A Unified Objective for Novel Class Discovery

Enrico Fini[1] Enver Sangineto[1] Stéphane Lathuilière[2] Zhun Zhong[1]* Moin Nabi[3] Elisa Ricci[1,4]
[1] University of Trento, Trento, Italy [2] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[3] SAP AI Research, Berlin, Germany [4] Fondazione Bruno Kessler, Trento, Italy

Abstract

In this paper, we study the problem of Novel Class Discovery (NCD). NCD aims at inferring novel object categories in an unlabeled set by leveraging from prior knowledge of a labeled set containing different, but related classes. Existing approaches tackle this problem by considering multiple objective functions, usually involving specialized loss terms for the labeled and the unlabeled samples respectively, and often requiring auxiliary regularization terms. In this paper we depart from this traditional scheme and introduce a UNified Objective function (UNO) for discovering novel classes, with the explicit purpose of favoring synergy between supervised and unsupervised learning. Using a multi-view self-labeling strategy, we generate pseudo-labels that can be treated homogeneously with ground truth labels. This leads to a single classification objective operating on both known and unknown classes. Despite its simplicity, UNO outperforms the state of the art by a significant margin on several benchmarks (≈+10% on CIFAR-100 and +8% on ImageNet). The project page is available at : https://ncd-uno.github.io.

1. Introduction

Deep learning has enabled astounding progresses in computer vision. However, the necessity of large annotated training sets for these models is often a limiting factor. For instance, training a deep neural network for classification requires a large amount of labeled data for each class of interest. This constraint is even more severe in scenarios where collecting sufficient data for each class is expensive or even impossible, as for instance in medical applications.

To alleviate these problems, *Novel Class Discovery (NCD)* [7, 6, 8] has recently emerged as a practical solution. NCD aims at training a network that can simultaneously classify a set of labeled classes while discovering new ones in an unlabeled image set. The basic motivation behind this setting is that the network can benefit from the supervision available on the labeled set to learn rich image representations that can be transferred to discover unknown classes in the unlabeled set. At training time, data are split into a set of labeled images and a set of unlabeled images, assuming disjoint sets of classes. These two sets are used to train a single network to classify both the known and the unknown classes. Note that this problem is similar but different from *semi-supervised learning* [22, 24], because, in the latter, the working assumption is that labeled and unlabeled sets share the same classes. Differently, in NCD, the two sets of classes are supposed to be disjoint. Moreover, differently from common *clustering* [1, 25] scenarios, in an NCD framework, labeled data can be utilized at training time, and the challenge consists in transferring the supervised knowledge on the known classes to improve clustering of the unknown ones.

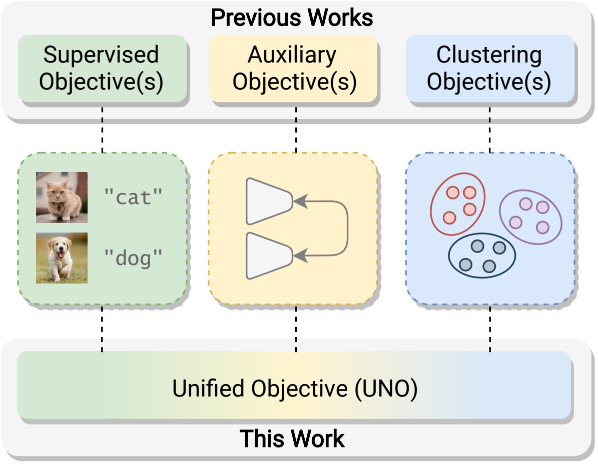

Figure 1: A visual comparison of our *UNified Objective (UNO)* with previous works. Existing approaches tackle NCD using multiple objective functions such as *supervised*, *clustering* and *auxiliary* objectives. On the contrary, we devise a single classification objective operating on both known and unknown classes.

Most NCD methods usually perform an initial *supervised pretraining* step on the labeled set, followed by a *clus-*

[*] Corresponding author

tering step on the unlabeled data [8, 10, 11]. This simple pipeline provides an effective means to transfer the representation capability from the labeled set to the unlabeled one. Generally speaking, these approaches combine two separated objectives. On the one hand, there is direct supervision through labels on the labeled set. On the other hand, a clustering objective is used to discover the novel categories. Clustering objectives are generally based on pseudo-labels [7, 12, 14, 29, 30, 31] estimated on the unlabeled set. In practice, these objectives are combined through independent losses such as cross-entropy (CE) and binary cross-entropy (BCE), respectively. Usually, the BCE loss is computed with pseudo pairwise labels often determined by setting an ad-hoc threshold which heavily influences the performance of these methods.

Additionally, NCD approaches generally require a strong semantic similarity between labeled and unlabeled classes in order to obtain expressive representation for discovering new concepts. In order to decrease the bias of the features toward known classes, Han *et al*. [7] propose to use an additional phase of *self-supervised pretraining* on *all* available images, both labeled and unlabeled, before the *supervised pretrain*. Moreover, the clustering stage is strengthened with another self-supervised objective (consistency), which enforces the model to output similar predictions for two different data augmentations of the same image. Adding an additional auxiliary objective makes optimization of this model even more cumbersome as it requires to further tune the hyper-parameters for each of these competing objectives. Moreover, this method assumes the availability of the unlabeled set in the pretraining stage. This is not suitable when it comes to learning in a sequential fashion as it requires to repeat the costly self-supervised pretraining stage every time that the unlabeled set changes.

Motivated by the need of simplifying NCD approaches, and inspired by the recent advancements in self-supervised learning [2, 3], in this paper we propose to eliminate the self-supervised pretraining step and unify all the objectives through a single loss function (see Fig. 1). Specifically, using a multi-view self-labeling strategy, we generate pseudo-labels that can be treated homogeneously with ground truth labels. This makes it possible to use a unified cross-entropy loss on both the labeled and the unlabeled set. In more detail, given a batch of images, we generate two views of each image using random transformations. Then, our network predicts a probability distribution over all classes (*labeled* + *unlabeled*) for each view. This results in two sub-batches that are independently clustered, so the cluster-assignment of each view is simply used as the pseudo-label for the other view. Ground truth and pseudo-labels are then used in combination to provide feedback to the network and update its parameters. Importantly, using a unified framework that operates on the complete class set enables us to learn a single model that can jointly recognize both labeled and unlabeled classes. We emphasize that this is a key point that is often neglected in the existing solutions for the NCD task.

Contributions. Our contributions can be summarized as follows: **(i)** we introduce a UNified Objective (UNO) for NCD where cluster pseudo-labels are treated homogeneously with ground truth labels, allowing a single CE loss to operate on both labeled and unlabeled sets; **(ii)** using multi-view, multi-head and over-clustering strategies we learn powerful representations while discovering new classes, de facto eliminating the need for self-supervised pretraining in NCD; **(iii)** experimentally, we show that our method surpasses all previous works on three publicly available benchmarks by a large margin. Notably, we outperform previous methods by 8% in accuracy on ImageNet, and by $\approx$+10% on CIFAR-100. **(iv)** Finally, we push NCD to the limit by changing the proportion of labeled and unlabeled samples, and find that our objective outperforms the state-of-the-art even more significantly on complex benchmarks.

2. Related Work

Novel Class Discovery. The concept of Novel Class Discovery was first formally introduced by Han *et al.* in [8], but the study of NCD can be dated back to the works in [10, 11]. In [10], Hsu *et al*. introduce the problem of transferring clustering models over tasks, which corresponds to the NCD setting: the goal is to cluster an unlabeled dataset given a labeled dataset without class-overlap. In [10, 11], a prediction network is trained on labeled data, which is then used to estimate pairwise similarities between unlabeled samples. Finally, the clustering network is trained to recognize novel classes in unlabeled dataset by using the predicted pairwise similarities. The main difference between [10] and [11] lies in the choice of the training loss applied to the prediction network.

More recently, Han *et al*. [8] address the same problem proceeding in two steps: a data embedding is learned on the labeled data using a metric learning technique, and then fine-tuned while learning the cluster assignments on the unlabeled data. Interestingly, they also tackle the problem of estimating the number of classes in the unlabeled dataset. Latter, many NCD works [7, 14, 29, 30, 31] are designed following the two-step training strategy. Han *et al*. [7] find that pretraining the backbone network in a self-supervised manner using rotation prediction can significantly improve the clustering accuracy. In addition, they employ rank statistics to identify data pairs that belong to the same class and minimize BCE to bring the network output closer for these pairs similarly to [11]. This pseudo-labeling loss is minimized together with a CE loss on the labeled set and a consistency loss that enforces network invariance to some random data transformations. OpenMix [31] generates virtual samples by mixing the labeled and unlabeled data, which can resist the noisy labels of unlabeled data. To leverage more positive samples, Zhong *et al.* [30] introduce Neighborhood Contrastive Learning (NCL) to aggregate pseudo-positive pairs with contrastive learning.

From this literature review, we observe that existing methods commonly need to (i) learn the classifier of the novel classes with pairwise relations between unlabeled samples, (ii) use a consistency loss to enforce network invariance to data transformations, and (iii) jointly train the network with several losses. Different from them, in this work, we propose a unified framework which enforces data-transformation consistency via a pseudo-labeling process by using a single objective.

Deep Clustering. Identifying classes in an unsupervised manner can be formalized as a clustering problem. Deep Cluster [1] can be considered as the first clustering method capable of learning rich image representations with deep networks without supervision. This approach alternates between a k-mean step that provides pseudo-labels and a network training step where cluster assignments are used as supervision. More recently, Van Gansbeke *et al.* [25] also show that a two-step approach where feature learning and clustering are decoupled can lead to state-of-the-art performance. Several approaches have been proposed to avoid this iterative process. The training stability of Deep Cluster is improved in [28] thanks to an online training formulation. A deep clustering network is trained in an end-to-end manner in [13] and [20] thanks to a mutual-information maximization objective. Other approaches propose more sophisticated pseudo-labeling strategies: Asano *et al.* [27] employ an optimal transport formulation to obtain robust pseudo-labels. This formulation is based on the Sinkhorn-Knopp algorithm [4] that maps sample representations to prototypes. Caron *et al.* [2] propose to use this clustering algorithm to introduce a ”swapped” prediction mechanism that uses two random transformations of the same images, referred to as views. Cluster assignments are estimated for each view and are used as a pseudo-label for the other view. In this work, we take advantage of this swapped prediction mechanism to obtain pseudo-labels for the unlabeled set but we incorporate to this mechanism the network-head corresponding to the labeled classes to guide clustering.

3. Method

In the NCD task, training data are split in two sets: a labeled set $D^l = \{(\mathbf{x}_1^l, y_1^l), ..., (\mathbf{x}_N^l, y_N^l)\}$ and an unlabeled set $D^u = \{\mathbf{x}_1^u, ..., \mathbf{x}_M^u\}$, where each $\mathbf{x}_i^l$ in D^l and $\mathbf{x}_j^u$ in D^u is an image and $y_i^l \in \mathcal{Y}^l = \{1, ..., C^l\}$ is a categorical label. The one-hot representation of y_i^l is denoted as $\boldsymbol{y}_i^l$. The goal is to use D^u to discover C^u clusters, where C^u is known a priori. The set of C^l labeled classes is assumed to be disjoint from the set of C^u unlabeled classes. Note that, at test time, we aim at classifying images corresponding to both labeled and unlabeled classes. We formulate this problem as learning a mapping from the image domain to the complete-label set $\mathcal{Y} = \{1, ..., C^l, C^l\!+\!1, ..., C^l\!+\!C^u\}$, where the first C^l elements correspond to $\mathcal{Y}^l$, while the subsequent C^u elements correspond to latent classes which should emerge from the clustering process.

In the following subsections, we first introduce how our Unified Objective learns this mapping (Sec. 3.1), then we explain how to use a multi-view self-labeling strategy to obtain strong pseudo-labels (Sec. 3.2), and finally we show how the performance of our method can be boosted by using multi-head clustering and overclustering (Sec. 3.3).

3.1. Unified Objective

To solve the NCD problem, we propose to train a neural network f_θ, parametrized by θ, which computes the posterior probabilities over $\mathcal{Y}$: $f_\theta(\mathbf{x}) = \{p(y|\mathbf{x}); y \in \mathcal{Y}\}$. Our network architecture is shown in Fig. 2: it is composed of a shared encoder E and two heads, h and g. The encoder E is a standard convolutional network (CNN) followed by an average pooling layer, and $\boldsymbol{z} = E(\mathbf{x})$, $\boldsymbol{z} \in \mathbb{R}^k$, is a feature vector representing the input image $\mathbf{x}$. The first head h is a linear classifier with C^l output neurons. On the other hand, g is implemented using a Multilayer Perceptron (MLP), that projects $\boldsymbol{z}$ to a lower dimensional representation $\boldsymbol{z}'$, and a linear classifier with C^u output neurons. Following [2, 26], we $l2$-normalize $\boldsymbol{z}$, $\boldsymbol{z}'$ and the linear classifiers.

Importantly, the logits $\boldsymbol{l}_h \in \mathbb{R}^{C^l}$ and $\boldsymbol{l}_g \in \mathbb{R}^{C^u}$ respectively produced by h and g are concatenated: $\boldsymbol{l} = [\boldsymbol{l}_h, \boldsymbol{l}_g]$. Then, they are fed to a shared softmax layer σ which outputs a posterior distribution over the complete-label set $\mathcal{Y}$: $\boldsymbol{p} = \sigma(\boldsymbol{l}/\tau)$, where τ is the temperature of the softmax. Once we have $\boldsymbol{p}$, we can train the whole network f using standard cross-entropy:

$$\ell(\mathbf{x}, \boldsymbol{y}) = -\sum_{c=1}^{C} \boldsymbol{y}_c \log(\boldsymbol{p}_c), \tag{1}$$

where $C = C^l + C^u$. $\boldsymbol{y}_c$ and $\boldsymbol{p}_c$ are the c-th elements of the label $\boldsymbol{y}$ and the network prediction $\boldsymbol{p} = f(\mathbf{x})$, respectively. The label $\boldsymbol{y}$ used for image $\mathbf{x}$ depends on whether $\mathbf{x} \in D^l$ or $\mathbf{x} \in D^u$. If $\mathbf{x}$ belongs to the labeled dataset we apply zero-padding to $\boldsymbol{y}^l$, while if $\mathbf{x} \in D^u$, we zero-pad the pseudo-label $\hat{\boldsymbol{y}}$ associated with $\mathbf{x}$:

$$\boldsymbol{y} = \begin{cases} [\boldsymbol{y}^l, \mathbf{0}_{C^u}] & \mathbf{x} \in D^u \\ [\mathbf{0}_{C^l}, \hat{\boldsymbol{y}}] & \mathbf{x} \in D^l. \end{cases} \tag{2}$$

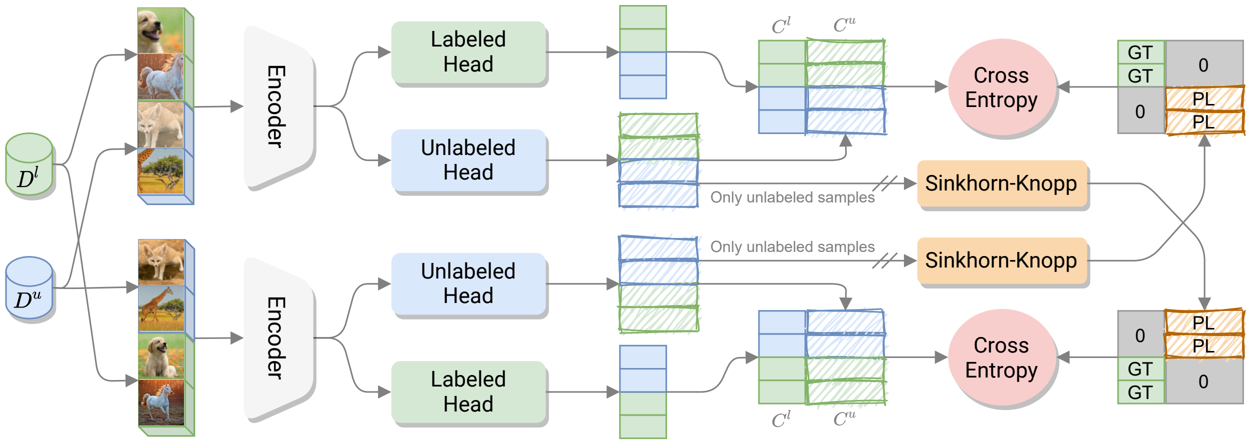

Figure 2: Overview of the proposed architecture. In green we represent the "labeled components" (labeled subset D^l, labeled head h, labeled samples), in blue their unlabeled counterparts (unlabeled subset D^u, unlabeled head g, unlabeled samples) and in orange the pseudo-labeling algorithm and its outputs. Sketchiness indicates uncertainty in the unlabeled logits and pseudo-labels. The parameters of the encoder E and the heads (h and g) are shared for the two views.

Here, $\mathbf{0}_{C^u}$ and $\mathbf{0}_{C^l}$ denote zero vectors of dimension C^u and C^l, respectively. This padding formulation is a natural choice, which derives from the assumption that the known and unknown classes are disjoint.

3.2. Multi-view and Pseudo-labeling

In this section, we show how a multi-view strategy can be leveraged to generate pseudo-labels for our Unified Objective. Given an image $\mathbf{x}$, we adopt common data-augmentation techniques, consisting in applying random cropping and color jittering to $\mathbf{x}$, and we obtain two different "views" of $\mathbf{x}$, which are resized to the original size and fed to f. These data-augmentation techniques, originally exploited in the self-supervised learning field [3], have recently been successfully applied also to standard supervised learning [16]. Coherently, we extract two views $\mathbf{v}_1$ and $\mathbf{v}_2$ from $\mathbf{x}$, both when $\mathbf{x} \in D^l$ and when $\mathbf{x} \in D^u$ (see Fig. 2).

In case $(\mathbf{x}, \boldsymbol{y}^l) \in D^l$, we associate $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ with the same label $\boldsymbol{y}_1 = \boldsymbol{y}_2 = [\boldsymbol{y}^l, \mathbf{0}_{C^u}]$. On the other hand, if $\mathbf{x} \in D^u$, then we use $\boldsymbol{v}_1$ to compute $\hat{\boldsymbol{y}}_1$ and $\boldsymbol{v}_2$ to compute $\hat{\boldsymbol{y}}_2$ and then we plug both pseudo-labels in Eq. (2). At this point, Eq. (1) can be applied independently for each view. However, this approach does not encourage the network to output consistent predictions for different views of the same image. In order to enforce such behavior, following [2], we use the *swapped prediction task*:

$$\ell(\boldsymbol{v}_1, \boldsymbol{y}_2) + \ell(\boldsymbol{v}_2, \boldsymbol{y}_1). \tag{3}$$

When we evaluate each term in the above formula, we apply a "stop-gradient" for the pseudo-label, *i.e.*, the gradient flows only though $f(\boldsymbol{v}_1)$. Note that these two loss terms are instances of the same objective applied to different views.

Regarding the computation of pseudo-labels, a naïve solution to obtain $\hat{\boldsymbol{y}}_1$ given $\boldsymbol{v}_1$, would be to simply use the predictions of $g(\boldsymbol{z}_1)$, with $\boldsymbol{z}_1 = E(\mathbf{v}_1)$. Let $\boldsymbol{p}_g^1 = \sigma(\boldsymbol{l}_g^1/\tau)$, where $\boldsymbol{l}_g^1$ are the logits computed by $g(\boldsymbol{z}_1)$ and the softmax operation is applied to *only* the C^u output neurons of $g(\boldsymbol{z}_1)$. We may set $\hat{\boldsymbol{y}}_2 = \boldsymbol{p}_g^1$ and use $\hat{\boldsymbol{y}}_2$ in Eq. (2) to get $\boldsymbol{y}_2$. However, as observed in [27], this pseudo-label assignment may lead to degenerate solutions, in which, *e.g.*, g always predicts the same logits vector, for any input. In this case, in Eq. (1), the network prediction and the labels are basically the same and there is no learning. Instead, inspired by [2, 27], when computing $\hat{\boldsymbol{y}}_2$, we add an entropy term which penalizes those situations in which all the logits are equal to each other and incentives a uniform partition of the pseudo-labels over all the C^u clusters. Specifically, let $\boldsymbol{L} = [\boldsymbol{l}_g^1, ..., \boldsymbol{l}_g^B]$ be the matrix whose columns are the logits computed by g with respect to a mini-batch of images of size B. Moreover, let $\hat{\mathbf{Y}} = [\hat{\boldsymbol{y}}_1, ..., \hat{\boldsymbol{y}}_B]^\top$ be the matrix whose rows are the unknown pseudo-labels of the current batch. $\hat{\mathbf{Y}}$ is found by solving:

$$\hat{\mathbf{Y}} = \max_{\mathbf{Y} \in \Gamma} \operatorname{Tr}(\mathbf{Y}\boldsymbol{L}) + \epsilon \operatorname{H}(\mathbf{Y}), \tag{4}$$

where $\epsilon > 0$ is an hyper-parameter, H is the entropy function which is used to "scatter" the pseudo-labels, Tr is the trace function, and Γ is the transportation polytope defined as:

$$\Gamma = \{\mathbf{Y} \in \mathbb{R}_+^{C^u \times B} | \mathbf{Y}\mathbf{1}_B = \frac{1}{C^u}\mathbf{1}_{C^u}, \mathbf{Y}^\top \mathbf{1}_{C^u} = \frac{1}{B}\mathbf{1}_B\}. \tag{5}$$

These constraints enforce that, on average, each cluster is selected $\frac{B^u}{C^u}$ times in the batch, where B^u is the number of unlabeled samples in the batch. The solution to Eq. (4) is obtained using the Sinkhorn-Knopp algorithm [4] (for more details, we refer to [27]). The resulting pseudo-labels, represented by each row $\hat{\boldsymbol{y}}_i$ in $\hat{\mathbf{Y}}$ can then be discretised. However, we found that the best performance can be achieved using soft pseudo-labels $\hat{\boldsymbol{y}}_i \in [0,1]^{C^u}$.

3.3. Multi-head Clustering and Overclustering

In order to boost the clustering performance, inspired by [13], jointly with the main clustering task, we also adopt overclustering, *i.e.*, we force f to produce an alternative partition of the unlabeled data which is more fine-grained. This is known to enhance the quality of the representations. Specifically, an overclustering head o, connected with E, is similar to g, but with $K = C^u \times m$ cluster output neurons.

Additionally, inspired by [1, 13], we also use multiple clustering $(g_1, ..., g_n)$ and overclustering $(o_1, ..., o_n)$ heads. This is useful because heads could converge to suboptimal clustering configurations. By using multiple heads we can smooth down this effect and increase the overall signal backpropagated to the shared part of the network. At training time, for a given batch of data, we iterate over $g_1, ..., g_n$ and, for each head g_i, we concatenate the logits produced by h ($\boldsymbol{l}_h$) with the logits produced by g_i (*i.e.* $\boldsymbol{l}_{g_i}$). We feed the result to the $C^l + C^u$-element softmax layer and, following the procedure described above, we compute Eq. (1) for each $\mathbf{x}$ in the batch. Similarly, for each o_j, we concatenate $\boldsymbol{l}_h$ with the logits produced by o_j (*i.e.* $\boldsymbol{l}_{o_j}$), we use a $C^l + K$-element softmax layer and we again compute Eq. (1) for each $\mathbf{x}$ in the batch.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate the performance of our method on three well-established NCD benchmarks following [8, 7], *i.e.*, CIFAR10 [17], CIFAR100 [17] and ImageNet [5]. Each dataset is divided into two subsets, the labeled set that contains labeled images belonging to a set of known classes, and an unlabeled set of novel classes for which we do not possess any supervision except for the number of classes. The details of the splits are shown in Tab. 1. Standard data splits used in the literature (1, 2 and 4 in Tab. 1) exhibit either (i) a *small number of classes* or (ii) a *strong imbalance* in the number of classes of the two subsets (the labeled set is much larger than the unlabeled set). However, these two assumptions do not hold in real-world scenarios, where the unlabeled data is far more abundant than its labeled counterpart. Hence, we introduce a new split that better approximates practical applications for NCD: CIFAR100-50. As shown in Tab. 1, CIFAR100-50 contains a large number of unlabeled classes (50), making the task more challenging.

Dataset	Labeled		Unlabeled	
	Images	Classes	Images	Classes
(1) CIFAR10	25K	5	25K	5
(2) CIFAR100-20	40K	80	10K	20
(3) CIFAR100-50	25K	50	25K	50
(4) ImageNet	1.25M	882	≈30K	30

Table 1: Statistics of the datasets and splits used in our novel class discovery benchmark.

In our experiments, we show that the performance of all methods in the proposed split drops considerably in comparison to the easier one (CIFAR100-20). This gives evidence that current solutions for NCD are not yet ready for deployment. Even more challenging settings are analyzed in Fig. 3.

Protocol. We evaluate our model using two evaluation settings: **task-aware** and **task-agnostic**. In the task-aware evaluation, we use the task information to exclude those outputs that are not relevant for the current sample. In other words, given an image belonging to the labeled classes we only consider the labeled classifier, and viceversa for images belonging to unlabeled classes we only consider the output of the unlabeled heads. This evaluation is typically used in the literature. However, in practical scenarios this evaluation is not very meaningful because it does not assess if the model is able to discriminate labeled from unlabeled classes. Hence, we also report task-agnostic accuracy, where the prediction is simply the most likely output after concatenating labeled and unlabeled logits. For the task-aware protocol, we report performance on the training set (Tab. 3) and test set (Tab. 2), while for the task-agnostic protocol and we report performance on the test set (Tab. 2 and Tab. 4). The results in Tab. 2 are averaged over 3 runs, while for Tab. 3 and 2 are averaged over 10 runs following the protocol of [7].

Metrics. We use the accuracy measure for labeled samples and the average clustering accuracy for unlabeled samples. The average clustering accuracy is defined as:

$$\text{ClusterAcc} = \max_{perm \in P} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left\{y_i = perm\left(\hat{y}_i\right)\right\}, \quad (6)$$

where y_i and $\hat{y}_i$ represent the ground-truth label and predicted label of a sample $\mathbf{x}_i \in D^u$, respectively. P is the set of all permutations, which is computed with the Hungarian algorithm [18]. Since we train the network using multiple heads, we compute evaluation metrics independently for each head and report both average accuracy and best head accuracy. We define the best head as the one exhibiting lowest training loss in the last epoch.

Implementation Details. For a fair comparison with existing methods, we use a ResNet18 [9] encoder for all datasets.

Method	Concat	Over	Aug	CIFAR10						CIFAR100-50					
				Task-aware			Task-agnostic			Task-aware			Task-agnostic		
				Lab	Unlab	All	Lab	Unlab	All	Lab	Unlab	All	Lab	Unlab	All
UNO	✗	✓	Strong	90.6	89.9	90.2	48.5	83.3	65.9	78.4	44.5	62.4	65.5	43.2	54.3
	✓	✗	Strong	96.4	93.0	94.7	**93.5**	90.5	92.0	**78.9**	49.8	64.4	**71.5**	48.4	59.0
	✓	✓	Weak	96.1	92.7	94.4	93.4	90.2	91.8	78.4	50.6	64.5	71.1	48.6	59.1
	✓	✓	Strong	**96.6**	**95.1**	**95.8**	**93.5**	**93.3**	**93.4**	78.8	**52.0**	**65.4**	**71.5**	**50.7**	**61.1**

Table 2: Ablation study. Each core component of our method is removed in isolation. “Concat” stands for the concatenation of the logits before evaluating the softmax layer, “Over” means overclustering and “Aug” is short for augmentation. We also report the performance of the full model to enable comparison. All the results reported in this table are measured on the test set and using the best head.

The labeled head h is an $l2$-normalized linear layer with C^l output neurons, while the unlabeled head g is composed of a projection head with 2048 hidden units and 256 output units, followed by a $l2$-normalized linear layer with C^u output neurons. We pretrain our model for 200 epochs on the labeled dataset and then train for 200 epochs in the discovery phase on both labeled and unlabeled datasets. For both phases we use SGD with momentum as optimizer, with linear warm-up and cosine annealing ($lr_{base} = 0.1$, $lr_{min} = 0.001$), and weight decay 10^{-4}. The batch size is always set to 512 for all experiments. Regarding the discovery phase, we use an overclustering factor $m = 3$ and a $n = 4$ heads for both clustering and overclustering. The temperature parameter τ is set to 0.1 for all softmax layers. For what concerns pseudo-labeling, we use the implementation of the Sinkhorn-Knopp algorithm [4] provided by [2] and we inherit all the hyperparameters from [2], *e.g.* $n\_iter = 3$ and $\epsilon = 0.05$.

Pretraining. In [7], a three stage pipeline was proposed, where the network is first trained with self-supervision on the combination of labeled and unlabeled sets, then fine-tuned on the labeled set, and finally used to discover new classes. This complex procedure makes NCD cumbersome and costly. In addition, it is based on the assumption that the unlabeled data is available at training time, which might not always hold in real world scenarios (*e.g.* online settings). For these reasons we decide not to use self-supervised pretraining in our method and show that it is enough to use a unified objective at discovery time in order to obtain the best performance. Nonetheless, for the sake of completeness, we examine the behavior of our model with self-supervised pretraining, finding no improvement with respect to simpler and more practical supervised pretraining. In our method, the labeled head h is composed by $l2$-normalized prototypes and the last layer computes the cosine similarity between each prototype and the $l2$-normalized features $\boldsymbol{z}$. For consistency and performance, we also normalize such features and prototypes during pretraining.

4.2. Ablation study

In Tab. 2 we report the results of an ablation study, obtained by removing each core component of our method in isolation, *i.e.*, logit concatenation, overclustering, and augmentation. For better inspecting the behavior of our model, we report clustering accuracy on both test subsets (labeled and unlabeled) as well as using the two proposed evaluation settings (task-aware and task-agnostic).

Logit Concatenation. As described in Sec. 3.1, our main contribution is a unified objective for training on labeled and unlabeled data jointly. In other words, our model works by concatenating labeled and unlabeled logits in order to predict a posterior probability distribution over all classes. We experimentally demonstrate that this design choice is indeed crucial for the final performance of our method. The ablation shows that treating clustering and supervised learning with different objectives is highly suboptimal with respect to using our unified objective. In particular, the results point out two aspects. First, it is clear that using separated objectives yields greater interference between the two tasks. Importantly, this effect is also present when evaluating them separately (task-aware). Second, performance drops more dramatically when using the task-agnostic evaluation, especially on the labeled set. This latter result is justified by the fact that training without concatenation does not encourage the network to distinguish labeled from unlabeled samples.

Overclustering. In Sec. 3.3 we described how we extract fine-grained clusters from the unlabeled data. This is known to significantly boost the quality of the representation [13, 2]. We investigate this effect in our framework, finding evidence that overclustering can be effectively leveraged in NCD. Tab. 2 shows that the performance on the unlabeled set is improved when the overclustering heads are used. Note that clustering heads are retained and used for evaluation, while overclustering heads are discarded at test time. Interestingly, the performance on the labeled classes does not benefit from fine-grained cluster extraction. This is

Method	CIFAR10	CIFAR100-20	CIFAR100-50	ImageNet
k-means [19]	72.5±0.0	56.3±1.7	28.3±0.7	71.9
KCL [10]	72.3±0.2	42.1±1.8	-	73.8
MCL [11]	70.9±0.1	21.5±2.3	-	74.4
DTC [8]	88.7±0.3	67.3±1.2	35.9±1.0	78.3
RS [7]	90.4±0.5	73.2±2.1	39.2±2.3	82.5
RS+ [7]	91.7±0.9	75.2±4.2	44.1±3.7	82.5
UNO (avg)	**96.1±0.5**	**84.5±1.0**	**52.8±1.4**	**89.2**
UNO (best)	**96.1±0.5**	**85.0±0.6**	**52.9±1.4**	**90.6**

Table 3: Comparison with state-of-the-art methods on CIFAR-10, CIFAR-100 and ImageNet for novel class discovery using task-aware evaluation protocol. Clustering accuracy is reported on the unlabeled set (training split). All methods except UNO initialize the encoder with self-supervised learning, except when evaluated on ImageNet. “RS+” is [7] with incremental classifier.

Method	CIFAR10			CIFAR100-20			CIFAR100-50		
	Lab	Unlab	All	Lab	Unlab	All	Lab	Unlab	All
KCL [10]	79.4	60.1	69.8	23.4	29.4	24.6	-	-	-
MCL [11]	81.4	64.8	73.1	18.2	18.0	18.2	-	-	-
DTC [8]	58.7	78.6	68.7	47.6	49.1	47.9	30.2	34.7	32.5
RS+ [7]	90.6	88.8	89.7	71.2	56.8	68.3	69.7	40.9	55.3
UNO (avg)	**93.5**	**93.3**	**93.4**	**73.2**	**72.7**	**73.1**	**71.5**	**50.6**	**61.0**
UNO (best)	**93.5**	**93.3**	**93.4**	**73.2**	**73.1**	**73.2**	**71.5**	**50.7**	**61.1**

Table 4: Comparison with state-of-the-art methods on CIFAR-10 and CIFAR-100 on both labeled and unlabeled classes, using task-agnostic evaluation protocol. Accuracy and clustering accuracy are reported on the test set.

reasonable because good representations of the labeled data are already learned using supervision. However, the overall accuracy is consistently higher when using overclustering, thus motivating our choice.

Data augmentation. Very recently, the usage of strong data augmentation techniques has been thoroughly investigated in the context of self-supervised learning [3]. At the same time, unsupervised clustering techniques based on self-supervision have appeared in the literature [25, 21]. We follow these works and investigate the benefits of using SimCLR-like augmentations for NCD. First, we found that using very small crops does not improve the performance. Rather, it prevents the network from learning meaningful clusters. We believe this behavior is reasonable, because cropping occludes important parts of the image, making it hard for the network to produce meaningful predictions, which in turn decreases the quality of the pseudo-labels. Instead, using moderate random cropping (as found in [7]) turned out to be the best choice in our experiments. Moreover, we discover that using strong color jittering and greyscale is beneficial for our method.

In Tab. 2, we evaluate two types of data augmentation: **weak** (moderate random crop and random flip) versus **strong** (moderate random crop, flip, jittering, and greyscale). On ImageNet we also make use of Gaussian blur. From the results, it emerges that by using strong augmentations we consistently increase the accuracy of our method on both labeled and unlabeled sets. For a fair comparison, we also apply these strong transformations to RS [7] without obtaining performance improvements.

4.3. Comparison with the state-of-the-art

We compare our approach with the current state-of-the-art in NCD: including KCL [10], MCL [11], DTC [8], RS [7] and RS+ [7] (“incremental classifier” version of RS). In addition, we report the performance of k-means applied on top of pretrained features.

In Tab. 3, we focus on the unlabeled classes, reporting clustering accuracy on the training set (common practice in the literature [7, 8]). For all related methods we report results using self-supervised pretraining, as described in Sec. 4.1, except for ImageNet, on which we only report results using supervised pretraining. Results using supervised pretrainig only for related methods are deferred to the supplementary material. Despite its simplicity and the lack of self-supervised pretraining, UNO considerably outperforms the state-of-the-art (RS+ with self-supervised pretraining [7]), in some cases by ≈10%. On CIFAR10 the clustering error is reduced to roughly half, coming very close to supervised accuracy. On ImageNet, UNO reaches over 90.0% accuracy, a remarkable result given the complexity of the dataset. We believe such strong results validate our hypothesis that unifying clustering and supervised objectives is a more effective solution for NCD. The difference in performance between “avg” and “best” is negligible for simple datasets, while when it becomes larger for complex dataset with a higher number of classes.

We also thoroughly compare our method with the state-of-the-art in the task-agnostic evaluation setting on the test set. The results on CIFAR10 and CIFAR100-50 are shown in Tab. 4. For UNO, we report the average accuracy over the clustering heads. These results show that not only our method is better at clustering the unlabeled data, but it also outperforms the state-of-the-art on the labeled test set, showing that our unified objective favors better cooperation and less interference between labeled and unlabeled heads. Moreover, we notice that on CIFAR100-20 the relative improvement in clustering accuracy with respect to RS+ [7] is larger when using the task-agnostic evaluation with respect to the task-aware evaluation. This means that UNO also discriminates labeled from unlabeled classes better than the related methods.

Finally, in Fig. 3 we inspect the behavior of the best methods (Ours, RS and RS+) with an increasing number of

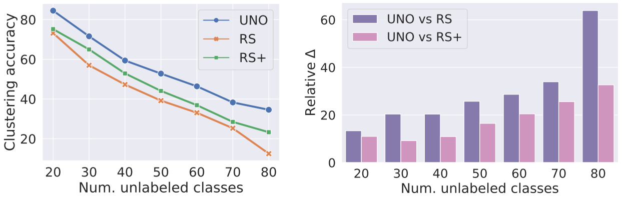

Figure 3: Clustering accuracy (left) and relative delta (right) with an increasing number of unlabeled classes. The relative delta is calculated as the margin between UNO and the related methods and normalized by the accuracy of UNO.

unlabeled classes C^u. The task-aware clustering accuracy of all methods decreases as the task becomes increasingly difficult. This happens for two reasons: first, clustering becomes harder with more classes; second, the number of labeled classes C^l decreases and in turn the quality of the representations learned with supervision is reduced. The latter is particularly disadvantageous for our model, since RS and RS+ are pretrained with self supervision on the union of the two datasets while ours only uses supervision on the labeled set. Despite this, our method always outperforms the other methods by large margins. Moreover, as shown in Fig. 3 (right), the relative gap between our method and the others grows larger when C^u increases, demonstrating the superiority of our objective even in complex scenarios.

4.4. Qualitative results

In addition to quantitative results, we also report a qualitative analysis showing the feature space learned by our unified objective on CIFAR10. In Fig. 4, we visualize the shared feature space (after the last convolutional block) and the concatenated of the logits $\boldsymbol{l}$ of the heads h and g. Since we have multiple unlabeled heads, for all data samples we concatenate the logits of all heads for visualization purposes. For the features, we run PCA [15] to reduce their dimensionality. Finally we project the data in two dimensions using t-SNE [23]. The same procedure is applied to the features produced by RS+ [7].

From the plot, it is clear that our model produces feature representations for samples of the same class that are more tightly grouped with respect to RS+. At the same time, in RS+ several classes are entangled together (*e.g.* cat, dog, horse), making it hard for a linear classifier to discriminate the samples. In accordance with our quantitative results, our method does a better job at separating the classes, both in the shared feature space and in the logits space.

Furthermore, we examine the influence that our architectural design choices have on the representations. It is clear from the plot that in the logits space samples are roughly uniformly distributed around the centroid of their class. On

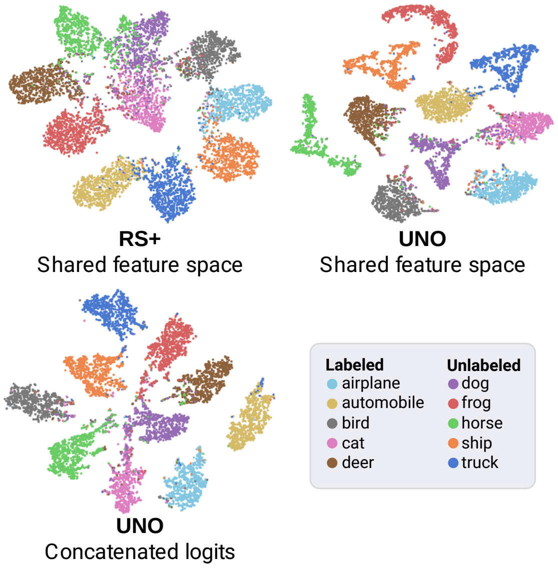

Figure 4: t-SNE visualization for all classes on CIFAR10. For both methods "Shared feature space" stands for the features after the last convolutional block.

the other hand, in the shared feature space, labeled samples are still organized in disk-shaped groups, while unlabeled samples exhibit more irregular shapes. This is justified by the fact that the labeled head h projects features linearly into logits, while the unlabeled head g contains multiple layers and non-linearities. Moreover, unlabeled samples in the feature space seem to be organized in subgroups. This is probably due to the use of overclustering.

5. Conclusions

We presented a simple approach for discovering and learning novel classes in an unlabeled dataset, while leveraging good features extracted with supervision in a labeled dataset. Our method stands out from the literature for the fact that we use pseudo-labels in combination with ground truth labels in a UNified Objective (UNO) that enables better cooperation and less interference between supervised and unsupervised learning. Moreover, we also removed the need for costly self-supervised pretraining, making NCD more practical. We demonstrated the effectiveness of our proposed approach through extensive experiments and careful analysis. We found that UNO outperforms all related methods significantly, despite being conceptually simpler and easier to implement and train.

Acknowledgements. This work is supported by the European Institute of Innovation & Technology (EIT) and the H2020 EU project SPRING - Socially Pertinent Robots in Gerontological Healthcare. This work was carried out under the "Vision and Learning joint Laboratory" between FBK and UNITN.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proc. ECCV*, 2018. 1, 3, 5

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proc. NeurIPS*, 2020. 2, 3, 4, 6

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. ICML*, 2020. 2, 4, 7

[4] Marco Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Proc. NeurIPS*, 2013. 3, 4, 6

[5] Jia Deng, R. Socher, Li Fei-Fei, Wei Dong, Kai Li, and Li-Jia Li. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 5

[6] K Han, SA Rebuffi, S Ehrhardt, A Vedaldi, and A Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *TPAMI*, 2021. 1

[7] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *Proc. ICLR*, 2020. 1, 2, 5, 6, 7, 8

[8] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proc. ICCV*, 2019. 1, 2, 5, 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 5

[10] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *Proc. ICLR*, 2018. 2, 7

[11] Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *Proc. ICLR*, 2019. 2, 7

[12] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proc. CVPR*, 2019. 2

[13] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proc. CVPR*, 2019. 3, 5, 6

[14] Xuhui Jia, Kai Han, Yukun Zhu, and Bradley Green. Joint representation learning and novel category discovery on single- and multi-modal data. In *Proc. ICCV*, 2021. 2

[15] Ian Jolliffe. *Principal Component Analysis*. 2011. 8

[16] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Proc. NeurIPS*, 2020. 4

[17] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *University of Tronto*, 2009. 5

[18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955. 5

[19] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proc. BSMSP*, 1967. 7

[20] Willi Menapace, Stéphane Lathuilière, and Elisa Ricci. Learning to cluster under domain shift. *Proc. ECCV*, 2020. 3

[21] Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Lsd-c: Linearly separable deep clusters. In *Proc. ICCV Workshop*, 2021. 7

[22] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Proc. NeurIPS*, 2020. 1

[23] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *JMLR*, 2014. 8

[24] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 2020. 1

[25] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Proc. ECCV*, 2020. 1, 3, 7

[26] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proc. ICML*, 2020. 3

[27] Asano YM., Rupprecht C., and Vedaldi A. Self-labelling via simultaneous clustering and representation learning. In *Proc. ICLR*, 2020. 3, 4, 5

[28] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proc. CVPR*, 2020. 3

[29] Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. *arXiv preprint arXiv:2107.03358*, 2021. 2

[30] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *Proc. CVPR*, 2021. 2, 3

[31] Zhun Zhong, Linchao Zhu, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *Proc. CVPR*, 2021. 2, 3