One-bit Active Query with Contrastive Pairs

Yuhang Zhang[1], Xiaopeng Zhang[3], Lingxi Xie[3], Jie Li[1], Robert C. Qiu[1]*, Hengtong Hu[2], Qi Tian[3]
[1]Shanghai Jiao Tong University [2]Hefei University of Technology [3]Huawei Inc.
{hang_universe, lijiecs, rcqiu}@sjtu.edu.cn, tian.qi1@huawei.com
{huhengtong.hfut, zxphistory, 198808xc}@gmail.com

Abstract

How to achieve better results with fewer labeling costs remains a challenging task. In this paper, we present a new active learning framework, which for the first time incorporates contrastive learning into recently proposed one-bit supervision. Here one-bit supervision denotes a simple Yes *or* No *query about the correctness of the model's prediction, and is more efficient than previous active learning methods requiring assigning accurate labels to the queried samples. We claim that such one-bit information is intrinsically in accordance with the goal of contrastive loss that pulls positive pairs together and pushes negative samples away. Towards this goal, we design an uncertainty metric to actively select samples for query. These samples are then fed into different branches according to the queried results. The* Yes *query is treated as positive pairs of the queried category for contrastive pulling, while the* No *query is treated as hard negative pairs for contrastive repelling. Additionally, we design a negative loss that penalizes the negative samples away from the incorrect predicted class, which can be treated as optimizing hard negatives for the corresponding category. Our method, termed as **ObCP**, produces a more powerful active learning framework, and experiments on several benchmarks demonstrate its superiority.*

1. Introduction

Active learning is particularly useful in many modern machine learning systems, where data may be sufficient, but labels are scarce or expensive to obtain [12, 14, 24, 25]. The key hypothesis of active learning is that one can build a good predictive model with less labeled samples if a model already knows which samples should be labeled to help improve predictive performance. Generally, an active learning algorithm iteratively selects samples to label, based on the prediction results of current model. Once the samples are labeled, they are added to the training set, and such process continues till the labeled budget is used up. However, conventional active learning strategies are usually constrained by annotating samples with accurate labels, *i.e.,* which exact class one sample belongs to. While it is difficult for the labeler to memorize and distinguish all categories especially when the number of categories scales up, like ImageNet [7] which consists of 1K categories.

*Corresponding author

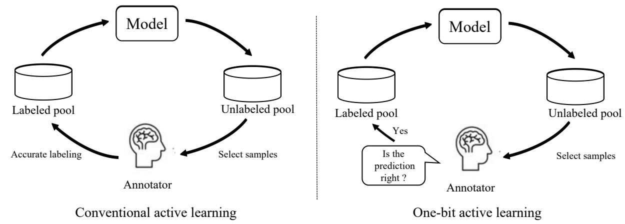

Figure 1. Illustration of differences between conventional active learning and one-bit supervision. The latter leverages a simple yes or no query about the correctness of the model's prediction instead of assigning an accurate label. In this paper, we adopt one-bit supervision in our active learning framework and combine it with contrastive learning for jointly optimization.

Recently, a novel active annotation method called one-bit supervision [18] is proposed, which labels samples with one bit information by answering a simple yes-or-no question, *i.e.,* whether an image belongs to a specified class c, this method can facilitate the learned procedure more efficiently under the same amount of supervision. For example, if we can get the model's prediction on unlabeled data, we only need to inquiry the labeler *"Does the model predict this sample right?"* It is a more efficient query strategy compared with traditional labeling procedure in terms of bit information, since for a C-classes classification problem, accurately labeling a sample needs $\log_2 C$ bits of information, while the one-bit query only needs one bit. This means that with the same amount of annotations, we are able to query more samples to get more information accordingly. The differences between conventional active learning and the one-bit method are shown in Figure 1. However, previous one-bit method [18] only makes use of the correct predictions like conventional supervised learning, while does not take full advantage of the information such as negative query, and hence limits its performance.

The one-bit supervision returns Yes or No results, based on this, we are able to obtain information whether the two samples (one is the queried sample, and the other is a random sample from the queried category) are from the same category or not, which is reminiscent of recent contrastive learning strategy [15]. In contrastive learning, the goal is to learn an encoder that is able to map positive pairs to similar representations while pushing away those negative samples in the embedding space. We claim that **contrastive loss is intrinsically in accordance with the one-bit query**, and we are able to simply treat the Yes query as positive pairs, while those No query as negative pairs. In this way, we are able to take full advantages of the queried results. Towards this goal, we develop a novel active learning approach that combines semi-supervised learning and contrastive learning with efficient one-bit supervision.

Specifically, firstly, we jointly train the model via combining supervised cross-entropy loss with contrastive loss, and obtain a pre-trained model for the next stage one-bit query. Then we develop an uncertainty measurement metric, which is based on the variance of the model's prediction during the training process, to help decide which samples to query. According to the queried results, for the correct prediction with Yes query, we extend the contrastive loss function to allow for multiple positive samples during each forward propagation so that images with the same ground truth label will be pulled together for compact representation. While for the incorrect prediction with No query results, we integrate this incorrect negative label information into contrastive learning to help these samples keep away from their queried class. Moreover, we design a negative loss that penalizes the negative samples away from the incorrect prediction class, which can be treated as optimizing hard negatives for the corresponding category. Integrating one-bit supervision into contrastive learning produces a much more powerful framework, and experimental results on several well-known datasets demonstrate its superiority.

Overall, we summarize our contributions as follows:

- We present a novel active learning framework, which combines contrastive learning with one-bit supervision for the first time, and take full advantages of the supervised information. We hope that such framework would shed light on active learning community and offer one possible direction for future research.

- We achieve significant leading performance on several widely used image classification benchmarks. Especially on ImageNet, with only 10% labels in terms of bit information, ObCP exceeds previous state-of-the-art that even uses 30% labels.

2. Related Work

Our approach is closely related to recent advances covering contrastive learning and active learning. We briefly review related works and clarify the differences between them and our method.

2.1. Active Learning

Active learning is an efficient way to use label information, which aims at minimizing the labeling cost by selecting high value data that can best improve the model performance, and plays an important role in modern machine learning systems. It can be classified into three categories. The first category is the uncertainty-based method which usually uses the probability distribution of prediction [6]. The diversity-based approach is the second category selecting diverse samples that expanse the input space maximally and represent the whole distribution of the unlabeled pool [14, 24]. The last category is based on model performance change, which selects the data points that would cause the greatest change to the current model parameters and encourage optimal model improvement [9, 26]. Besides the above three categories, [18] proposed one-bit supervision, which can be considered as a novel type of active learning that only inquires the most informative part in the class level. Different from the mentioned works, our approach can utilize the label information more efficiently by combining one-bit supervision with semi-supervised contrastive learning.

2.2. Contrastive Learning

In recent years, contrastive learning [4, 5, 8, 13, 15, 17, 23] has attracted a lot of attention. It regards each image and its augmentations as the same class and others as negative ones. The researchers used different ways to maintain a training queue. A memory bank is used in [30] to store the pre-computed representations from which positive examples are repossessed given a query. Based on this, a momentum update mechanism is used in [15] to maintain a long queue of negative examples for contrastive learning, while [4] uses a large batch to produce enough negative samples. [3] adopted a set of trainable "code" vector to compute different views of image, while [13] predicted previous version of itself. [22] solves transfer issue with few-label based on contrastive pretraining. These works prove that contrastive learning reaches better performance on learning data characteristics. Previous contrastive based semi-supervised learning works are almost two-stage ones, *i.e.,* using contrastive learning to pretrain a backbone model and then using few shot labeled data to fine-tune it. However, the proposed method integrates contrastive learning into semi-supervised learning in an end-to-end way, and introduces one-bit supervision to help contrastive learning generate its positive and negative samples, which will help the model maximize the use of annotations.

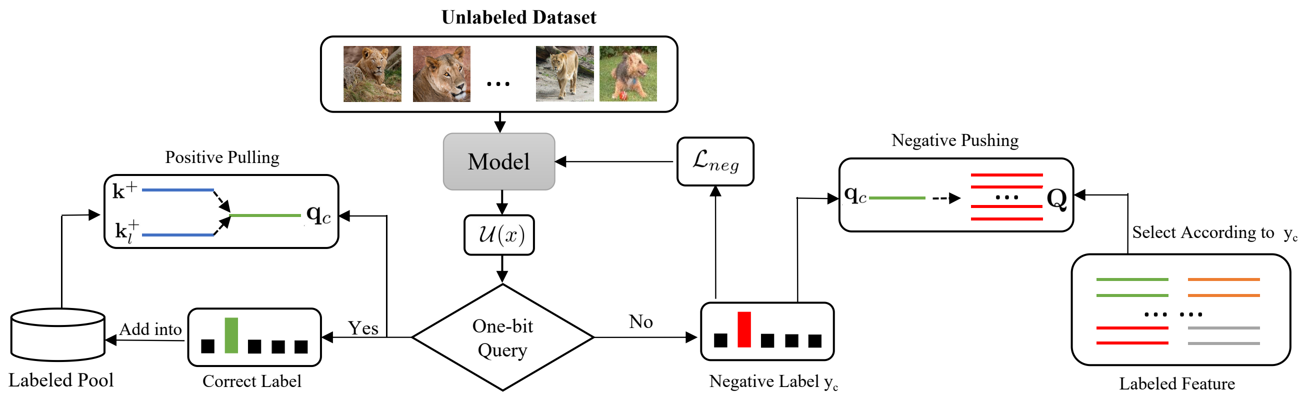

Figure 2. An illustration of our pipeline in query stage. The one-bit query will be operated in the unlabeled samples selected by our uncertainty measurement module. After that, the correct prediction will be added into the labeled pool and used not only for the supervised loss but also for contrastive learning. As for the incorrect prediction, they will be leveraged for generating negative samples for contrasting and be optimized by the designed negative loss.

3. Method

3.1. Framework Overview

We first give an overview of the proposed active learning framework. For better understanding, we split the whole procedure into two stages. The model is firstly trained with ground truth labels, using a conventional cross entropy supervised loss $\mathcal{L}_{sup}$ for labeled data as well as the original contrastive loss $\mathcal{L}_{ctr}$ for unlabeled data, so as to generate pretrained model for the next data mining stage. Then as illustrated in Figure 2, once the model is initialized, we deliberately select samples from the unlabeled pool for query, using the proposed uncertainty estimation criterion. The active learning process only needs to return whether the queried question is correct or not, which we denote as one-bit query. According to the queried *Yes* or *No* results, the samples are fed to different branches for training, *i.e.*, for the *Yes* query with accurate labels, we add samples to the supervised loss term $\mathcal{L}_{sup}$ and contrastive loss term $\mathcal{L}_{ctr}$, and for the *No* query that we only know samples do not belong to a certain category, we treat these samples as hard negative ones for the queried category. These samples are used for training in two-fold, on one hand, we incorporate them into the contrastive loss term $\mathcal{L}_{ctr}$ and highlight as hard negative samples of the queried category, and on the other hand, we design a negative loss function $\mathcal{L}_{neg}$ to make these samples away from the *No* queried label. The overall loss can be formulated as:

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda_{ctr}\mathcal{L}_{ctr} + \lambda_{neg}\mathcal{L}_{neg}. \tag{1}$$

where the supervised loss is defined as $\mathcal{L}_{sup} = H(p, y)$, p and y denote the prediction and ground truth labels, respectively, and $H(\cdot)$ denotes the conventional cross-entropy loss. $\mathcal{L}_{ctr}$ is the modified contrastive loss appears with different formats according to the labeled status of the samples. $\mathcal{L}_{neg}$ is the negative loss term that keeps the hard negative samples away from the queried categories. λ_{ctr} and λ_{neg} are two balancing factors that control the weights of the two losses respectively.

Note that similar with conventional active learning procedure, the query and model training are interchanged, and we are able to conveniently control the query times during the model training until the query budget is used up. Please refer to Figure 6 for detailed analysis. Besides, those samples that have incorrect prediction may be queried in the next query stage. The whole pipeline is presented in Algorithm 1. In the following parts, we first elaborate how to deal with different query results by designing different loss branches, as well as how to update the corresponding loss. Then we describe our uncertainty estimation criterion for sample selection.

3.2. Contrastive Loss with One-bit Query

In this section, we will elaborate how to construct the contrastive loss term $\mathcal{L}_{ctr}$ for different levels of annotated samples. For a C category classification task, denote a binary label $y_c \in \{0,1\}^C$ that only the cth dimension is 1 while others are 0. Given the queried category c, an unlabeled sample x can be categorized into three conditions during the active query procedure, *i.e.*, the *Yes* query means that y_c is the accurate label; the *No* query means that y_c is the negative label, which also represents that x definitely does not belong to category c; and an unlabeled status that does not undergo the active query procedure.

$\mathcal{L}_{ctr}$ for unlabeled data. For unlabeled data that we do not have any available human feed back label prior, we simply use original contrastive loss $\mathcal{L}_{c\_u}$ [15] to optimize it.

Algorithm 1 Contrastive learning based one-bit supervision framework

Input: Labeled pool as $\mathcal{D}_l$, unlabeled pool as $\mathcal{D}_u$, epochs for stage 1 as T_1, epochs for stage 2 as T_2.

```
for t = 0, ..., T1 − 1 do
    Stage 1
    Train the model with L_sup + λ_ctr L_c_u
end for
for t = T1, ..., T2 − 1 do
    Stage 2
    Select K samples from D_u to form a query pool D_q using U(x)
    for sample x ∉ D_q do
        use x to calculate L_c_u
    end for
    for sample x and its prediction ŷ ∈ D_q do
        if ŷ is correct then
            D_l ∪ x, D_u \ x
            Use ŷ to generate positives for L_c_l
        else
            Use ŷ to generate negatives for L_c_n
            Use ŷ to calculate L_neg
        end if
    end for
    Build contrastive loss by L_ctr = L_c_u + L_c_l + L_c_n
    Train the model with L_sup + λ_neg L_neg + λ_ctr L_ctr
end for
```

Given an unlabeled sample x, we denote its feature of two augmented views as $\mathbf{q}$ and $\mathbf{k}^+$, and let $\mathbf{M}$ be a memory bank of size K. The objective can be defined as:

$$\mathcal{L}_{c\_u} = -\log \frac{\exp(\mathbf{q} \cdot \mathbf{k}^+ / \tau)}{\exp(\mathbf{q} \cdot \mathbf{k}^+ / \tau) + \sum_{\mathbf{k} \in \mathbf{M}} \exp(\mathbf{q} \cdot \mathbf{k} / \tau)}, \tag{2}$$

where τ is a temperature parameter.

$\mathcal{L}_{ctr}$ **for *Yes* query.** Once we get the *Yes* answer in the query stage, which exactly corresponds to accurate labels as in conventional labeling strategy. Denote the labeled dataset as Ω, for the view of labeled data $\mathbf{q}_\Omega$ (including correct prediction sample), we not only use its different views $\mathbf{k}^+$ as positives, but also generate its positives by randomly selecting samples $\mathbf{k}_\Omega^+$ from Ω which have the same labels with them. The function is defined as:

$$\mathcal{L}_{c\_l} = -\frac{1}{2} \Bigg[\log \frac{\exp(\mathbf{q}_\Omega \cdot \mathbf{k}^+ / \tau)}{\exp(\mathbf{q}_\Omega \cdot \mathbf{k}^+ / \tau) + \sum_{\mathbf{k} \in \mathbf{M}} \exp(\mathbf{q}_\Omega \cdot \mathbf{k} / \tau)} + \log \frac{\exp(\mathbf{q}_\Omega \cdot \mathbf{k}_\Omega^+ / \tau)}{\exp(\mathbf{q}_\Omega \cdot \mathbf{k}_\Omega^+ / \tau) + \sum_{\mathbf{k} \in \mathbf{M}} \exp(\mathbf{q}_\Omega \cdot \mathbf{k} / \tau)} \Bigg] \tag{3}$$

where each labeled sample $\mathbf{q}_\Omega$ is pulled with two views $\mathbf{k}^+$ and $\mathbf{k}_\Omega$, and pushed away with other samples in the memory bank $\mathbf{M}$. There is no doubt that incorporate class specific priors into contrastive training will be better for classification task. The specific benefits will be shown in Sec. 4.4.

$\mathcal{L}_{ctr}$ **for *No* query.** If the prediction y_c for sample x is incorrect in query stage, which means that sample x definitely does not belong to category c. Denote y_c as its negative label for sample x, we can infer from this incorrect prediction that the model cannot distinguish x from class c easily since they suffer certain similarities in feature space, so pushing away the feature of class c will benefit the model training. Towards this goal, We redesigned a reasonable generation method of negative samples in contrastive learning for those samples with incorrect predictions.

In previous contrastive learning method, negative pairs are simply formed by sampling views from other images, which is denoted as $\mathbf{M} = \{\mathbf{k}_1, ..., \mathbf{k}_K\}$. While in our method, we first create a labeled feature set $\mathbf{L}$ to store the features of Ω, then sample N features $\mathbf{n}_i, i = 1, ..., N, N < K$ from $\mathbf{L}$ whose label is y_c. Then we use these features to replace the random N samples in $\mathbf{M}$. After that, the negative queue $\mathbf{Q}$ for the samples have incorrect prediction becomes $\mathbf{Q} = \{\mathbf{n}_1, ..., \mathbf{n}_N, \mathbf{k}_{N+1}, ..., \mathbf{k}_K\}$, while the loss function is defined as:

$$\mathcal{L}_{c\_n} = -\log \frac{\exp(\mathbf{q} \cdot \mathbf{k}^+ / \tau)}{\exp(\mathbf{q} \cdot \mathbf{k}^+ / \tau) + \sum_{\mathbf{k} \in \mathbf{Q}} \exp(\mathbf{q} \cdot \mathbf{k} / \tau)} \tag{4}$$

It is worth mentioning that since the labeled data is limited, unlike previous contrastive learning methods that make use of large size of memory bank (*e.g*, 65536), we are able to substantially reduce K (*e.g*, down to 256) without much influence to the model performance.

In summary, the function $\mathcal{L}_{ctr}$ used in Eq. (1) can be formulated as following, where different forms of contrastive function correspond to different types of annotations.

$$\mathcal{L}_{ctr} = \mathcal{L}_{c\_u} + \mathcal{L}_{c\_l} + \mathcal{L}_{c\_n} \tag{5}$$

3.3. Negative Loss

In the other way, we can use the negative label y_c to teach the model that *"the sample does not belong to this class"*. Different from other random noisy labels [20], these negative labels illustrate that the model is easily to mispredict the samples to these categories in the training process. Since that, it is very helpful for the model to keep the samples away from these negative labels. Inspired by [20], we design the negative loss function to overcome this issue:

$$\mathcal{L}_{neg} = -\mathbf{1}(p_c \geq \frac{1}{C}) \sum_i^C y_c \log(1 - p_i), \tag{6}$$

where p denotes the probability which represents the prediction output by the model, and p_i represents the ith element of p. Eq. (6) enables the probability value of the incorrect label to be optimized as zero, resulting in an increase in the probability values of other classes, which meets our purpose. Besides, we adopt an indicator function $\mathbf{1}$ that only train the model with those predictions whose confidence is above threshold $1/C$. This operation can prevent the model from over minimizing those incorrect samples.

3.4. Uncertainty Estimation

We now return to the sample selection process to elaborate how to select valuable samples from the unlabeled pool for active learning. In a nutshell, the selected samples fed for active query should be endowed with high uncertainty prediction with respect to current model, thus we are able to reduce uncertainty as much as possible after the active query procedure. To achieve this goal, we define the model uncertainty metric via diagnosing the output of a sample during consecutive training epochs.

In this part, we explain how to measure the uncertainty of a sample during the training procedure. Intuitively, the uncertain samples are defined as those predicted sometimes incorrectly during training and correctly at other times, as illustrated in Figure 3. Formally, during the training procedure, we maintain a running average of the model's prediction on all unlabeled data over the last m epochs, we refer to it as $\tilde{p}(y)$, which represents an average probability distribution of the sample's prediction during the training procedure. Let q represent the model's prediction on all unlabeled data at the current epoch, we use the probability vector of each sample during the last t epochs ($q_i, i = 1, 2..., t$) to calculate the uncertainty $\mathcal{U}(x)$, which is defined as:

$$\mathcal{U}(x) = -\sum_{i}^{t} \tilde{p}(y) \log(q_i) \tag{7}$$

After calculating the uncertainty value, we sort the unlabeled samples in descending order according to their uncertainty and choose top ranked samples to query.

4. Experiment

In this section, we conduct extensive experiments on several widely-used benchmarks to validate the effectiveness of our proposed active learning framework, as well as detailed ablation studies to uncover how each module affects the performance. All experiments are trained on 8 NVIDIA Tesla-V100 GPUs, and the results are reported over 3 times with random initial network weights and labeled pool.

Dataset. We evaluate our method on several standard image classification benchmarks, including CIFAR-10/100 [21] and ImageNet [7]. Specifically, both CIFAR-10 and CIFAR-100 consist of 60K images, of which 50K are used for training and the rest 10K for testing, while CIFAR-10 contains 10 classes and CIFAR-100 has 100 classes. ImageNet is a large-scale dataset that contains about 1.2M images ranging 1K classes.

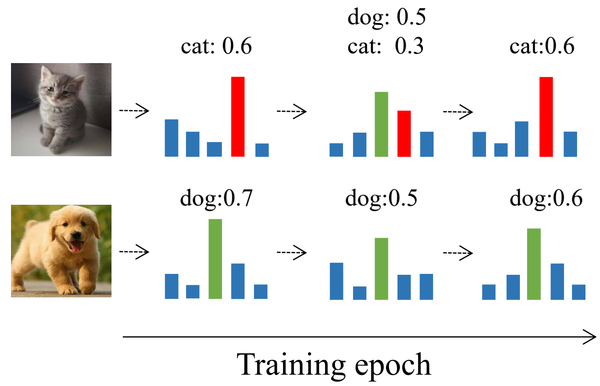

Figure 3. Some toy cases of sample's prediction performance during the training process. The top sub-figure represents an unstable prediction case, whose result often changes with training iteration while the bottom is stable. Labeling uncertain samples will provide greater help for model training.

Table 1. The correspondence of information bits between one-bit and conventional annotations on different dataset.

	CIFAR-10	CIFAR-100	ImageNet
$\log_2 C$	3.3219	6.6439	9.9658
$\lvert\mathcal{P}^L\rvert$	10K	10K	128K
bits of information	33.2K	66.4K	1276K
$\lvert\mathcal{I}^S\rvert$	5K	5K	13K
$\lvert\mathcal{T}^O\rvert$	16.6K	33.2K	1146K
bits of information	33.2K	66.4K	1276K

Active learning settings. We conduct our experiments on different number of labeled pool $\mathcal{P}^L$. For fair comparison, we denote the initial labeled data used for training Stage 1 as $\mathcal{I}^S$, and one-bit query budgets as $\mathcal{T}^O$. Based on the description in Sec. 1, these three items satisfy: $\lvert\mathcal{P}^L\rvert \approx \lvert\mathcal{I}^S\rvert + \lvert\mathcal{T}^O\rvert / \log_2 C$. The details about labeled number and its corresponding bits of information and query times are shown in Table 1.

4.1. Comparison with Active Learning Methods

4.1.1 CIFAR-10/100

Baseline method. We compare ObCP against several well-known methods, which including VAAL [28], LL4AL

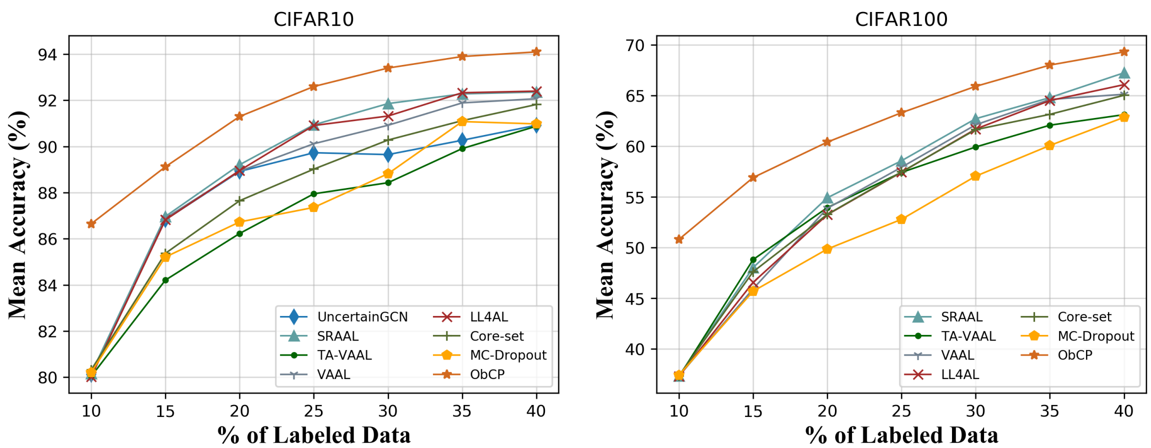

Figure 4. The comparison results of image classification on CIFAR10/100.

[31], Core-set [24], Ensembles using Variation Ratios (Ensembles w.VarR) [1], UncertainGCN [2], SRAAL [32], TA-VAAL [19], DBAL [11] and MC-Dropout [10].

Implementation details. Following the conventional practices in deep active learning [19], for fair comparison, we adopt ResNet-18 [16] as the base model. The total training epochs is 200 while the first 100 epoch is used for Stage 1 shown in Algorithm 1. The momentum coefficient is set to 0.9 with a weight decay of 5e-4. We use SGD optimizer to update the parameters of the model, and the learning rate is set to 0.03 with cosine decay schedule with batch size of 64. The data augmentation methods we used for contrastive learning are consistent with that used in [15]. We conduct experiments on different labeled budgets and $\mathcal{I}^S$ is set to 10%, then in query stage we use up the total query budget $|\mathcal{T}^O|$ in four times. The m is set to 30 while t set to 5. Besides, we simply set $N = K/2$ during experiments, and the λ_{ctr} and λ_{neg} are set to 0.1.

Results. The comparison results of image classification performance are shown in Figure 4. It can be observed that our method outperforms all other active learning methods on both two benchmark datasets at each active cycle. Additionally, we have the following observations. For CIFAR-10, we obtain an accuracy of 92.6% with 25% labels, even better than other methods (*e.g.,* SRAAL [32], 92.3%) which need 40% labels. The accuracy is 94.1% when using 40% labels, which surpasses other method by a large margin. Similar trends can be observed on more difficult dataset CIFAR-100, ObCP shows robust superior performance with respect to different cycles. We achieve an accuracy of 68% with 35% labels, which has already outperformed all other methods with 40% labels, and the result can be further improved to 69.2% with 40% labels. This mainly owes to the designed one-bit query strategies and loss, which is able to efficiently make use of returned answers.

Table 2. Comparison of accuracy on ImageNet dataset, where 10% labels bits of information is used.

ImageNet				
Method	Backbone	$\|\mathcal{I}^S\|$	$\|\mathcal{P}^L\|$	Acc.(%)
VAAL	VGG16	10%	30%	53.3
Core-set	VGG16	10%	30%	52.2
MC-Dropout	VGG16	10%	30%	48.9
DBAL	VGG16	10%	30%	50.4
Ensembles w. VarR	VGG16	10%	30%	52.0
ObCP	VGG16	1%	10%	**61.3**
ObCP	Res50	1%	10%	**64.9**

4.1.2 ImageNet

Implementation details. For fair comparison, we conduct experiments using VGG-16 [27] as our backbone follow other baseline methods. For model training, we set the learning rate as 0.01 with batch size of 128. The λ_{ctr} is set to 5 while λ_{neg} is set to 0.1. We train our model for 100 epochs and start the one-bit query at 60. Other details are consistent with experiments on CIFAR10/100. Besides, for baseline methods, their results are borrowed from [28].

Results. As can be seen from Table 2, the performance of our model also demonstrates its superiority. Specifically, when we set the total labeled pool $|\mathcal{P}^L|$ as only 10%, we achieve the result of 61.3%, which improves the best baseline VAAL [28] that conducts over 30% labels by 8% accuracy. What's more, with a deeper architecture ResNet50

Table 3. Comparison of our method with the previous one-bit supervision method.

Dataset	Settings	One-bit	ObCP
CIFAR100	$\lvert\mathcal{I}^S\rvert$: 3K, $\lvert\mathcal{T}^O\rvert$: 47K	59.5%	**62.1%**
ImageNet	$\lvert\mathcal{I}^S\rvert$: 30K, $\lvert\mathcal{T}^O\rvert$: 977K	60.4%	**63.8%**

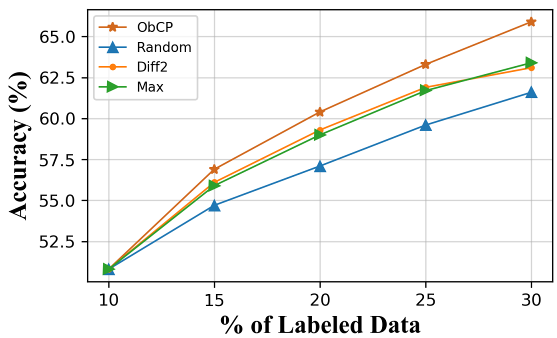

Figure 5. Model performance comparison with different sample selection methods on CIFAR-100.

[16], our result can be further improved to 64.9%, which demonstrates the robustness of our approach.

4.2. Comparison with One-bit Supervision Method

In order to better demonstrate the superiority of our designed strategies, we compare our approach with previous one-bit supervision [18] under the same protocol. We re-implemented the one-bit method by using the source-code [1] and all hyper-parameters follow its original settings. We conduct over ResNet-18 for CIFAR-100 dataset and train for 200 epochs, and ResNet-50 is used for the ImageNet dataset, which is trained for 60 epochs.

As shown in Table 3, we can observe that ObCP surpasses previous one-bit approach by a large margin on both CIFAR100 and ImageNet dataset. The improvement is 59.5% to 62.1% for CIFAR100, and 60.4% to 63.8% for ImageNet. The reason may be that the previous one-bit method only adopts correct prediction in the supervised term, which is optimized by cross-entropy loss, and does not effectively leverage the information of negative label. However, our designed pipeline maximizes the revenue generated by inquiry and significantly improves the performance of the model.

4.3. Comparisons of Different Sample Selection Strategies

In this section, we focus on comparing the performance with several selection methods re-implemented in our framework. For a C-classification problem, we denote p as the probability which represents the prediction output by the model, p_c is the cth element of p which represents the probability of assigning x to class c. We summarize the compared metrics [29] as follows:

- *Random*: which means each unlabeled sample is selected for active query under same probability.
- *diff2*: measuring the gap margin between the two most likely classes in prediction, here $\mathcal{U}(x) = 1 - (p_{c1} - p_{c2})$, where $c1$ and $c2$ denote the classes with 1st and 2nd highest probabilities.
- *max*: measuring the maximum confidence that the model has in any one label, here $\mathcal{U}(x) = 1 - \max_c p_c$.

We conduct the comparison experiments on CIFAR-100 and all settings are consistent except for sample selecting methods, and the results are shown in Figure 5. It can be observed that our method outperforms other baselines by a clear margin. Specifically, with 30% labels budgets, our method is superior to *Random* by $\sim$ 4.1% and *max* by $\sim$ 2.3%, as well as outperforms *diff2* by $\sim$ 2.4%. What's more, our selection method is more labeling-efficient, and we can get better results even with 25% labels than other methods which have 30% labels budgets. These results demonstrate that under the same label budget, our measurement can select more valuable samples, thus greatly improving the performance of the model.

4.4. Ablation Study

In this section, we conduct detailed ablation studies to inspect how each module affects the performance. For efficiency, **unless specified, all experiments are conducted on ResNet-50 with 100 epochs using a randomly selected subset of 100 categories in ImageNet**, and we call it ImageNet-100. We report top-1 accuracy in each experiment, where we achieve 79.3% accuracy with our proposed active learning strategy using 10% labeling budgets in total.

4.4.1 The Effectiveness of Each Module

We first conduct ablations to reveal the effectiveness of each module. The main modules include adding the correct prediction into the supervised loss, the modified contrastive loss, and the designed negative module. The results are summarized in Table 4, and we explain them as follows.

- The first and second rows represent the influence of our designed negative learning strategies. We notice that the negative contrasting and negative loss can bring about 0.8% and 2.2% improvement, respectively. Our guess is that this strategy helps the model eliminate a difficult option and enables the model with more powerful discriminative ability.

[1] https://github.com/huhengtong/one-bit-supervision

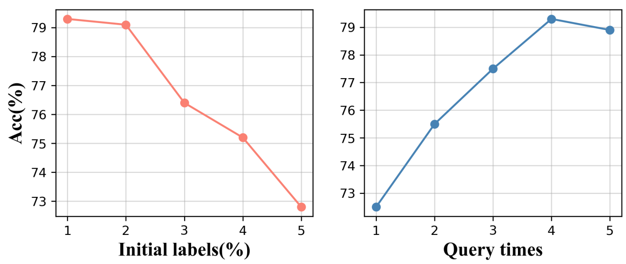

Figure 6. The influence of different parameters. From left to right are (1) Varying the number of labels we used in stage 1. (2) Varying the query times.

Table 4. The influence of different modules. Results are based on ImageNet-100 for 100 epochs.

Adding in sup.	Adding in ctr.	Neg ctr.	Neg loss.	Acc.
✓	✓	✗	✓	78.5% (-0.8%)
✓	✓	✓	✗	77.1% (-2.2%)
✓	✗	✓	✓	76.8% (-2.5%)
✗	✓	✓	✓	72.9% (-6.4%)
✓	✓	✓	✓	79.3%

- The third row means we include the correct label into contrastive loss, as shown in Eq. (3). We observe that this operation can also improve the model performance by 2.5%. It is because that incorporating class prior into contrasive makes the samples belong to the same category closer in feature space, which is beneficial for the classification task.
- The fourth row means we use the correct query label in the supervised loss. we can observe that this module clearly enhances the model's performance (+6.4%), which demonstrates that giving a correct label to uncertainty samples is more beneficial for classification.

4.4.2 Hyperparameters Analysis

Then we analyze the impact of several hyper-parameters, which include query times and initial labeled number $\mathcal{I}^S$.

Query times. The query time means that how many times we need to run out the one-bit query budgets $\mathcal{T}^O$. For simplicity, we equally assign the query quota according to the number of query times. The results are summarized in Figure 6. The advantage of using quota in multi-stage is clear, and the result of running out the quota $\mathcal{T}^O$ in 4 times is $\sim$6% higher than using up them at once. This mainly owes to the reason that the intermediate model is strengthened by ObCP and thus can find more positive labels than the initial model. We can also observe that too many query times may not bring positive impact on performance, and finally we choose it as 4 for our experiments.

Initial labeled number $\left|\mathcal{I}^S\right|$. Here we study the impact of different sizes of $\left|\mathcal{I}^S\right|$ when given the fixed label budget, which is used for the training of stage one. We adjust its size to 1% - 5% while given a fixed labeled pool $\left|\mathcal{P}^L\right|$ of 10%. As shown in Figure 6, the larger full-bit supervision data may not bring positive results, for it may reduce the advantage of our designed active learning cycle and one-bit learning strategies, which may make the framework degenerate to a regular semi-supervised learning method.

5. Limitations

As a new active learning framework, ObCP still suffers some limitations. For example, ObCP and previous one-bit supervision method [18] all focus on classification task. However, how to transfer this idea to other tasks, such as object detection, is still an open problem. For example, we can annotate image by querying "Does the bounding box is correct?" rather than giving an accurate box-level labels. But there exists many details to be fixed due to the introduction of localized bounding boxes. This remains to be studied in the future research.

6. Conclusion

This paper proposed a novel active learning framework, which integrates contrastive learning into a novel active learning method named one-bit query. We claim that the yes-or-no setting in one-bit query is intrinsically in accordance with contrastive learning that pulls positive pairs together and pushes negative samples away. Towards this goal, we design an uncertainty-based sample selection metric according to the variance of the model's prediction during the training process. Then in order to make better use of queried information, we design two branches according to different queried situations. The correct prediction will be added into the labeled pool and help improve the performance of contrastive learning. As for the incorrect prediction, we not only use this incorrect label information to generate hard negative samples for contrastive repelling, but also design a negative loss to keep samples away from the queried category. Experiments on several benchmarks demonstrate the effectiveness of the proposed framework, especially on the large-scale dataset.

Acknowledgement. This work has been partially supported by NSFC Grant 61932014, NSFC Grant 12141107, National Key R&D Program of China No.2020YFB1806700, Project BE2020026, the Key R&D Program of Jiangsu, China. We thank Yuang Deng, Siyan Li, Yinzhao Zheng and Zhao Jin for the assistance.

References

[1] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. In *CVPR*, June 2018. 6

[2] Razvan Caramalau, Binod Bhattarai, and Tae-Kyun Kim. Sequential graph convolutional network for active learning. In *CVPR*, pages 9583–9592, June 2021. 6

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33, 2020. 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[6] D.D.Lewis and W.A.Gale. A sequential algorithm for training text classifiers. *SIGIR'94*, 1994. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, and Fei Fei Li. Imagenet: A large-scale hierarchical image database. In *ICCV*, 2009. 1, 5

[8] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE TPAMI*, 38(9):1734–1747, 2015. 2

[9] A. Freytag, E. Rodner, and J. Denzler. Selecting influential examples: Active learning with expected model output changes. In *ECCV*. Springer, 2014. 2

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. pages 1050–1059, 2016. 6

[11] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR, 06–11 Aug 2017. 6

[12] Mingfei Gao, Zizhao Zhang, Guo Yu, Larry S. Davis Sercan O. Arik, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *ECCV*. Springer, 2020. 1

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33, 2020. 2

[14] Mahmudul Hasan and Amit K. Roy-Chowdhury. Context aware active learning of activity recognition models. In *ICCV*, pages 4543–4551, 2015. 1, 2

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 2, 3, 6

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6, 7

[17] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 2

[18] Hengtong Hu, Lingxi Xie, Zewei Du, Richang Hong, and Qi Tian. One-bit supervision for image classification. In *NeurIPS*, 2020. 1, 2, 7, 8

[19] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *CVPR*, pages 8166–8175, June 2021. 6

[20] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *ICCV*, 2019. 4

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009. 5

[22] Suichan Li, Dongdong Chen, Yinpeng Chen, Lu Yuan, Lei Zhang, Qi Chu, B. Liu, and Nenghai Yu. Improve unsupervised pretraining for few-label transfer. *ICCV*, pages 10181–10190, 2021. 2

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[24] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018. 1, 2, 6

[25] B. Settles. Active learning literature survey. *University of Wisconsinmadison*, 2010. 1

[26] M. Settles, B.and Craven and S. Ray. Multiple-instance active learning. In *NeurIPS*, 2008. 2

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 6

[28] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *ICCV*, October 2019. 5, 6

[29] Shuang Song, David Berthelot, and Afshin Rostamizadeh. Combining mixmatch and active learning for better accuracy with fewer labels. *arXiv preprint arXiv:1912.00594*, 2019. 7

[30] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 2

[31] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *CVPR*, June 2019. 6

[32] Beichen Zhang, Liang Li, Shijie Yang, Shuhui Wang, Zheng-Jun Zha, and Qingming Huang. State-relabeling adversarial active learning. In *CVPR*, June 2020. 6