Long-Tailed Generalized Class Discovery using Vision Transformers and Sharpness-Aware Optimization

Lecture
Practical Applications of Deep Learning
Summer Semester 2022

Furkan Fahrettin Simsek
Niklas Kaspareit
Maximilian Kleissl

Supervisor:
PD Dr. Haojin Yang
Ziyun Li
Jona Otholt

September 4, 2022

Contents

1 Introduction

Deep learning has shown great success in various real-world applications and especially in the field of computer vision. However, the need for huge annotated training sets is regularly a limiting factor. Training a deep convolutional neural network for the task of image classification typically requires much labelled data. In domains such as medicine, where collecting data is expensive, this constraint requirement makes the training of the models hard.

Different approaches have been made to tackle this problem. Han et al. [HVZ19] formally introduce the image classification problem of Novel Class Discovery (*NCD*), where classes in the training set (*known classes*) and in the test set (*novel classes*) are disjoint. However, a more general problem is the task of Generalized Category Discovery (*GCD*) [VHVZ22], which requires the classes in the training set (known classes) to be a subset of the classes used in the test set (known and novel classes). Consequently, the final model has to be able to recognize both: novel and known classes. Fini et al. consider the problem of GCD as a realistic use case in many machine vision applications and overcome the assumption that for each sample in the test set it is known that the category is novel.

However, they assume the classes to be distributed evenly. This assumption is a restrictive prerequisite on many domains, where the natural distribution is not uniform. For instance, in medical applications, some pathologies occur much more frequently than others.

In order to overcome this limitation as well, we consider the problem of Long-Tailed Generalized Category Discovery (*LT-GCD*), where—in contrast to GCD—the test set is imbalanced and contains less novel classes then known classes.

To tackle this problem, we adapt the work of Fini et al. [VHVZ22] by replacing the first step of their classification-pipeline with ideas of Liu et al. [LHGM21a] in order to make their approach more robust to imbalanced data. Using this adaptation, we outperform the original approach on the LT-GCD task.

We summarize our contributions as follows: 1. the formalization of Long-Tailed Generalized Category Discovery (*LT-GCD*), a new and realistic setting for image recognition; 2. the establishment of strong baselines by adapting state-of-the-art techniques from standard GCD to this task; 3. a method to sample the CIFAR-100 according to our problem definition; and 4. the evaluation on a long-tailed version of the standard image recognition dataset CIFAR-100.

2 Related Work

2.1 Novel Class Discovery

Zhao and Han [ZH21] propose a two-branches model. One branch is used for global feature learning and the other for local feature learning. They conduct dual ranking statistics and mutual learning with these two branches for improved representation learning and new class discovery.

Rankstats [HRE+20, HRE+21] uses a three stage method to tackle the problem of NCD. The first stage is training the model with self-supervision on all data. This stage enables low-level representation learning. Then, the model is further trained with full supervision on labelled data. This enables high-level representation learning. The third stage is a joint learning stage and helps to transfer knowledge from labelled to unlabelled data using ranking statistics.

While most of the methods used to treat the NCD problem take into account several objective functions, which typically incorporate specialized loss terms for labelled and unlabelled samples, respectively, and often require auxiliary regularization terms. Fini et al. [FSL+21] use a multi-view self-labeling strategy, by generating 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. However, these approaches for NCD and their adapted versions are not able to perform well on the GCD task mostly due to the assumption in NCD, that the unlabelled data consists of novel classes only which does not have to hold true for GCD.

In addition to NCD, Vaze et al. [VHVZ22] very recently researched a more generalized setting in which, given a labelled and unlabelled set of images, the task is to categorize all images in the unlabelled set. Here, the unlabelled images may come from known classes and from novel ones as well. Our work mainly builds up on the findings and approaches presented in the aforementioned paper. Furthermore, Cao et al. [CBL21] have also researched this open-world setting and developed a semi-supervised learning approach called "'ORCA"' that uses an uncertainty adaptive margin mechanism to reduce the bias towards seen classes which is caused by learning discriminative features for seen classes faster than for the novel classes.

2.2 Reweighted Sharpness Aware Minimization (rwSAM)

Probably everyone can imagine that the problem of NCD is quite a complex task. Tasks like these are becoming more common in the area of Machine Learning. When the environment becomes more complex and uses more data, the algorithms require more and more parameters to optimize the models. It can be quite tedious to choose and fit those parameters and there is a high risk of overfitting. Additionally, loss functions are becoming more complex. Usually they are not convex and therefore they have multiple minima. Hence, an optimizer like stochastic gradient descent, Adam, or something

similar is required. However, we know that optimizers do not necessarily find global minima, but often end up in local minima. Pierre Foret et al. [FKMN20] suggest the new approach SAM that uses the geometry of loss-functions for better generalization. So, instead of going into a local sharp minimum, the algorithm prefers wide minima. The idea to achieve this is that the loss value and loss sharpness will be minimized simultaneously, by finding parameters with uniformly low loss values within a neighborhood. Hong Liu et al. [LHGM21a] use and improve the approach of SAM by also introducing and integrating a way of reweighting to the inner maximization step of the original approach. The idea here is to apply stronger regularization on rare samples. This will reduce the likelihood of overfitting on the more frequent samples.

3 Preliminary Background

In Chapter 2 we provided a general overview of the papers and algorithms that are required for understanding the following contents of this paper. With the given foundation we can have a closer look at the problem of long-tailed generalizes class discovery. So there is research that handles class discovery in a long-tailed setting. There is also research that can solve the problem of Generalized Category Discovery reasonably well. However combining these two problems yields new difficulties that have not been researched yet. We analyze how the existing algorithms actually perform within this new setting, then we will discuss possibilities for improvement and we show the results of experimenting with different combinations of algorithms to improve the metrics.

3.1 Class Distribution

Firstly we have to look at the dataset that has been used primarily for our experiments. We subsample the dataset CIFAR-100, which is described by Krizhevsky et al. [KH$^+$09], to obtain a long-tailed data set. The CIFAR-100 is a widely used dataset containing tiny images, ordered into 100 classes containing 500 training- and 100 test images each. For the Generalized Category Discovery task, we consider the first 80 classes as known, and the other 20 classes as novel classes. For long-tailed real-world use cases, we assume novel-classes to be less common than the known-classes in the training dataset. Taking this assumption into account in our sampling algorithm, we decided to use a step function s that assigns a class to the number of samples used in the long-tailed data set.

$$s(c) = \begin{cases} 500; \text{if } c \text{ is a known class} \\ 125; \text{if } c \text{ is a novel class} \end{cases}$$

We do not subsample the test set, because We decided to keep the labelled data set balanced, as (i) most scientifically used datasets are balanced, (ii) we expect better results, and (iii) we keep a maximum number of samples from the CIFAR-100, which

we subsampled from.. Therefore we can use the whole range of known classes in the test set. The first and third arguments may not hold for every real-world use case, as in datasets which are not artificially created; we might already have a long-tailed labelled dataset and do not want to artificially balance it. However, dealing with a long-tailed labelled dataset is left for future work. The GCD framework expects two training datasets, a labelled dataset and an unlabelled dataset. Consequently, we split the long-tailed training data set into a labelled set of 250 samples per known class and an unlabelled data set of 250 samples per known class and 125 samples per novel class. Obviously, only the unlabelled dataset is long-tailed then.

3.2 Generalized Category Discovery Pipeline

As mentioned in the related works section, we approached our *LT-GCD* problem by mainly building upon the work by Vaze et al. [VHVZ22]. They proposed a pipeline (see Figure 1) of different steps and components that we will shortly introduce in the following.

- **Transformer based backbone network.** The network is used to learn visual representations from the input images. Instead of using a classification head, the networks embedding space is the core component on which the actual class discovery task is performed on.
- **Supervised and Self-Supervised Contrastive Learning.** Methods (see Sections 3.3.1 and 4.1) used to learn to distinguish between classes based on visual similarities and dissimilarities of the images. Supervised contrastive learning is applied only on the labelled part of the train data while the self-supervised component is applied on all train data. These two learning techniques are then combined through a joint total loss function.
- **Feature Extraction and Semi-Supervised K-Means.** After the training is finished, GCD continues with extracting features from the original input images and then uses these features to perform clustering of the unlabelled train data using K-Means. These and above components of the GCD pipeline will be discussed in more detail in the following sections.

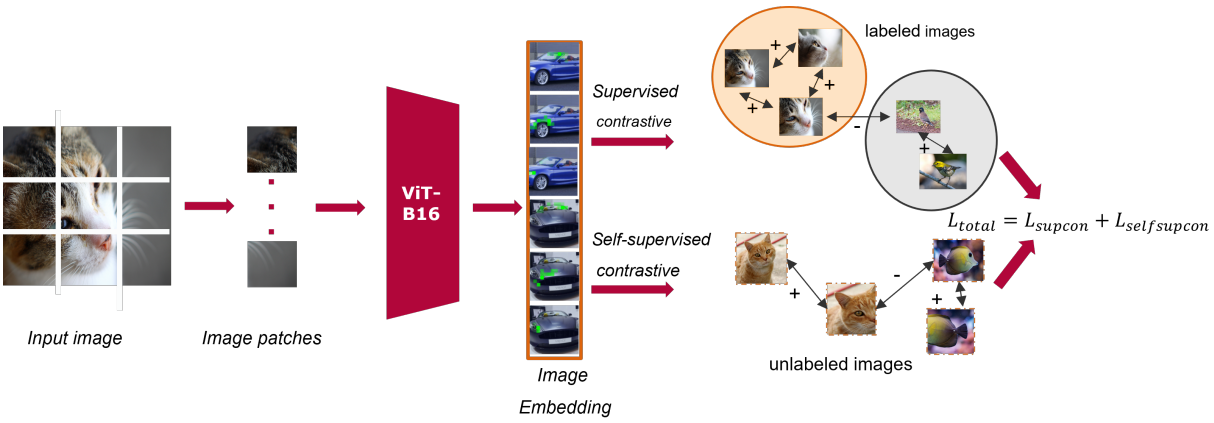


Fig. 1: Pipeline visualization of GCD.

3.3 Pretraining

In the following, we will give a brief overview on the main ideas and concepts behind the methods used during the pretraining phase in GCD as well as the idea behind sharpness-aware minimization.

3.3.1 Contrastive Learning

In GCD and many other works around Novel Class Discovery, approaches usually contain a pre-training phase in which a model learns discriminative features from input samples. To achieve this goal, the method of contrastively learning visual representations of an input image has recently been shown to yield strong performance in the unsupervised training of deep image models [WXYL18, CKNH20, HFLM+18]. As also described in [KTW+20], the common idea in contrastive learning is to pull together an anchor and a "positive" sample in the embedding space of a network, and push apart the anchor from many "negative" samples. In an unsupervised setting where labels are not available, a positive pair often consists of data augmentations of the sample (or the anchor), and negative pairs are formed by randomly chosen samples from the mini-batch. For the unlabelled part of the train data, GCD mostly follows the self-supervised contrastive learning framework *SimCLR* by Chen et al. [CKNH20]. In this framework, the goal is to learn representations by maximizing agreement between differently augmented views of the same data example via contrastive loss. Four major components simply illustrate this framework:

- **Data Augmentation**: Set of augmentations used to generate two correlated *views* x_i and x_j of an input image. Augmentations usually include *random cropping*, *resizing*, *random color distortions* and *gaussian blur*. The correlated views are considered as positive pairs, while the views generated from other input images are considered as negative pairs.

- **Encoder Neural Network** $f(\cdot)$: Used to extract representation vectors h_i and h_j from the augmented views of an input image, i.e. $h_i = f(x_i)$
- **Neural Network projection head** $g(\cdot)$: Usually a small multilayer perceptron (MLP) used to map the representations from the encoder network to a space where contrastive loss can be applied, i.e. to obtain $z_i = g(h_i)$.
- **Contrastive Loss Function** $L_{i,j}$: Defined for a contrastive prediction task. Given a mini-batch of N input images, the loss is applied on the pairs of augmented examples derived from the mini-batch yielding $2N$ data points. Since here the other augmented pairs are treated as negative examples, this leads to a total of $2(N-1)$ negative examples. Given x_i, the contrastive prediction task aims to identify the other positive example x_j in the mini-batch. Using the cosine similarity function $sim(u,v) = u^T v/\|u\|\|v\|$, the loss term for a positive pair of examples (i,j) is defined as:

$$L_{i,j} = -\log \frac{\exp(sim(z_i, z_j/\tau))}{\sum_{k=1}^{2N} \mathbf{1}_{[k\neq i]} \exp(sim(z_i, z_j/\tau))}, \tag{1}$$

  where $\mathbf{1}_{[k\neq i]} \in \{0,1\}$ is an indicator function evaluating to 1 iff $k \neq i$ and τ denotes a temperature parameter. Figure 2 illustrates the utilization of these components.

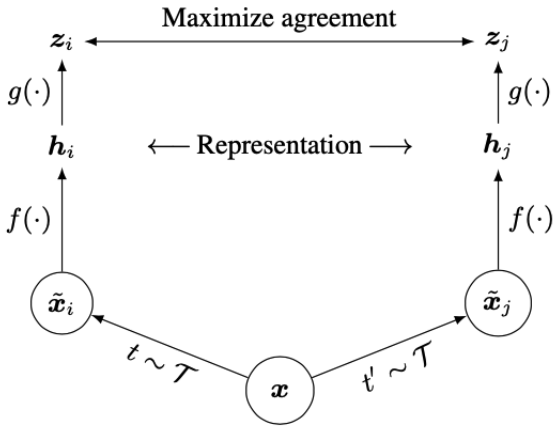


Fig. 2: The simCLR framework. First, two separate augmentation operators t and t' are generated from the same set of augmentations T. These operators are then applied to the input image x to obtain two correlated views. The encoder network $f(\cdot)$ together with the projection head $g(\cdot)$ is used to maximize agreement using a contrastive loss [CKNH20]. After the training is finished, $g(\cdot)$ is discarded. Our GCD pipeline also follows this approach of discarding the projection head after the training and only uses the encoder network $f(\cdot)$ for the actual class categorization task.

3.3.2 reweighted Sharpness-Aware Minimization

To be more precise, the authors of SAM seek flatter minima to achieve this. The idea is that the loss value und loss sharpness will be minimized simultaneously, by finding paramaters with uniformly low loss values within a neighborhood. The paper suggests an informally stated theorem:

Theorem 3.1. *For any $p > 0$, with probability over training set S generated from distribution $\mathscr{D}$.*

$$\begin{aligned} L_{\mathscr{D}}(\boldsymbol{w}) &\leq \max_{||\boldsymbol{\varepsilon}||_2 \leq \rho} L_S(\boldsymbol{w}+\boldsymbol{\varepsilon}) + h(||w||_2^2/\rho^2) \\ &\leq [\max_{||\boldsymbol{\varepsilon}||_2 \leq \rho} L_S(\boldsymbol{w}+\boldsymbol{\varepsilon}) + L_S(\boldsymbol{w})] - L_S(\boldsymbol{w}) + h(||w||_2^2/\rho^2) \end{aligned}$$

where h: $\mathbb{R}_+ \to \mathbb{R}_+$ is a strictly increasing function.

The term in the square brackets covers the sharpness of L_S at $\boldsymbol{w}$, which is summed up with the value loss with a regularizer of magnitude $\boldsymbol{w}$. Then the second term is substituted with $\lambda||w||_2^2$ introducing the hyperparameter λ. Essentially this is a L2 regularization. Using these information the idea is to solve the following Sharpness Aware Minimization problem:

$$\min_{\boldsymbol{w}} L_S^{SAM}(\boldsymbol{w}) + \lambda||\boldsymbol{w}||_2^2 \quad \text{where } L_S^{SAM}(\boldsymbol{w}) = \max_{||\boldsymbol{\varepsilon}||_p \leq \rho} L_S(\boldsymbol{w}+\boldsymbol{\varepsilon}),$$

with $\rho \geq 0$ and $p \in [1,\infty]$. For minimization Pierre Foret et al. [FKMN20] suggest to approximate the gradient by just differentiating the inner maximization using a first-order Taylor expansion around $\mathbf{0}$, such that the following formula is obtained:

$$\boldsymbol{\varepsilon}^*(\boldsymbol{w}) = \operatorname*{arg\,max}_{||\boldsymbol{\varepsilon}||_p \leq \rho} L_S(\boldsymbol{w}+\boldsymbol{\varepsilon}) \approx \operatorname*{arg\,max}_{||\boldsymbol{\varepsilon}||_p \leq \rho} \boldsymbol{\varepsilon}^T \nabla_{\boldsymbol{w}} L_S(\boldsymbol{w}).$$

Liu et al. [LHGM21a] uses and improves the approach of SAM by also introducing and integrating a way of reweighting to the inner maximization step of the original approach. The idea here is to apply stronger regularization on rare samples. To prevent any confusions with the notation, in rwSAM the weights from the paper SAM are denoted as $\boldsymbol{\phi}$ instead of $\boldsymbol{w}$. Therefore the formula for SAM becomes:

$$\min_{\boldsymbol{\phi}} L_S^{SAM}(\boldsymbol{\phi}+\boldsymbol{\varepsilon}), \quad \text{where } \boldsymbol{\varepsilon} = \operatorname*{arg\,max}_{||\boldsymbol{\varepsilon}||_p \leq \rho} \boldsymbol{\varepsilon}^T \nabla_{\boldsymbol{\phi}} L_S(\boldsymbol{\phi}).$$

Let us denote L_S^{SAM} as $\hat{L}$ for convenience in the following parts. Now a new weight vector is introduced: $w \in \mathbb{R}^n$ and we apply this vector to the loss function of SAM, such

that the formula becomes $\hat{L}_w(\phi) = \frac{1}{n}\sum_{j=1}^{n} w_j l(x_j, \phi)$. This function is only applied to the regularization, but not to the training loss:

$$\min_{\phi} \hat{L}(\phi + \varepsilon_w), \quad \text{where } \varepsilon_w = \underset{||\varepsilon||_p \leq \rho}{\arg\max}\, \varepsilon^T \nabla_\phi \hat{L}_w(\phi).$$

3.4 Metrics

For the evaluation of this paper's results, different metrics have been selected. Including the commonly known metrics recall, precision and accuracy. Now since this is a multi-class context, it is required to elaborate those metrics a little bit further. For the simple binary classification problem the metrics recall, precision and F1-Score are very simple. We will not consider the popular metric accuracy as we will go into the specific adaption of accuracy, called clustering accuracy. We will only recap the formulas following Grandini et al.[GBV20] for those basic metrics and not go into any further detail.

$$precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

$$F1\text{-}Score = 2 \cdot \left(\frac{precision \cdot recall}{precision + recall} \right)$$

For the multiclass case a multilabel classification matrix has to be created. An example for a simple mulit-class matrix might look like Figure 3. With this we have

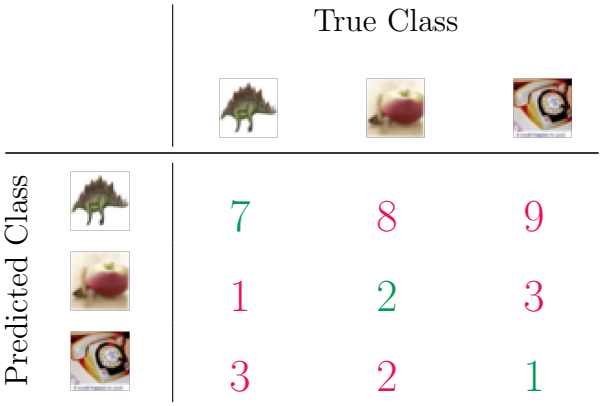


Fig. 3: Example visualization for a confusion matrix in a multi class classification setting based on images from the CIFAR100 dataset.

to consider True Positives (TP), False Positives (FP), True Negatives (TN) and False

Negatives(FN) individually for each class. Using a classification matrix the metrics like recall, precision and F1-Score have to be calculated individually for each class. And then one can use the average over all individually computed metrics to get an overall metric. We will mainly consider the metrics per class to show trends for known and novel classes. As a tool for the computation we use the library scikit-learn described by Pedregosa et al. [PVG+11] and the explicit function *classification_report*. The library automatically computes the metrics automatically per class for the multi-class case, by first creating a multiclass confusion matrix and then using this for calculating the metrics as described above.

In the GCD paper by Vaze et al. [VHVZ22] two versions of a more complex metric called the clustering accuracy have been introduced.

$$ACC = \max_{p \in P(Y_U)} \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}\{y_i = p(\hat{y}_i)\}$$

with $M = |D_U|$ and $P(Y_U)$ being all permutations of the class labels in the unlabelled set. The computation of this is using the Hungarian optimal assignment algorithm.
Now two the difference between the versions. In version one the clustering accuracy is determined individually for the old classes and the new classes. So they solve a linear assignment problem for each subset of classes and then compute the clustering accuracies individually. And the total accuracy is determined by summing these values and weighting them individually. The weight is selected based on the distribution of old and new classes. So it is just $w_1 = \frac{\#old\_class}{\#total}$ and $w_2 = 1 - w_1$. So the metrics are only computed based on the minimized cost within their subsets.
In version two for the cluster accuracy of the GCD paper they start by solving the linear assignment problem with the Hungarian optimal assignment over the whole set of classes. And then they determine the clustering accuracies by only looking into the subset of known and novel classes within this general result. This approach yields in our opinion a more generalizable result, because essentially the linear assignment finds the minimum cost within the given set. Now if you split the set for the linear assignment into subsets it might affect the choice of minimal cost. Therefore we think that it is more comprehensible to solve the linear assignment on the whole dataset and then individually compute the clustering accuracies. Hence, we will focus on optimizing the version two of the clustering accuracy.

the metric is computed indiviudally for each instance. Therefore the formula is applied to the old classes, new classes and total classes.

3.5 Extract Features

As mentioned in Section 3.3, the pretraining of the encoder network is performed on correlated pairs of augmented data points that are derived from the same original input image. However, the categorization task in GCD aims to categorize the original

images into classes and not the augmented views created from them. Therefore, after the pretraining is completed, GCD performs an intermediate step, where this time the pretrained encoder network is used to extract features directly from the original input images. These features are then stored and can now be used to perform the actual K-Means clustering.

3.6 K-Means

To cluster the data in the right class, the well known algorithm k-means is used in the paper GCD. The idea of k-means is to partition data into k sets. The k has to be given. There is a lot of research on how to estimate the k, but for ourpurpose we know the exact amount of classes to be predicted. So we will not immerse any further into this. The fundamental algorithm of k-means is defined nicely by Hamerly et al. [HE02] as this in a few steps:

1. Initialize the algorithm with guessed centers C.
2. For each data point x_i, compute its membership $m(c_j|x_i)$ in each center c_j and its weight $w(x_i)$.
3. For each center c_j recompute its location from all data points x_i according to their memberships and weights:

$$c_j = \frac{\sum_{i=1}^{n} m(c_j|x_i)w(x_i)x_i}{\sum_{i=1}^{n} m(c_j|x_i)w(x_i)}$$

4. Repeat steps 2 and 3 until convergence.

where $x_i \in X$ and X defines the number of n d-dimensional data points and $c_j \in C$ and C defines the set of k d-dimensional centroids. Now the most optimization problem can be defined as this according to Hamerly et al.[HE02]:

$$KM(X,C) = \sum_{i=1}^{n} \min_{j \in \{1 \ldots k} ||x_i - c_j||^2,$$

Now we can enhance the algorithm quite a bit if we choose the initialization method which is called $k-means++$ by Arthur et al. [AV06]. With this method of initialization the goal is not just to choose k centroids at random, but to choose the centroids based on a probability such that the centroids are chosen further apart. For the algorithm, we need to introduce the shortest distance $D(x)$ from a data point to the closest already chosen center. With this we can have a look at the algorithm for $k-means++$ described by Arthur et al. [AV06]:

1. Take one center c_1, chosen uniformly at random from X

2. Take a new center c_j, choosing $x \in X$ with probability $\frac{D(x)^2}{\sum_{x \in X} D(x)}$
3. Repeat Step 2, until all k centroids are chosen
4. Proceed with the step 2 of the standard $k-means$ algorithm as described before.

This version of $k-means$ is chosen for determining the clusters in the long-tailed GCD scenario.

4 Methods

The problem that we investigate consists of how to categorize a set of unlabelled image samples that may come from known as well as unknown (novel) classes. The known classes refer to classes for which labels exist in the data set. Combined with the utilization of unlabelled sets during the training, this leads to a semi-supervised learning task. Current image recognition settings for Novel Class Discovery make assumptions on the classes of the unlabelled images. The most important assumption is that all of the unlabelled images come from novel categories. Because this is rather unrealistic in the real-world, this NCD setting has limitations and therefore we put our focus on the less restrictive setting described in the beginning.

Additionally, existing methods assume that the sample distributions between novel and known classes in the unlabelled set are rather balanced. However, most of the real world data is rather imbalanced. For example, the iNaturalist [VHMAS+17] dataset impressively shows the heavily imbalanced distribution of images across different species in the natural world, as some species are more abundant and easier to photograph than others. Therefore, we focused on this more challenging but also more open-world setting.

4.1 Combining self-supervised and supervised contrastive learning

In our work, we studied two methods for this categorization task. The first method consists of combining self-supervised contrastive learning as explained in Section 3.5 with a supervised contrastive learning component during the pretraining of our vision transformer model. We mainly followed the supervised contrastive (*SupCon*) learning approach by Khosla et al. [KTW+20] from 2021. This is motivated by the fact that in our GCD setting we also have access to a labelled set of images and therefore it appears reasonable to leverage label information during the training as well.

Although the fully-supervised approach by Khosla et al. [KTW+20] has similarities to the *SimCLR* self-supervised approach, there are a few important differences. First, as also discussed in Section 3.5, the *SimCLR* framework contrasts *one* positive augmented sample against all other remaining augmented samples from the minibatch. However, the *SupCon* approach considers *many* samples as positive samples. More specifically, *all*

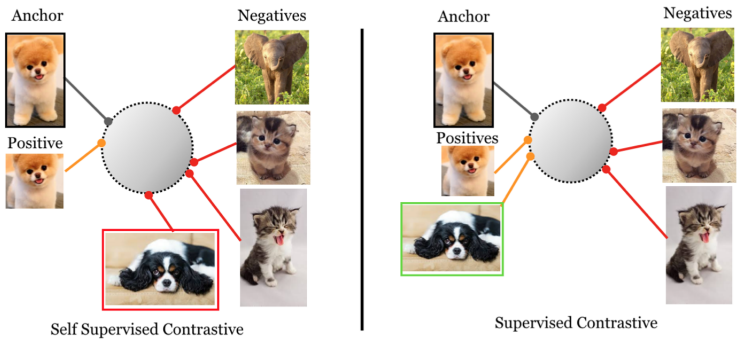


Fig. 4: Supervised (right) vs. self-supervised (left) contrasting approaches. The presence of label information allows the supervised contrasting approach to cosider more than one example as positive. This supports the model in pulling together clusters of data points belonging to the same class and pushing apart clusters of data points belonging to different classes [KTW+20].

samples that belong to the same class as the anchor (and the derived view from the anchor) are contrasted against the remainder of the minibatch. Utilizing label information in this way results in an embedding space where data points of the same class are more likely to be closely aligned than compared to the self-supervised contrasting approach. Figure 4 visualizes this difference between self-supervised and supervised contrasting. Consequently, the loss function used for this setting is also slightly different from the loss function in (1) in order to be able to handle the scenario in which more than one sample can belong to the same class in a minibatch. Although Khosla et al. [KTW+20] have proposed two different equations for the loss function, after thorough investigation of the gradients, they concluded to consider only one of them for the training. Similar to (1), let $I = \{1..2N\}$ be the set of all augmented view pairs generated from N input images in the minibatch and $i \in I = \{1..2N\}$ be the index of the anchor from which the positive example is derived. Additionally, let $A(i) \equiv I \backslash \{i\}$ and $P(i) \equiv \{p \in A(i) : \hat{y}_p = \hat{y}_i\}$ be the set of all indices of the views that belong to the same class as the anchor, i.e. the set of all positives distinct from i. $|P(i)|$ is the cardinality of $P(i)$. Then, the supervised contrastive loss where the summation of the positives is done outside the log is defined as:

$$L_{out}^{sup} = \sum_{i \in I} L_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p)/\tau}{\sum_{a \in A(i)} \exp(z_i \cdot z_p)/\tau} \tag{2}$$

The $\cdot$ symbol denotes the inner (dot) product and τ is temperature parameter. One important advantage of using this function is its property to generalize well to an arbitrary number of positives, because now for any anchor, all positives in the minibatch will contribute to the numerator due to the sum function. As also mentioned in [KTW+20], consequently, this will encourage the model to give closely aligned representations to *all*

data points from the same class resulting in a more robust clustering when performed on the feature space of the model.

Since in GCD the self-supervised contrastive loss from SimCLR (1) as well as this supervised contrastive loss are used in combination, the final loss term is defined as:

$$L = (1-\lambda)\sum_{i\in B} L_{unsup,i} + \lambda \sum_{i\in B_L} L^{sup}_{out,i} \tag{3}$$

where B corresponds to a minibatch, B_L corresponds to the labelled subset of the minibatch B and λ corresponds to a weight coefficient. The idea from [VHVZ22] to use a contrastive loss rather than standard cross-entropy loss leads to a setting where labelled and unlabelled data are treated similarly. This is suggested to reduce the risk of overfitting to the labelled classes. This is also particularly important for our setting where the samples from the long-tailed unlabelled dataset are less frequent which already naturally causes a bias towards learning from the more frequent samples. In our case the more frequent samples come from the balanced and labelled part of train dataset.

4.2 Using rwSAM optimization in GCD

Contrastive Learning is a very good tool for pretraining in Generalized Category Discovery. However we had a closer look at it and experimented on how it performs on Generalized Category Discovery in a setting where the unlabelled data is long-tailed. Running these expermints clearly showed that Contrastive Learning does not handle the long-tailed scenario very well. This happens because the data distribution is not fully recognized by the algorithm. The loss function does not consider the bias. Seeing this challenge, we decided to use a completely different approach for the pretraining by replacing the algorithm by Imbalanced SSL with rwSAM while reusing the other parts of the GCD pipeline. This decision has been made as rwSAM handles biased data much better and therefore the assumption is that it will perform better on long-tailed data. In Chapter 3.3.2 we explained in detail how rwSAM works and how it simultanuously minimizes the loss value and the loss sharpness to find flatter minima.

5 Experiments

5.1 Backbones

Different possible backbone networks were considered in previous works and in the following we give a brief overview on transformer based and convolutional neural networks.

5.1.1 VitDINO

We use VitDINO [CTM+21] as a backbone for GCD. VitDINO is a vision transformer [DBK+21] which is pretrained self-supervised. The model consists of a student and a teacher model,

sharing the same architecture but having different parameters. The teacher model is not given a priori but is built from the past from the student's parameters of the last epochs. While the teacher model is fed randomly transformed global views (i.e., at least 50% of the image size) of a sample, the student model can be fed both, randomly transformed local and global views. Each network outputs a k dimensional feature that is normalized with a softmax. The output of the teacher network is then centered with a mean computed over the batch. The student's and teacher's network's output similarity is then measured with a cross-entropy loss. The teacher parameters θ_t are updated with an exponential moving average of the student parameters θ_s: $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$. λ follows a cosine-schedule from 0.996 to 1 during the training.

5.1.2 ResNet

ResNet [HZRS15] is a deep convolutional neural network whose main components are residual blocks, which use skip connections. That means, in contrast to sequential models, the output of a layer x is not only fed into the next layer $x+1$, but can also skip some layers and be fed to a layer $x+y+1$, where y is the number of skipped layers. Skip connections add neither learnable parameters nor computational complexity to the network. However, they can help to train deeper networks [TLLG17].

5.2 Experiment Setup

Dataset and Evaluation Protocol We used the CIFAR-100 [KH+09] dataset throughout our whole experiments as the only dataset. For our main experiments, we only used our long-tail sampling algorithm for the unlabelled part of the train dataset during the pretraining and during the clustering and did not change the test set at all.

For the evaluation, we followed the same evaluation protocol as in the original GCD paper [VHVZ22]. During test-time, we measure the clustering accuracies $V1$ and $V2$ that we discussed previously in Section 3.4. Additionally, we also implemented and reported per-class metrics (recall, precision and F1-Score) which we derived from the predicted cluster indices. The evaluation protocol from [VHVZ22] already provides a mapping table from the predicted clusters to the CIFAR-100 image classes and we used this table to translate the cluster predictions after the last K-Means iteration accordingly.

Implementation details and hyperparameters Similar to [VHVZ22], all of our experiments were conducted on the ViT-B-16 backbone that was pretrained using DINO [CTM+21] self-supervision on unlabelled ImageNet data [DDS+09] and we used the output [CLS] token as our extracted features. Additionally, in all experiments we trained our models for 200 epochs. We experminted with both using the best model and using the most recent model. The original code from the paper for GCD provides the option to save the best model based on the minimal loss. After multiple tests we concluded that the most

recent model performs much better though for the clustering accuracy on the unlabelled train data, which is our research focus in this paper.

We also did not fine tune the entire backbone, but finetuned only the last block of the vision transformer with an initial learning rate of 0.1. Additionally, we decay this learning rate with a cosine anneal schedule and experimented with different batch sizes ranging from 64 up to 512. We found 128 to work best for our setting and thus, used this size for our results. Our weight coefficient λ was set constantly to 0.35 (see (3) for the experiments where we used contrastive learning). Moreover, as it is also commonly done in other self-supervised training settings, we use the small 3-Layer MLP that was also used during the DINO pretraining [CTM+21] to project the backbones output through before we apply the contrastive loss. For the imbalanced-SSL training using rwSAM, we use the same nonlinear heads as in [LHGM21b], one 2-Layer projection head applied on the extracted features and another 2-Layer classification head. However, we adjusted the ouput and feature dimensions of these heads to fit to our vision transformer model. The last prediction head is also only used to conform with the initially proposed method for rwSAM by Liu et al. [LHGM21b] and all heads are discarded at test-time.

5.3 Results

To actually get the results, we did multiple deviation runs and calculated the average for every class and every metric to produce more reliable results. With this we compute the basic metrics precision, recall and F1-Score, which we have introduced earlier in this paper:

The trend definitely shows that both sides perform quite similar for the balanced known classes. But for the long-tailed unknown classes using rwSAM for the pretraining appears to have a postitive impact. Over all 3 metrics the rwSAM adapted GCD pipeline appears to perform better for the unlabelled novel classes. As suggested in the paper for GCD by Vaze et al. [VHVZ22] we also compare the clustering accuracy, which has been introduced in chapter 3.4. Now for this metric the results can be found in Table 2. It can be observed that the accuracy for the ”New” novel classes in V2 performs better for our adaptation of the code, which was our main goal. So when we calculate the clustering accuracy by solving the linear assignment problem on all data, we perform better. Interestingly for the accuracy for the ”New” novel classes in V1 we perform worse. In this scenario the clustering accuracy is only calculated by solving the linear assignment on the subset of novel classes. So for the overall classification problem we perform better on the novel classes and for the classification problem within the subset of novel classes we perform worse than the original GCD pipeline. Also improving the accuracy on the ”New” novel classes yields a trade-off as the accuracy for the ”Old” known classes becomes worse. The idea is that Contrastive Learning tries to maximize agreement using the constrastive loss function. This works very well for Novel Class Discovery in general. However for our long-tailed scenario this algorithm struggles a little bit as it might overfit based on the more frequently occuring samples from the

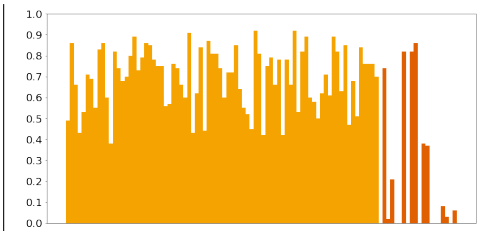
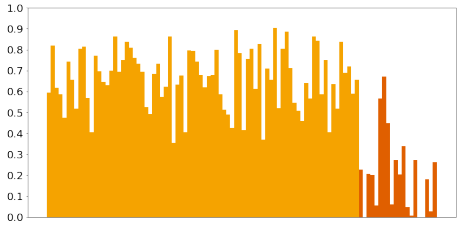
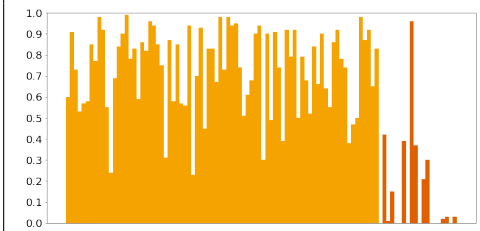
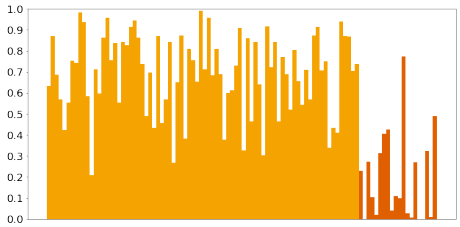
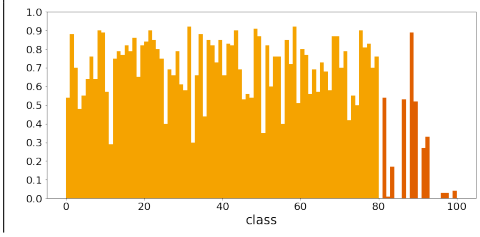
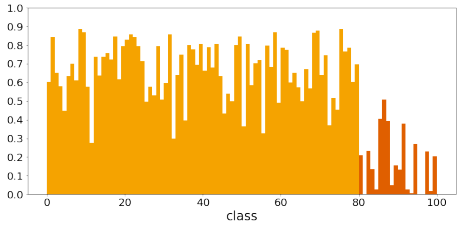
Metric	Original GCD	rwSAM adapted GCD
Recall		
Precision		
F1-Score		

Table 1: Comparison of different classification metrics between the original GCD pretraining using contrastive learning and the rwSAM adapated pretraining. The bars in orange refer to the known classes

head classes. Now the algorithm SAM might struggle with overfitting here, but with rwSAM Liu et al. [LHGM21b] propose an algorithm that is supposed to reduce overfitting using weights. Therefore we achieve better results on the V2 clustering accuracy with this.

6 Conclusion & Future Work

In this paper we explained and showed the algorithm for generalized category discovery that has been implemented in a setting with balanced data. We have shown a way to improve this algorithm to a setting with long-tailed data by replacing the contrastive learning in the pretraining by the rwSAM algorithm. This reduces overfitting on very frequent classes and is therefore more likely to perform better on rare classes. A few open questions remain to be addressed in different categories.
Currently we are only considering a long-tailed scenario where the data is distributed based on a step function. For further research it would be interesting to research how the algorithms behave for long-tailed data distributed based on an exponential function.

	Pretraining		Train ACC V2			Train ACC V1		
Methods	Contrastive	rwSAM	Old	New	All	Old	New	All
DINO	No	No	69.52	17.40	63.37	69.52	51.68	67.54
GCD	Yes	No	84.07	21.96	77.26	84.07	57.20	81.08
Ours	No	Yes	66.95	23.09	62.08	66.96	50.19	65.09

Table 2: Accuracies for GCD pipeline with Contrastive Pretraining in comparison to accuracies for GCD pipeline with rwSAM pretraining. The unlabelled data is distributed long-tailed. We focus on improving on the "New" novel classes V2 clustering accuracy as this evaluates the clustering accuracy based on solving the linear assignment problem on all classes rather than just a subset. Therefore it provides a better representation of computing the overall accuracy.

So one open question is: How does the steepness of the data distribution affect the efficiency of our pipeline?
The algorithm has been optimized on the Cifar100 dataset. We have also considered other datasets, but within this research there was no time to dive any deeper into experminenting with different long-tailed datasets. So it remains open to check: How well does the pipeline perform on different long-tailed datasets? Interesting datasets to immerse further into are iNaturalist and a long-tailed version of ImageNet.
During our experiments we encountered that overall the clustering accuracy improves for each epoch however there is quite a large deviation for the clustering accuracy between each epoch if you compute k-means in every epoch. Doing the step for k-means in every epoch slows down the training quite a bit, but it might ensure that a better model can be saved as the best model. It might be worth to look into this for future work.

References

[AV06] Arthur, David ; Vassilvitskii, Sergei: k-means++: The Advantages of Careful Seeding / Stanford InfoLab. Version: June 2006. http://ilpubs.stanford.edu:8090/778/. Stanford, June 2006 (2006-13). – Technical Report

[CBL21] Cao, Kaidi ; Brbic, Maria ; Leskovec, Jure: *Open-World Semi-Supervised Learning*. http://dx.doi.org/10.48550/ARXIV.2102.03526. Version: 2021

[CKNH20] Chen, Ting ; Kornblith, Simon ; Norouzi, Mohammad ; Hinton, Geoffrey: *A Simple Framework for Contrastive Learning of Visual Representations*. http://dx.doi.org/10.48550/ARXIV.2002.05709. Version: 2020

[CTM+21] Caron, Mathilde ; Touvron, Hugo ; Misra, Ishan ; Jégou, Hervé ; Mairal, Julien ; Bojanowski, Piotr ; Joulin, Armand: *Emerging Properties in Self-Supervised Vision Transformers*. http://dx.doi.org/10.48550/ARXIV.2104.14294. Version: 2021

[DBK+21] Dosovitskiy, Alexey ; Beyer, Lucas ; Kolesnikov, Alexander ; Weissenborn, Dirk ; Zhai, Xiaohua ; Unterthiner, Thomas ; Dehghani, Mostafa ; Minderer, Matthias ; Heigold, Georg ; Gelly, Sylvain ; Uszkoreit, Jakob ; Houlsby, Neil: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021

[DDS+09] Deng, Jia ; Dong, Wei ; Socher, Richard ; Li, Li-Jia ; Li, K. ; Fei-Fei, Li: ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), S. 248–255

[FKMN20] Foret, Pierre ; Kleiner, Ariel ; Mobahi, Hossein ; Neyshabur, Behnam: Sharpness-Aware Minimization for Efficiently Improving Generalization. In: *CoRR* abs/2010.01412 (2020). https://arxiv.org/abs/2010.01412

[FSL+21] Fini, Enrico ; Sangineto, Enver ; Lathuilière, Stéphane ; Zhong, Zhun ; Nabi, Moin ; Ricci, Elisa: A Unified Objective for Novel Class Discovery. In: *CoRR* abs/2108.08536 (2021). https://arxiv.org/abs/2108.08536

[GBV20] Grandini, Margherita ; Bagli, Enrico ; Visani, Giorgio: Metrics for Multi-Class Classification: an Overview. In: *ArXiv* abs/2008.05756 (2020)

[HE02] Hamerly, Greg ; Elkan, Charles: Alternatives to the K-Means Algorithm That Find Better Clusterings. In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*. New York, NY, USA : Association for Computing Machinery, 2002 (CIKM '02). – ISBN 1581134924, 600–607

[HFLM+18] Hjelm, R D. ; Fedorov, Alex ; Lavoie-Marchildon, Samuel ; Grewal, Karan ; Bachman, Phil ; Trischler, Adam ; Bengio, Yoshua: *Learning deep representations by mutual information estimation and maximization*. http://dx.doi.org/10.48550/ARXIV.1808.06670. Version: 2018

[HRE+20] Han, Kai ; Rebuffi, Sylvestre-Alvise ; Ehrhardt, Sébastien ; Vedaldi, Andrea ; Zisserman, Andrew: Automatically Discovering and Learning New Visual Categories with Ranking Statistics. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020

[HRE+21] Han, Kai ; Rebuffi, Sylvestre-Alvise ; Ehrhardt, Sébastien ; Vedaldi, Andrea ; Zisserman, Andrew: AutoNovel: Automatically Discovering and Learning Novel Visual Categories. In: *CoRR* abs/2106.15252 (2021). https://arxiv.org/abs/2106.15252

[HVZ19] Han, Kai ; Vedaldi, Andrea ; Zisserman, Andrew: Learning to Discover Novel Visual Categories via Deep Transfer Clustering. In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, IEEE, 2019, 8400–8408

[HZRS15] He, Kaiming ; Zhang, Xiangyu ; Ren, Shaoqing ; Sun, Jian: Deep Residual Learning for Image Recognition. In: *CoRR* abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385

[KH+09] Krizhevsky, Alex ; Hinton, Geoffrey u. a.: Learning multiple layers of features from tiny images. (2009)

[KTW+20] Khosla, Prannay ; Teterwak, Piotr ; Wang, Chen ; Sarna, Aaron ; Tian, Yonglong ; Isola, Phillip ; Maschinot, Aaron ; Liu, Ce ; Krishnan, Dilip: *Supervised Contrastive Learning*. http://dx.doi.org/10.48550/ARXIV.2004.11362. Version: 2020

[LHGM21a] Liu, Hong ; HaoChen, Jeff Z. ; Gaidon, Adrien ; Ma, Tengyu: Self-supervised Learning is More Robust to Dataset Imbalance. In: *CoRR* abs/2110.05025 (2021). https://arxiv.org/abs/2110.05025

[LHGM21b] Liu, Hong ; HaoChen, Jeff Z. ; Gaidon, Adrien ; Ma, Tengyu: *Self-supervised Learning is More Robust to Dataset Imbalance*. https://arxiv.org/abs/2110.05025. Version: 2021

[PVG+11] Pedregosa, F. ; Varoquaux, G. ; Gramfort, A. ; Michel, V. ; Thirion, B. ; Grisel, O. ; Blondel, M. ; Prettenhofer, P. ; Weiss, R. ; Dubourg, V. ; Vanderplas, J. ; Passos, A. ; Cournapeau, D. ; Brucher, M. ; Perrot, M. ; Duchesnay, E.: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830

[TLLG17] Tong, Tong ; Li, Gen ; Liu, Xiejie ; Gao, Qinquan: Image Super-Resolution Using Dense Skip Connections. In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, 4809–4817

[VHMAS+17] Van Horn, Grant ; Mac Aodha, Oisin ; Song, Yang ; Cui, Yin ; Sun, Chen ; Shepard, Alex ; Adam, Hartwig ; Perona, Pietro ; Belongie, Serge: *The iNaturalist Species Classification and Detection Dataset*. http://dx.doi.org/10.48550/ARXIV.1707.06642. Version: 2017

[VHVZ22] Vaze, Sagar ; Han, Kai ; Vedaldi, Andrea ; Zisserman, Andrew: *Generalized Category Discovery*. http://dx.doi.org/10.48550/ARXIV.2201.02609. Version: 2022

[WXYL18] Wu, Zhirong ; Xiong, Yuanjun ; Yu, Stella ; Lin, Dahua: *Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination*. http://dx.doi.org/10.48550/ARXIV.1805.01978. Version: 2018

[ZH21] Zhao, Bingchen ; Han, Kai: Novel Visual Category Discovery with Dual Ranking Statistics and Mutual Knowledge Distillation. In: Ranzato, Marc'Aurelio (Hrsg.) ; Beygelzimer, Alina (Hrsg.) ; Dauphin, Yann N. (Hrsg.) ; Liang, Percy (Hrsg.) ; Vaughan, Jennifer W. (Hrsg.): *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, 22982–22994