparameters which decreases the probability of overfitting on the hidden classes during the k-fold CV procedure.

[2] There are no widely accepted approaches, as the concept of an "elbow" is subjective. In this study, we employed the *kneedle* algorithm (Satopaa et al. 2011) through the *kneed* Python Library (Arvai 2023).

8 Conclusion

In this article, we have shown that in the NCD setting, unsupervised clustering algorithms can benefit from knowledge of the known classes and reliably improve their performance by implementing simple modifications. We have also introduced a novel NCD algorithm called PBN, which is characterized by its simplicity and low number of hyperparameters, which proved to be a decisive advantage under realistic conditions. In addition, we have proposed an adaptation of the k-fold cross-validation process to tune the hyperparameters of NCD methods without depending on the labels of the novel classes. Finally, we have demonstrated that the number of novel classes can be accurately estimated within the latent space of PBN. These two previous contributions have shown that the NCD problem can be solved in realistic situations where no prior knowledge of the novel classes is available during training.

Appendix A: Pseudocode of the methodologies

In this section, we provide the pseudocode of the NCD k-means and NCD Spectral Clustering methods proposed in Sections 3.2 and 3.3, respectively. With this addition, we hope to improve the clarity of our research and make it easier to understand the available open-source code.

Algorithm 2 NCD k-means

Require: Labeled set of known classes $D^l = \{X^l, Y^l\}$, unlabeled set of novel classes $D^u = \{X^u\}$, number of clusters k.

Initialize: L^l and L^u empty lists of centroids for the known and novel classes

Phase 1 - Initialization of the centroids

// Initialize the centroids of known classes using the ground truth labels:
for each class c in D^l **do**
    Let $X^l_c = \{x^l_i \mid x^l_i \in X^l, y^l_i = c\}$ be the subset of points belonging to class c
    Calculate $\mu^l_c = \frac{1}{|X^l_c|} \sum_{x^l_i \in X^l_c} x^l_i$ the average point of class c
    Add μ^l_c to the list L^l of centroids of the known classes
end for
// Initialize the centroids of the novel classes in L^u following k-means++:
while $|L^u| < k$ **do**
    For each point x^u_i in D^u, calculate the distance $d(x^u_i)$ between x^u_i and its nearest centroid in $L^l \cup L^u$
    Choose the next centroid from the points in D^u with probability proportional to $d(x^u_i)^2$
    Add the chosen centroid to L^u
end while

Phase 2 - Follow the usual k-means algorithm

Make the centroids of L^u converge, using only the data of D^u
Partition the points in D^u using only the centroids of the novel classes in L^u

Algorithm 3 NCD Spectral Clustering

Require: Labeled set of known classes $D^l = \{X^l, Y^l\}$, unlabeled set of novel classes $D^u = \{X^u\}$, number of clusters k, number of hyperparameter optimization runs n_{opt}.

Phase 1 - Determination of the optimal hyperparameters

Initialize $ARI_{opt} = 0$ the best ARI obtained and $\{u_{opt} = 0, \sigma_{opt} = 0\}$ the optimal hyperparameters
for n_{opt} random combinations of hyperparameters $\{u_{rand}, \sigma_{rand}\}$ **do**
    Compute the adjacency matrix A of the points in $D^l \cup D^u$, where $A_{i,j} = \exp(-\|x_i - x_j\|_2^2/(2\sigma_{rand}^2))$
    Define the spectral embedding U as the first u_{rand} eigenvectors of the symmetric normalized Laplacian of A
    Partition all the points in U using k-means into $k + C^l$ classes
    Calculate ARI_{rand}, the clustering ARI of this k-mean only on the points of D^l, using the ground truth labels Y^l
    if $ARI_{rand} > ARI_{opt}$ **then**
        $ARI_{opt} \leftarrow ARI_{rand}$, $u_{opt} \leftarrow u_{rand}$ and $\sigma_{opt} \leftarrow \sigma_{rand}$
    end if
end for

Phase 2 - Clustering of the data of the novel classes

Get U_{opt} the spectral embedding of $D^l \cup D^u$ with the parameters $\{u_{opt}, \sigma_{opt}\}$
Partition the points of D^u embedded in U_{opt} into k clusters with k-means

Appendix B: Additional result metrics

In addition to the ACC results discussed in Sect. 7.2.2, we present the NMI (Table 6) and ARI (Table 7) for the NCD methods when the number of novel classes C^u is unknown and has to be estimated. Our results are consistent with those shown in Table 5: the PBN method largely outperforms its competitors in this realistic scenario. In particular, PBN achieves an average NMI higher than the baseline, NCD SC and TabularNCD by 23.5%, 8.6% and 13.5% respectively. Similarly, the ARI is 24.5%, 12.4% and 17.5% higher on average.

Table 6 Test NMI averaged over 10 runs. With C^u estimated with the Silhouette coefficient

Dataset	Baseline	NCD SC	TabularNCD	PBN
Human	56.1 ± 8.9	45.8 ± 3.0	**75.2 ± 0.0**	**75.2 ± 0.0**
Letter	54.4 ± 3.3	52.6 ± 1.0	37.2 ± 3.2	**59.2 ± 2.4**
Pendigits	41.0 ± 6.7	**79.5 ± 1.4**	48.0 ± 6.5	73.4 ± 2.5
Census	59.7 ± 0.6	33.5 ± 1.0	42.6 ± 1.5	**60.6 ± 0.5**
m feat	50.0 ± 3.9	71.9 ± 2.9	44.8 ± 2.1	**79.1 ± 3.3**
Optdigits	28.5 ± 6.0	77.3 ± 2.0	**93.1 ± 2.0**	84.9 ± 2.0
CNAE-9	23.0 ± 1.5	**56.5 ± 2.0**	42.0 ± 1.3	45.2 ± 13.4
Average	44.7	59.6	54.7	**68.2**

Best scores are in bold

Table 7 Test ARI averaged over 10 runs. With C^u estimated with the Silhouette coefficient

Dataset	Baseline	NCD SC	TabularNCD	PBN
Human	48.7 ± 4.0	20.3 ± 3.0	**61.4 ± 0.0**	**61.4 ± 0.0**
Letter	44.3 ± 4.1	29.5 ± 2.2	23.4 ± 5.6	**48.9 ± 3.1**
Pendigits	29.8 ± 5.7	**73.7 ± 1.7**	37.6 ± 6.5	65.3 ± 3.4
Census	**42.6 ± 3.7**	23.0 ± 3.8	26.7 ± 0.8	35.5 ± 0.2
m feat	40.9 ± 5.3	64.1 ± 4.5	21.5 ± 0.2	**79.1 ± 4.2**
Optdigits	16.9 ± 6.5	75.6 ± 3.6	**94.1 ± 2.7**	84.4 ± 4.4
CNAE-9	11.6 ± 1.9	**33.2 ± 4.0**	18.8 ± 0.4	31.5 ± 1.7
Average	33.5	45.6	40.5	**58.0**

Best scores are in bold

Appendix C: Impact of the ratio of novel classes

In the literature on Novel Class Discovery, the ratio of the number of novel classes to the number of known classes is usually set arbitrarily (e.g. 0.5 if there are few classes, and 0.2 if there are many). In reality, this ratio is not known in advance, so NCD methods should ideally be robust over a wide range of ratios. To compare the robustness of the 5 methods discussed in this paper, we perform experiments on the *Letter Recognition* (Frey and Slate 2005) dataset when the number of novel classes increases. For each number of novel classes that was evaluated, we defined 5 random combinations of known/novel classes. For example, in Fig. 7, the second point from the left corresponds to the average performance of the methods over 5 random combinations of 3 novel and 23 known classes.

From the results displayed in Fig. 7, we find that the performance of the methods is almost always uniformly decreasing as the ratio of novel classes increases. It can also be observed that the accuracy of the deep-based methods (PBN, TabularNCD and the baseline) start decreasing faster than NCD k-means and NCD SC as the number of novel classes becomes too large. As the number of known classes C^l decreases, the quality of their latent representation is reduced and their clustering is negatively affected. Of course, this is especially true for the baseline, which trains its latent space solely on the known classes.

The performance of TabularNCD peaks around 7 novel classes, which is the number of novel classes for which it was specifically optimized. On the other hand, PBN is more stable and outperforms the competitors over a wide range of values. Its small number of hyperparameters, coupled with its reconstruction loss, makes it less prone to overfitting, and it still extracts relevant information into its latent space even when there are few known classes.

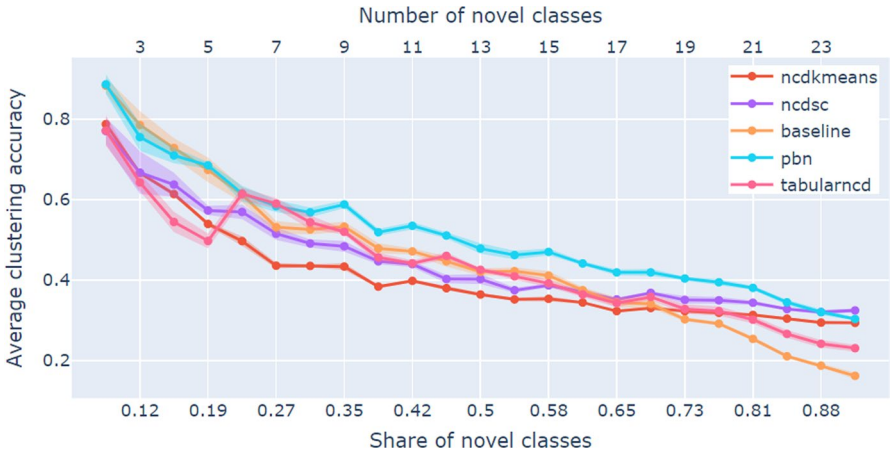


Fig. 7 Comparison of the performance of the 5 methods discussed in this paper when the number of novel classes increases

Appendix D: Hyperparameters details

The Table 8 shows the hyperparameters found by the full procedure described in Sect. 6. The type of distributions and value spaces explored for each hyperparameter are given in the "Range" and "Distribution" columns. The variable d in the range of the *latent dim* refers to the dimension of the points in the datasets.

Appendix E: PBN coupled with Spectral Clustering

In Sect. 3.3, we proposed the Projection Based NCD (PBN) method where a latent representation is first trained with a reconstruction loss on all the points, and a classification loss on the labeled data. After training of the representation, the simple k-means algorithm is used to cluster the unlabeled data of the novel classes. It was chosen mainly for its fast execution time, as the idea behind PBN is to obtain a representation where the novel classes are already well separated, so a sophisticated clustering algorithm should not be needed. But naturally, any other unsupervised clustering method can be employed and might improve the performance. Therefore, in this section, we investigate the performance of PBN when k-means is replaced with Spectral Clustering (SC).

In these experiments, the number of novel classes C^u is considered to be known in advance. In Table 9, three variations of the PBN method are compared:

- *KM PBN:* The model used in the article, where k-means is used in the latent space.

Table 8 Hyperparameters of the methods found when C^u is estimated

Parameter		Range	Distribution	Human	Letter	Pendigits	Census	m features	Optdigits	CNAE-9
Base-line	Latent dim	[5, d]	Int uniform	560	9	9	30	34	42	569
	lr	[0.0001, 0.1]	Log uniform	0.000154	0.001333	0.005517	0.002021	0.000118	0.003330	0.000167
	Dropout	[0.0, 0.6]	Uniform	0.168044	0.140095	0.052505	0.467836	0.564855	0.075180	0.257765
NCD SC	s_{min}	]0.0, 1.0[	Uniform	0.29396	0.98137	0.86147	0.63534	0.70005	0.20412	0.83132
	u	[1, 200]	Int uniform	144	14	18	53	16	26	14
TabularNCD	k neighbors	[2, 100]	Int uniform	70	29	73	45	15	48	7
	Latent dim	[5, d]	Int uniform	372	14	13	56	98	62	791
	lr	[0.0001, 0.1]	Log uniform	0.00961	0.00036	0.00464	0.00124	0.00840	0.00755	0.00129
	Dropout	[0.0, 0.6]	Uniform	0.15436	0.06672	0.27823	0.27489	0.01384	0.09158	0.08220
	Top k	[0.0, 1.0]	Uniform	0.61097	0.16517	0.46739	0.93956	0.59044	0.99586	0.94210
	w_1	[0.0, 1.0]	Uniform	0.04745	0.41367	0.13992	0.31247	0.30193	0.78532	0.19631
	w_2	[0.0, 1.0]	Uniform	0.70265	0.97095	0.60544	0.88265	0.83427	0.93356	0.64128
PBN	Latent dim	[5, d]	Int uniform	504	22	12	13	172	53	579
	lr	[0.0001, 0.1]	Log uniform	0.00010	0.00058	0.00107	0.00211	0.00467	0.00036	0.00440
	Dropout	[0.0, 0.6]	Uniform	0.50984	0.02745	0.01126	0.16777	0.23934	0.06606	0.08131
	w	[0.0, 1.0]	Uniform	0.46714	0.72241	0.10671	0.96086	0.67034	0.18917	0.69815

- *SC PBN default:* k-means is directly replaced by SC with default hyperparameters. The similarity s_{min} is set to 0.6, and the number of components u of the spectral embedding is set to the number of novel classes C^u, as is usually done in the literature on Spectral Clustering.
- *SC PBN full opt:* Here, SC is also used for clustering, and its hyperparameters s_{min} and u are optimized together with PBN's hyperparameters. So for each datasets, 6 hyperparameters are optimized (4 from PBN and 2 from SC).

In Table 9, it can be seen that using SC with its default hyperparameters results in a clustering accuracy that is on average 1.1% higher than PBN coupled with k-means. But when the hyperparameters of SC are also optimized, the average accuracy drops by 4.1% instead. This shows that, as expressed throughout this article, having too many hyperparameters in the context of the NCD problem makes it difficult to effectively transfer the results on the known classes to the novel classes, and the hyperparameters tend to overfit the known classes.

These results suggest that the best approach is to apply SC with its default hyperparameters to cluster the projection of the novel data. However, it must be noted that the number of novel classes is used to define the number of components of the spectral embedding. Thus, in the case where C^u is not considered to be known in advance and has to be estimated, the performance may be degraded.

Appendix F: Cluster validity indices results details

An estimate of the number of clusters in the 7 datasets considered in this paper can be found in Table 10. Among the 6 CVIs reported here, the Silhouette coefficient performed the best. Furthermore, compared to the original feature space, its average estimation error significantly decreased in the latent space, validating our approach. For some datasets, the Davies-Bouldin index continued to decrease and the Dunn index continued to increase as the number of clusters

Table 9 Test accuracy of the 3 variations of PBN averaged over 10 runs

Dataset	KM PBN	SC PBN default	SC PBN full opt
Human	76.7 ± 1.8	**80.2 ± 0.7**	76.6 ± 6.1
Letter	62.4 ± 2.0	**64.4 ± 2.8**	50.9 ± 2.7
Pendigits	**82.8 ± 0.6**	82.0 ± 2.2	79.3 ± 4.0
Census	**62.4 ± 0.9**	61.1 ± 2.0	53.0 ± 2.4
m feat	91.7 ± 0.8	**92.6 ± 1.6**	89.7 ± 1.5
Optdigits	92.6 ± 2.3	**94.8 ± 2.1**	94.6 ± 3.6
CNAE-9	72.6 ± 4.6	**73.5 ± 6.1**	68.0 ± 4.2
Average	77.3	**78.4**	73.2

Best scores are in bold

increased, resulting in very large overestimations. Note that the estimates of the number of novel classes in Table 10 are not needed in the experiments of Sect. 7.2.2, since Algorithm 1 directly incorporates such estimates in the training procedure. This table has only helped us to identify the most appropriate CVI for our problem. The only exception is the TabularNCD method, which requires an a priori estimation of the number of novel classes in the original feature space.

Appendix G: NCD k-means centroids convergence

In this appendix, we aim to determine how to achieve the best performance with NCD *k*-means. Specifically, after the centroid initialization described in Sect. 3.2, we investigate:

1. whether it is more effective to update the centroids of both known and novel classes, or only the centroids of novel classes;
2. whether the centroids need to be updated using data from both known and novel classes, or only using data from novel classes.

The results are presented in Table 11 and show that for 5 out of 7 datasets, the best results are obtained when only the centroids of the novel classes are updated on the unlabeled data. Updating the centroids of the known classes always leads to worse performance, as the class labels are not used in this process. Thus, the centroids of the known classes run the risk of capturing data from the novel classes (and vice versa).

Table 10 An estimation of the number of novel classes with some CVIs in the latent space of PBN

Dataset	Human	Letter	Pendigits	Census	m feat	Optdigits	CNAE-9
Ground-truth	3	7	5	6	5	5	5
PBN latent space							
Silhouette	2	8	5	3	5	5	5
CH	2	3	5	4	2	2	5
Dunn	2	3	98	3	2	95	2
KM ACC	1	2	3	3	5	6	1
Davies-B	2	63	6	3	99	4	96
Elbow	9	14	10	7	16	13	9
Original feature space							
Silhouette	2	45	5	3	2	9	2

Table 11 ACC of NCD *k*-means averaged over 10 runs

Dataset	Converging the Centroids	On unlabeled Data only	On labeled and Unlabeled data
Human	Novel only	75.9 ± 0.0	**77.4 ± 0.0**
	Known and Novel	–	75.1 ± 0.5
Letter	Novel only	**51.9 ± 2.3**	39.5 ± 1.9
	Known and novel	–	42.3 ± 2.6
Pendigits	novel only	**81.7 ± 0.0**	72.7 ± 0.9
	Known and novel	–	75.3 ± 4.0
Census	novel only	**50.4 ± 1.1**	50.4 ± 4.8
	Known and novel	–	44.6 ± 8.3
m feat	novel only	**89.7 ± 0.4**	69.1 ± 0.2
	Known and novel	–	84.1 ± 7.0
Optdigits	Novel only	**94.2 ± 0.0**	70.8 ± 7.8
	Known and novel	–	74.0 ± 14.7
CNAE-9	Novel only	61.2 ± 4.5	48.3 ± 8.5
	Known and novel	–	**68.1 ± 7.5**

Best scores are in bold

Author contributions Colin Troisemaine, Alexandre Reiffers-Masson, Stéphane Gosselin, Vincent Lemaire and Sandrine Vaton contributed to the manuscript equally.

Funding Colin Troisemaine, Alexandre Reiffers-Masson, Stéphane Gosselin, Vincent Lemaire and Sandrine Vaton received funding from Orange SA.

Availability of data and materials All data used is this study are available publicly online. The datasets were extracted directly in the repositories available with the links in the corresponding section.

Code availability The code for experiments is available at the following url: https://github.com/Orange-OpenSource/PracticalNCD.

Declarations

Conflict of interest Colin Troisemaine, Stéphane Gosselin and Vincent Lemaire received research support from Orange SA. Alexandre Reiffers-Masson and Sandrine Vaton received research support from IMT Atlantique.

Ethics approval Not applicable.

Consent to participate All authors have read and approved the final manuscript.

Consent for publication Not applicable.

References

Anguita D, Ghio A, Oneto L, Parra X, Reyes-Ortiz JL (2013) A public domain dataset for human activity recognition using smartphones. In: ESANN

Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez JM, Perona I (2013) An extensive comparative study of cluster validity indices. Pattern Recogn 46(1):243–256

Arthur D, Vassilvitskii S (2007) K-means++ the advantages of careful seeding. In: ACM-SIAM SODA, pp 1027–1035

Arvai K (2023) Kneed. Zenodo. https://doi.org/10.5281/zenodo.7873825

Cao K, Brbic M, Leskovec J (2022) Open-world semi-supervised learning. In: ICLR

Caron M, Touvron H, Misra I, Jegou H., Mairal J, Bojanowski P, Joulin A (2021) Emerging properties in self-supervised vision transformers. In: ICCV, pp. 1–21

Chen Y, Zhu X, Li W, Gong S (2020) Semi-supervised learning under class distribution mismatch. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 3569–3576

Chi H, Liu F, Yang W, Lan L, Liu T, Han B, Niu G, Zhou M, Sugiyama M (2022) Meta discovery: learning to discover novel classes given very limited data. In: ICLR

Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In: CVPR, pp 248–255

Dua D, Graff C (2017) UCI machine learning repository

Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, vol 96, pp 226–231

Fei Y, Zhao Z, Yang S, Zhao B (2022) Xcon: learning with experts for fine-grained category discovery. In: British machine vision conference (BMVC)

Frey PW, Slate DJ (2005) Letter recognition using Holland-style adaptive classifiers. Mach Learn 6:161–182

Gidaris S, Singh P, Komodakis N (2018) Unsupervised representation learning by predicting image rotations. In: ICLR

Guo L-Z, Zhang Z-Y, Jiang Y, Li Y-F, Zhou Z-H (2020) Safe deep semi-supervised learning for unseen-class unlabeled data. In: ICML

Han K, Rebuffi S-A, Ehrhardt S, Vedaldi A, Zisserman A (2021) Autonovel: automatically discovering and learning novel visual categories. In: PAMI

Han K, Vedaldi A, Zisserman A (2019) Learning to discover novel visual categories via deep transfer clustering. In: ICCV

Hsu Y-C, Lv Z, Kira Z (2018) Learning to cluster in order to transfer across domains and tasks. In: ICLR

Hsu Y-C, Lv Z, Schlosser J, Odom P, Kira Z (2019) Multi-class classification without multi-class labels. In: ICLR

Khan AA, Mohanty SK (2022) A fast spectral clustering technique using MST based proximity graph for diversified datasets. Inf Sci 7:1113–1131

Kuhn HW, Yaw B (1955) The Hungarian method for the assignment problem. Naval Res Logist Quart 6:83–97

Le L, Patterson A, White M (2018) Supervised autoencoders: improving generalization performance with unsupervised regularizers. Adv Neural Inf Process Syst 31:72

Li Z, Otholt J, Dai B, Hu D, Meinel C, Yang H (2022) A closer look at novel class discovery from the labeled set. In: NeurIPS 2022 workshop on distribution shifts: connecting methods and applications

Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17:395–416

Ng A, Jordan M, Weiss Y (2001) On spectral clustering: analysis and an algorithm. Adv Neural Inf Process Syst 8:14

Satopaa V, Albrecht J, Irwin D, Raghavan B (2011) Finding a "kneedle" in a haystack: detecting knee points in system behavior. In: ICDCS workshops. IEEE, pp 166–171

Scheirer WJ, Rezende Rocha A, Sapkota A, Boult TE (2013) Toward open set recognition. IEEE Trans Pattern Anal Mach Intell 35(7):1757–1772

Stuetzle W (2003) Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. J Classif 20(1):25–47

Sun Y, Shi Z, Liang Y, Li Y (2023) When and how does known class help discover unknown ones? Provable understanding through spectral analysis. ICML 202:3014–33043

Sun Y, Li Y (2023) Opencon: open-world contrastive learning. In: TMLR

Troisemaine C, Flocon-Cholet J, Gosselin S, Vaton S, Reiffers-Masson A, Lemaire V (2022) A method for discovering novel classes in tabular data. In: ICKG, pp 265–274

Troisemaine C, Lemaire V, Gosselin S, Reiffers-Masson A, Flocon-Cholet J, Vaton S (2023) Novel class discovery: an introduction and key concepts

Vaze S, Han K, Vedaldi A, Zisserman A (2022) Generalized category discovery. In: CVPR, pp. 7492–7501

Von Luxburg U, Williamson RC, Guyon I (2012) Clustering: science or art? In: ICML workshop on unsupervised and transfer learning, pp 65–79
Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis. ICML 48:478–487
Yang M, Wang L, Deng C, Zhang H (2023) Bootstrap your own prior: towards distribution-agnostic novel class discovery. In: CVPR, pp 3459–3468
Yang M, Zhu Y, Yu J, Wu A, Deng C (2022) Divide and conquer: compositional experts for generalized novel class discovery. In: CVPR, pp 14268–14277
Zhang L, Qi L, Yang X, Qiao H, Yang M-H, Liu Z (2022) Automatically discovering novel visual categories with self-supervised prototype learning
Zhao B, Han K (2021) Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. Adv Neural Inf Process Syst 34:22982–22994
Zheng J, Li W, Hong J, Petersson L, Barnes N (2022) Towards open-set object detection and discovery. In: CVPR, pp 3961–3970
Zhong Z, Fini E, Roy S, Luo Z, Ricci E, Sebe N (2021) Neighborhood contrastive learning for novel class discovery. In: CVPR
Zhong Z, Zhu L, Luo Z, Li S, Yang Y, Sebe N (2021) Openmix: reviving known knowledge for discovering novel visual categories in an open world. In: CVPR, pp 9462–9470

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.