Agreement-Discrepancy-Selection: Active Learning with Progressive Distribution Alignment

Mengying Fu[†], Tianning Yuan[†], Fang Wan[†*], Songcen Xu[‡], Qixiang Ye[†*]

[†] University of Chinese Academy of Sciences, Beijing, China
[‡] Noah's Ark Lab, Huawei Technologies, Shenzhen, China
{fumengying19, yuantianning19}@mails.ucas.ac.cn, xusongcen@huawei.com, {wanfang, qxye}@ucas.ac.cn

Abstract

In active learning, the ignorance of aligning unlabeled samples' distribution with that of labeled samples hinders the model trained upon labeled samples from selecting informative unlabeled samples. In this paper, we propose an agreement-discrepancy-selection (ADS) approach, and target at unifying distribution alignment with sample selection by introducing adversarial classifiers to the convolutional neural network (CNN). Minimizing classifiers' prediction discrepancy (maximizing prediction agreement) drives learning CNN features to reduce the distribution bias of labeled and unlabeled samples, while maximizing classifiers' discrepancy highlights informative samples. Iterative optimization of agreement and discrepancy loss calibrated with an entropy function drives aligning sample distributions in a progressive fashion for effective active learning. Experiments on image classification and object detection tasks demonstrate that ADS is task-agnostic, while significantly outperforms the previous methods when the labeled sets are small.

Introduction

The key idea behind active learning is that a machine learning algorithm can achieve better performance with fewer training labels if it is allowed to choose the data it wants to learn from. Despite of the rapid progress of learning methods with less supervision, *e.g.*, weakly supervised learning and semi-supervised learning, active learning remains the cornerstone of many artificial intelligence applications for its simplicity and higher performance bound.

The majority of previous researches suggests that active learning is an empirical method which generalizes models trained on a labeled set to an unlabeled set by iterative sample selection. Uncertainty-based methods define various metrics to select informative samples to adapt the trained model to the unlabeled set (Gal, Islam, and Ghahramani 2017). Distribution-based approaches aim at estimating the layout of unlabeled samples for selecting samples of large diversity or loss. Expected model change methods (Freytag, Rodner, and Denzler 2014; Käding et al. 2016) find out samples which can cause the greatest change to the model parameters or prediction samples' loss (Yoo and Kweon 2019).

*Correspond author.

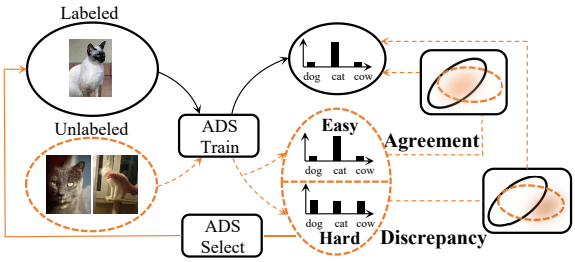

Figure 1: Overview of ADS, which leverages the prediction agreement and discrepancy to select informative unlabeled samples.

Despite of the great progress, most existing methods remain simply generalizing the models trained on the labeled set to the unlabeled set while ignoring the distribution alignment issue. This is problematic when there is a significant distribution bias between the labeled and unlabeled sets (Gudovskiy et al. 2020). Active learning fused with deep learning has alleviated this problem by sharing an implicit feature space. However, there remains lacking an explicit way to unify distribution alignment with sample selection, which hinders the model trained upon labeled samples from selecting informative unlabeled samples.

In this paper, we propose the ADS approach[1] and target at unifying distribution alignment with sample selection in a continuous and explainable manner. Considering the unreliability of predictions themselves, we propose to leverage the prediction agreement and discrepancy of two classifiers to estimate the distribution continuity, Fig. 1. The motivation behind ADS is that maximizing the prediction agreement and discrepancy upon unlabeled samples makes it possible to quantify the distribution overlap and bias, while avoiding directly predicting the uncertainty or diversity of samples.

To fulfill this purpose, we introduce adversarial classifiers atop the convolutional neural network (CNN). During training, minimizing the prediction discrepancy (maximizing their agreement) of fixed classifiers' drives learning CNN features to align the distribution of easy unlabeled samples. Maximizing classifiers' prediction discrepancy upon fixed features finds out the hard samples, which

[1]Code is enclosed in the supplementary material.

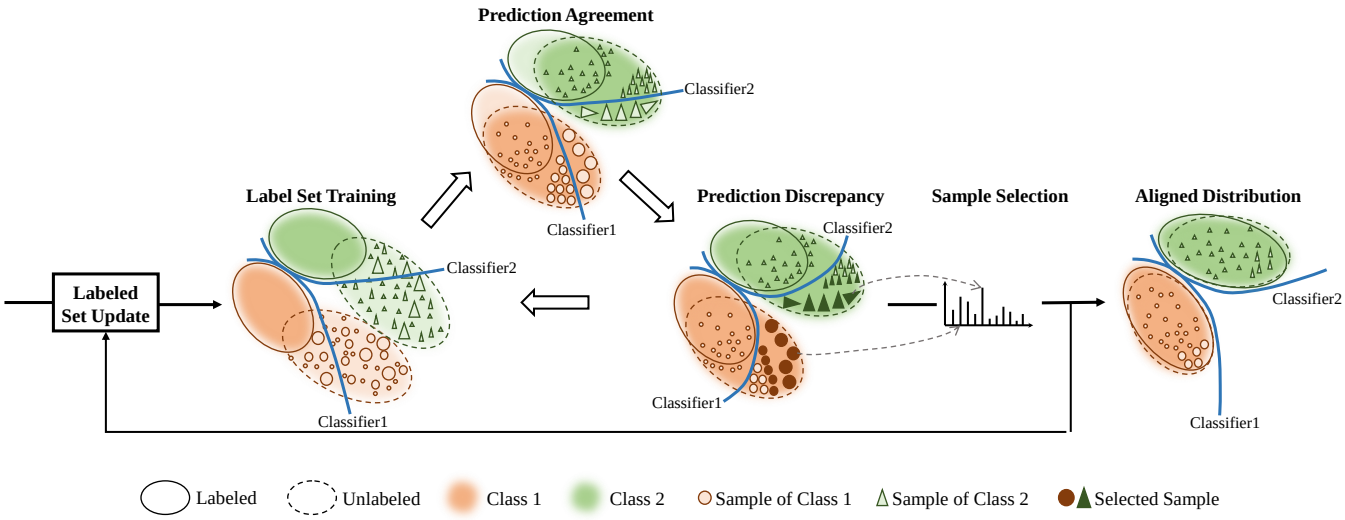

Figure 2: ADS flowchart.The prediction agreement step "pulls" the distributions of labeled and unlabeled samples with low and mid entropy together by updating features while the prediction discrepancy step "push" the distribution of unlabeled samples with high and middle entropy out of the alignment area by updating classifiers. Iterative agreement-discrepancy progressively aligns distributions of unlabeled samples with those of labeled samples. Larger circles/triangles denote more informative samples with larger entropy.

are highlighted by an entropy-based calibration function. Iterative agreement-discrepancy progressively aligns distributions of unlabeled samples with those of labeled samples in a way like domain adaptation for active learning, Fig. 2.

The contributions of this paper include:

- We propose an agreement-discrepancy-selection (ADS) approach, solving the active learning problem by aligning the distributions of unlabeled samples with those of labeled samples in a continuous and progressive fashion.
- We design an entropy-based metric to measure the distribution alignment and discrepancy. Based on the metric, we further propose entropy-based calibration functions to differentiate informative samples with easy samples.
- We apply ADS to image classification and object detection, improving the state-of-the-arts with significant margins.

Related Work

Uncertainty-based Method. Active learning, for its practical application value, has been one of the most important research topic in machine learning and artificial intelligence. Conventional methods used uncertainty as a metric to select samples for active learning (Settles 2012). Uncertainty can be defined as the the posterior probability of a predicted class (Lewis and Gale 1994; Lewis and Catlett 1994), or the margin between posterior probabilities of a predicted class and the secondly predicted class (Joshi, Porikli, and Papanikolopoulos 2009; Roth and Small 2006). It can also be defined upon entropy (Settles and Craven 2008; Luo, Schwing, and Urtasun 2013; Joshi, Porikli, and Papanikolopoulos 2009), which measures the posterior probability of unlabeled samples.

Combined with deep learning, an improved uncertainty approach (Gal, Islam, and Ghahramani 2017) used Monte Carlo Dropout and multiple forward passes to estimate uncertainty. Despite of its effectiveness, the efficiency is significantly reduced for the usage of dense dropout layers which hinders the network convergence.

Distribution-based Method. This line of methods targets at estimating the distribution of unlabeled samples for selecting diverse and informative samples. Clustering methods (Nguyen and Smeulders 2004) have been applied to build the unlabeled sample distribution while discrete optimization methods (Guo 2010; Elhamifar et al. 2013; Yang et al. 2015) were employed to perform sample subset selection. By considering the distances to their surrounding samples, the context-aware methods (Hasan and Roy-Chowdhury 2015; Aodha et al. 2014) select the samples that can represent to the global distribution. The expected model change methods (Roy and McCallum 2001; Settles, Craven, and Ray 2007) utilized the present model to estimate expected gradient, or expected output changes (Freytag, Rodner, and Denzler 2014; Käding et al. 2016), which guide the sample selection.

Core-set (Sener and Savarese 2018) suggested that many of the active learning heuristics in the literature were not effective when applied to CNNs with batch setting. It thus defined the problem of active learning as core-set selection, $i.e.$, choosing a set of points such that a model learned over the labeled subset captures the diversity of the unlabeled samples.

Learning Loss Method. In the deep learning era, the active learning methods remain falling into the uncertainty-based and distribution-based routines (Lin et al. 2018; Wang et al. 2017; Beluch et al. 2018; Lin et al. 2020). Sophisticated

methods have extended active learning to open sets (Liu and Huang 2019), or combined it with self-paced learning (Tang and Huang 2019). Nevertheless, it remains questionable whether or not the intermediate feature representation is effective for sample selection. Recent learning loss approach (Yoo and Kweon 2019) can be categorized to either an uncertainty-based or a distribution-based approach. By introducing the network structure to predict the "loss" of unlabeled samples, it estimates sample uncertainty and distribution, and selects samples of large "loss" in a fashion like hard negative mining.

Despite of the great progress, the continuous distribution of labeled and unlabeled samples remain not well modeled, which causes the gap between trained model and the unlabeled samples to be predicted. Recent active learning combined with self-supervised learning provided an interesting solution, but is difficult to be extended to other tasks like object detection (Gudovskiy et al. 2020). Motivated by the model ensemble method (Beluch et al. 2018), our study solves this problem by iterative prediction agreement and discrepancy (Saito et al. 2018). Our work is also inspired by the uncertainty-aware graph Gaussian process (Liu et al. 2020) which models continuous distribution with graph. The difference between our approach the adversarial learning approaches (Sinha, Ebrahimi, and Darrell 2019; Zhang et al. 2020) lies in that they learn how to discriminate between sample dissimilarities in the latent space while we focus on modeling the predictions upon unlabeled samples.

The Proposed Approach

The core of ADS is leveraging the prediction agreement and discrepancy of adversarial classifiers to estimate the distribution of unlabeled samples. In each training iteration, three steps are successively performed: (1) Training the backbone network (feature extractor) and the classifiers using the labeled set; (2) Fixing the classifiers, fine-tuning the feature extractor to maximize the prediction agreement ($i.e.$, to minimize the prediction discrepancy) on the unlabeled set to align the distributions of unlabeled samples with those of labeled samples; (3) Fixing the feature extractor, fine-tuning the classifiers to maximize the prediction discrepancy on the unlabeled set and highlight informative samples. After each training iteration, an entropy-based metric is used to select informative samples, which will be used to update the label set for the next iteration of active learning, Fig. 2.

Label Set Training

Let L denotes the labeled set, U the unlabeled set, and C the number of classes. A sample $x_l \in \mathbb{R}^{H*W*3}$ from the L has the label y_l. To quantify the distribution bias and distribution alignment between L and U, we introduce two adversarial classifiers after the last convolutional layer, Fig. 3(a), where g denotes the feature extractor parameterized by θ_g, and f_1 and f_2 are two adversarial classifiers parameterized by θ_{f_1} and θ_{f_2}, respectively.

Given the labeled set L, the purpose of network training is to optimize both θ_g and θ_{f_1} and θ_{f_2} by minimizing the Binary Cross Entropy (BCE) loss using the Stochastic Gradient Descent (SGD) algorithm, as

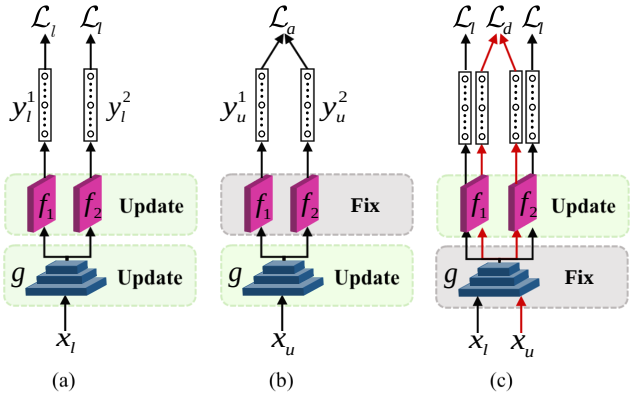

Figure 3: Network architectures. (a) Label set training. (b) Prediction agreement. (c) Prediction discrepancy.

$$\begin{aligned}&\underset{\theta_g,\theta_{f_1},\theta_{f_2}}{\operatorname{argmin}} \mathcal{L}_l(\hat{y}_l^1, \hat{y}_l^2)\\ &= -\sum_{c=1}^{C}(y_{l,c} log \hat{y}_{l,c}^1 + (1-y_{l,c}) log(1-\hat{y}_{l,c}^1)\\ &\quad + y_{l,c} log \hat{y}_{l,c}^2 + (1-y_{l,c}) log(1-\hat{y}_{l,c}^2)).\end{aligned} \tag{1}$$

Given an unlabeled sample for test, the network with two adversarial classifiers takes x_u as input and generates two predictions, $\hat{y}_u^1$ and $\hat{y}_u^2$, where $\hat{y}_u^1 = f_1(g(x_u))$ and $\hat{y}_u^2 = f_2(g(x_u))$, $\hat{y}_u \in [0, 1]$.

When there is a significant distribution bias between L and U, unlabeled samples in the biased regions are difficult to be classified by the models trained on the labeled set. In previous studies, samples were selected by empirically designed metrics to handle the bias. However, when the bias is significant, the model experience difficulty to precisely predict the classification probability of unlabeled samples, which could waste the quota for labeling. In what follows, the prediction agreement and prediction discrepancy modules are proposed to solve this problem.

Prediction Agreement: Distribution Alignment

To select informative samples from the unlabeled set using the model trained on the labeled set, the key is to find a reasonable way to reduce the distribution bias and increase distribution alignment between labeled and unlabeled samples. To this end, we proposed a method to update the network parameters and the feature representation so that the prediction agreement of f_1 and f_2 upon unlabeled samples is maximized. By forcing the two classifiers to agree with each other on predictions, the feature representation is updated so that some unlabeled samples can "move" towards L in the feature space, Fig. 2. The unlabeled samples "staying" in the biased regions are considered informative and selected for labelling.

Particularly, each sample $x_u \in \mathbb{R}^{H*W*3}$ from the unlabeled set is taken as the input of the network to predict

classification outputs, Fig. 3(b). This is an efficient inference procedure given network parameters trained on the labeled set. The adversarial classifiers f_1 and f_2 are both used in the forward propagation to predict classification results. When performing back-propagation, the parameters θ_{f_1} and θ_{f_2} of the classifiers are fixed so that solely the parameters θ_g of the feature extractor are fine-tuned. This actually defines a procedure to update the feature space for sample alignment, when the parameters θ_g of the feature extractor are optimized by minimizing the prediction agreement loss, as

$$\underset{\theta_g}{\operatorname{argmin}} \mathcal{L}_{a'}(\hat{y}_u^1, \hat{y}_u^2) = \frac{1}{C}\sum_{c=1}^{C}\left|\hat{y}_{u,c}^1 - \hat{y}_{u,c}^2\right|, \qquad (2)$$

where $\hat{y}_{u,c}^1$ and $\hat{y}_{u,c}^2$ are the predictions of the sample $x_{u,c}$ output by f_1 and f_2, respectively.

To qualify the prediction alignment, we propose an entropy-based metric, $E(u)$, which is defined as the mean entropy based on the classifiers' predictions, as

$$E(u) = -\frac{1}{2}(\sum_{c=1}^{C}\hat{y}_{u,c}^1 log\hat{y}_{u,c}^1 + \sum_{c=1}^{C}\hat{y}_{u,c}^2 log\hat{y}_{u,c}^2), \qquad (3)$$

where $\hat{y}_{u,c}^1$ and $\hat{y}_{u,c}^2$ respectively denote the prediction probability of the c-th class by f_1 and f_2 on the unlabeled set. The entropy assigns each unlabeled sample a weight indicating how well it aligns with the labeled set, providing a way to measure the distribution alignment. Based on the entropy, a calibration weight is designed to differentiate the samples with large entropy from those with small entropy. Considering that entropy is non-negative while the Sigmoid function used by f_1 and f_2 has the largest slope near the origin, we use the Sigmoid function to calculate the calibration weight w_a, as

$$w_a = \frac{1}{n}\big(1 - \delta(E(u) - \tau)\big), \qquad (4)$$

where n denotes the batch size and $\delta(x) = \frac{1}{1+e^{-x}}$ the Sigmoid function. τ is a hyper-parameter, which is experimentally set to 0.1.

According to Eq. 4, we assign larger alignment weights to the samples with smaller entropy, and smaller alignment weights to the samples with larger entropy. The advantages of entropy calibration are two-folds: (1) It prevents hard samples from alignment process and avoid the negative effect of them on feature learning; (2) By selecting easy samples but leaving hard samples out, the distance between easy samples and hard samples is further enlarged, which facilitates highlighting the hard samples. These advantages are shown in the last three rows of Tab. 1. Accordingly, the prediction agreement is implemented by minimizing the calibrated loss $\mathcal{L}_a$ to optimize the network parameters, as

$$\underset{\theta_g}{\operatorname{argmin}} \mathcal{L}_a = w_a \mathcal{L}_{a'}(\hat{y}_u^1, \hat{y}_u^2). \qquad (5)$$

Prediction Discrepancy: Highlighting Informative Samples

By maximizing the prediction agreement, L and U are aligned in the feature space as much as possible. In the following step, we propose to maximize the discrepancy of predictions to highlight informative unlabeled samples, Fig. 2.

Particularly, we fix the feature extractor's parameters θ_g and fine-tune the classifiers' parameters θ_{f_1} and θ_{f_2} to minimize a discrepancy loss, Fig. 3(c). The fine-tuning procedure drives the two classifiers, f_1 and f_2, to output discrepant predictions on each unlabeled sample $x_{u,c}$, as

$$\underset{\theta_{f_1},\theta_{f_2}}{\operatorname{argmin}} \mathcal{L}_{d'}(\hat{y}_u^1, \hat{y}_u^2) = 1 - \frac{1}{C}\sum_{c=1}^{C}\left|\hat{y}_{u,c}^1 - \hat{y}_{u,c}^2\right|, \qquad (6)$$

where $\hat{y}_{u,c}^1$ and $\hat{y}_{u,c}^2$ are the predictions of classifier f_1 and f_2 of the unlabeled sample $x_{u,c}$. When optimizing Eq. 6, it requires to simultaneously minimize the classification loss defined in Eq. 1 to prevent the performance degradation on the labeled samples. The loss of labeled examples and unlabeled examples are mixed together and train the model once.

Under the constraint of discrepancy loss, not all samples have discrepant predictions, $i.e.$, some easy samples remain outputting similar predictions. We design a discrepancy calibration weight to handle them. Considering that smaller entropy means smaller prediction discrepancy, the discrepancy calibration weight w_d is defined as

$$w_d = \frac{1}{n}(\delta(E(u) - \tau)), \qquad (7)$$

where $E(u)$ follows Eq. 3 on the prediction of the current epoch. The prediction discrepancy is implemented by optimizing the calibrated discrepancy loss $\mathcal{L}_d$, as

$$\underset{\theta_{f_1},\theta_{f_2}}{\operatorname{argmin}} \mathcal{L}_d = w_d \mathcal{L}_{d'}(\hat{y}_u^1, \hat{y}_u^2). \qquad (8)$$

Algorithm 1: ADS Training Procedure

Require: Network parameters θ_g, classifiers' parameters θ_{f_1} and θ_{f_2}, labeled set L and unlabeled set U.
for *iteration* **do**
  for *epoch* **do**
    if *epoch == 0* **then**
      Training on L using Eq. 1;
    Compute the calibration weight w_a;
    Maximize prediction agreement upon U using Eq. 5;
    Compute the calibration weight w_d;
    Maximize prediction discrepancy on U and L using Eq. 8 and Eq. 1;
    Training on L using Eq. 1;
  Select samples using the entropy metric, Eq. 3;
  update L and U.

Entropy-based Sample Selection

After each iteration with multiple epochs of training, a small proportion of samples staying in the bias distribution region would be informative samples to be selected. To quantify how informative each sample is, we propose the entropy-based sample selection metric, $E(u)$, following Eq. 3, which

is based on the fact that larger entropy implies larger uncertainty of probabilistic predictions. The selected samples will be added to the labeled samples for next iteration of active learning, Fig. 2.

The learning procedure (described in Alg. 1) of ADS is an adversarial min-max discrepancy procedure, which aims to progressively push the unlabeled distribution towards the labeled distribution by leveraging their overlap and bias. This is like a kind of continuous domain adaptation where the "source" domain is the labeled samples and the "target" domain is the unlabeled samples. During the learning procedure, by using the calibration weights to highlight informative samples and filter out easy samples, the major proportion of unlabeled samples are aligned with the labeled samples.

Experiments

We evaluate the proposed approach upon image classification and object detection tasks. In experiments, a labeled dataset L_k^0 is initialized by randomly sampling k_0 data points from the whole dataset U_N, where N denotes the number of samples. In the i-th iteration of active learning, we add k_i labeled samples to the labeled set and then re-train the network. To report the mean and standard deviation of performance, the experiment repeats for three times.

Experimental Settings

Dataset. The commonly used CIFAR-10 and CIFAR-100 datasets are used in the image classification task, following the experimental settings (Yoo and Kweon 2019; Sinha, Ebrahimi, and Darrell 2019; Zhang et al. 2020). CIFAR-10 consists of 60000 images of $32 \times 32 \times 3$ pixels. The training and test sets contain 50000 and 10000 images, respectively. CIFAR-100 is a fine-grained dataset, which consists of 100 categories containing 600 images each.

Training Settings. We respectively use ResNet-18 (He et al. 2016) and VGG-16 (Simonyan and Zisserman 2015) as backbone networks by removing the fully-connected (FC) layers and adding two classifiers atop the backbone network to implement ADS. Considering the budget of the labeled set, we set k_0=1000 and k_i=1000 when using ResNet-18 on CIFAR-10 while set k_0=5000 and k_i=2500 when using ResNet-18 on CIFAR-100 or using VGG-16. Data augmentation strategies including 32×32 random image crop and random image horizontal flip. The images are normalized using the channel mean and standard deviation vectors estimated over the training set.

For each learning iteration, we train the model for 200 epochs with the mini-batch size 128 and the initial learning rate 0.1. After 160 epochs, the learning rate decrease to 0.01. The momentum and the weight decay are respectively set to 0.9 and 0.0005.

Sub-set Sampling. The training set is regarded as the initial unlabeled set U_N, where N=50000 denote the sample number. According to (Settles 2012; Sener and Savarese 2018), selecting top-k_i samples from U_N directly is inefficient because of the information overlap among images. To tackle this issue, we follow the settings in (Beluch et al. 2018) to randomly select a subset $S_M \subset U_N$ first and then choose top-k_i representative samples from S_M. The sample number (M) is empirically set to 10000.

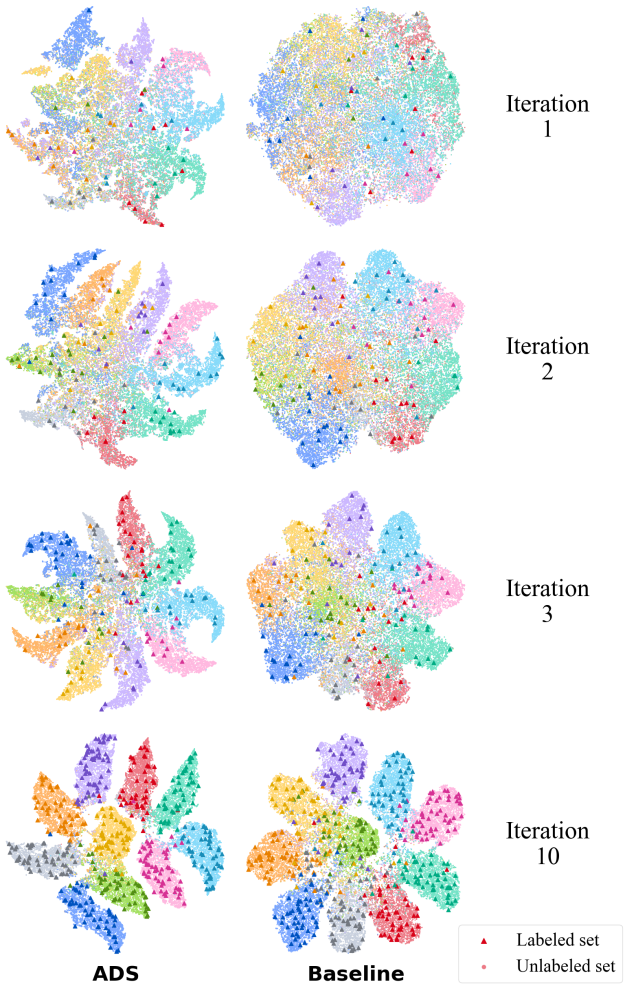

Figure 4: T-SNE visualization of sample distributions. "Triangles" and "dots" respectively denote labeled and unlabeled samples. The "triangles" are progressively aligned with "dots", showing that ADS aligns the distributions of unlabeled samples with those of labeled samples while training discriminative models. The baseline method uses a single classifier to randomly select samples. (Best viewed in color with zoom).

Ablation Study

Prediction Agreement and Discrepancy. To evaluate the effect of the distribution agreement and discrepancy, we conduct ablation study on CIFAR-10. As shown in Tab. 1, by using the prediction agreement and discrepancy module without entropy weights and selecting samples randomly, ADS significantly boosts the performance at early iterations. Particularly, it improves the accuracy of the second iteration by 6.27%, from 61.48% to 67.75%. By using the entropy weight to calibrate the agreement loss (Eq. 5), ADS im-

Table 1: Module evaluation on CIFAR-10 using ResNet-18. "Non", "Ent.(w.)" and "Cal." respectively denote ADS without entropy weight, with entropy weight and calibration weight. "Ent.(sel.)" denotes ADS using the entropy metric to select samples.

ADS				Accuracy (%) on Proportion (%) of Labeled Samples									
Non	Ent. (w.)	Cal.	Ent. (sel.)	2	4	6	8	10	12	14	16	18	20
				51.01	61.48	69.14	75.14	79.77	82.83	84.77	85.78	86.89	87.27
✓				**58.07**	67.75	74.91	78.88	80.96	83.23	84.66	85.29	86.50	87.24
✓			✓	54.28	66.23	74.61	80.18	82.89	85.99	**88.00**	88.86	89.86	90.41
	✓		✓	55.43	67.21	75.49	80.08	83.46	85.40	87.13	88.55	89.72	90.02
		✓	✓	**57.22**	**70.08**	**78.18**	**82.30**	**83.97**	**86.78**	87.82	**89.05**	**90.03**	**90.63**

Table 2: Comparison of prediction alignment metrics on CIFAR-10. "ADS(non/max/min/mean)" respectively denote ADS without entropy weight, with the max entropy weight, the min entropy weight, the mean entropy weight of the two classifiers.

Metric	Accuracy (%) on Proportion (%) of Labeled Samples									
	2	4	6	8	10	12	14	16	18	20
Baseline	51.01	61.48	69.14	75.14	79.77	82.83	84.77	85.78	86.89	87.27
ADS (non)	54.28	66.23	74.61	80.18	82.89	85.99	88.00	88.86	89.86	90.41
ADS (max)	54.73	66.51	74.7	79.89	83.13	85.03	86.56	88.29	89.31	89.87
ADS (min)	54.31	65.61	73.87	79.65	82.89	**85.49**	86.85	88.36	89.41	**90.04**
ADS (mean)	**55.43**	**67.21**	**75.49**	**80.08**	**83.46**	85.40	**87.13**	**88.55**	**89.72**	90.02

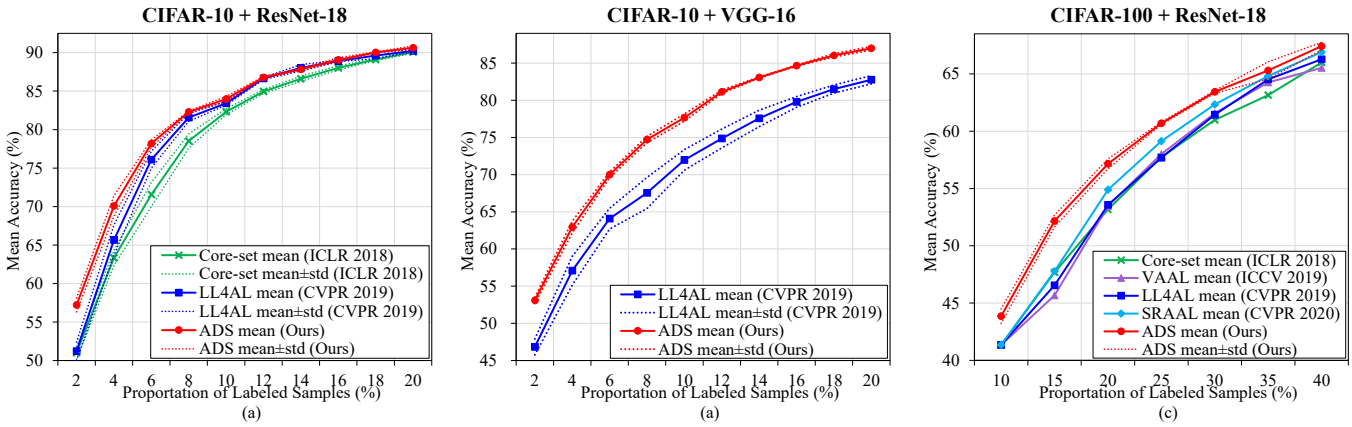

Figure 5: Comparison of ADS with Core-set (Sener and Savarese 2018), VAAL (Sinha, Ebrahimi, and Darrell 2019), LL4AL (Yoo and Kweon 2019) and SRAAL (Zhang et al. 2020): (a) on CIFAR-10 using the ResNet-18 backbone, (b) on CIFAR-10 using the VGG-16 backbone, (c) on CIFAR-100 using the ResNet-18 backbone.

proves performance quickly when using 2%∼10% samples. When using the calibration entropy weights (Eqs. 4 and 7), ADS achieves the highest performance in almost all training iterations. This confirms the effectiveness of the proposed calibration metric in differentiating hard and easy samples.

The primary reason for the performance improvement of ADS at early iterations lies in the usage of unlabeled samples. The prediction discrepancy on unlabeled samples, as a supervision signal, can improve feature representation, particularly when the labeled set is very small. With the increase of the labeled samples, the effect of such supervision signal would decay.

Progressive Distribution Alignment. In Fig. 4, we visualize and compare the samples' distributions in four learning iterations. It can be seen that ADS can classify the samples quickly and clearly. Meanwhile, it can align the distributions of unlabeled samples with those of labeled samples more efficiently, and purposefully select the informative samples around class boundary.

Prediction Alignment Metric. The CIFAR-10 dataset

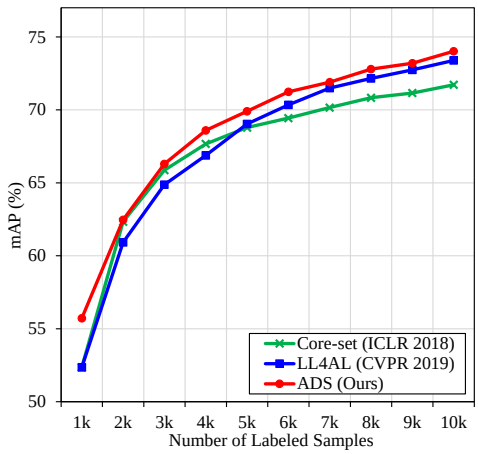

Figure 6: Comparison of ADS with Core-set (Sener and Savarese 2018) and LL4AL (Yoo and Kweon 2019) on PASCAL VOC using the VGG-16 backbone.

is also used to evaluate the effect of prediction alignment/discrepancy metric. In Tab. 2, "ADS(mean)", which uses the mean entropy of the two classifiers as the metric, significantly outperforms other metrics, *e.g.*, "ADS(non)", "ADS(min)", and "ADS(max)". The reason lies in that using the mean entropy can take into account the prediction outputs of both classifiers and avoid one classifier overwhelming the other.

Performance and Comparison

CIFAR-10. As shown in Fig. 5(a), ADS significantly outperforms state-of-the-arts, particularly at the early iterations. It respectively outperforms the state-of-the-art LL4AL by 6.00%, 4.43% and 2.09% when using 2%, 4% and 6% samples. Such improvements validate that ADS can align the distributions of unlabeled with those of labeled samples and select representative samples using a small training sets, which is very important for active learning. In the last iteration, with 20% samples, ADS achieves 90.63% accuracy, which has been very close to that on the full training set. Using VGG-16 as a backbone network, ADS has higher accuracy and less standard deviation than LL4AL, Fig. 5(b), demonstrating its robustness to random initialization.

CIFAR-100. Compared with CIFAR-10, CIFAR-100 is a more challenging dataset for the larger category number. Therefore, larger proportions of training samples are required to obtain acceptable performance. Fig. 5(c) shows that ADS significantly outperforms all other methods with a smaller standard deviation. Particularly, it respectively outperforms the SRAAL method by 2.51%, 4.40%, and 2.25% using 10%, 15% and 20% samples. The SRAAL has a comparable performance with ADS in the last two iteration. SRAAL solely uses the label state information of samples while ignoring aligning the continuous distribution of unlabeled samples with those of labeled samples, which makes it selecting less representative samples in early iterations.

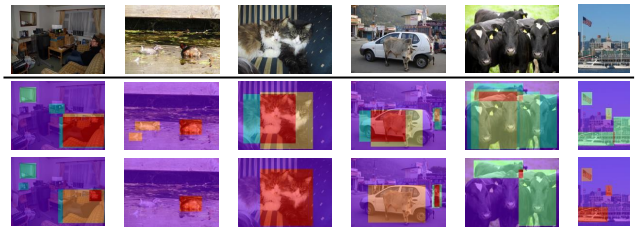

Figure 7: Visualization of object predictions of the two adversarial classifiers. The first row shows the original images, the second and the third rows show the predictions of classifier 1 and classifier 2 respectively. Redder colors indicate higher scores. (Best viewed in color with zoom)

Object Detection

Following the settings in LL4AL (Yoo and Kweon 2019), we apply ADS to object detection using the SSD detector (Liu et al. 2016). We add two fully convolutional layers as the adversarial classifiers, which are 3×3 kernels with the stride of 1 and the padding of 1. For each image, all the feature vectors generated by the detector are fed to the ADS module for entropy calculation. For multiple feature vectors, the vector of the top-1000 largest entropy is used as the metric to determine whether the image is informative or not.

The experiments are conducted on PASCAL VOC 2007 and 2012 (Everingham et al. 2010), where 1000 samples are selected from the training set as the initial labeled set. In each iteration, 1000 samples are selected and added to the labeled set. The agreement-discrepancy module is trained for 30 epochs, while the labeled set training module for 150 epochs. The agreement-discrepancy module only works for the positive images which contain at least one object. As shown in Fig. 6, ADS outperforms the state-of-the-arts, validating its effectiveness for object detection.

In Fig. 7, we visualize the predictions (classification scores) of the two adversarial classifiers with score threshold 0.3 after the NMS step with IoU threshold 0.5. It can be seen that ADS produces informative predictions with discrepant scores. The images containing larger discrepancy are selected for data annotation.

Conclusion

We proposed the Agreement-Discrepancy-Selection (ADS) approach for active learning and unify the distribution alignment with sample selection by introducing adversarial classifiers. By operating the classifiers' prediction agreement and discrepancy, ADS quantified the distribution overlap and bias without directly predicting the uncertainty or diversity of samples. With well-designed calibration weights, ADS further differentiated the alignment and discrepancy of unlabeled samples, which facilitates informative sample selection. Experiments on image classification and object detection benchmarks demonstrate the task-agnostic advantage of ADS. The ADS approach provides a fresh insight to the classical active learning problem.

Acknowledgement

This work was supported in part by National National Natural Science Foundation of China (NSFC) under Grant 61836012, 61771447 and 62006216, Strategic Priority Research Program of Chinese Academy of Science under Grant XDA27010303, and Post Doctoral Innovative Talent Support Program of China under Grant 119103S304.

References

Aodha, O. M.; Campbell, N. D. F.; Kautz, J.; and Brostow, G. J. 2014. Hierarchical Subquery Evaluation for Active Learning on a Graph. In *IEEE CVPR*, 564–571.

Beluch, W. H.; Genewein, T.; Nürnberger, A.; and Köhler, J. M. 2018. The Power of Ensembles for Active Learning in Image Classification. In *IEEE CVPR*, 9368–9377.

Elhamifar, E.; Sapiro, G.; Yang, A. Y.; and Sastry, S. S. 2013. A Convex Optimization Framework for Active Learning. In *IEEE ICCV*, 209–216.

Everingham, M.; Gool, L. V.; Williams, C. K. I.; Winn, J. M.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comp. Vis.* 88(2): 303–338.

Freytag, A.; Rodner, E.; and Denzler, J. 2014. Selecting Influential Examples: Active Learning with Expected Model Output Changes. In Fleet, D. J.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *ECCV*, volume 8692, 562–577.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep Bayesian Active Learning with Image Data. In Precup, D.; and Teh, Y. W., eds., *ICML*, volume 70, 1183–1192.

Gudovskiy, D. A.; Hodgkinson, A.; Yamaguchi, T.; and Tsukizawa, S. 2020. Deep Active Learning for Biased Datasets via Fisher Kernel Self-Supervision. In *IEEE CVPR*, 9038–9046.

Guo, Y. 2010. Active Instance Sampling via Matrix Partition. In Lafferty, J. D.; Williams, C. K. I.; Shawe-Taylor, J.; Zemel, R. S.; and Culotta, A., eds., *NeurIPS*, 802–810.

Hasan, M.; and Roy-Chowdhury, A. K. 2015. Context Aware Active Learning of Activity Recognition Models. In *IEEE ICCV*, 4543–4551.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE CVPR*, 770–778.

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N. 2009. Multi-class active learning for image classification. In *IEEE CVPR*, 2372–2379.

Käding, C.; Rodner, E.; Freytag, A.; and Denzler, J. 2016. Active and Continuous Exploration with Deep Neural Networks and Expected Model Output Changes. *CoRR* abs/1612.06129.

Lewis, D. D.; and Catlett, J. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning. In Cohen, W. W.; and Hirsh, H., eds., *Machine Learning*, 148–156.

Lewis, D. D.; and Gale, W. A. 1994. A Sequential Algorithm for Training Text Classifiers. In Croft, W. B.; and van Rijsbergen, C. J., eds., *SIGIR*, 3–12.

Lin, L.; Wang, K.; Meng, D.; Zuo, W.; and Zhang, L. 2018. Active Self-Paced Learning for Cost-Effective and Progressive Face Identification. *IEEE PAMI* 40(1): 7–19.

Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; and Shao, L. 2020. HRank: Filter Pruning using High-Rank Feature Map. In *IEEE CVPR*, 1529–1538.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *ECCV*, 21–37.

Liu, Z.; and Huang, S. 2019. Active Sampling for Open-Set Classification without Initial Annotation. In *AAAI*, 4416–4423.

Liu, Z.; Li, S.; Chen, S.; Hu, Y.; and Huang, S. 2020. Uncertainty Aware Graph Gaussian Process for Semi-Supervised Learning. In *AAAI*, 4957–4964.

Luo, W.; Schwing, A. G.; and Urtasun, R. 2013. Latent Structured Active Learning. In Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; and Weinberger, K. Q., eds., *NeurIPS*, 728–736.

Nguyen, H. T.; and Smeulders, A. W. M. 2004. Active learning using pre-clustering. In Brodley, C. E., ed., *ICML*.

Roth, D.; and Small, K. 2006. Margin-Based Active Learning for Structured Output Spaces. In Fürnkranz, J.; Scheffer, T.; and Spiliopoulou, M., eds., *ECML*, volume 4212, 413–424.

Roy, N.; and McCallum, A. 2001. Toward Optimal Active Learning through Sampling Estimation of Error Reduction. In Brodley, C. E.; and Danyluk, A. P., eds., *ICML*, 441–448.

Saito, K.; Watanabe, K.; Ushiku, Y.; and Harada, T. 2018. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *IEEE CVPR*, 3723–3732.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *ICLR*.

Settles, B. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning.

Settles, B.; and Craven, M. 2008. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *EMNLP*, 1070–1079.

Settles, B.; Craven, M.; and Ray, S. 2007. Multiple-Instance Active Learning. In Platt, J. C.; Koller, D.; Singer, Y.; and Roweis, S. T., eds., *NeurIPS*, 1289–1296.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Bengio, Y.; and LeCun, Y., eds., *ICLR*.

Sinha, S.; Ebrahimi, S.; and Darrell, T. 2019. Variational Adversarial Active Learning. In *IEEE ICCV*, 5971–5980.

Tang, Y.; and Huang, S. 2019. Self-Paced Active Learning: Query the Right Thing at the Right Time. In *AAAI*, 5117–5124.

Wang, K.; Zhang, D.; Li, Y.; Zhang, R.; and Lin, L. 2017. Cost-Effective Active Learning for Deep Image Classification. *IEEE CSVT* 27(12): 2591–2600.

Yang, Y.; Ma, Z.; Nie, F.; Chang, X.; and Hauptmann, A. G. 2015. Multi-Class Active Learning by Uncertainty Sampling with Diversity Maximization. *IJCV* 113(2): 113–127.

Yoo, D.; and Kweon, I. S. 2019. Learning Loss for Active Learning. In *IEEE CVPR*, 93–102.

Zhang, B.; Li, L.; Yang, S.; Wang, S.; Zha, Z.; and Huang, Q. 2020. State-Relabeling Adversarial Active Learning. In *IEEE CVPR*, 8753–8762.