Exploiting Counter-Examples for Active Learning with Partial labels

Fei Zhang[1,2], Yunjie Ye[3], Lei Feng[4], Zhongwen Rao[3],
Jieming Zhu[3], Marcus Kalander[3], Chen Gong[5], Jianye Hao[3],
Bo Han[1*]

[1]Shanghai Jiao Tong University, Shanghai, 200240, China.
[2*]Hong Kong Baptist University, Hong Kong, 999077, China.
[3]Huawei Noah's Ark Lab, Shenzhen, 518129, China.
[4]Nanyang Technological University, Shenzhen, 639798, Singapore.
[5]Nanjing University of Science and Technology, Nanjing, 210094, China.

*Corresponding author(s). E-mail(s): bhanml@comp.hkbu.edu.hk;
Contributing authors: ferenas@sjtu.edu.cn; yejunjie4@huawei.com;
lfengqaq@gmail.com; raozhongwen@huawei.com; jiemingzhu@ieee.org;
marcus.kalander@huawei.com; chen.gong@njust.edu.cn;
jianye.hao@tju.edu.cn;

Abstract

This paper studies a new problem, *active learning with partial labels* (ALPL). In this setting, an oracle annotates the query samples with partial labels, relaxing the oracle from the demanding accurate labeling process. To address ALPL, we first build an intuitive baseline that can be seamlessly incorporated into existing AL frameworks. Though effective, this baseline is still susceptible to the *overfitting*, and falls short of the representative partial-label-based samples during the query process. Drawing inspiration from human inference in cognitive science, where accurate inferences can be explicitly derived from *counter-examples* (CEs), our objective is to leverage this human-like learning pattern to tackle the *overfitting* while enhancing the process of selecting representative samples in ALPL. Specifically, we construct CEs by reversing the partial labels for each instance, and then we propose a simple but effective WorseNet to directly learn from this complementary pattern. By leveraging the distribution gap between WorseNet and the predictor, this adversarial evaluation manner could enhance both the performance of the predictor itself and the sample selection process, allowing the predictor to capture more accurate patterns in the data. Experimental results

1

on five real-world datasets and four benchmark datasets show that our proposed method achieves comprehensive improvements over ten representative AL frameworks, highlighting the superiority of WorseNet. The source code will be available at https://github.com/Ferenas/APLL.

1 Introduction

The community of artificial intelligence has witnessed great progress owing to deep learning, whose success heavily relies on the quality and volume of accurately annotated datasets. To ease the pressure of such costing labeling work, numerous researchers have been investigating *active learning* (AL) [1], which aims to achieve as high-performance gain as possible by labeling as few samples as possible. A popular setting in AL is pool-based AL [1], where a fixed number of samples selected by a selector are sent to an oracle for labeling iteratively until the exhaustion of the sampling budget. Pool-based AL has a wide range of applications, including but not limited to semantic segmentation [2] and object detection [3].

Most existing pool-based AL frameworks [4–9] assume that the oracle is perfect, i.e., the oracle always provides accurate labels for selected samples. However, due to inherent label ambiguity and noise, we cannot expect such a "perfect" oracle to exist in real-world applications [10]. Let us consider a birdsong classification problem [11]. The songs of different bird species are usually recorded simultaneously in one field-collected recording. Thus, it would be difficult for experts to localize each specie to the corresponding spectrogram simply by virtue of this recording. To apply AL in a more practical way, we turn to a new type of imperfect oracle, which would provide the selected samples with a special but prevailing form of the weak label, i.e., partial label. A partial label of an instance, essentially a set of candidate labels that includes the true label, is intuitively adaptable to various real-world tasks, including image retrieval [12] and face recognition [13]. With the full potential of partial labels seen in these real-world scenarios, *partial-label learning* (PLL), has naturally emerged and boomed in the community [14–17]. Motivated by the industrial and academic value of PLL, we propose a new setting for AL, i.e., *active learning with partial labels* (ALPL). Formally, ALPL is built on a pool-based AL learning problem but with only one imperfect oracle that assigns partial labels to samples. Figure 1 illustrates the pipelines of AL and ALPL. Compared with AL, the oracle in ALPL shall provide noise-tolerant partial labels instead of the exact true label when annotating confusing objects, highly improving the labeling efficiency while easing the annotation pressure of the oracle.

To address ALPL, we first focus on building a group of promising baselines by adopting the RC loss [18], as one of the state-of-the-art milestones in PLL [16, 17, 19, 20], to train the predictor with the given partial labels from the oracle. By doing so, we are able to establish a robust baseline for ALPL that can be seamlessly integrated into various pool-based AL frameworks. Though encouraging and effective, ALPL with RC
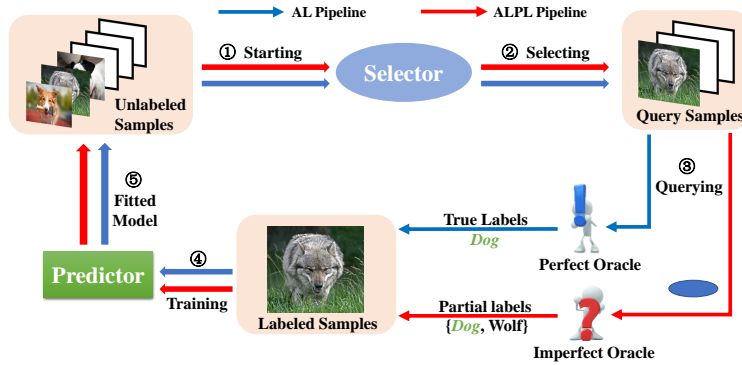
Fig. 1 Comparison of pool-based AL (blue arrow) and our proposed ALPL framework (red arrow). The core difference between these two settings is the label form provided by the oracle.

loss, similar to all AL frameworks, confronts the inevitable *overfitting* challenge [21–23] during the training process with simply few annotated samples provided. Besides, this simple baseline also falls short of the selection of the representative samples with partial labels during the query process.

To move toward better prediction, we turn to an interesting concept from cognitive science named *counter-examples* (CEs). According to the *mental models* in cognitive science [24–26], humans are able to assess the deductive validity of inference with the help of CEs, leading to drawing an accurate conclusion. Inspired by such an adversarial working mechanism, we aim to excavate useful knowledge from CEs to address ALPL by guiding the predictor to deduce in an explicit way. Firstly, we construct CEs for the predictor by directly reversing their partial labels to the inverse version. Building upon the proposed CEs, we propose a simple but effective WorseNet to learn in a way complementary to the predictor. To this end, we propose Worse loss, which contains the *inverse RC* (IRC) loss and the *Kullback-Leibler divergence* (KLD) regularization, to guide WorseNet to learn from the inverse partial labels from CEs. Figure 2 illustrates the overall framework. Compared with the predictor, WorseNet would possess lower confidence toward the labels inside the partial label.

Based on the complementary learning pattern between WorseNet and the predictor, we propose to take advantage of the predicted probability gap between these two networks to separately improve the evaluating and selecting process (shown in Figure 2). To improve the predicting accuracy, we treat the class with the maximum distribution gap, rather than the maximum predictor score, as the predicted true label during the evaluation. On the other hand, we propose to enhance the sample selector by focusing solely on labels with positive probability gaps, as these labels predominantly cover the true label. This narrows down the range for calculating the uncertainty score, thereby refining the selection process and reducing uncertainty. Consequently, we propose three new selectors in ALPL by adopting this selecting strategy. Experimental results on benchmark-simulated and real-world datasets validate the effectiveness and superiority of our proposed WorseNet in improving both the selector and the predictor in ALPL. Our main contributions are summarized here:

- We for the first time propose a practical setting, i.e., *active learning with partial labels* (ALPL), to economically facilitate the annotation process for the experts. In this way, we provide a solid baseline on top of any AL approach to address ALPL.
- We turn to exploring and exploiting the learning pattern from *counter-examples* (CEs), and propose a simple but effective WorseNet to explicitly improve the predictor and the selector in ALPL in a complementary manner.
- Experimental results on four benchmark datasets and five real-world datasets show that our proposed WorseNet achieves promising performance elation over compared baseline methods, achieving state-of-the-art performance in ALPL.

2 Related Work

2.1 Pool-based Active Learning

According to the different query types between the oracle and the predictor, *active learning* (AL) normally can be divided into membership query synthesis, stream-based query, and pool-based query [1]. Pool-based AL, where the selector decides on the annotated samples from a large pool of unlabeled datasets, has drastically appealed to many scholars from academia and industry because of its huge potential value in practical application. With the development of deep learning, pool-based AL has simultaneously experienced the stage from model-driven to data-driven.

For the prevailing model-driven category, the selector heavily relies on handcrafted features or metrics to query the data. Uncertainty sampling, as the most used metric for the selector, aims to pick out the samples with low confidence from the predictor. Often, such uncertainty could be modeled in three following ways: the posterior probability of a predicted class [27], the margin between posterior probabilities of a predicted class and the secondly predicted class [28], or the entropy [5]. Furthermore, all these uncertainty metrics could be improved, though time-consuming as it is, by using Monte Carlo Dropout and multiple forward passes based on Bayesian inference [7, 29]. Some methods also modeled the impacts of the selected sample on the current model through Fisher information [30], mutual information [7, 29], or expected gradient length [31]. Specifically, [31] proposed to select the samples that were disparate and high magnitude in a hallucinated gradient space constructed by using the model parameters of the predictor. Another important metric for the selector is diversity sampling, which aims to select representative and diverse samples for the predictor to better learn from the datasets. To this end, some methods using discrete optimization [32, 33] focused on sample subset selection while [34] aimed at mining out the center points of subsets by clustering. Besides, such informative samples could also be highlighted by measuring the expected output changes [35], or the distribution distance between the unlabeled pool and the selected samples [36].

The methods in the data-driven category describe that the selector, often equipped with deep models, is trained to automatically learn features or metrics. To learn the auto-feature or auto-metric, some methods adopted a generative model-based selector, such as VAE or GAN, to learn to distinguish unlabeled samples from labeled ones [8, 37]. Moreover, some methods turned to adopting or designing data augmentation to help the selector better learn the input space [9]. [6] introduced an auxiliary deep

network, predicting the "loss" of the unlabeled samples, to select the samples with large "loss" to help the query process.

2.2 Active Learning with imperfect oracle

Most works in AL assumed that the oracle would always yield the accurate label, overlooking that the oracle could practically not be infallible in some real-world applications. Therefore, a few researchers have investigated AL with an imperfect oracle, where the oracle could provide a wrong (noise) label to the selected sample [38–41]. Early works [38] assumed that there were two oracles in the system with one always returning the correct label, while the other returned an incorrect label with a fixed probability. [39] modeled a human-like oracle that would provide noisy labels for the samples with low confidence from the predictor. [40] studied a case where the oracle could choose to return incorrect labels or abstain from labeling. Some works [41] focused on active learning with multiple noisy oracles and formed the query process as a constrained optimization problem. In this paper, we work towards a new setting for active learning with simply one imperfect oracle involved in the query process, who would annotate the selected samples with partial labels.

2.3 Partial-Label Learning

In this part, we concisely give an introduction to the two mainstream strategies for *partial-label learning* (PLL), i.e., the *averaged-based strategy* (ABS) and the *identification-based strategy* (IBS). This method in this paper belongs to the ABS.

ABS treats all candidate labels equally and then averages the model outputs of all candidate labels for evaluation. Some non-parametric methods [42, 43] focused on predicting the label by using the outputs of its neighbors. Moreover, some approaches [44, 45] concentrated on leveraging the labels outside the candidate set to discriminate the potential true label. Some recent works [18–20] focused on the data generation process and proposed a classifier-consistent method based on a transition matrix. [20] proposed a family of loss functions, introducing a leverage parameter to consider the trade-off between losses on partial labels and non-partial labels.

IBS focuses on identifying the most possible true label from the candidate label set to eliminate label ambiguity. Early works treated the potential truth label as a latent variable, optimizing the objective function by the maximum likelihood criterion [46] or the maximum margin criterion [47]. Later, many researchers engaged in leveraging the representation information of the feature space to generate the score for each candidate label [15–17]. [16] turned to a contrastive learning framework to eliminate the label disambiguation and reinforce the feature representation learning. [17] proposed to use the class activation map, discriminating the learning pattern of the classifier, to distinguish the potential true label from the candidate label set.

3 Preliminaries

3.1 Symbols and Notations on Pool-based AL

Pool-based AL depicts a learning process where the performance gain of the system is achieved through active interaction between the human and the target predictor. Formally, we are given a bunch of training samples $\mathbb{X} = \{\boldsymbol{x}_i\}_{i=1}^n \in \mathbb{R}^d$ with a total number of n, which is initially split into a small set of labeled samples $\mathbb{L} = \{\boldsymbol{x}_i\}_{i=1}^l \in \mathbb{R}^d$ and a large pool of unlabeled samples $\mathbb{U} = \{\boldsymbol{x}_i\}_{i=1}^u \in \mathbb{R}^d$. Note that here d denotes the input dimension, and $\mathbb{U} \cup \mathbb{L} = \mathbb{X}, \mathbb{U} \cap \mathbb{L} = \varnothing$. Let $\mathbb{Y} = \{1, 2, ..., k\} \in \mathbb{R}$ denote the label space with k classes, and $y_i \in \mathbb{Y}$ denote the ground truth for each $\boldsymbol{x}_i$. A classifier (predictor) $f : \mathbb{R}^d \to \mathbb{R}^k$ is then trained by using the original labeled samples $\mathbb{L}$. Afterwards, a specifically-designed selector $\Psi(\mathbb{L}, \mathbb{U}, f)$ evaluates the samples in $\mathbb{U}$ and selects $\triangle\mathbb{U} = \{\boldsymbol{x}_i\}_{i=1}^b \in \mathbb{U}$ samples to be labeled by an oracle (human expert). Then samples in $\triangle\mathbb{U}$ with *oracle-annotated true labels* are added to $\mathbb{L}$, leading to a group of new labeled samples ($\mathbb{L} = \mathbb{L} \cup \triangle\mathbb{U}$), which are further reused to train the classifier f. This cycle of predictor-oracle-based interaction is repeated continuously until a well-performed metric is achieved or the sampling budget is exhausted. The sampling budget aims to restrict the total number of labeled samples for training the classifier, so the overall size of the sampling budget is denoted as B such that $B << u$.

A well-suited selecting metric Ψ could help elate the performance of the model by using as few labeled examples as possible, achieving a win-win situation for the human oracle and the predictor. *Uncertainty* is one of the most prevailing metrics in active learning, arguing that the oracle-annotated samples are able to confound the model most. To mine out those "uncertain samples", the selector firstly calculates the uncertainty score for each sample in $\mathbb{U}$. Typically there are three simple ways to obtain the uncertainty scores by using the model outputs, which are *minimum confidence uncertainty* (MCU), *minimum margin uncertainty* (MMU) and *entropy uncertainty* (EU). These three metrics can be sequentially expressed as follows [1]:

$$\boldsymbol{x}_{\text{MCU}}^* = \arg\max_{\boldsymbol{x}_i \in \mathbb{U}}\{1 - \arg\max_{y_i \in \mathbb{Y}} P(y_i|\boldsymbol{x}_i)\}, \tag{1}$$

$$\boldsymbol{x}_{\text{MMU}}^* = \arg\min_{\boldsymbol{x}_i \in \mathbb{U}}\{\max_{y_i \in \mathbb{Y}}^1 P(y_i|\boldsymbol{x}_i) - \max_{y_i \in \mathbb{Y}}^2 P(y_i|\boldsymbol{x}_i)\}, \tag{2}$$

$$\boldsymbol{x}_{\text{EU}}^* = \arg\max_{\boldsymbol{x}_i \in \mathbb{U}}\{\sum\nolimits_{y_i \in \mathbb{Y}} P(y_i|\boldsymbol{x}_i) \log(P(y_i|\boldsymbol{x}_i))\}, \tag{3}$$

where $P(y_i|\boldsymbol{x}_i)$ refers to class-conditional probability and $\boldsymbol{x}^*$ denotes the selected uncertain samples. Consequently, uncertainty samples handed over to the oracle could be picked by ranking the uncertainty score of each sample in $\mathbb{U}$ in descending order, resulting in a new labeled dataset to retrain the classifier.

---

[1] In Eq. (2), $\max^1$ ($\max^2$) means the (second) maximum item.

3.2 Symbols and Notations on PLL

Formally, let us denote $\mathbb{C} = \{2^{\mathbb{Y}} \backslash \varnothing \backslash \mathbb{Y}\}$ as the candidate label space where $2^{\mathbb{Y}}$ is the power set of $\mathbb{Y}$, and $|\mathbb{C}| = 2^k - 2$ means that the candidate label set is neither the empty set nor the whole label set. For each training instance $\boldsymbol{x}_i$, let $S_i \in \mathbb{C}$ be the partial labels. We denote $P(\boldsymbol{x}, y)$ and $P(\boldsymbol{x}, S)$ as the probability densities of fully labeled examples and partially labeled examples. Building upon the critical assumption of PLL that the candidate label set of each instance must include the correct label, we have $y_i \in S_i$. PLL targets at learning a predictor f with training examples sampled from $P(\boldsymbol{x}, S)$ to make correct predictions for test examples. Practically, there are two common ways to generate the partial label sets: *(I) uniformly sampling strategy* (USS). Uniformly sampling the partial label for each training instance from all the possible candidate label sets [17, 18]. *(II) Flip Probability Strategy* (FPS). By setting a flip probability q to any false label, the false label could be selected as a candidate label with a probability q [16, 19, 20, 48, 49]. In this paper, we adopt both of them to generate partial labels. Refer to the **Appendix** file for more details.

4 Active Learning with partial labels

In this section, we introduce in detail a new but practical setting based on AL, namely *active learning with partial labels* (ALPL). Different from the previous AL settings, which may be impractical and demanding for the oracle, requiring the oracle to provide the true labels [8, 29, 50] to the selected samples, ALPL regulates that the oracle is asked to label the samples with partial labels that are widely used in real-world scenarios [12, 13]. Compared with AL, ALPL eases the annotation pressure for the oracle when facing confusing samples. Therefore, we believe that ALPL is full of research significance, and a formal definition of ALPL is given as

Definition of ALPL. *Active learning with partial labels (ALPL) trains a predictor with initial training samples annotated with partial labels, uses its selector to select the samples from the unlabeled samples, sends them to an oracle who only provides partial labels, adds them into the labeled training samples, and then retrains the predictor.*

Figure 1 illustrates the pipelines of AL and our proposed ALPL. Note that the key difference between ALPL and AL is the label supervision, so it is intuitive to address ALPL by simply adopting a PLL-based loss function to train the predictor, relieving the negative effects caused by the false positive labels in the candidate label sets. In this case, we use RC loss [18, 19], as one of the most prevailing state-of-the-art loss functions [16, 17, 20], to address ALPL in a simple but effective manner. The empirical risk function $\hat{\mathcal{R}}_{\mathrm{rc}}$ is defined as

$$\hat{\mathcal{R}}_{\mathrm{rc}} = \sum\nolimits_{i=0}^{l} \sum\nolimits_{j \in S_i} \frac{P(y_i = j | \boldsymbol{x}_i)}{\sum_{z \in S_i} P(y_i = z | \boldsymbol{x}_i)} \mathcal{L}(f(\boldsymbol{x}_i), j). \tag{4}$$

Here $\mathcal{L}(f(\boldsymbol{x}), s), s \in S$ refers to the cross entropy loss. As shown in Eq. (4), RC loss is essentially a form of weighted cross entropy among the labels in the candidate set, which is theoretically proved to reach risk consistency in PLL, i.e., achieving comparable performance when compared to the fully supervised methods. Therefore, here we train the predictor f with RC loss to serve as the baseline of ALPL. In

this way, we could seamlessly apply any AL-based frameworks to address ALPL (ten approaches implemented in our paper, see Section 6 for more details).

5 WorseNet: learning from Counter Examples

In this section, we introduce our proposed method to address ALPL in detail. Figure 2 illustrates the overall framework of our proposed WorseNet. Section 5.1 introduces the training procedure of our WorseNet. Section 5.2 and Section 5.3 introduce how WorseNet could address ALPL in both prediction and selection processes.

5.1 Constructing Counter-Examples

Though effective, it is observed two potential issues for the baseline method in ALPL. The first goes to the *overfitting* [21–23], which is a common challenge in both AL and ALPL due to the utilization of a relatively small set of annotated samples. Meanwhile, the sample selection process, as the fundamental part of ALPL, aims to select representative samples that are successively annotated with partial labels, and such distinction sets ALPL apart from conventional AL.

To address these two problems, we turn to an interesting concept in human reasoning. When humans perceive and learn the world, vision yields a mental model to help understand the things described in the scene, and builds a prior knowledge base to proceed further reasoning. Specifically, when evaluating the deductive validity of an *inference*, humans search for *counter-examples* (CEs) to help disapprove the conjecture [24–26]. For instance, the fact that "John Smith is not a lazy student" is one CE to the *inference* "all students are lazy". Therefore, we can tell that "all students are lazy" is a false conclusion because of "John Smith". Intuitively, CEs occupy on an important position in human reasoning. Inspired by the effectiveness of CEs in the mental model, we are driven to draw an interesting question: *can the predictor also benefit from CEs*? Thus, here we aim to explore and exploit CEs from the data, explicitly assisting the predictor to improve its performance in ALPL.

The first question goes to how to construct CEs for the predictor. It is emphasized that CEs rigorously deplore the *inference*. Let us consider that we classify an image of a dog with a one-hot label, and assume that the *inference* here is "The image has a dog". In this way, this conjecture is rejected once this image is annotated "0" at the "Dog" index. Here the simple inverse on the true label intuitively leads to a CE, which violates the original accurate *inference*, leading to a complementary conclusion. Motivated by this, we propose to build up CEs for the predictor by adopting label inversion to the selected samples. Formally, we are given a set of data samples $\mathbb{W} = \{\boldsymbol{x}_i\}_{i=1}^l \in \mathbb{R}^d$ such that $\mathbb{W} = \mathbb{L}$, and the assigned label of each sample in $\mathbb{W}$ is defined as follows:

$$\overline{S}_i = \mathbb{Y} - S_i, \tag{5}$$

where $\overline{S}_i$ denotes the candidate label set for the instance in $\mathbb{W}$. Intuitively, $\overline{S}_i$ is complementary to S_i, i.e., $\overline{S}_i = \complement_{\mathbb{Y}} S_i$, meaning that there is no true label within $\mathbb{W}$. For convenience, we name the candidate label set $\overline{S}$ as the *inverse partial label* (IPL). Note that IPL is different from the *complementary label* [51]. The former provides a wrong indicator to the samples while the latter aims to train a true-label predictor by specifying the classes that the example does not belong to.
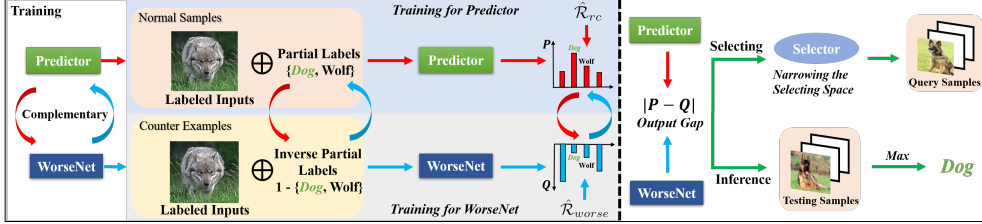
8

Fig. 2 The overall framework of our proposed method to address ALPL. A strong baseline for ALPL is achieved by directly using RC loss to train the predictor (red arrows). To further improve the performance, we propose WorseNet (blue arrows) to extract the useful knowledge from the constructed *counter examples*, individually learning in a complementary way to the predictor. With the help of the distribution gap between the predictor and WorseNet, the selecting and inference process (green arrows) in ALPL could be improved in an explicit way.

There are two benefits to forming IPL by following Eq. (5) in ALPL. Firstly, it is convenient and efficient to construct CEs with a label-based operation to the selected label samples $\mathbb{L}$. Secondly, IPL considers that all false labels outsides $\overline{S}_i$ shall become the inverse knowledge to the instance $\boldsymbol{x}_i$, enriching the label variety of CEs.

5.2 Predicting better with WorseNet

In this section, we introduce how to assist the predictor with the help of the proposed CEs in ALPL. Firstly, an extra classifier apart from the predictor is needed to learn from CEs obtained from $\mathbb{W}$ annotated with IPL. Formally, let us name such a classifier as the WorseNet and denote it as $w : \mathbb{R}^d \to \mathbb{R}^k$. Note that w shares the same input and output space as the predictor f since w is trained with training samples from $Q(\boldsymbol{x}, \overline{S})$, which denotes the probability densities of samples with IPL. To help w extract the inverse knowledge from $Q(\boldsymbol{x}, \overline{S})$, we formulate this learning process, treating the IPL as the normal partial labels, to a similar PLL problem, where we propose *inverse RC* (IRC) loss to address it as follows:

$$\hat{\mathcal{R}}_{\text{irc}} = \sum\nolimits_{i=1}^{l} \sum\nolimits_{j \in \overline{S}_i} \frac{Q(y_i = j|\boldsymbol{x}_i)}{\sum_{z \in \overline{S}_i} Q(y_i = z|\boldsymbol{x}_i)} \mathcal{L}(w(\boldsymbol{x}_i), j), \tag{6}$$

where $\hat{\mathcal{R}}_{\text{irc}}(\mathcal{L}, w)$ denotes the empirical risk function for w, and $Q(y|\boldsymbol{x})$ denotes the class-conditional probability modeled by w. Clearly, IRC loss focuses on the labels outside the candidate label set in a way complementary to RC loss.

Supported by the IRC loss, WorseNet is able to latch on to a pattern that is complementary to the predictor. To improve the predictor with WorseNet, we leverage the output distribution gap between w and f to predict the true label during the inference. Since the original true label only lies in the candidate label set S, we should intuitively aim at enlarging the gap of the output distribution on S between f and w. To this end, we further add a *Kullback-Leibler divergence* (KLD) regularization item for w, regulating its learning process toward the gainful direction to the predictor. Specifically, the KLD item is expressed as

$$\text{KLD} = \sum\nolimits_{i=1}^{l} \sum\nolimits_{j \in \overline{S}_i} P(y_i = j|\boldsymbol{x}_i) \log \frac{P(y_i = j|\boldsymbol{x}_i)}{Q(y_i = j|\boldsymbol{x}_i)}. \tag{7}$$

Note that here we stop the gradient backpropagation of P when training w. As shown in Eq. (7), we calculate the KLD between the predictor and WorseNet by

merely using their outputs inside $\overline{S}$, which could be minimized to implicitly enlarge the output distribution of the candidate set between f and w. In all, the learning loss function for WorseNet, denoted as Worse loss, could be expressed as follows:

$$\hat{\mathcal{R}}_{\text{worse}} = \hat{\mathcal{R}}_{\text{irc}}(\mathcal{L}, w) + \alpha\text{KLD}. \tag{8}$$

where α is a regularized parameter and we empirically set $\alpha = 1$. After training by Eq. (8), the predictor during the inference could predict the potential true label by

$$y_i^* = \arg\max_{y_i \in \mathbb{Y}}\{P(y_i|\boldsymbol{x}_i) + (1 - Q(y_i|\boldsymbol{x}_i))\}, \tag{9}$$

where y_i^* denotes the predicted true label of $\boldsymbol{x}_i$. Note that here we use $1 - Q$ to help the predictor recognize the true label. As WorseNet is trained independently of the predictor, the proposed WorseNet is able to benefit the predictor on top of any selector in ALPL. To better illustrate this, we provide the following theorem.

Theorem 1. *Assume that the posterior probability of WorseNet satisfies $Q(y = j|\boldsymbol{x}_i) + P(y = j|\boldsymbol{x}_i) = 1$ for any label $j \in \mathbb{Y}$ of sample $\boldsymbol{x}_i$, and the loss function $\mathcal{L}$ is the standard cross entropy. Then the Worse loss $\hat{\mathcal{R}}_{\text{worse}}$ holds*

$$\hat{\mathcal{R}}_{\text{worse}} \propto \sum\nolimits_{i=1}^{l} \sum\nolimits_{j \in \overline{S}_i} -n(Q_{ij})\log(Q_{ij}), \tag{10}$$

where Q_{ij} represents $Q(y = j|\boldsymbol{x}_i)$ for simplicity and $n(Q_{ij}) > 0, \forall Q_{ij} \in [0, 1]$. The proof and analysis of Theorem 1 is in the **Appendix** file. Theorem 1 shows that the WorseNet is learned to approximate the false labels in $\overline{S}$ in an entropy-based manner. As $\hat{\mathcal{R}}_{\text{worse}}$ decreases and $Q_{ij} \to 1$, the predictor is correspondingly pushed away from $\overline{S}$ ($P_{ij} \to 0$). In all, the Worse loss could serve as an auxiliary module to the predictor by considering the extra supervision on the elements outside the partial labels. For convenience, we denote this improvement of WorseNet to the predictor during the evaluation as WorseNet-Predictor (WP), and its pseudo-code is given in Algorithm 1.

5.3 Selecting better with WorseNet

In this section, we illustrate that the proposed WorseNet can also promote the sampling metric of some uncertainty-based selectors. As shown in Section 3.1, a selector $\Psi(\mathbb{L}, \mathbb{U}, f)$ needs to calculate the uncertainty score of $\boldsymbol{x}_i$ in the entire class space since it has no prior knowledge about the class of this sample. We argue that such a strategy could be further improved if the class space for obtaining the uncertainty could be narrowed down, bringing well inductive bias to the selector. As shown in Eq. (9), we test our proposed framework during the inference by measuring the gap of the output distribution between f and w. In particular, we assume that the true label is the class with the maximum probability distance between f and w. As f focuses on the candidate label set S while w learns from CEs, the former one shall have a higher response to the labels in S than the latter one. Hence, it reveals that the potential true label must satisfy $P > Q$ since the true label absolutely lies on S. Based on this, we construct a pseudo partial label candidate set S' for each unlabeled sample in $\mathbb{U}$ as follows:

$$S_i' = \{z|P(y_i = z|\boldsymbol{x}_i) - Q(y_i = z|\boldsymbol{x}_i) \geq 0, z \in \mathbb{Y}\}. \tag{11}$$

Building upon S', a selector could narrow the class range of acquiring the uncertainty score in $\mathbb{U}$. To this end, we propose three sampling strategies based on MCU (Eq. (1)),

10

Algorithm 1 ALPL with WorseNet-Predictor (WP)

Input: Predictor f, WorseNet w, iterations T, unlabeled examples $\mathbb{X}$, an oracle $\mathcal{O}$, a selector $\Psi(\mathbb{L}, \mathbb{U}, f)$, initial sampling size b_0, query size b, sampling budget B.

1: **Label** b_0 samples drawn uniformly at random from $\mathbb{X}$ with partial labels S, forming the initial labeled samples $\mathbb{L}$, and all the remaining samples in $\mathbb{X}$ compose the unlabeled samples $\mathbb{U}$;

2: **Train** an initial f on $\mathbb{L}$ by $\hat{\mathcal{R}}_{rc}$ in Eq. (4);

3: **Label** the samples from $\mathbb{L}$ with IPL $\overline{S}$ by Eq. (5), forming the initial CEs $\mathbb{W}$;

4: **Train** an initial w on $\mathbb{W}$ by $\hat{\mathcal{R}}_{worse}$ in Eq. (8);

5: **while** $t < T$ and $B > 0$ **do**

6:     **Select** b samples from $\mathbb{U}$ by $\Psi(\mathbb{L}, \mathbb{U}, f)$, building the query samples $\triangle \mathbb{U}$;

7:     **Label** $\triangle \mathbb{U}$ with S by $\mathcal{O}$, forming the labeled query samples $\triangle \mathbb{L}$;

8:     **Label** $\triangle \mathbb{U}$ with $\overline{S}$ by Eq. (5), forming the IPL-annoatated query samples $\triangle \mathbb{W}$;

9:     $\mathbb{U} \Leftarrow \mathbb{U} - \triangle \mathbb{U}$; $\mathbb{L} \Leftarrow \mathbb{L} \cup \triangle \mathbb{L}$; $\mathbb{W} \Leftarrow \mathbb{W} \cup \triangle \mathbb{W}$;

10:     **Train** f on $\mathbb{L}$ labeled with S by $\hat{\mathcal{R}}_{rc}$ in Eq. (4);

11:     **Train** w on $\mathbb{W}$ labeled with $\overline{S}$ by $\hat{\mathcal{R}}_{worse}$ in Eq. (8);

12:     $t \Leftarrow t + 1$; $B \Leftarrow B - b$;

13: **end while**

14: **(Inference):** Predict the true label y^* in Eq. (9).

Output: f, w.

MMU (Eq. (2)), and EU (Eq. (3)) by directly substituting $\mathbb{Y}$ with S'. For convenience, we denote the improvement of WorseNet on the selector as WorseNet-Selector (WS), and denote these three methods as WS-MCU, WS-MMU, and WS-EU.

6 Experiments

In this section, we evaluate our proposed WP, WS-MCU, WS-MMU, and WS-EU against several algorithms from the literature, and extensive experiments are implemented to verify the correctness and effectiveness of our proposed modules. More details could be found in the **Appendix** file.

6.1 Benchmark datasets comparisons

Datasets and backbones. Our proposed WorseNet-based modules are evaluated on four popular benchmark datasets, which are MNIST [52], Fashion-MNIST [53], SVHN [54] and CIFAR-10 [55]. Note that it is necessary for the oracle to manually generate the candidate label sets for these datasets, which are supposed to be used for single-classification problems. Recall that we introduce two different candidate label generation approaches, i.e., USS and FPS. For FPS, we set $q \in \{0.3, 0.5\}$ to represent different ambiguity degrees. For MNIST and Fashion-MNIST, we adopt a 3-layer MLP and a simple CNN-based network denoted as C-Net (similar to the network used in [7, 29]) as the backbones for the predictor. For SVHN and CIFAR-10, we follow most works [6, 8, 31] and choose ResNet18 [56] and VGG11 [57] as the base models. Note that WorseNet w follows the identical architecture to the predictor f.

Compared methods and training settings. We compare our proposed modules with ten approaches which contains seven model-driven methods: 1) Random Sampling

(RS), 2) MCU, 3) MMU, 4) EU, 5) Coreset [36], 6) BALD [7], 7) BADGE [31], and three data-driven methods: 8) LL4AL [6], 9) VAAL [37] and 10) TA-VAAL [8]. For the seven model-driven methods, we adopt the Adam optimizer [58] with a learning rate of 0.001 to train f. We take a mini-batch size of 256 images and train all seven methods for 200 epochs. For three data-driven methods, we strictly follow the reported training hyper-parameters in their papers [6, 8, 37]. Besides, we simply adopt ResNet18 as the backbone for f and w in these three data-driven methods. For the ALPL setting, we construct an initial labeled set $\mathbb{L}$ with the size $b_0 = 20$, and acquire $b = 100$ instances ($b = 1000$ for SVHN and CIFAR-10) from $\mathbb{U}$ in each query round, following prior works [7, 29, 59]. We repeat the query process 10 times such that the overall budget size $B = 1000$ ($B = 10000$ for SVHN and CIFAR-10). Note that we directly adopt RC loss on these ten methods to build the baselines (see Section 4 for more details). To guarantee comparison fairness, we repeatedly conduct all experiments 5 times and report the average test accuracy using the model achieving the maximum performance on a validation set, which is constructed by randomly selecting 100 instances from the training datasets. Here the validation performance of w is measured by Eq. (9). All the implemented methods are trained on 2 RTX3090 GPUs each with 24 GB memory.

Experiment results. As shown in Table 1, following the default settings, our proposed WorseNet shows its effectiveness and superiority in addressing ALPL on these four benchmark datasets. Firstly, WP can bring a constant gain to the classifier regardless of the backbone and the adopted AL methods. Moreover, the improvement by WP shall be witnessed in both USS and FPS cases, validating that our WP does not rely on any data generation assumption. Our approach could also deliver promising performance with full access to the datasets, which means that WP is also an effective way to address PLL. Particularly, we would like to highlight a counter-intuitive phenomenon that RS may perform better than some methods in some cases. RS (70.73%) performs far better than EU (64.58%) and Coreset (53.17%) in Fashion-MNIST. This counter-intuitive could also be seen in [6, 31, 37, 59]. This phenomenon can be attributed to the instability caused by a relatively small number of labeled samples.

For three WS-based selectors, i.e., WS-MMU, WS-MCU, and WS-EU, they are found to better elate the performance of the classifier in ALPL when compared to the original version. Additionally, these three improved uncertainty-based approaches show competitive performance compared with the other ten AL methods, and such performance could be further improved by reusing WP to reach state-of-the-art performance in ALPL. As shown in Figure 3, we select 6 classes and visualize the selected samples of EU and WS-EU. Compared to EU, our WS module could enforce the selector to select more representative and diverse samples. Specifically, our proposed selectors are able to select more samples (marked by the red circle) that nearby the class boundary. Besides, more samples near the center of the class cluster are also selected to ensure the accuracy (marked by the blue circle), illustrating that our WS could help ALPL to select more representative samples with partial labels. Overall, the experimental results on four benchmark datasets reasonably verify the generalization and effectiveness in addressing ALPL.

Table 1 Test performance of the methods on benchmark datasets using label generation by FPS ($q = 0.5$). The best results are marked in **bold**. -/+ WP denotes whether the predictor is helped by WorseNet. The underline points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 1%. The backbones for MNIST and Fashion-MNIST are C-Net, and for SVHN and CIFAR-10 are ResNet18. *Fully-supervised PLL* denotes the training performance with full access to the partially labeled datasets. Here the standard deviation is ignored.

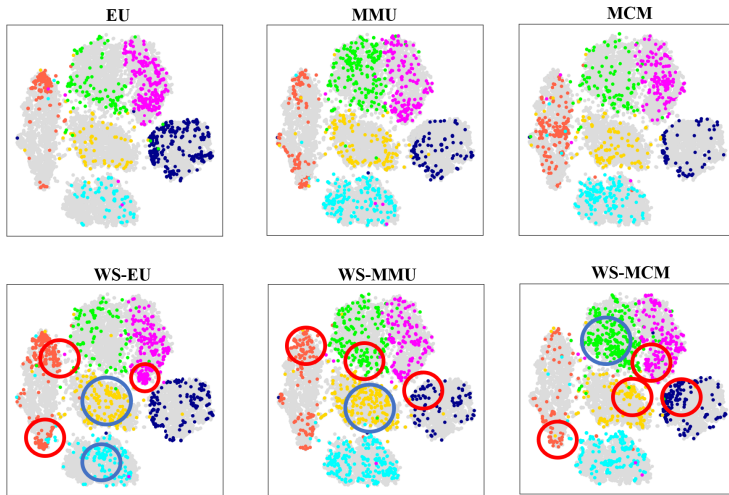| Methods (-/+ WP)	MNIST	Fashion-MNIST	SVHN	CIFAR-10
RS	89.26 / 90.11	70.73 / 71.17	71.63 / 72.23	54.57 / 55.41
MMU	95.18 / 96.37 ↑	74.22 / 76.44 ↑	75.13 / 76.21 ↑	57.65 / 58.67 ↑
MCU	93.75 / 94.68	64.59 / 65.75 ↑	76.28 / 77.09	58.41 / 59.51 ↑
EU	90.83 / 91.28	64.58 / 65.16	75.17 / 76.08	57.58 / 58.79 ↑
Coreset	86.05 / 87.65 ↑	53.14 / 61.62 ↑	75.32 / 76.10 ↑	59.25 / 60.37 ↑
BALD	94.08 / 95.11 ↑	70.95 / 72.95 ↑	77.15 / 77.82	59.09 / 60.13 ↑
BADGE	96.01 / 96.49	76.75 / 77.10	77.23 / 78.76 ↑	59.04 / 60.30 ↑
LL4AL	81.91 / 82.75	60.91 / 61.62	76.69 / 77.80 ↑	55.81 / 56.97 ↑
VAAL	90.68 / 91.08	75.18 / 75.44	77.81 / 78.05	56.69 / 57.32
TA-VAAL	90.93 / 91.26	75.21 / 75.90	78.07 / 78.40	56.81 / 57.94 ↑
WS-MMU	95.74 / **96.66**	77.08 / **77.75**	77.51 / 78.18	58.63 / 59.36
WS-MCU	94.96 / 95.17	68.36 / 69.77 ↑	78.81 / **79.61**	59.39 / 60.83 ↑
WS-EU	93.90 / 94.80	66.01 / 67.75 ↑	76.09 / 77.12	58.45 / **59.12**
Fully-supervised PLL	97.61 / 98.27	84.49 / 85.87 ↑	92.36 / 93.01	71.89 / 73.58 ↑

Fig. 3 Visualized tSNE results of EU and WS-EU in MNIST with FPS ($q = 0.5$). The red circles mark that more samples near the class boundary are selected, and the blue circle mark that more samples near the center of the class cluster are selected.

6.2 Real-World datasets comparisons

Datasets and backbones. Apart from benchmark datasets whose candidate label set needs to be self-generated, here we evaluate our proposed WorseNet-based modules on five real-world datasets that are widely used in PLL: Lost [12], MSRCv2 [60], BirdSong [61], Soccer Player [13] and Yahoo!News [62]. Note that all five of these real-world datasets are annotated with the given candidate label sets, and most samples, as a realistic scenario, are annotated with similar semantic labels. Thus, we simply use

Table 2 Test performance of compared methods on five real-world datasets. The <u>underline</u> points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 3%. Note that three data-driven methods are not implemented here due to the framework incompatibility.

Methods (-/+ WP)	Lost	MSRCV2	BirdSong	SoccerPlayer	Yahoo!News
RS	51.68 / <u>55.93</u> ↑	40.91 / <u>44.59</u> ↑	57.01 / <u>62.05</u> ↑	48.31 / <u>49.94</u>	52.59 / <u>56.20</u> ↑
MMU	53.58 / <u>56.75</u> ↑	44.32 / <u>46.59</u>	58.60 / <u>62.80</u> ↑	50.17 / <u>51.92</u>	56.85 / <u>59.37</u>
MCU	53.06 / <u>56.25</u>	43.18 / <u>46.02</u> ↑	63.39 / <u>66.43</u>	51.32 / <u>53.06</u>	55.42 / <u>58.55</u> ↑
EU	48.21 / <u>54.46</u> ↑	41.32 / <u>45.14</u> ↑	63.22 / <u>66.60</u> ↑	49.19 / <u>51.06</u>	54.94 / <u>57.98</u> ↑
Coreset	52.32 / <u>56.79</u> ↑	41.92 / <u>44.32</u>	60.15 / <u>66.43</u> ↑	50.03 / <u>50.83</u>	50.98 / <u>52.02</u>
BALD	52.79 / <u>54.57</u>	40.91 / <u>47.73</u> ↑	62.80 / <u>65.20</u> ↑	48.94 / <u>52.38</u> ↑	54.21 / <u>58.24</u> ↑
BADGE	52.00 / <u>53.79</u> ↑	50.57 / <u>**53.98**</u> ↑	64.61 / <u>68.05</u> ↑	50.72 / <u>53.47</u>	57.72 / <u>**60.98**</u> ↑
WS-MMU	54.09 / <u>**57.14**</u> ↑	46.59 / <u>50.00</u> ↑	62.42 / <u>65.57</u> ↑	51.32 / <u>52.58</u>	57.55 / <u>59.98</u>
WS-MCU	53.57 / <u>57.10</u> ↑	44.48 / <u>47.16</u> ↑	64.40 / <u>67.20</u> ↑	52.12 / <u>**53.58**</u>	56.33 / <u>59.07</u> ↑
WS-EU	51.79 / <u>54.46</u>	42.32 / <u>46.32</u> ↑	64.61 / <u>**68.68**</u> ↑	49.80 / <u>51.81</u>	55.81 / <u>57.89</u>

them as the oracle annotation. For these five datasets, we adopt the same 3-layer MLP used in Section 6.1 as the sole backbone since these real-world datasets are not limited to image input (simple vector inputs), which also follows conventions in [16–20, 48, 63]. **Compared methods and training settings.** Due to the simplicity of these five real-world datasets, we adopt a simple MLP as the backbone for both the predictor and WorseNet, so here we compare our methods with seven model-driven methods, 1) - 7), the architecture of which does not necessarily build upon the deep models. Based on the different data quantities, we specifically design different settings for these five datasets. Specifically, we set the size of the initial labeled set $\mathbb{L}$ to 5, and repeat the query process 5 times. We repeatedly conduct all experiments 10 times, and record the average testing accuracy by using the model achieving maximum performance on a validation set built by randomly selecting 10 instances from the training datasets. Other settings are similar to Section 6.1.

Experiment results. The experimental results in Table 2 validate that our proposed WorseNet is also effective in dealing with ALPL in five real-world datasets. Specifically, our WP is capable of delivering promising performance gains to the predictor with any baseline method. Furthermore, the three improved metrics (WS-MMU, WS-MCU, and WS-EU) in the selector also show competitive performance compared to the baselines.

6.3 Ablation studies on WorseNet

Comparison with different DAs. In Table 3, we implement three *data augmentations* (DAs), i.e., Random Erasing, Mixup (The mixing parameter of MixUp is 0.5), and Random Crop on top of RS to evaluate the generalization of WorseNet. Clearly, our proposed WorseNet could improve predictor performance with any of three DAs on two evaluated datasets, which further validates the superiority and effectiveness of our proposed WorseNet. We also notice that Random Erasing and Mixup could make harmful performance degradation to the predictor. This may be due to that these DAs further damage the training samples built on the partial labels, especially for Mixup (the partial labels are mixed together), troubling the learning of the predictor in ALPL. In conclusion, our WorseNet is a promising method to address *overfitting*.

Number of selected samples. As shown in Figure 4, with the increase of queried samples (100 samples in each round), all methods achieve steady performance enhancement throughout the whole training time. Clearly, it is noticed that all baseline

Table 3 Testing performance with different data augmentation. The <u>underline</u> points out the improved accuracy by WP. Other settings are similar with Table 1.

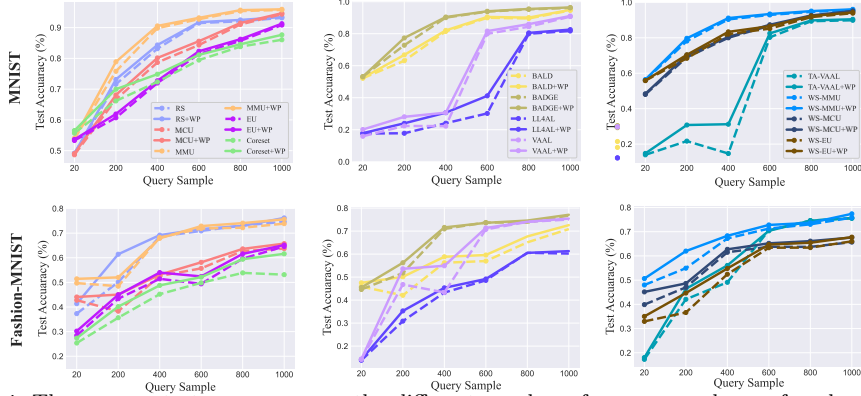| Data Augmentation + RS	Fashion-MNIST	CIFAR-10
Random Erasing	66.15 ± 5.98 / <u>67.12 ± 4.25</u>	49.87 ± 3.22 / <u>51.02 ± 2.58</u>
Mixup	16.15 ± 4.67 / <u>16.56 ± 4.87</u>	10.33 ± 4.74 / <u>12.19 ± 5.23</u>
Random Crop	71.29 ± 1.00 / <u>72.82 ± 0.51</u>	56.33 ± 1.05 / <u>57.62 ± 1.74</u>

Fig. 4 The average test accuracy over the different number of query samples on four benchmark datasets during the training time. Note that here the settings are the same with Table 1.

methods (dashed lines) are comparably strengthened by our proposed WP (solid lines) in each query round. Besides, the three new proposed selectors could also achieve competitive performance. More relevant results can be found in the **Appendix** file.

7 Conclusion

We have proposed and investigated a new and practical setting, *active learning with partial labels* (ALPL), where the oracle is requested to provide partial labels for the selected samples during the query process. To address ALPL, we first adopt RC loss on different prevailing AL frameworks to establish a strong and effective baseline. Motivated by the salutary effects of *counter examples* (CEs) in human reasoning, we turn to such a human-based adversarial learning process to relieve the *overfitting* and improve the partially-labeled sample selection process in ALPL. In this regard, we designed CEs by reversing the original partially-labeled examples. Furthermore, we introduced WorseNet that directly learns such complementary knowledge by using the proposed Worse loss. By capitalizing on the probability gap between the predictor and WorseNet, our proposed WorseNet not only explicitly enhances the evaluation performance of the predictor but also improves the selector's ability to query partially-labeled samples more precisely. Comprehensive experimental results on various datasets demonstrate that our WorseNet yields state-of-the-art performance in ALPL, and validates the superiority of such an adversarial learning pattern. Additionally, PLL could also be well addressed by this method, which warrants further investigation in the future.

Appendix A Generation of PLL

In Section 3, we introduce two different generation ways for the candidate label sets, i.e., USS, uniformly sampling a label set from the full partial label space $\mathbb{C}$ for each instance. FPS, setting a flip probability q for any irrelevant labels which could possibly become an item in the candidate label set with probability q.

A.1 USS

For USS, each partially-labeled example $(\boldsymbol{x}, S)$ is independently drawn from a probability distribution with the following density:

$$\widetilde{P}(\boldsymbol{x}, S) = \sum\nolimits_{i=1}^{k} P(S|y=i)P(\boldsymbol{x}, y=i), P(S|y=i) = \begin{cases} \frac{1}{2^{k-1}-1} & i \in S, \\ 0 & i \notin S. \end{cases} \tag{A1}$$

The generation process assumes that the candidate label set S is independent of the instance $\boldsymbol{x}$. There are a total of $2^k - 1$ possible candidate label sets that contain the specific true label y. Therefore, Eq. (A1) illustrates that the candidate label set for each instance is uniformly sampled.

A.2 FPS

For FPS, we set a flip probability q to any irrelevant label that possibly entries the candidate label set. Here, we introduce the class transition matrix (denoted by T) for partially labeled data, where T_{ij} refers to the probability that the label j is a candidate label given the true label i for each instance. Note that $T_{ii} = 1$ always holds since the true label always belongs to the candidate set. $T_{ij} = q, i \neq j$ holds for other elements.

Appendix B Proof and Analysis of Theorem 1

Since the WorseNet is regulated to learn the inverse partial-label set $\overline{S}$, the WorseNet shall have a high confidence on the false labels in $\overline{S}$, which is complementary to the predictor. Therefore, we assume the following equality:

$$Q(y_i = j|\boldsymbol{x}_i) + P(y_i = j|\boldsymbol{x}_i) = 1, j \in \overline{S}_i, i \in \{1, ..., l\}. \tag{B2}$$

Assume that the loss function $\mathcal{L}$ is implemented with a standard cross entropy loss (which is also the practical achievement in our experiments and [18]). In this way, we have the following equation for the IRC loss:

$$\hat{\mathcal{R}}_{\mathrm{irc}} = \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -\frac{Q(y_i = j|\boldsymbol{x}_i)}{\sum_{j \in \overline{S}_i} Q(y_i = j|\boldsymbol{x}_i)} \log Q(y_i = j|\boldsymbol{x}_i). \tag{B3}$$

Based on Eq. (B2) and (B3), we could express the Worse loss $\hat{\mathcal{R}}_{worse}$ as

$$
\begin{aligned}
\hat{\mathcal{R}}_{\text{worse}} &= \hat{\mathcal{R}}_{\text{irc}}(\mathcal{L}, w) + \alpha \text{KLD} \qquad (\alpha = 1) \\
&= \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -\frac{Q(y_i = j|\boldsymbol{x}_i)}{\sum_{j \in \overline{S}_i} Q(y_i = j|\boldsymbol{x}_i)} \log Q(y_i = j|\boldsymbol{x}_i) + \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} P(y_i = j|\boldsymbol{x}_i) \log \frac{P(y_i = j|\boldsymbol{x}_i)}{Q(y_i = j|\boldsymbol{x}_i)} \\
&= \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -\frac{Q(y_i = j|\boldsymbol{x}_i)}{\sum_{j \in \overline{S}_i} Q(y_i = j|\boldsymbol{x}_i)} \log Q(y_i = j|\boldsymbol{x}_i) + \underbrace{P(y_i = j|\boldsymbol{x}_i) \log P(y_i = j|\boldsymbol{x}_i)}_{\text{constant term when minimizing } \hat{\mathcal{R}}_{\text{worse}}} - P(y_i = j|\boldsymbol{x}_i) \log Q(y_i = j|\boldsymbol{x}_i) \\
&= \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -\frac{Q(y_i = j|\boldsymbol{x}_i)}{\sum_{j \in \overline{S}_i} Q(y_i = j|\boldsymbol{x}_i)} \log Q(y_i = j|\boldsymbol{x}_i) - \underbrace{(1 - Q(y_i = j|\boldsymbol{x}_i))}_{\text{substitution for } P(y_i = j|\boldsymbol{x}_i)) \text{ by Eq. (B2)}} \log Q(y_i = j|\boldsymbol{x}_i) + c \\
&= \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -\frac{Q(y_i = j|\boldsymbol{x}_i)}{\sum_{j \in \overline{S}_i} Q(y_i = j|\boldsymbol{x}_i)} \log Q(y_i = j|\boldsymbol{x}_i) - (1 - Q(y_i = j|\boldsymbol{x}_i)) \log Q(y_i = j|\boldsymbol{x}_i) + c \\
&= \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -\underbrace{\frac{Q(y_i = j|\boldsymbol{x}_i) + (1 - Q(y_i = j|\boldsymbol{x}_i)) \sum_{j \in \overline{S}_i} Q(y_i = j|\boldsymbol{x}_i)}{\sum_{j \in \overline{S}_i} Q(y_i = j|\boldsymbol{x}_i)}}_{n(Q(y_i = j|\boldsymbol{x}_i))} \log Q(y_i = j|\boldsymbol{x}_i) + c \\
&= \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -n(Q(y_i = j|\boldsymbol{x}_i) \log Q(y_i = j|\boldsymbol{x}_i) + c.
\end{aligned}
$$

(B4)

Intuitively, minimizing the Worse loss $\hat{\mathcal{R}}_{\text{worse}}$ is equal to minimize the loss function below:

$$
\min \hat{\mathcal{R}}_{\text{worse}} \Longrightarrow \min \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} -n(Q(y_i = j|\boldsymbol{x}_i)) \log Q(y_i = j|\boldsymbol{x}_i), Q(y_i = j|\boldsymbol{x}_i) \in [0, 1].
$$

(B5)

Here, the proof of Theorem 1 is complete. Figure B1 illustrates the graph of the above function, from which we can easily observe that this loss is a monotone-decreasing function. While optimizing the Worse loss, the probability of WorseNet $Q(y_i = j|\boldsymbol{x}_i)$ gradually approaches "1", indicating high confidence in the false labels in the inverse partial-label set $\overline{S}$. Since learning the WorseNet is complementary to the predictor, the predictor is expected to decrease its confidence in predicting labels in $\overline{S}$ ($P(y_i = j|\boldsymbol{x}_i) \to 0$). In other words, WorseNet acts as an auxiliary regularization to the RC loss, further keeping the predictor away from the false labels in $\overline{S}$. We believe this is also why this learning mechanism can help the predictor alleviate *overfitting* problems, as the predictor is able to generalize to the entire label space (whereas RC loss only considers the labels in the partial label set S).
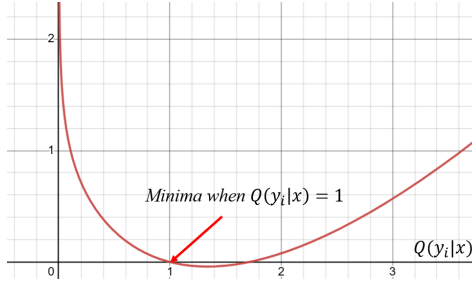
Fig. B1 The loss graph of the simplified Worse loss $\hat{\mathcal{R}}_{\text{worse}}$ in Eq. (B5).

Appendix C   Experiments

C.1   Benchmark Datasets

In Section 5.1, we use four widely-used benchmark datasets, i.e., MNIST [52], Fashion-MNIST [53], SVHN [54], and CIFAR-10 [55]. Table C1 lists the characteristics of these datasets. We respectively describe these datasets as follows.

- MNIST [52]: It is a 10-class dataset of handwritten digits. Each data is a 28×28 grayscale image.
- Fashion-MNIST [53]: It is also a 10-class dataset. Each instance is a fashion item from one of the 10 classes, which are T-shirt/top, trouser, pullover, dress, sandal, coat, shirt, sneaker, bag, and ankle boot. Moreover, each image is a 28×28 grayscale image.
- SVHN [54]: Each instance is a $32 \times 32 \times 3$ colored image in RGB format. It is a 10-class dataset of digits.
- CIFAR-10 [55]: Each instance is a $32 \times 32 \times 3$ colored image in RGB format. It is a ten-class dataset of objects including airplane, bird, automobile, cat, deer, frog, dog, horse, ship, and truck.

Table C1 Characteristics of benchmark datasets

Datasets	#Train	#Test	#Features	#Classes
MNIST [52]	60,000	10,000	784	10
Fashion-MNIST [53]	60,000	10,000	784	10
SVHN [54]	73,257	26,032	3,072	10
CIFAR-10 [55]	50,000	10,000	3,072	10

C.2   Real Datasets

In Section 5.2, we select five real-world datasets including Lost [12], MSRCv2 [60], BirdSong [61], Soccer Player [13], and Yahoo!News [62]. According to the different data quantities of these datasets, we specifically design the unique query setting for each

of them, and the detailed parameters are in Table C3. Here, we give a comprehensive description of them as follows.

- Lost, Soccer Player, and Yahoo!News: They crop faces in images or video frames as instances, and the names appearing on the corresponding captions or subtitles are considered as candidate labels.
- MSRCv2: Each image segment is treated as a sample, and objects appearing in the same image are regarded as candidate labels.
- BirdSong: The singing syllables of birds are regarded as instances, and bird species that are jointly singing during any ten seconds are represented as candidate labels.

Table C2 Characteristics of the real-world datasets.

Datasets	Application Domain	#Examples	#Features	#Classes	Avg #CLs
Lost [12]	Automatic face naming	1,122	108	16	2.23
MSRCv2 [60]	Object classification	1,758	48	23	3.16
BirdSong [61]	Bird song classification	4,998	38	13	2.18
Soccer Player [13]	Automatic face naming	17,472	279	171	2.09
Yahoo! News [62]	Automatic face naming	22,991	163	219	1.91

Table C3 The explicit query size b and budget size B on five real-world datasets in ALPL. The percentage number(%) depicts the proportion of query budget in the total unlabeled data.

Parameters	Query size (b)	Query budget (B)
Lost [12]	40	200 (17.8%)
MSRCV2 [60]	60	300 (17.1%)
BirdSong [61]	200	1000 (20.0%)
SoccerPlayer [13]	600	3000 (17.2%)
Yahoo!News [62]	900	4500 (19.6%)

C.3 Compared Methods

In this section we will briefly introduce ten compared methods used in Section 5.5, containing seven model-based modules and three data-driven modules. The compared methods are list as follows:

1) Random Sampling (RS): In each query round, it randomly selects b samples from the unlabeled pool, and then hand over these samples to the oracle for annotation.
2) Minimum confidence uncertainty (MCU): Similar to MMU, it calculates the uncertainty score but using Eq. 1 and selects the b samples with the highest uncertainty scores in the unlabeled pool and then sends these samples to the oracle for annotation.

3) Minimum margin uncertainty (MMU): In each query round, it calculates the uncertainty score using Eq. 2 and selects the b samples with the highest uncertainty scores in the unlabeled pool and then sends these samples to the oracle for annotation.

4) Entropy uncertainty (EU): Similar to MMU, it uses Eq. 3 to obtain the uncertainty score in each round, and selects the b samples with the highest uncertainty scores in the unlabeled pool and then sends these samples to the oracle for annotation.

5) Coreset [36]: In each query round, it selects b samples by solving a b-center issues on the full unlabeled space, using the embedding of the unlabeled samples generated from the penultimate layer of the predictor.

6) BALD [7]: It is developed based on [29]. The original version [29] is a Bayesian modelling-based method, combining the Bayesian modelling to calculate the uncertainty score in each query round. BALD improves this mechanism and proposes an acquisition function to select multiple informative points jointly for AL.

7) BADGE [31]: It selects b samples by adopting the $k-$Means++ to group the features in the unlabeled space, and the feature is generated in a hallucinated gradient space.

8) LL4AL [6]: It introduces an extra module to learn the loss of the predictor, and selects b samples by the loss distance between the predictor and the extra module, and then hands these samples to the oracle for annotation.

9) VAAL [37]: It proposes to train a VAE, latching on to the representing information of both the labeled and unlabeled data. With the help of adversarial learning, the selector could choose b samples with high diversity compared to the labeled samples.

10) TA-VAAL [8]: Building upon VAAL, it further exploits the space difference between the labeled data and the unlabeled data, and incorporate the "learning loss" [6] module to select better representative samples in each query round.

C.4   Ablation results on WorseNet

Different Backbones and partial label generation approaches. In Section 5.1, we list the test performance of our proposed Worsenet and ten AL-based approaches with C-Net (ResNet18) for MINIST and Fashion-MNIST (SVHN and CIFAR-10), and the partial labels are generated using FPS ($q = 0.5$). Here we show the corresponding results implemented based on different backbones and partial label generation methods among Tables C4-C8. As shown in these tables, we could tell that our proposed WP achieves global improvements on all proposed AL-based methods among all backbones and partial label generation methods. Specifically, our proposed WP could achieve performance elation in both FPS with $q = 0.3$ and $q = 0.5$ cases, illustrating that WP is robust to the label noise in the candidate set.

C.5   Ablation results on WorseNet

Different Backbones and partial label generation approaches. In Section 5.1, we list the test performance of our proposed Worsenet and ten AL-based approaches with C-Net (ResNet18) for MINIST and Fashion-MNIST (SVHN and CIFAR-10), and the partial labels are generated using FPS ($q = 0.5$). Here we show the corresponding results implemented based on different backbones and partial label generation methods among Tables C4-C8. As shown in these tables, we could tell that our proposed WP

achieves global improvements on all proposed AL-based methods among all backbones and partial label generation methods. Specifically, our proposed WP could achieve performance elation in both FPS with $q = 0.3$ and $q = 0.5$ cases, illustrating that WP is robust to the label noise in the candidate set.

Discussion about WorseNet-Selector module. For the three newly designed uncertainty-based selectors, i.e., WS-MMU, WS-MCU, and WS-EU, it is found that they could achieve a much higher performance gain in some cases compared to the original version. For instance, WS-MMU achieves about 18% accuracy elation compared to MMU in Table C8. However, it is admitted that WS sometimes degrades the original selection strategies. As shown in Table C7, we can see that WS-MCU are inferior (about 1% accuracy decline) to MCU in Fashion-MINST. More similar phenomenon inordinately appears in different situations in Tables C4-C8.

C.6   Ablation studies on the number of selected samples on WorseNet

In Section 5.3, we study the influence of the number of selected samples during the training period over all modules. Here we present more relevant results in different cases. Figure C3 (Figure C2) shows the results in FPS with $q = 0.3$ (USS), and we can find that our proposed WP (solid lines) could achieve sustainable improvements in all baseline methods (dashed lines) regardless of the partial label generation approach. Besides, we can find that the enhancements are not obvious for some data-driven methods such as LL4AL and VAAL, which means our proposed WP module could be further refined.

Table C4 Test performance of the proposed WorseNet modules and other methods on benchmark datasets using label generation by USS. The best results among all methods with the same backbone are marked in **bold**. -/+ WP denotes whether the predictor is helped by WorseNet. The underline points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 1%. The backbones for MNIST and Fashion-MINIST are C-Net, and for SVHN and CIFAR-10 are ResNet18. Here the standard deviation is ignored.

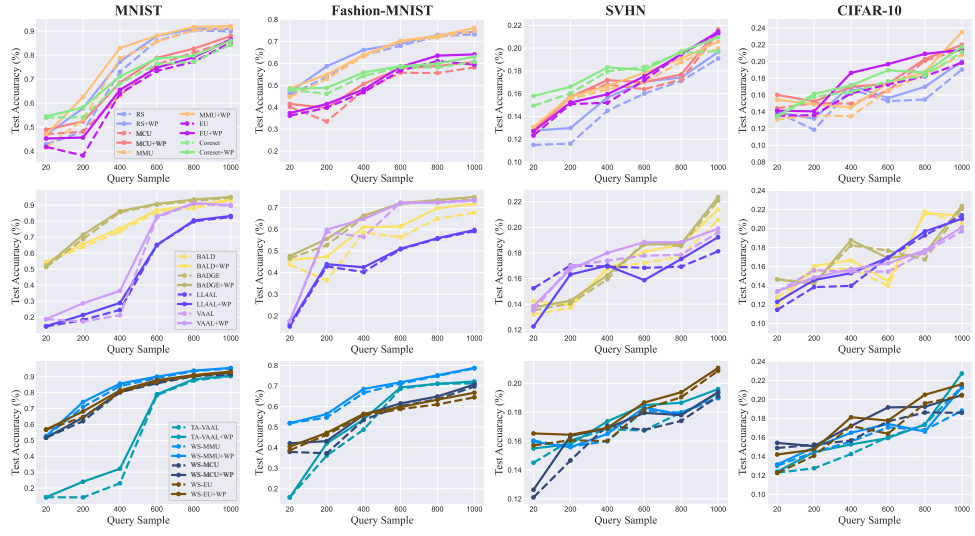| Methods (-/+ WP)	MNIST	Fashion-MINIST	SVHN	CIFAR-10
RS	90.95 / 91.82	74.05 / 74.66	79.19 / 79.64	56.91 / 58.62 ↑
MMU	92.66 / 93.90 ↑	74.80 / 76.19 ↑	80.14 / 81.73 ↑	56.77 / **59.54** ↑
MCU	85.91 / 87.97 ↑	59.64 / 61.60 ↑	79.88 / 80.44	57.45 / 57.99
EU	85.33 / 86.59 ↑	62.36 / 64.73 ↑	79.97 / 81.06	58.15 / 60.46 ↑
Coreset	84.33 / 86.10 ↑	64.34 / 65.79 ↑	79.73 / 81.05 ↑	55.43 / 58.05
BALD	93.50 / 93.90	69.55 / 72.68 ↑	80.48 / 81.39	57.80 / 58.17
BADGE	95.00 / 95.25	74.82 / 75.75	82.09 / **82.47**	58.18 / 58.58
LL4AL	82.74 / 83.31	59.10 / 59.65	79.73 / 79.94	57.02 / 58.87 ↑
VAAL	90.98 / 91.21 ↑	73.12 / 73.83	79.11 / 79.75	57.72 / 58.12
TA-VAAL	90.85 / 91.13	71.94 / 72.45	79.07 / 80.33	58.14 / 58.73
WS-MMU	95.21 / **95.54**	78.55 / **78.80**	80.06 / 80.78	57.38 / 58.48 ↑
WS-MCU	92.44 / 92.90	70.52 / 71.50	81.10 / 81.39	57.17 / 59.14
WS-EU	93.56 / 94.03	65.62 / 67.99 ↑	80.87 / 81.07	58.36 / 60.36 ↑

Fig. C2 The average test accuracy over the different number of query samples on four benchmark datasets during the training time. Note that here settings are corresponding to Table C4 (USS).
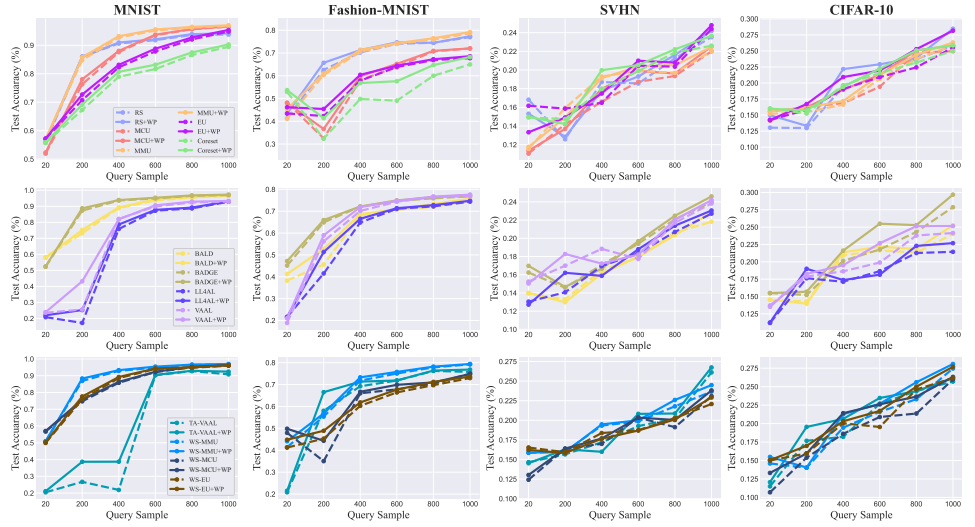

Fig. C3 The average test accuracy over the different number of query samples on four benchmark datasets during the training time. Note that here settings are corresponding to Table C5 (FPS with $q = 0.3$).

Table C5 Test performance of the proposed WorseNet modules and other methods on benchmark datasets using label generation by FPS ($q = 0.3$). The best results among all methods with the same backbone are marked in **bold**. -/+ WP denotes whether the predictor is helped by WorseNet. The underline points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 1%. The backbones for MNIST and Fashion-MINIST are C-Net, and for SVHN and CIFAR-10 are ResNet18. Here the standard deviation is ignored.

Methods (-/+ WP)	MNIST	Fashion-MINIST	SVHN	CIFAR-10
RS	94.18 / 94.51	77.53 / 77.82	80.52 / 81.19	58.83 / 61.46 ↑
MMU	96.99 / 97.21	79.35 / 79.44	82.10 / 82.61	60.96 / 61.85
MCU	96.65 / 96.76	72.16 / 72.35	82.05 / 82.46	62.84 / 63.31
EU	94.84 / 95.41	68.99 / 70.51 ↑	84.40 / 84.79	61.34 / 62.17 ↑
Coreset	89.71 / 90.76	64.98 / 68.26 ↑	82.65 / 83.65 ↑	61.02 / 62.88
BALD	96.61 / 96.74	75.59 / 75.84	81.82 / 82.85 ↑	60.12 / 61.35 ↑
BADGE	97.08 / **97.37**	77.86 / 78.30	83.88 / 84.61	63.88 / **64.69**
LL4AL	92.85 / 93.11	75.09 / 75.58	82.75 / 83.15	57.44 / 58.79 ↑
VAAL	93.36 / 93.61	77.72 / 77.98	83.83 / 84.19	60.15 / 61.16 ↑
TA-VAAL	93.07 / 93.30	76.94 / 77.44	86.11 / **86.74**	61.69 / 62.19
WS-MMU	97.11 / 97.35	79.47 / **79.80**	83.66 / 84.49	63.46 / 64.07
WS-MCU	96.15 / 96.41	74.96 / 75.28	83.08 / 83.81	61.98 / 63.32
WS-EU	96.10 / 96.33	74.01 / 74.51	84.08 / 84.91	62.16 / 63.69 ↑

Table C6 Test performance of the proposed WorseNet modules and other methods on benchmark datasets using label generation by USS. The best results among all methods with the same backbone are marked in **bold**. -/+ WP denotes whether the predictor is helped by WorseNet. The underline points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 1%. The backbones for MNIST and Fashion-MINIST are MLP, and for SVHN and CIFAR-10 are VGG11. Here the standard deviation is ignored.

Methods (-/+ WP)	MNIST	Fashion-MINIST	SVHN	CIFAR-10
RS	84.71 / 85.05	76.32 / 76.76	83.15 / 84.35 ↑	59.64 / 60.19
MMU	85.76 / 86.03	77.76 / 78.24	84.70 / 86.40 ↑	62.89 / 63.17
MCU	77.77 / 78.23	68.53 / 69.04	84.31 / 86.01 ↑	61.45 / 62.66
EU	78.36 / 78.81	63.70 / 64.52	84.38 / 86.88 ↑	61.53 / 61.97
Coreset	70.73 / 71.58	67.18 / 67.86	85.57 / 86.25	60.66 / 60.98
BALD	67.18 / 67.56	73.52 / 74.25	87.07 / 88.37 ↑	61.88 / 62.22
BADGE	86.37 / 86.90	76.82 / 77.36	86.18 / 87.19	63.59 / **64.05**
WS-MMU	87.94 / **88.03**	78.45 / **78.97**	86.98 / 87.80 ↑	61.67 / 62.01
WS-MCU	82.38 / 82.67	72.61 / 73.12	86.17 / 87.32 ↑	61.84 / 62.14
WS-EU	83.18 / 83.40	67.85 / 68.23	86.45 / 87.29	61.42 / 61.92

References

[1] Settles, B.: Active learning literature survey. Science **10**(3), 237–304 (1995)

[2] Cai, L., Xu, X., Liew, J.H., Foo, C.S.: Revisiting superpixels for active learning in semantic segmentation with realistic annotation costs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10988–10997 (2021)

[3] Haussmann, E., Fenzi, M., Chitta, K., Ivanecky, J., Xu, H., Roy, D., Mittel, A., Koumchatzky, N., Farabet, C., Alvarez, J.M.: Scalable active learning for object detection. In: 2020 IEEE Intelligent Vehicles Symposium, pp. 1430–1435 (2020)

Table C7 Test performance of the proposed WorseNet modules and other methods on benchmark datasets using label generation by FPS ($q = 0.3$). The best results among all methods with the same backbone are marked in **bold**. -/+ WP denotes whether the predictor is helped by WorseNet. The <u>underline</u> points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 1%. The backbones for MNIST and Fashion-MINIST are MLP, and for SVHN and CIFAR-10 are VGG11. Here the standard deviation is ignored.

Methods (-/+ WP)	MNIST	Fashion-MINIST	SVHN	CIFAR-10
RS	87.30 / <u>87.96</u>	79.13 / <u>79.63</u>	82.19 / <u>84.93</u> ↑	57.63 / <u>58.98</u> ↑
MMU	91.44 / **<u>91.91</u>**	80.14 / **<u>80.83</u>**	85.06 / <u>86.36</u> ↑	63.19 / <u>64.49</u> ↑
MCU	88.60 / <u>89.12</u>	73.66 / <u>74.12</u>	82.34 / <u>83.37</u>	64.73 / <u>65.14</u>
EU	85.64 / <u>86.41</u>	66.79 / <u>67.53</u>	86.82 / <u>87.14</u>	64.69 / <u>65.08</u>
Coreset	73.47 / <u>74.24</u>	65.75 / <u>66.21</u>	85.86 / <u>87.01</u> ↑	64.93 / <u>65.33</u>
BALD	90.19 / <u>90.80</u>	79.10 / <u>79.68</u>	87.55 / **<u>89.35</u>** ↑	64.61 / <u>65.91</u> ↑
BADGE	91.09 / <u>91.41</u>	78.09 / <u>79.91</u> ↑	86.94 / <u>88.46</u> ↑	65.97 / <u>66.43</u> ↑
WS-MMU	91.08 / <u>91.48</u>	78.84 / <u>79.42</u>	84.13 / <u>84.44</u>	64.42 / **<u>64.74</u>**
WS-MCU	89.14 / <u>89.98</u>	72.47 / <u>73.12</u>	86.25 / <u>88.17</u>	61.74 / <u>61.81</u>
WS-EU	88.11 / <u>88.98</u>	74.07 / <u>74.59</u>	87.93 / <u>88.30</u>	63.13 / <u>63.95</u>

Table C8 Test performance of the proposed WorseNet modules and other methods on benchmark datasets using label generation by FPS ($q = 0.5$). The best results among all methods with the same backbone are marked in **bold**. -/+ WP denotes whether the predictor is helped by WorseNet. The <u>underline</u> points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 1%. The backbones for MNIST and Fashion-MINIST are MLP, and for SVHN and CIFAR-10 are VGG11. Here the standard deviation is ignored.

Methods (-/+ WP)	MNIST	Fashion-MINIST	SVHN	CIFAR-10
RS	85.45 / <u>86.17</u>	77.18 / <u>77.85</u>	73.45 / <u>76.07</u> ↑	52.91 / <u>56.21</u>
MMU	89.14 / **<u>89.48</u>**	78.14 / **<u>78.89</u>**	74.86 / <u>75.95</u> ↑	58.04 / <u>58.17</u>
MCU	82.13 / <u>82.94</u>	59.16 / <u>59.69</u>	75.15 / <u>75.48</u>	59.11 / <u>59.91</u>
EU	79.74 / <u>80.06</u>	64.97 / <u>65.02</u>	74.25 / <u>74.53</u>	58.03 / <u>58.71</u>
Coreset	72.18 / <u>72.59</u>	63.52 / <u>64.85</u> ↑	78.66 / <u>80.09</u> ↑	60.08 / <u>61.09</u> ↑
BALD	87.40 / <u>87.67</u>	74.67 / <u>75.58</u>	77.68 / <u>78.91</u> ↑	61.79 / <u>62.47</u>
BADGE	88.91 / <u>89.12</u>	76.97 / <u>77.19</u>	78.14 / <u>79.87</u> ↑	61.54 / **<u>62.50</u>**
WS-MMU	88.45 / <u>88.56</u> ↑	77.34 / <u>78.19</u>	78.53 / <u>79.07</u>	59.95 / <u>60.14</u>
WS-MCU	85.10 / <u>85.40</u>	70.13 / <u>71.69</u> ↑	80.69 / **<u>81.45</u>**	60.73 / <u>61.06</u>
WS-EU	82.80 / <u>83.91</u> ↑	66.36 / <u>66.73</u>	77.18 / <u>77.58</u>	59.34 / <u>60.97</u> ↑

[4] Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2372–2379 (2009)

[5] Luo, W., Schwing, A.G., Urtasun, R.: Latent structured active learning. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 1, pp. 728–736 (2013)

[6] Yoo, D., Kweon, I.S.: Learning loss for active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 93–102 (2019)

[7] Kirsch, A., Amersfoort, J.v., Gal, Y.: Batchbald: efficient and diverse batch acquisition for deep bayesian active learning. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, vol. 32, pp. 7026–7037

(2019)

[8] Kim, K., Park, D., Kim, K.I., Chun, S.Y.: Task-aware variational adversarial active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8166–8175 (2021)

[9] Parvaneh, A., Abbasnejad, E., Teney, D., Haffari, G.R., Hengel, A., Shi, J.Q.: Active learning by feature mixing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12237–12246 (2022)

[10] Fang, M., Zhu, X.: I don't know the label: Active learning with blind knowledge. In: Proceedings of the 21st International Conference on Pattern Recognition, pp. 2238–2241 (2012)

[11] Briggs, F., Lakshminarayanan, B., Neal, L., Fern, X.Z., Raich, R., Hadley, S.J., Hadley, A.S., Betts, M.G.: Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. The Journal of the Acoustical Society of America **131**(6), 4640–4650 (2012)

[12] Cour, T., Sapp, B., Taskar, B.: Learning from partial labels. The Journal of Machine Learning Research **12**, 1501–1536 (2011)

[13] Zeng, Z., Xiao, S., Jia, K., Chan, T.-H., Gao, S., Xu, D., Ma, Y.: Learning by associating ambiguously labeled images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 708–715 (2013)

[14] Feng, L., An, B.: Leveraging latent label distributions for partial label learning. In: International Joint Conference on Artificial Intelligence, pp. 2107–2113 (2018)

[15] Wang, D.-B., Li, L., Zhang, M.-L.: Adaptive graph guided disambiguation for partial label learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 83–91 (2019)

[16] Wang, H., Xiao, R., Li, Y., Feng, L., Niu, G., Chen, G., Zhao, J.: Pico: Contrastive label disambiguation for partial label learning. In: International Conference on Learning Representations (2022)

[17] Zhang, F., Feng, L., Han, B., Liu, T., Niu, G., Qin, T., Sugiyama, M.: Exploiting class activation value for partial-label learning. In: International Conference on Learning Representations (2022)

[18] Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., Sugiyama, M.: Provably consistent partial-label learning. In: Advances in Neural Information Processing Systems (2020)

[19] Lv, J., Xu, M., Feng, L., Niu, G., Geng, X., Sugiyama, M.: Progressive identification of true labels for partial-label learning. In: International Conference on

Machine Learning, pp. 6500–6510 (2020)

[20] Wen, H., Cui, J., Hang, H., Liu, J., Wang, Y., Lin, Z.: Leveraged weighted loss for partial label learning. In: Proceedings of the 38th International Conference on Machine Learning, vol. 139, pp. 11091–11100 (2021)

[21] Chen, J., Schein, A., Ungar, L., Palmer, M.: An empirical study of the behavior of active learning for word sense disambiguation. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pp. 120–127 (2006)

[22] Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning (2017). Preprint at https://arxiv.org/abs/1712.04621

[23] Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of big data **6**(1), 1–48 (2019)

[24] De Neys, W., Schaeken, W., d'Ydewalle, G.: Working memory and everyday conditional reasoning: Retrieval and inhibition of stored counterexamples. Thinking & Reasoning **11**(4), 349–381 (2005)

[25] Verschueren, N., Schaeken, W., d'Ydewalle, G.: Everyday conditional reasoning: A working memory—dependent tradeoff between counterexample and likelihood use. Memory & Cognition **33**(1), 107–119 (2005)

[26] Johnson-Laird, P.N.: Mental models and human reasoning. Proceedings of the National Academy of Sciences **107**(43), 18243–18250 (2010)

[27] Lewis, D.D., Catlett, J.: Heterogenous uncertainty sampling for supervised learning. In: Proceedings of the Eleventh International Conference on International Conference on Machine Learning, pp. 148–156 (1994)

[28] Roth, D., Small, K.: Margin-based active learning for structured output spaces. In: European Conference on Machine Learning, pp. 413–424 (2006)

[29] Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: International Conference on Machine Learning, pp. 1183–1192 (2017)

[30] Settles, B., Craven, M., Ray, S.: Multiple-instance active learning. Advances in neural information processing systems **20** (2007)

[31] Ash, J.T., Zhang, C., Krishnamurthy, A., Langford, J., Agarwal, A.: Deep batch active learning by diverse, uncertain gradient lower bounds. In: International Conference on Learning Representations (2020)

[32] Elhamifar, E., Sapiro, G., Yang, A., Sasrty, S.S.: A convex optimization framework for active learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 209–216 (2013)

[33] Yang, Y., Ma, Z., Nie, F., Chang, X., Hauptmann, A.G.: Multi-class active learning by uncertainty sampling with diversity maximization. International Journal of Computer Vision **113**(2), 113–127 (2015)

[34] Nguyen, H.T., Smeulders, A.: Active learning using pre-clustering. In: Proceedings of the Twenty-first International Conference on Machine Learning, p. 79 (2004)

[35] Freytag, A., Rodner, E., Denzler, J.: Selecting influential examples: Active learning with expected model output changes. In: European Conference on Computer Vision, pp. 562–577 (2014). Springer

[36] Sener, O., Savarese, S.: Active learning for convolutional neural networks: A coreset approach. In: International Conference on Learning Representations (2018)

[37] Sinha, S., Ebrahimi, S., Darrell, T.: Variational adversarial active learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5972–5981 (2019)

[38] Donmez, P., Carbonell, J.G.: Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 619–628 (2008)

[39] Du, J., Ling, C.X.: Active learning with human-like noisy oracle. In: 2010 IEEE International Conference on Data Mining, pp. 797–802 (2010)

[40] Yan, S., Chaudhuri, K., Javidi, T.: Active learning from imperfect labelers. Advances in Neural Information Processing Systems **29** (2016)

[41] Chakraborty, S.: Asking the right questions to the right users: Active learning with imperfect oracles. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 3365–3372 (2020)

[42] Hüllermeier, E., Beringer, J.: Learning from ambiguously labeled examples. Intelligent Data Analysis **10**(5), 419–439 (2006)

[43] Gong, C., Liu, T., Tang, Y., Yang, J., Yang, J., Tao, D.: A regularization approach for instance-based superset label learning. IEEE Transactions on Cybernetics **48**(3), 967–978 (2017)

[44] Cour, T., Sapp, B., Jordan, C., Taskar, B.: Learning from ambiguously labeled images. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 919–926 (2009)

[45] Yao, Y., Deng, J., Chen, X., Gong, C., Wu, J., Yang, J.: Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. In:

Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12669–12676 (2020)

[46] Liu, L., Dietterich, T.: Learnability of the superset label learning problem. In: International Conference on Machine Learning, pp. 1629–1637 (2014)

[47] Yu, F., Zhang, M.-L.: Maximum margin partial label learning. In: Asian Conference on Machine Learning, pp. 96–111 (2016)

[48] Feng, L., An, B.: Partial label learning by semantic difference maximization. In: International Joint Conference on Artificial Intelligence, pp. 2294–2300 (2019)

[49] Yan, Y., Guo, Y.: Partial label learning with batch label correction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 6575–6582 (2020)

[50] Tran, T., Do, T.-T., Reid, I., Carneiro, G.: Bayesian generative active deep learning. In: International Conference on Machine Learning, pp. 6295–6304 (2019)

[51] Ishida, T., Niu, G., Hu, W., Sugiyama, M.: Learning from complementary labels. Advances in neural information processing systems **30** (2017)

[52] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[53] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017). Preprint at https://arxiv.org/abs/1708.07747

[54] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)

[55] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)

[56] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[57] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). Preprint at https://arxiv.org/abs/1409.1556

[58] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (2015)

[59] Kim, Y.-Y., Song, K., Jang, J., Moon, I.-C.: Lada: Look-ahead data acquisition via augmentation for deep active learning. Advances in Neural Information

Processing Systems **34**, 22919–22930 (2021)

[60] Liu, L., Dietterich, T.G.: A conditional multinomial mixture model for superset label learning. In: Advances in Neural Information Processing Systems, pp. 548–556 (2012)

[61] Briggs, F., Fern, X.Z., Raich, R.: Rank-loss support instance machines for miml instance annotation. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 534–542 (2012)

[62] Guillaumin, M., Verbeek, J., Schmid, C.: Multiple instance metric learning from automatically labeled bags of faces. In: European Conference on Computer Vision, pp. 634–647 (2010)

[63] Feng, L., An, B.: Partial label learning with self-guided retraining. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3542–3549 (2019)