A Method for Discovering Novel Classes in Tabular Data

Colin Troisemaine*†, Joachim Flocon-Cholet*, Stéphane Gosselin*,
Sandrine Vaton†, Alexandre Reiffers-Masson†, Vincent Lemaire*
* *Orange Labs, Lannion, France*
† *Department of Computer Science, IMT Atlantique, Brest, France*
colin.troisemaine@orange.com

Abstract—**In Novel Class Discovery (NCD), the goal is to find new classes in an unlabeled set given a labeled set of known but different classes. While NCD has recently gained attention from the community, no framework has yet been proposed for heterogeneous tabular data, despite being a very common representation of data. In this paper, we propose TabularNCD, a new method for discovering novel classes in tabular data. We show a way to extract knowledge from already known classes to guide the discovery process of novel classes in the context of tabular data which contains heterogeneous variables. A part of this process is done by a new method for defining pseudo labels, and we follow recent findings in Multi-Task Learning to optimize a joint objective function. Our method demonstrates that NCD is not only applicable to images but also to heterogeneous tabular data. Extensive experiments are conducted to evaluate our method and demonstrate its effectiveness against 3 competitors on 7 diverse public classification datasets.**

Index Terms—**novel class discovery, clustering, tabular data, heterogeneous data, open world learning**

I. Introduction

The recent success of machine learning models has been enabled in part by the use of large quantities of labeled data. Many methods currently assume that a large part of the available data is labeled and that all the classes are known. However, these assumptions don't always hold true in practice and researchers have started considering scenarios where unlabeled data is available [1], [2]. They belong to Weakly Supervised Learning, where methods that require all the classes to be known in advance can be distinguished from those that are able to manage classes that have never appeared during training. As an example, Semi-Supervised Learning [3] combines a small amount of labeled data with larger amounts of unlabeled data. Nonetheless, some labeled data is still needed for each of the classes in this case. Another example is Zero-Shot Learning [4], where the models are designed to accurately predict classes that have never appeared during training. But some kind of description of these unknown classes is needed to be able to recognize them.

Very recently, Novel Class Discovery (NCD) [5] has been proposed to fill these gaps and attempts to identify new classes in unlabeled data by exploiting prior knowledge from known classes. In this specific setup, the data is split in two sets. The first is a labeled set containing known classes and the second is an unlabeled set containing unknown classes that must be discovered. Some solutions have been proposed for the NCD problem in the context of computer vision [6]–[9] and have displayed promising results.

However, research in this domain is still recent, and to the best of our knowledge, NCD has not directly been addressed for tabular data. While audio and image data have been of great interest in recent scientific publications, tabular data remains a very common information structure that is found in many real world problems, such as the information systems of companies. In this paper, we will therefore focus on NCD for tabular data.

Tabular data can come in large unlabeled quantities due to the labelling process being often costly and time-consuming, in part due to the difficulty of visualization, which is why a lot of unlabeled data is often left unexploited. In [10], the authors review the primary challenges that come with tabular data and identify three main obstacles to the success of deep neural networks on this type of data. The first is the quality of the data, that often includes missing values, extreme data (outliers), erroneous or inconsistent data and class imbalance. Then, the lack of spatial correlation between features makes it difficult to use techniques based on inductive biases, such as convolutions or data augmentation [11]. And finally, the heterogeneous and complex nature of the data (dense continuous and sparse categorical features) that can require considerable pre-processing, leading to information loss compared to the original data [12].

Our proposal: To address the NCD problem in the challenging tabular data environment, we propose TabularNCD (for Tabular Novel Class Discovery). In the first step of our method, an unbiased latent representation is initialized by taking advantage of the advances in Self-Supervised Learning (SSL) for tabular data [13]. Then, we build on the idea that the local neighborhood of an instance in the latent space is likely to belong to the same class, and a clustering of the unlabeled data is learned through similarity measures. The process in this second step is jointly optimized with a classifier on the known classes to include the relevant features from the already discovered classes.

Our key contributions are summarized as follows:

- We propose TabularNCD, a new method for Novel Class Discovery. To the best of our knowledge, this is the first attempt at solving the NCD problem in the context of tabular data with heterogeneous features. Thus, we do not depend on the spatial inductive bias of features, which other NCD methods rely heavily on when using convolutions and specialized data augmentation techniques.

- We empirically evaluate TabularNCD on seven varied public classification datasets. These experiments demonstrate the superior performance of our method over common fully unsupervised methods and a baseline that exploits known classes in a naive way.
- We also introduce an original approach for the definition of pairs of pseudo labels of unlabeled data. This approach exploits the local neighborhood of an instance in a pre-trained latent space by considering that its k most similar instances belong to the same class. We study its robustness and the influence of its parameters on performance.
- Lastly, we conduct experiments to understand in depth the reasons for the advantage that the proposed method has over simpler approaches.

II. Novel Class Discovery and related works

The process of discovering new classes depends on the final application in the real world (offline vs online learning, nature of the data, etc). In the literature of "concept drift" [14], changes in the distribution of known classes can appear either gradually or suddenly. Some analogies can be made with the present paper, where we consider the case of the "sudden drift", as the labeled dataset D^l and unlabeled dataset D^u do not share any classes (as described in Sec. I). Furthermore, we assume that a form of knowledge can be extracted from D^l to be then used on D^u. In this sense, "Novel Class Discovery (NCD)" methods mainly lie at the intersection of several other lines of research that we briefly review here.

Transfer Learning [15]: To solve a problem faster or with better generalization, transfer learning leverages knowledge from a different (but related) problem. Transfer learning can either be *cross-domain*, when a model trained to execute a task on one domain is retrained to solve the same task but on another domain. Or it can be *cross-task*, when a model trained to recognize some classes is retrained to distinguish other classes of the same domain. NCD can be regarded as a cross-task transfer learning problem, in the sense that it aims to cluster unlabeled data by leveraging knowledge from different labeled data. However, most of the works in transfer learning require labeled data in the source and target domains to train a classifier for the new task. When there are no labels in the target domain, cross-task transfer learning methods usually employ unsupervised clustering approaches trained on features extracted from labeled data.

Open World Learning: Unlike in the traditional *closed-world learning*, the problem of Open World Learning expects at test time instances from classes that were not seen during training. The objective is therefore threefold: 1) to obtain good accuracy on the known classes, 2) to recognize instances from novel classes and optionally, 3) to incrementally learn the new classes. In practice, most methods [16], [17] assume that the discovery of the new classes in the rejected examples is either the duty of a human or a task that is outside the scope of their research. To the best of our knowledge, [18], [19] are exceptions in this regard. However, Open World Learning still diverges from NCD, as classes in the unlabeled set can be unknown or not. This is not the case for NCD, where it is assumed that it is already known if an instance belongs to a unknown class or not. In other words, the focus of NCD is only on the third task of Open World Learning.

Semi-Supervised Learning: When a training set is partially labeled, classical supervised methods can only exploit the labeled data. Semi Supervised Learning [3] is a special instance of *weak supervision* where additional information in the form of a labeled subset or some sort of known relationship between instances is available. Using this information, models can learn from the full training set even when not all of the instances are labeled. However, all of these methods build on the assumption that labeled and unlabeled sets contain instances of the same classes, which is not the case in NCD.

Novelty Detection: From a training set with only instances of known classes, the goal of *Novelty Detection* (ND) is to identify if new test observations come from a novel class or not. This problem is concerned with semantic shift (i.e. the apparition of new classes). The authors of [20] conclude that the goal of ND is only to distinguish novel samples from the training distribution, and not to actually discover the novel classes. This is the first major difference with Novel Class Discovery, as the ultimate goal of NCD is to explore the novel samples. The second difference is that in NCD, the known and unknown classes are already split, which is not the case in ND.

Novel Class Discovery (NCD): In a large part of the literature related to NCD, the goal is to cluster classes in an unlabeled set when a labeled set of known classes is available. Unlike in semi-supervised learning, the classes of these two sets are disjoint. The labeled data is used to reduce the ambiguity that comes with the many potentially valid clustering criteria of the unlabeled data. It is a challenging problem, as the patterns learned from the labeled data may not be relevant or sufficient to cluster the unlabeled data. In [21], the authors explore the assumptions behind NCD and give a formal definition of NCD. They find that this is a theoretically solvable problem under certain assumptions, the most important one being that the labeled and unlabeled data must share good high-level features and that it must be meaningful to separate observations from these two sets.

NCD methods have mainly been developed for the computer vision problem, and the pioneering works involve *Constrained Clustering Network* (CCN) [5], where the authors approach NCD as a transfer learning problem and rely on the KL-divergence to evaluate the distance between the cluster assignments distributions of two data points. In *Deep Transfer Clustering* (DTC) [7], the authors extend the *Deep Embedded Clustering* method [22] by taking the available labeled data into account, allowing them to learn more relevant high-level features before clustering the unlabeled data. Another important work is *AutoNovel* [6], where two neural networks update a shared latent space. Finally, *Neighborhood Contrastive Learning* [9], which is based on *AutoNovel*, adds two contrastive learning terms to the loss to improve the quality of the learned representation. The common denominators of these works are the learning of a latent space and the definition of pairwise pseudo labels for the unlabeled data. Despite showing competitive performance, they are all solving NCD for computer vision problems. Thus, they are able to take advantage of the spatial correlation between the features with specialized architectures and they are not affected by the other challenges of tabular data. Because data augmentation is a crucial component to regularize the network, it is not possible to directly transfer their work on tabular data.

In the next section, the architecture of the proposed method is depicted, along with the two main steps of the training process.

III. A method for Novel Class Discovery on tabular data

Given a labeled set $D^l = \{X^l, Y^l\}$ where each instance $x^l \in \mathbb{R}^d$ has a label $y^l \in \{0,1\}^{C^l}$ (representing the one-hot encoding of the C^l classes) and an unlabeled set $D^u = \{X^u\}$, the goal is to identify and predict the classes of D^u. In this paper, our setting assumes that the number C^u of classes of D^u is known in advance and that the classes of D^l and D^u are disjoint. Additionally, we follow the formulation of the NCD problem from [21] and suppose that $P_{D^l}(Y|X)$ and $P_{D^u}(Y|X)$ are statistically different and separable, but still share high-level semantic features, so that we can extract general knowledge from D^l of what constitutes a pertinent class. So to *identify* the classes in D^u, a transformation ϕ must be learned, such that $\phi(X^u)$ is separable.

The proposed method includes two main steps: (i) first, the representation is initialized by pre-training the encoder ϕ on $D^l \cup D^u$ without using any label. Then (ii), a supervised classification task and an unsupervised clustering task are jointly solved on the previously learned representation, further updating it. Each of these two steps has its own architecture and training procedure which are described below and in the next sections.

Similarly to [9], our method is based on AutoNovel [6], and the representation is obtained using a model that includes a feature extractor $\phi(x) = z$, which is a simple combination of multiple dense layers with non-linear activation functions (see Fig. 1). In the second step, the latent representation of the feature extractor is forwarded to two linear classification and clustering layers followed by Softmax layers (see Fig. 2).

Despite the fact that for the tasks of classification or regression on tabular data, neither deep nor shallow neural networks have caught up with the performance of classical methods such as XGBoost [23], LightGBM [24] or CatBoost [25], we believe they are still relevant in our context. Neural networks on tabular data suffer from sparse values, high dimensionality and heterogeneous features. The authors of [10] insist that encoders are a good solution to these challenges, as data is projected in a homogeneous latent space of reduced dimensionality. Furthermore, in a well-defined latent space, instances close to each other are likely to belong to the same class. This means that data augmentation and similarity measures relying on this inductive bias can be used with more confidence after projection.

The following subsections describe in more details the two main training steps of the method and the consistency regularization term, which is required in the kind of architecture that is presented here.

A. Initialization of the representation

This first step aims at capturing a common and informative representation of both D^l and D^u. This is important because the representation is used in the next step, among other things, to compute the similarity of pairs of instances and thus determine if examples should belong to the same cluster or not. Among the possibilities offered in the literature, the way we decided to elaborate this representation is to project examples in a latent space produced by a deep architecture trained on D^l and D^u.

To pre-train the latent space using both labeled and unlabeled data, we take advantage of *Self-Supervised Learning* (SSL), and apply the *Value Imputation and Mask Estimation* (VIME) method [13]. As the name suggests, VIME defines two pretext tasks to train the encoder ϕ. From an input vector $x \in \mathbb{R}^d$ that has been corrupted, the objective is to 1) recover the original values and 2) recover the mask used to corrupt the input. The corruption process begins with the generation of a binary mask $m = [m_1, ..., m_d]^\top \in \{0,1\}^d$, with m_j randomly sampled from a Bernoulli distribution with probability p_m. The corrupted vector $\tilde{x}$ is then created by replacing each dimension j of x where $m_j = 1$ with the dimension j of a randomly sampled instance from the training set. So $\tilde{x} = (1-m) \odot x + m \odot \bar{x}$, where $\odot$ is the element-wise product and each $\bar{x}_j$ of $\bar{x}$ has been randomly sampled from the empirical marginal distribution of the j-th feature of the training set.

The pre-training framework is illustrated in Fig. 1. The encoder and the mask and feature vector estimators are jointly trained with the following optimization problem:

$$\mathcal{L}_{VIME} = l_{recons.} + \alpha l_{mask} \tag{1}$$

where α is a trade-off parameter. Following [13], we use $\alpha = 2.0$. The *mean squared error* (MSE) loss is used to optimize the feature vector estimator:

$$l_{recons.} = \frac{1}{d} \sum_{j=1}^{d} (x_j - \hat{x}_j)^2 \tag{2}$$

where $\hat{x}$ is the input reconstructed from the corrupted vector $\tilde{x}$. The *binary cross-entropy* (BCE) loss is used to optimize the mask estimator:

$$l_{mask} = -\frac{1}{d} \sum_{j=1}^{d} [m_j \log(\hat{m}_j) + (1-m_j)\log(1-\hat{m}_j)] \tag{3}$$

where $\hat{m}$ is the estimated mask.

The authors argue that by training an encoder to solve these two tasks, it is possible to capture the correlation between the

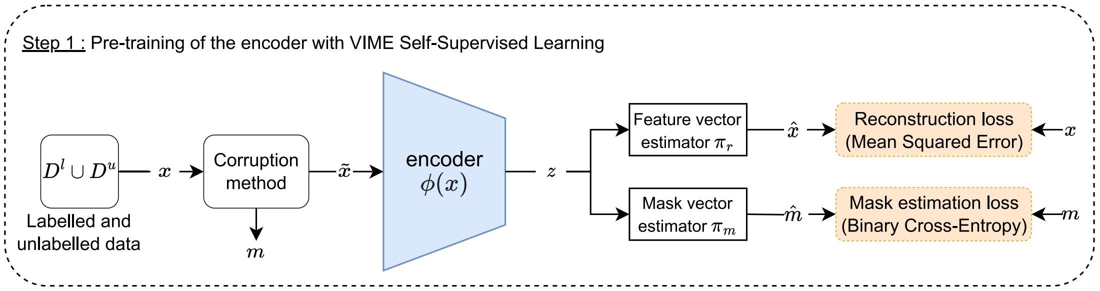

Step 1 : Pre-training of the encoder with VIME Self-Supervised Learning
$D^l \cup D^u$
Labelled and unlabelled data
x
Corruption method
m
$\tilde{x}$
encoder $\phi(x)$
z
Feature vector estimator π_r
Mask vector estimator π_m
$\hat{x}$
$\hat{m}$
Reconstruction loss (Mean Squared Error)
Mask estimation loss (Binary Cross-Entropy)
x
m

Fig. 1. Architecture of the pre-training step.

features and create a latent representation z that contains the necessary information to recover the input x. While the general idea of the VIME method is similar to that of a *denoising auto-encoder* (DAE), it reports superior performance in [13] and there are two major differences. The first is the addition of the mask estimation task, and the second lies in the noise generation process. A DAE will create noisy inputs by adding Gaussian noise or replacing values with zeroes, while VIME randomly selects values from the empirical marginal distribution of each feature. This means that the noisy $\tilde{x}$ of VIME will be harder to distinguish from the input.

Note that a simple approach to initialize the encoder would be to train a classifier using the known classes. However, the resulting representation would rapidly overfit on these classes and the unique features of the unknown classes would be lost. Instead, by generating pseudo labels from pretext tasks on unlabeled data, a model can be trained to learn informative representations and high level features that are unbiased towards labeled data. This is the general idea behind Self-Supervised Learning and it is a technique commonly used in NCD [6], [9], [26]. An example of it is RotNet [27], where the model has to predict the rotation that was applied to an image from the possible 0, 90, 180 and 270 degrees values. Unfortunately, only a few works have attempted to adapt this technique for tabular data [13], [28], [29] and haven't had the same success.

B. Joint training on the labeled and unlabeled data

We formulate the novel class discovery process as multi-task optimization problem. In this step, two new networks are added on top of the previously initialized encoder, each solving different tasks on different data (see Fig. 2). The first is a classification network $\eta^l(z) \in \mathbb{R}^{C^l+1}$ trained to predict 1) the C^l known classes from D^l with the ground-truth labels and 2) a single class formed of the aggregation of the unlabeled data. The second is another classification network trained to predict the C^u novel classes from D^u. It will be referred as the *clustering* network $\eta^u(z) \in \mathbb{R}^{C^u}$. These two networks share the same embedding and both update it through back-propagation, sharing knowledge with one another.

The classification network is optimized with the *cross-entropy* loss using the ground-truth one-hot labels y:

$$l_{class.} = -\sum_{c=1}^{C^l+1} y_c \log(\eta^l_c(z)) \tag{4}$$

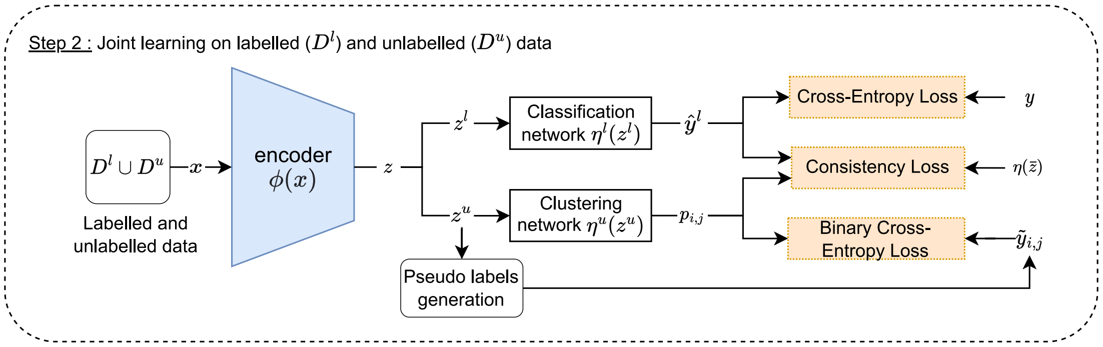
where $z = \phi(x)$. Its role is to guide the representation to include the features of the known classes that are relevant for the supervised classification task.

To train the clustering network η^u with unlabeled data in a supervised manner, pseudo labels $\tilde{y}_{i,j} \in \{0,1\}$ are generated for each pair (x_i, x_j) of unlabeled data in a mini-batch. This is comparable to a pretext task in Self-Supervised Learning. Another possible approach would be to predict the similarity of instances directly, but using pseudo-labels has the advantage of directly associating classes to the instances and does not require an additional clustering of the embedding. In this case, pseudo labels indicate if a pair of instances should belong to the same class or not, independently of the class number predicted by the network. They are defined as $\tilde{y}_{i,j} = 1$ if z_i and z_j are similar, and $\tilde{y}_{i,j} = 0$ if they are dissimilar. After the mini-batch X has been projected in the encoder ($Z = \phi(X)$), there are $|Z| - 1$ pairwise pseudo labels defined for each instance $z_i \in Z$. The clustering network is optimized with the *binary cross-entropy* loss, which is for a single instance z_i:

$$l_{clust.} = \frac{1}{|Z|-1} \sum_{\substack{j=1 \\ j \neq i}}^{|Z|} [-\tilde{y}_{i,j} \log(p_{i,j}) - (1-\tilde{y}_{i,j}) \log(1-p_{i,j})] \tag{5}$$

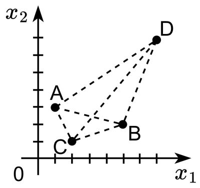
where $p_{i,j} = \eta^u(z_i) \cdot \eta^u(z_j)$ is a score close to 1 if the clustering network predicted the same class for z_i and z_j and close to 0 otherwise. For this reason, $p_{i,j}$ can directly be compared to the pairwise pseudo labels $\tilde{y}_{i,j}$. The intuition is that instances similar to each other in the latent space are likely to belong to the same class. Therefore, η^u will create clusters of similar instances, guided by η^l with the knowledge of the known classes.

Note: Predicting the unlabeled data as a single new class in the classification network is not done in AutoNovel [6] and is one of the main differences with our work. We found that having an overlap of the instances in the classification and clustering networks would improve the performance and result in the definition of cleaner latent space that does not mix labeled and unlabeled data.



Step 2 : Joint learning on labelled (D^l) and unlabelled (D^u) data
$D^l \cup D^u$
Labelled and unlabelled data
x
encoder $\phi(x)$
z
z^l
z^u
Classification network $\eta^l(z^l)$
Clustering network $\eta^u(z^u)$
$\hat{y}^l$
$p_{i,j}$
Pseudo labels generation
Cross-Entropy Loss
Consistency Loss
Binary Cross-Entropy Loss
y
$\eta(\bar{z})$
$\tilde{y}_{i,j}$

Fig. 2. Architecture of the joint learning step. After the input data x has been projected, labeled data z^l is used to train the classification network η^l and unlabeled data is used to train the clustering network η^u with the generated pseudo labels as its target.



(a) Representation of the data points of the batch.

	A	B	C	D
A	1	0.59	0.78	0.28
B	0.59	1	0.68	0.46
C	0.78	0.68	1	0.22
D	0.28	0.46	0.22	1

(b) Pairwise similarity matrix.

	A	B	C	D
A	0	1	1	0
B	1	0	1	0
C	1	1	0	0
D	1	1	0	0

(c) Pairwise pseudo labels matrix for $k = 2$.

Fig. 3. The pairwise pseudo labels definition process.

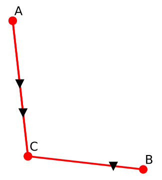


Fig. 4. Illustration of the SMOTE data augmentation technique. The red dots are the batch samples, and the black triangles are the synthetic new samples.

Definition of pseudo labels. Pseudo labels are defined for each unique pair of unlabeled instances in a mini-batch based on similarity. This idea has been employed in many NCD works (e.g. [6], [9], [30]), where the most common approach is to define a threshold λ for the minimum similarity of pairs of instances to be assigned to the same class.

However, we found[1] that defining for each point the top k most similar instances as positives was a more reliable method. So, for each pair (x_i, x_j) in the projected batch of unlabeled data Z, the pseudo labels are assigned as follows:

$$\tilde{y}_{i,j} = \mathbb{1}[j \in \underset{\substack{r \in \{1,\dots,|Z|\} \\ r \neq i}}{\text{argtop}_k} \; \delta(z_i, z_r)] \tag{6}$$

where $\delta(z_i, z_r) = \frac{z_i \cdot z_r}{\|z_i\| \|z_r\|}$ is the cosine similarity and argtop_k is the subset of indices of the k largest elements. This means that each observation has k positive pairs and $|Z| - 1 - k$ negative pairs. This process is illustrated in Fig. 3. From a mini-batch of a few observations (Fig. 3(a)), the pairwise similarity matrix (Fig. 3(b)) is derived. Then, the pairwise pseudo labels matrix is defined (Fig. 3(c)) where in each row, positive relations are set for the $k = 2$ most similar pairs and the rest are negatives.

C. Consistency regularization

During the training of the two classification networks, the latent representation of the encoder changes. And since the pseudo labels are defined according to the similarity of instances in the latent space, this causes a *moving target* phenomenon, where the pairwise pseudo labels can differ from one epoch to another. To limit the impact of this problem, a regularization term is needed. Commonly employed in semi-supervised learning [31], we use "data augmentation". The idea is to encourage the model to predict the same class for an instance x and its augmented counterpart $\bar{x}$. But while data augmentation has become a standard in computer vision, the same cannot be said for tabular data. The authors of [10] consider the lack of research in tabular data augmentation (along with the difficulty to capture the dependency structure of the data) to be one of the main reasons for the limited success of neural networks in tabular data. Nevertheless, we find that consistency regularization is an essential component for the performance of the method proposed here.

In this paper we use SMOTE-NC (for Synthetic Minority Oversampling Technique [32]), which is one of the most widely used tabular data augmentation techniques. It synthesizes new samples from the minority classes to reduce imbalance, thus improving the predictive performance of models on these classes. Synthetic samples are generated by taking a point along the line between the considered sample and a random instance among the k closest of the same class (in the original input space). Fig. 4 illustrates this technique. The larger the k is, the wider and less specific the decision regions of the learned model will be, resulting in better generalization.

This method can be applied directly to instances of the labeled set, however we relax the constraint of selecting same-class instances for the unlabeled set, as no labels are available. It can also be noted that the nearest neighbors for

[1]Experimentally, see the supplementary material in https://github.com/ColinTr/TabularNCD

the observations of the unlabeled set are not selected from the labeled set since one of the hypotheses of our problem is that $C^l \cap C^u = \emptyset$ (the same goes for the labeled set).

The *mean squared error* (MSE) is used as the consistency regularization term for both the classification and the clustering network. For the clustering network, it is written as:

$$l_{reg.} = \frac{1}{C^u} \sum_{c=1}^{C^u} (\eta_c(z) - \eta_c(\bar{z}))^2 \tag{7}$$

where $\bar{z} = \phi(\bar{x})$ is the projection of x augmented with the method inspired from SMOTE-NC. For the classification network, it is averaged over $C^l + 1$ instead.

D. Overall loss

The method proposed here falls under the domain of Multi-Task Learning (MTL) where instead of focusing on the only task that solves the problem at hand (discovering new classes in D^u), additional related tasks are jointly learned (classification of samples in D^l). By introducing new inductive biases, the model will prefer some hypotheses over others, guiding the model towards better generalization [33]. Our architecture falls under the *hard parameter sharing* domain of MTL, where the first layers are shared between the different tasks (i.e. the encoder), and the last few layers are task-specific.

We chose to adopt an alternating optimization strategy, which was introduced in [34]. In this case, we define one objective function and one individual optimizer per task. The loss of the classification network is:

$$\mathcal{L}_{classification} = w_1 l_{classif.} + (1 - w_1) l_{reg.} \tag{8}$$

And the loss of the clustering network can be written as:

$$\mathcal{L}_{clustering} = w_2 l_{clust.} + (1 - w_2) l_{reg.} \tag{9}$$

where w_1 and w_2 are trade-off hyper-parameters for balancing the weight of the consistency regularization terms.

Description of the training process: The classification network and the clustering network are trained alternately. For each mini-batch during training, the classification network and the encoder are first updated through back-propagation with the loss from (8). Then, the unlabeled data of the same mini-batch is forwarded once again through the encoder to compute the loss from (9), which is back-propagated to update the clustering network and the encoder once more.

IV. Experiments

A. Datasets and experimental details

Datasets. To evaluate the performances of the method proposed here, a variety of datasets in terms of application domains and characteristics have been chosen (see Table I). Six public tabular classification datasets were selected: Forest Cover Type [35], Letter Recognition [36], Human Activity Recognition [37], Satimage [38], Pen-Based Handwritten Digits Recognition [38] and 1990 US Census Data [38], as well as MNIST [39], which was flattened to transform the 28×28 grayscale images into vectors of 784 attributes. We pre-process

TABLE I
Statistical information of the selected datasets.

Dataset name	Attri-butes	# classes C^l / C^u	# train labeled	# train unlabeled	# test labeled	# test unlabeled
MNIST [39]	784	5 / 5	30,596	29,404	5,139	4,861
Forest Cover type [35]	54	4 / 3	6,480	4,860	36,568	13,432
Letter recognition [36]	16	19 / 7	10,229	3,770	4,296	1,704
Human activity [37]	562	3 / 3	3,733	3,619	1,494	1,453
Satimage [38]	36	3 / 3	2,525	1,976	1,042	887
Pendigits [38]	16	5 / 5	3,777	3,717	1,764	1,734
1990 US Census [38]	67	12 / 6	50,000	50,000	31,343	18,657

the numerical features of all the datasets to have zero-mean and unit-variance, while the categorical features are one-hot encoded. If the training and testing data were not already split, 70% were kept for training while the remaining 30% were used for testing.

Following the same procedure found in Novel Class Discovery articles [6], [9], [26], we hide the labels of some classes to create the *unknown* classes. And the capacity of the compared methods to recover these classes is then evaluated. Around 50% of the classes are hidden, resulting in further splitting of the training and testing sets into labeled and unlabeled subsets. The partitions of the 7 datasets can be found in Table I.

Evaluation metrics. To thoroughly assess the performance of the methods compared in the experiments described below, we report four different metrics. The first two are the clustering accuracy (ACC) and balanced accuracy (BACC), which can be computed after the optimal linear assignment of the class labels is solved with the Hungarian algorithm [40]. The use of the balanced accuracy is justified by the unbalanced class distribution of some datasets, which resulted in inflated accuracy scores that were not truly representative of the performance of the method. Another reported metric is the Normalized Mutual Information (NMI). It measures the correspondence between two sets of label assignments and is invariant to permutations. Finally, we compute the Adjusted Rand Index (ARI). It measures the similarity between two clustering assignments.

These four metrics range between 0 and 1, with values closer to 1 indicating a better agreement to the ground truth labels. The ARI is an exception and can yield values in $[-1, +1]$, with negative values when the index is less than the $Expected\_RI$. The metrics reported in the next section are all computed on the unlabeled instances from the test sets.

Competing methods. As expressed in Section I, to the best of our knowledge, there is no other method that solves the particular Novel Class Discovery problem for tabular datasets. Nonetheless, we can compare ourselves to unsupervised clustering methods. This will also allow us to show that our method provides an effective way of incorporating knowledge from known classes. We choose the k-means algorithm for its simplicity and wide adoption, as well as the Spectral Clustering [41] method for its known good results to discover new patterns as in Active Learning [42]. We set k to the ground truth (i.e. $k = C^u$) for both clustering methods, as it was already assumed that it is known in the proposed method.

We also define a simple baseline that clusters unlabeled data while still capitalizing on known classes: (i) first, a usual classification neural network is trained on the known classes from D^l; (ii) then the classifier's penultimate layer is used as a feature embedding for D^u. In this projection, a k-means is applied to assign labels to the unlabeled test instances. Compared to the unsupervised approaches, this technique has the advantage of learning the patterns of the known classes in a homogeneous latent space. However, we still expect it to perform sub-optimally as the unique features and patterns of the unlabeled data might be lost in the last hidden layer after training.

Implementation details. For all datasets, we employ an encoder composed of 2 dense hidden layers that keeps the dimension of the input with activation functions and dropout values that are optimized hyper-parameters. Following VIME [13] and AutoNovel [6], we use a single linear layer for the mask estimator, vector estimator, classification network and clustering network.

The hyper-parameters (see Table VI in appendix) of the proposed method are optimized with a Bayesian search on a validation set which represents 20% of the training set. During the Self-Supervised Learning step (see Section III-A), we observe little impact on the final performance of the model when optimizing the hyper-parameters specific to the VIME method, and therefore use the values suggested in the original paper: a learning rate of 0.001, a corruption rate of 30% and a batch size of 128. In the joint training step (see Section III-B), we use a larger batch size of 512 as the observations are randomly sampled from the merged labeled and unlabeled data. The hyper-parameters of this step are the learning rates of both classification and clustering network optimizers, along with the trade-off parameters of their respective losses. The top k number of positive pairs defined by (6) and the k $neighbors$ considered in the data augmentation method of Sec. III-C are also optimized and can all be found in Table VI in appendix. We use AdamW [43] as the optimizer for the pre-training step and for the two optimizers of the joint training step, and find that 30 epochs are enough to converge for both steps on any of the tabular datasets used.

We implemented our method under Python 3.7.9 and with the PyTorch 1.10.0 [44] library. The experiments were conducted on Nvidia 2080 Ti and Nvidia Quadro T1000 GPUs. The networks are initialized with random weights, and following [6], the results are averaged over 10 runs. The code is publicly available at https://github.com/ColinTr/TabularNCD.

B. Results

Comparison to the competing methods. In Table II, we report the performance of the 4 competing methods and on the 7 datasets for the clustering task. The results show that TabularNCD achieves higher performance than the baseline and the two unsupervised clustering methods, on all considered datasets and on all metrics. The improvements in accuracy over competing methods ranges between 1.6% and 21.0%. This proves that even for tabular data, useful knowledge can be extracted from already discovered classes to guide the novel class discovery process.

TABLE II
PERFORMANCE OF TABULARNCD ON THE UNKNOWN CLASSES.

Dataset	Method	BACC (%)	ACC (%)	NMI	ARI
MNIST	Baseline	57.7±4.7	57.6±4.5	0.37±0.2	0.31±0.3
	Spect. clust	-	-	-	-
	k-means	60.1±0.0	61.1±0.0	0.48±0.0	0.38±0.0
	TabularNCD	**91.5±4.1**	**91.4±4.2**	**0.82±0.06**	**0.81±0.04**
Forest	Baseline	55.6±2.0	68.5±1.4	0.27±0.02	0.15±0.01
	Spect. clust	32.1±1.4	85.8±4.0	0.01±0.01	0.09±0.01
	k-means	32.9±0.0	62.0±0.0	0.04±0.00	0.05±0.00
	TabularNCD	**66.8±0.6**	**92.2±0.2**	**0.37±0.09**	**0.56±0.09**
Letter	Baseline	55.7±3.6	55.9±3.6	0.49±0.04	0.33±0.04
	Spect. clust	45.3±4.0	45.3±4.0	0.48±0.03	0.18±0.03
	k-means	50.2±0.6	49.9±0.6	0.40±0.01	0.28±0.01
	TabularNCD	**71.8±4.5**	**71.8±4.5**	**0.60±0.04**	**0.54±0.04**
Human	Baseline	80.0±0.5	78.0±0.6	0.64±0.01	0.62±0.01
	Spect. clust	70.2±0.0	69.4±0.0	0.72±0.00	0.60±0.00
	k-means	75.3±0.0	77.0±0.0	0.62±0.00	0.59±0.00
	TabularNCD	**98.9±0.2**	**99.0±0.2**	**0.95±0.01**	**0.97±0.01**
Satimage	Baseline	53.8±3.4	53.9±4.2	0.25±0.03	0.22±0.03
	Spect. clust	82.2±0.1	77.8±0.1	0.51±0.00	0.46±0.00
	k-means	73.7±0.3	69.2±0.2	0.30±0.00	0.28±0.00
	TabularNCD	**90.8±4.0**	**91.4±5.0**	**0.71±0.11**	**0.79±0.07**
Pendigits	Baseline	72.8±5.5	72.8±5.4	0.62±0.06	0.54±0.07
	Spect. clust	84.0±0.0	84.0±0.0	**0.78±0.00**	0.67±0.00
	k-means	82.5±0.0	82.5±0.0	0.72±0.00	0.63±0.00
	TabularNCD	**85.5±0.7**	**85.6±0.8**	0.76±0.02	**0.71±0.02**
Census	Baseline	53.0±3.5	**55.0±6.5**	0.49±0.02	0.30±0.03
	Spect. clust	23.6±3.3	51.3±5.5	0.24±0.11	0.18±0.09
	k-means	38.5±2.6	49.8±3.6	0.41±0.05	0.28±0.03
	TabularNCD	**61.9±0.6**	50.1±0.9	0.48±0.01	0.30±0.00

The standard deviation is computed over 10 executions. The 2 unsupervised clustering methods (Spect. clust and k-means) are only fitted to the test instances belonging to the unknown classes. Values for the spectral clustering of MNIST are missing as the execution did not complete under 1 hour.

While the baseline competitor (k-means on a projection learned on the labeled data, see Sec. IV-A) improves the performance of k-means for some datasets (notably the Census and Forest datasets), it still lags behind our method in general. The baseline obtains a lower score than the simple k-means on only 3 datasets, which means that in some cases, the features extracted from the known classes are insufficient to discriminate novel unseen data. This is the case for the *Satimage* dataset, where the performance of the baseline is much lower compared to the simple k-means on the original data. To understand this phenomenon, we compare the importance of the features used to predict the known classes to the importance of the features of the unknown classes. Because the features have been standardized, we can simply look at the coefficients of a logistic regression. Fig. 5 illustrates this experiment: the 5 most important coefficients in the prediction of one of the unknown classes are more than twice as large as the addition of the coefficients of the attributes used to predict the known classes. Because these attributes that are critical in the discrimination of this unknown class are relatively unimportant in the discrimination of the known classes, they are lost during the training of the classifier that serves as a feature extractor in the baseline, which results in a projection of the unlabeled instances of poor quality. This issue is more

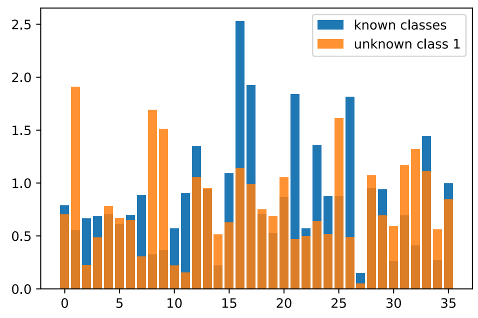


Fig. 5. Per-attribute coefficients of a logistic regression trained to predict the known and unknown classes. The dataset is *Satimage*.

TABLE III
PERFORMANCE OF THE PROPOSED METHOD AGAINST CD-KNET [8].

Dataset	MNIST		
Method	ACC (%)	NMI	ARI
TabularNCD	91.4	0.823	0.812
CD-KNet	86.9	0.683	0.707
CD-KNet-Exp	**94.5**	**0.856**	**0.869**

pronounced as the known and unknown classes are more dissimilar. The latent representation of TabularNCD is pre-trained using SSL on all the available data, which explains why it is unaffected by the phenomenon described above.

Comparison to a NCD method on MNIST. CD-KNet-Exp [8] (for *Class Discovery Kernel Networks with Expansion*) is one of the first methods that attempts to solve the NCD problem. They define their problem as an Open World Class Discovery problem, which is actually another name for Novel Class Discovery. In their proposed method, they start by training a classifier on the known classes of D^l, and then fine-tune it on both D^l and D^u before applying a k-means on the learned latent representation to get the cluster assignments. Similarly to our experimental protocol, they select the first 5 digits of MNIST as known classes and use the last 5 as new unlabeled classes to discover. For this reason, we can directly compare their reported performance to ours in Table III. From this table, we see that in terms of clustering accuracy, our method performs 4.5% better than their simpler implementation CD-KNet, and 3.1% worse than their complete approach CD-KNet-Exp, where the previously learned classifier is re-trained with the pseudo labels assigned using k-means. Obtaining comparable results on MNIST is very encouraging because unlike our approach, CD-KNet-Exp takes advantage of convolutional neural networks as they only use image data. Their re-training technique could also be explored in future work to improve performance.

Ablation study. The usefulness of each of the components of the proposed method is estimated in Table IV by ablating them and comparing the metrics after training to the metrics of full method. The first observation that can be made is that each component has a positive influence, and removing any of them results in a drop in performance. The most important one is the BCE (5), since without it the clustering network degenerates to a trivial solution where it predicts the same class for all observations. Next is the MSE from the consistency loss (7). The substantial drop in performance was expected as its role is crucial to (1) reduce the moving target phenomenon introduced in Sec. III-C and (2) regularize the network to improve its generalization. We also observe an important decrease in performance when removing the CE (5), meaning that in the case of the Satimage dataset, jointly learning a classification network with the clustering network helps guide the clustering process as intended. Finally, while it is beneficial to pre-train the encoder with SSL, the impact is small. This can be explained in part by the limited success that SSL works have had in the challenging area of tabular data.

TABLE IV
ABLATION STUDY OF THE PROPOSED METHOD.

Method	BACC (%)	ACC (%)	NMI	ARI
TabularNCD	90.8±4.0	91.4±5.0	0.71±0.11	0.79±0.07
- w/o SSL	88.4±5.3	88.6±7.0	0.67±0.15	0.67±0.10
- w/o CE	72.0±6.1	69.5±6.0	0.44±0.12	0.49±0.08
- w/o BCE	33.3±0.0	51.7±0.0	0.00±0.00	0.00±0.00
- w/o MSE	66.7±5.7	63.9±4.4	0.44±0.02	0.37±0.02

SSL: Self-Supervised Learning, **CE:** Cross Entropy loss of the classification network, **BCE:** Binary Cross Entropy loss of the clustering network, **MSE:** Mean Squared Error consistency loss. The dataset is Satimage [38].

Visualization. In addition to quantitative results, we also report a qualitative analysis showing the feature space that is learned. In Fig. 6, we visualize the evolution of the representation of all the classes during the training process of the proposed method. Fig 6(a) corresponds to the original data, while the next figures are the data projected in the latent space. Classes overlap in the original representation, as well as in the latent space after Self-Supervised Learning (Fig 6(b)). However, the SSL step initialized the encoder to a reasonable representation. This means that at the beginning of the joint training, the pseudo labels defined on the unlabeled data using (6) will be accurate. During the joint training in the Figures 6(c) and 6(d), the classes are more and more separated, showing that the proposed method is able to successfully discover novel classes using knowledge from known classes. The plot clearly shows that our model produces feature representations where samples of the same class are tightly grouped.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a first solution to the Novel Class Discovery problem in the challenging environment of tabular data. We demonstrated the effectiveness of our proposed approach, TabularNCD, through extensive experiments and careful analysis on 7 public datasets against unsupervised clustering methods. The greater performance of our method has shown that it is possible to extract knowledge from already discovered classes to guide the discovery process of novel classes, which demonstrates that NCD is not only applicable to images but also on tabular data. Lastly, the original method of defining pseudo labels proposed here has proven to be reliable even in the presence of unbalanced classes.

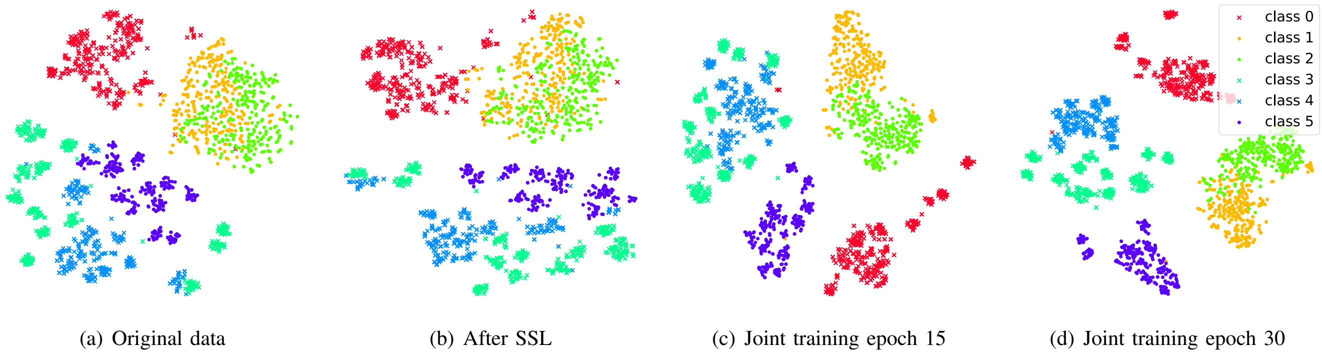


Fig. 6. Evolution of the t-SNE during the joint training of the model on the *Human Activity Recognition* dataset.

The recent advances of deep learning in tabular data are worth investigating in future work. Specifically, we will explore Generative Adversarial Networks and Variational Auto-Encoders as substitutes for the model's current simple encoder, which is a core component of our method. Furthermore, the assumption that the number of novel classes is known is a limitation of our method, and will certainly be a future direction of our work.

References

[1] P. Nodet, V. Lemaire, A. Bondu, A. Cornuéjols, and A. Ouorou, "From weakly supervised learning to biquality learning: an introduction," in *IJCNN*. IEEE, 2021, pp. 1–10.
[2] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, 08 2017.
[3] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. The MIT Press, 2006.
[4] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, jan 2019.
[5] Y.-C. Hsu, Z. Lv, and Z. Kira, "Learning to cluster in order to transfer across domains and tasks," *ArXiv*, vol. abs/1711.10125, 2018.
[6] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman, "Autonovel: Automatically discovering and learning novel visual categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
[7] K. Han, A. Vedaldi, and A. Zisserman, "Learning to discover novel visual categories via deep transfer clustering," in *Proceedings of the IEEE/CVF ICCV*, 2019, pp. 8401–8409.
[8] Z. Wang, B. Salehi, A. Gritsenko, K. Chowdhury, S. Ioannidis, and J. Dy, "Open-world class discovery with kernel networks," in *IEEE International Conference on Data Mining (ICDM)*, 2020, pp. 631–640.
[9] Z. Zhong, E. Fini, S. Roy, Z. Luo, E. Ricci, and N. Sebe, "Neighborhood contrastive learning for novel class discovery," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
[10] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *arXiv preprint arXiv:2110.01889*, 2021.
[11] Y. Zhu, T. Brettin, F. Xia, A. Partin, M. Shukla, H. Yoo, Y. Evrard, J. Doroshow, and R. Stevens, "Converting tabular data into images for deep learning with convolutional neural networks," *Scientific Reports*, vol. 11, 05 2021.
[12] E. Fitkov-Norris, S. Vahid, and C. Hand, "Evaluating the impact of categorical data encoding and scaling on neural network classification performance: The case of repeat consumption of identical cultural goods," in *Communications in Computer and Information Science*, vol. 311, 09 2012, pp. 343–352.
[13] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar, "Vime: Extending the success of self- and semi-supervised learning to tabular domain," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 11 033–11 043.
[14] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, pp. 2346–2363, 2019.
[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
[16] Q. Qin, W. Hu, and B. Liu, "Text classification with novelty detection," *ArXiv*, vol. abs/2009.11119, 2020.
[17] H. Xu, B. Liu, L. Shu, and P. Yu, "Open-world learning and application to product classification," in *The World Wide Web Conference*. ACM Press, 2019, p. 3413–3419.
[18] X. Guo, A. Alipour-Fanid, L. Wu, H. Purohit, X. Chen, K. Zeng, and L. Zhao, "Multi-stage deep classifier cascades for open world recognition," *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Nov 2019.
[19] L. Shu, H. Xu, and B. Liu, "Unseen class discovery in open-world classification," *ArXiv*, vol. abs/1801.05609, 2018.
[20] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *arXiv preprint arXiv:2110.11334*, 2021.
[21] H. Chi, F. Liu, W. Yang, L. Lan, T. Liu, B. Han, G. Niu, M. Zhou, and M. Sugiyama, "Meta discovery: Learning to discover novel classes given very limited data," in *ICLR*, 2022.
[22] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International Conference on Machine Learning (ICML)*, vol. 48, 2016, pp. 478–487.
[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[25] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[26] K. Cao, M. Brbic, and J. Leskovec, "Open-world semi-supervised learning," in *ICLR*, 2022.
[27] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *ArXiv*, 2018.
[28] D. Bahri, H. Jiang, Y. Tay, and D. Metzler, "Scarf: Self-supervised contrastive learning using random feature corruption," *ArXiv*, 2021.
[29] T. Ucar, E. Hajiramezanali, and L. Edwards, "Subtab: Subsetting features of tabular data for self-supervised representation learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.
[30] Y.-C. Hsu, Z. Lv, J. Schlosser, P. Odom, and Z. Kira, "Multi-class classification without multi-class labels," *ArXiv*, 2019.
[31] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[32] N. Chawla, K. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[33] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, 07 1997.

[34] L. Pascal, P. Michiardi, X. Bost, B. Huet, and M. A. Zuluaga, “Optimization strategies in multi-task learning: Averaged or independent losses?” *arXiv*, vol. abs/2109.11678, 2021.
[35] J. A. Blackard and D. J. Dean, “Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables,” *Computers and Electronics in Agriculture*, vol. 24, no. 3, pp. 131–151, 1999.
[36] P. W. Frey and D. J. Slate, “Letter recognition using holland-style adaptive classifiers,” *Machine Learning*, vol. 6, pp. 161–182, 2005.
[37] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones,” in *ESANN*, 2013.
[38] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[39] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
[40] H. W. Kuhn and B. Yaw, “The hungarian method for the assignment problem,” *Naval Res. Logist. Quart*, pp. 83–97, 1955.
[41] U. von Luxburg, “A tutorial on spectral clustering.” *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[42] X. Wang and I. Davidson, “Active spectral clustering,” in *IEEE International Conference on Data Mining (ICDM)*, 2010, pp. 561–568.
[43] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations (ICLR)*, 2019.
[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

Hyper-parameters influence.

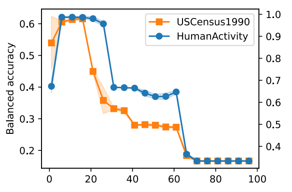
The method proposed here has a few hyper-parameters that must be tuned to get optimal performance. In Fig 7, we study some of the most important hyper-parameters. The first is the top_k from (6), it represents the percentage of pairs that will be regarded as positives for each instance. It varies in the range [1, 100] in Fig. 7(a), with ideal values between 6 and 16% for the two datasets considered. The performance is shaped in “steps”, decreasing as top_k increases. This shows that after certain thresholds the model merges some of the novel classes together, with values superior to 70% causing the model to predict a single class for all observations.

Then, we vary the $k\_neighbors$ in [1, 90]. This parameter controls the number of closest neighbors considered in the data augmentation (see Sec. III-C). Fig. 7(b) shows under 70, all values produce good results. The *Human* datasets appears to have high variance, but this is simply a result of the very small range of the performance (between 0.970 and 0.995).

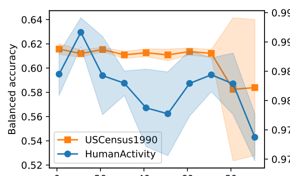
Finally, we study $w1$ and $w2$ in Figs. 7(c) and 7(d) by varying them in [0, 1]. They control the trade-off between the consistency loss and loss of the classification or clustering networks (see Sec.III-D). Because the classification task is secondary to the clustering problem, $w1$ has less influence on performance, and good values range between 0.3 and 0.9 for both datasets. On the other hand, $w2$ is very important for the accuracy of the method, and good values range between 0.8 and 0.9 for the *Census* dataset and between 0.4 and 0.9 for the *Human* dataset.

Influence of data representation for the k-means

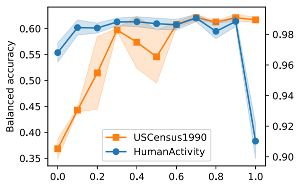
In table V, the quality of different data representations of MNIST are compared using k-means. We observe that after



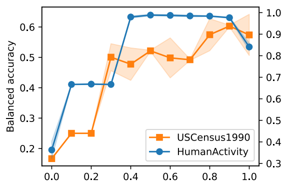
(a) Pseudo label top_k (in %)



(b) Data aug. $k\_neighbors$



(c) Trade-off $w1$



(d) Trade-off $w2$

Fig. 7. Visualization of the influence of each parameter on the performance on the Human Activity Recognition and 1990 US Census datasets.

Self-Supervised Learning, the encoder learned a representation of the original data where the k-means algorithm performs slightly better than on the original data (+2.8% in accuracy). And after joint training, the unknown classes are even more cleanly separated, as k-means gains +24.3% in accuracy over the original data representation.

TABLE V
CLUSTERING PERFORMANCE ON NEW CLASSES OF k-MEANS ON DIFFERENT DATA REPRESENTATIONS.

Method	k-means on MNIST			
Data representation	BACC (%)	ACC (%)	NMI	ARI
(1) Original data	60.1	61.1	0.48	0.38
(2) Latent w/ SSL	63.8	63.9	0.501	0.414
(3) Latent w/ joint	84.9	85.4	0.748	0.728

Performance of the k-means algorithm on different data representation of the MNIST dataset. (1) is the original data, (2) is the data projected by an encoder trained with Self-Supervised Learning (Sec. III-A), and (3) is the data projected by an encoder jointly trained by the proposed method (Sec. III-B)

Hyper-parameters values

We optimize the most important hyper-parameters for each dataset and report their values in table VI. The constants are: *batch size* = *512*, *encoder layers sizes* = *[d, d, d]*, α = *2.0*, *ssl lr* = *0.001*, *epochs* = *30* and p_m = *0.30*.

TABLE VI
BEST HYPER-PARAMETERS FOUND.

Parameter	MNIST	Forest	Letter	Human	Satimage	Pendigits	Census
cosine topk	15.015	19.300	2.019	15.277	6.214	5.609	13.96
w1	0.8709	0.1507	0.4887	0.4560	0.80	0.7970	0.8104
w2	0.6980	0.8303	0.9350	0.6335	0.8142	0.8000	0.9628
$lr_{classif.}$	0.009484	0.001876	0.009906	0.001761	0.007389	0.006359	0.005484
$lr_{cluster.}$	0.0009516	0.007191	0.007467	0.001017	0.008819	0.009585	0.0008563
k neighbors	4	9	6	15	11	10	7
dropout	0.01107	0.09115	0.07537	0.2959	0.4210	0.01652	0.06973
activation fct.	Sigmoid	Sigmoid	ReLU	ReLU	ReLU	ReLU	ReLU

“cosine topk” is here a percentage of the max number of pairs of instances in a batch.