ACTIVE LEARNING WITH PARTIAL LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we for the first time study a new problem setting called *active learning with partial labels* (ALPL), where an oracle provides the query samples with a set of candidate labels that contains the true label, relaxing the oracle from the demanding accurate labeling process. To address ALPL, we firstly propose a firm and intuitive baseline by directly adapting a state-of-the-art method for learning with partial labels to train the predictor, which can be seamlessly incorporated into existing AL frameworks. Inspired by human inference in cognitive science, we propose to improve the baseline by exploiting and exploring *counter examples* (CEs) to relieve the *overfitting* caused by a few number of training samples in ALPL. Specifically, we propose to construct CEs by reversing the partial labels for each instance, and then we propose a simple but effective WorseNet to learn from such designed knowledge. By leveraging the distribution gap between WorseNet and the predictor, both the predictor itself and the sample selection process can be improved. Experimental results on five real-world datasets and four benchmark datasets show that our proposed methods achieve comprehensive improvements over ten representative AL frameworks, highlighting the superiority and effectiveness of CEs and WorseNet.

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 labelling work, numerous researchers have been investigating *active learning* (AL) (Settles, 2009), which aims to achieve as high performance gain as possible by labelling as few samples as possible. A popular setting in AL is pool-based AL (Settles, 2009), where a fixed number of samples selected by a selector are sent to an oracle for labelling iteratively until the exhaustion of the sampling budget. Pool-based AL has a wide range of applications, including but not limited to semantic segmentation (Cai et al., 2021) and object detection (Haussmann et al., 2020).

Most existing pool-based AL frameworks (Joshi et al., 2009; Luo et al., 2013; Yoo & Kweon, 2019; Kirsch et al., 2019; Kim et al., 2021a; Parvaneh et al., 2022) 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 exist in real-world applications (Fang & Zhu, 2012). Let us consider a birdsong classification problem (Briggs et al., 2012b). 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.

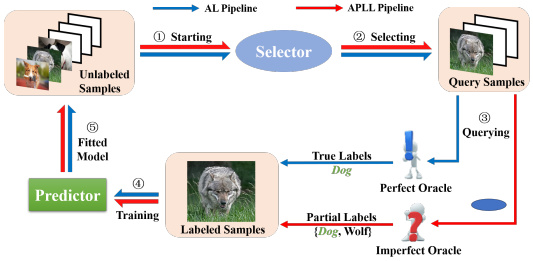

Figure 1: Comparison of pool-based AL (blue arrow) and our proposed ALPL framework (red arrow). The core difference between these two settings is the label form provided from the oracle.

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 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 (Cour et al., 2011), web mining (Luo

& Orabona, 2010), and face recognition (Cour et al., 2011; Zeng et al., 2013). For example, face recognition aims to learn a face recognition system from online images associated with text captions and video scripts. In this way, the face image is often labeled with multiple names since a caption or script usually contains multiple annotations. With the full potential of partial labels seen in these real-world scenarios, *partial label learning* (PLL), aiming at solving a multi-class classification problem where each training instance is assigned a partial label, has naturally emerged and boomed in the community (Feng & An, 2018; Wang et al., 2019; 2022; Zhang et al., 2022). Motivated by such industrial and academical value of partial labels, we propose a new setting for active learning, 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 2 illustrates the pipelines of AL and ALPL. Compared with AL, the oracle in ALPL shall provide noise-tolerant partial labels when annotating confusing objects, highly improving the labeling efficiency while easing the annotation pressure of the oracle during the query process.

To address ALPL, we firstly focus on building a promising baseline. RC loss (Feng et al., 2020), as one of the state-of-the-art milestones in PLL, has been not only theoretically proved to achieve risk-consistency in PLL, but also experimentally evaluated to show competitive performance compared with various works (Lv et al., 2020; Wen et al., 2021; Wang et al., 2022; Zhang et al., 2022). Motivated by this, we directly adopt RC loss to train the predictor with the given partial labels from the oracle, seamlessly switching ALPL to normal pool-based AL. Correspondingly, we are able to form a firm and strong baseline for ALPL on top of various AL frameworks. Though encouraging and effective as it is, ALPL with RC loss, similar to all AL frameworks, face the inevitable *overfitting* challenge (Chen et al., 2006; Perez & Wang, 2017; Shorten & Khoshgoftaar, 2019) during the training process with simply few annotated samples provided.

To relieve the above *overfitting* issue, we turn to an interesting concept from cognitive science named *counter examples* (CEs). According to *mental models* in cognitive science (De Neys et al., 2005; Verschueren et al., 2005; Johnson-Laird, 2010), humans are able to assess the deductive validity of an inference with the help of CEs, leading to draw an accurate conclusion. Inspired by such a human working mechanism, we aim to explore and exploit the useful knowledge from CEs to address ALPL. 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 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 improve the selector by simply considering the labels whose probability gaps are greater than zero, narrowing down the range of uncertainty score calculation. Specifically, we propose three new selectors in ALPL by adopting this selecting strategy on three basic uncertainty-based selectors. Experimental results on benchmark-simulated and real-world datasets validate the effectiveness and superiority of our proposed WorseNet on improving both the selector and the predictor in ALPL.

Our main contributions are summarized here:

- We for the first time propose a practical setting for *active learning* (AL), i.e., *active learning with partial labels* (ALPL), where the oracle could provide the query samples with partial labels, economically facilitating the annotation process for the experts.
- We provide a firm and strong baseline to address ALPL via adopting RC loss for training the predictor, which could be built on any AL approaches. Furthermore, we turn to exploring and exploiting *counter examples* (CEs), and propose a simple but effective WorseNet with Worse loss to improve the predictor and the selector in ALPL.
- Experimental results on four benchmark datasets and five real-world datasets show that our proposed WorseNet achieves promising performance elation over all compared baseline methods, achieving the state-of-the-art performance in ALPL.

2 PRELIMINARIES

2.1 POOL-BASED ACTIVE LEARNING

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* (MCM), *minimum margin uncertainty* (MMU) and *entropy uncertainty* (EU). These three metrics can be sequentially expressed as follows [1]:

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

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

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

where $P(y_i|\boldsymbol{x}_i)$ refers to class-conditional probability modeled by the model outputs and $\boldsymbol{x}^*$ denotes the selected uncertain samples. In this way, uncertainty samples handed over to the oracle could be easily 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.

2.2 PARTIAL-LABEL LEARNING

Partial-label learning (PLL) addresses a multi-class classification problem, where each training instance is assigned a set of candidate labels that include the true label. 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 $y_i \in S_i$, i.e.,

$$P(y_i \in S_i | y = y_i, \boldsymbol{x} = \boldsymbol{x}_i) = 1, \forall y_i \in \mathbb{Y}, \forall S_i \in \mathbb{C}. \tag{4}$$

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 (Feng et al., 2020; Zhang et al., 2022). *(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 (Feng & An, 2019a; Yan & Guo, 2020; Lv et al., 2020; Wen et al., 2021; Wang et al., 2022). In this paper we adopt both of them to generate the partial labels. Refer to Appendix B for more details.

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

3 LEARNING ACTIVELY 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, that may be impractical and demanding for the oracle, which require the oracle to provide the true labels (Gal et al., 2017; Tran et al., 2019; Kim et al., 2021a) 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 (Cour et al., 2011; Zeng et al., 2013). Compared with AL, ALPL eases the annotation pressure for the oracle when facing confusing samples, effectively reducing the labeling efforts. Therefore, we believe that ALPL is full of research significance. We give a formal definition of ALPL as follows:

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 (Lv et al., 2020; Feng et al., 2020), as one of the most prevailing state-of-the-art loss functions, to address ALPL in a simple but effective manner. The empirical risk function $\hat{\mathcal{R}}_{\mathrm{rc}}$ is defined as

$$\hat{\mathcal{R}}_{\mathrm{rc}}(\mathcal{L}, f) = \frac{1}{l} \sum\nolimits_{i=1}^{l} \sum\nolimits_{j \in S_i} \frac{P(y_i = j|\boldsymbol{x}_i)}{\sum_{z \in S_i} P(y_i = z|\boldsymbol{x}_i)} \mathcal{L}(f(\boldsymbol{x}_i), j). \tag{5}$$

Here $\mathcal{L}(f(\boldsymbol{x}), s), s \in S$ refers to the cross entropy loss. As shown in Eq. (5), RC loss is essentially a form of weighted cross entropy, which is theoretically proved to reach risk consistency in PLL, i.e., achieving comparable performance when compared to the fully supervised methods. Due to its superior performance, RC loss is known as one of the milestones to address PLL, forming the baseline or the comparison in other works (Wen et al., 2021; Wang et al., 2022; Zhang et al., 2022). Therefore, here we directly train the predictor f with RC loss to serve as the baseline of ALPL. In this way, we are also able to seamlessly apply any AL-based frameworks (ten approaches implemented in our paper, see Section 5 for more details) to address ALPL.

4 WORSENET: LEARNING FROM COUNTER EXAMPLES

4.1 CONSTRUCTING COUNTER-EXAMPLES

The common point of AL and ALPL is to train a well-performing predictor by leveraging a relatively small set of annotated samples. Therefore, such learning process would confront the *overfitting* problem (Chen et al., 2006; Perez & Wang, 2017; Shorten & Khoshgoftaar, 2019) on these selected samples regardless of the metric design in the selector. In this paper, we would like to relieve this *overfitting* problem through enriching the input data. Rather than the *data augmentation* (DA) technique adopted in previous works (Perez & Wang, 2017; Shorten & Khoshgoftaar, 2019), 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 (De Neys et al., 2005; Verschueren et al., 2005; Johnson-Laird, 2010). 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 due to the existence 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 learned from the original samples, 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

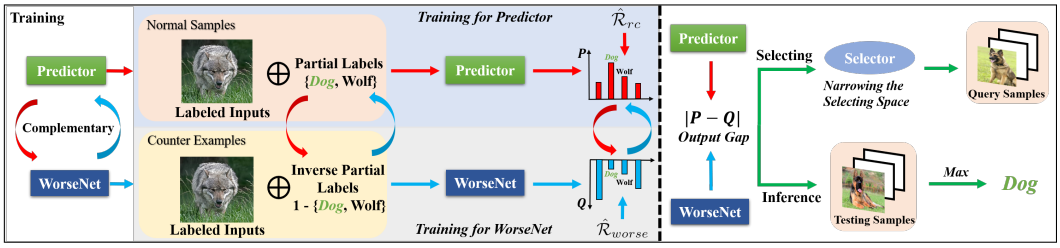

Figure 2: The overall framework of our proposed method to address ALPL. A strong baseline for ALPL is achieved by directly using RC loss to train the predictor (red arrows). To further improve the performance, we propose WorseNet (blue arrows) to extract the useful knowledge from the constructed *counter 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.

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. Note that here the simple inverse annotation forms a CE violating 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:

$$\overline{S}_i = \mathbb{Y} - S_i, \tag{6}$$

where $\overline{S}_i$ denotes the candidate label set for the instance in $\mathbb{W}$. Intuitively, $\overline{S}_i$ is the complementary to S_i, i.e., $\overline{S}_i = \complement_{\mathbb{Y}} S_i$, meaning that the candidate label sets for $\mathbb{W}$ do not contain any true label. For convenience, we name the candidate label set $\overline{S}$ as the *inverse partial label* (IPL).

There are two benefits to forming IPL by following Eq. (6) in ALPL. Firstly, it is convenient and efficient to construct CEs simply by implementing 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 $\boldsymbol{x}_i$, forming CEs in a wealthy manner. Therefore, our proposed IPL takes account of both operational simplicity and label diversity to make CEs in ALPL.

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 assume that there is potentially a *fake true label* inside $\overline{S}$. In this way, we formulate this learning process to a similar PLL problem, where we propose *inverse RC* (IRC) loss to address it as follows:

$$\hat{\mathcal{R}}_{\mathrm{irc}}(\mathcal{L}, w) = \frac{1}{l} \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} \frac{Q(y_i = j|\boldsymbol{x}_i)}{\sum_{z \in \overline{S}_i} Q(y_i = z|\boldsymbol{x}_i)} \mathcal{L}(w(\boldsymbol{x}_i), j), \tag{7}$$

where $\hat{\mathcal{R}}_{\mathrm{irc}}(\mathcal{L}, w)$ 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.

4.2 Predicting Better with WorseNet

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 towards the gainful direction to the predictor. Specifically, the KLD regularization item is expressed as

$$\mathrm{KLD}(P||Q) = \frac{1}{l} \sum_{i=1}^{l} \sum_{j \in \overline{S}_i} P(y_i = j|\boldsymbol{x}_i) \log \frac{P(y_i = j|\boldsymbol{x}_i)}{Q(y_i = j|\boldsymbol{x}_i)}. \tag{8}$$

Algorithm 1 Active learning with partial labels with WorseNet-Predictor (WP)

Input: Predictor f, WorseNet w, iterations T, unlabeled pool of 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.

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 $\mathbb{U}$;
Train an initial f on $\mathbb{L}$ by minimizing the RC loss $\hat{\mathcal{R}}_{\text{rc}}$ in Eq. (5);
Label the samples from $\mathbb{L}$ with IPL $\overline{S}$ by Eq. (6), forming the initial CEs $\mathbb{W}$;
Train an initial w on $\mathbb{W}$ by minimizing the loss function $\hat{\mathcal{R}}_{\text{worse}}$ in Eq. (9);
while $t < T$ and $B > 0$ **do**
  Select b samples from $\mathbb{U}$ by $\Psi(\mathbb{L}, \mathbb{U}, f)$, building the query samples $\triangle\mathbb{U}$;
  Label $\triangle\mathbb{U}$ with S by $\mathcal{O}$, forming the labeled query samples $\triangle\mathbb{L}$;
  Label $\triangle\mathbb{U}$ with $\overline{S}$ by Eq. (6), forming the IPL-annoatated query samples $\triangle\mathbb{W}$;
  $\mathbb{U} \leftarrow \mathbb{U} - \triangle\mathbb{U}$;
  $\mathbb{L} \leftarrow \mathbb{L} \cup \triangle\mathbb{L}$;
  $\mathbb{W} \leftarrow \mathbb{W} \cup \triangle\mathbb{W}$;
  Train f on $\mathbb{L}$ labeled with S by minimizing $\hat{\mathcal{R}}_{\text{rc}}$ in Eq. (5);
  Train w on $\mathbb{W}$ labeled with $\overline{S}$ by minimizing the loss function $\hat{\mathcal{R}}_{\text{worse}}$ in Eq. (9);
  $t \leftarrow t + 1$;
  $B \leftarrow B - b$;
end while
(Inference): Predict the true label y^* by using f and w in Eq. (10).

Output: f, w.

Note that here we stop the gradient backpropagation of P when training w. As shown in Eq. (8), 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:

$$\hat{\mathcal{R}}_{\text{worse}} = \hat{\mathcal{R}}_{\text{irc}}(\mathcal{L}, w) + \text{KLD}(P||Q). \tag{9}$$

Following Eq. (9) to train WorseNet, the predictor during the inference is able to predict the potential true label by

$$y_i^* = \arg\max_{y_i \in \mathbb{Y}} \{P(y_i|\boldsymbol{x}_i) + (1 - Q(y_i|\boldsymbol{x}_i))\}, \tag{10}$$

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, since w, learning directly from CEs, is supposed to show a low response to the elements in the candidate label set. As WorseNet is trained independently of the predictor, the proposed WorseNet is able to benefit the predictor on top of any selector in ALPL. 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.

4.3 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 2.1, 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. (10), 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 MCM (Eq. (1)), 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 correspondingly denote these three methods as WS-MCM, WS-MMU, and WS-EU.

5 EXPERIMENTS

In this section, we evaluate our proposed WP, WS-MCM, 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. Refer to Appendix C for more details.

5.1 BENCHMARK DATASETS COMPARISONS

Datasets and backbones. Our proposed WorseNet-based modules are evaluated on four popular benchmark datasets, which are MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011) and CIFAR-10 (Krizhevsky et al., 2009). 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 (refer to Appendix B for details), 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 Gal et al. (2017); Kirsch et al. (2019)) as the backbones for the predictor. For SVHN and CIFAR-10, we follow most works (Yoo & Kweon, 2019; Ash et al., 2020; Kim et al., 2021a) and choose ResNet18 (He et al., 2016) and VGG11 (Simonyan & Zisserman, 2014) 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) MCM, 3) MMU, 4) EU, 5) Coreset (Sener & Savarese, 2018), 6) BALD (Kirsch et al., 2019), 7) BADGE (Ash et al., 2020), and three data-driven methods: 8) LL4AL (Yoo & Kweon, 2019), 9) VAAL (Sinha et al., 2019) and 10) TA-VAAL (Kim et al., 2021a). The specific training and ALPL settings refer to Appendix C.2. Note that we directly adopt RC loss on these ten methods to build the baselines (see Section 3 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. (10). All the implemented methods are trained on 2 RTX3090 GPUs each with 24 GB memory.

Experiment results. As shown in Table 6 (more results could be found in Appendix C.5), our proposed WorseNet shows its effectiveness and superiority in addressing ALPL on the 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. For three WS-based selectors, i.e., WS-MMU, WS-MCM, 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 diversity samples. Therefore, the experimental results on four benchmark datasets reasonably verify the generalization and effectiveness in addressing ALPL.

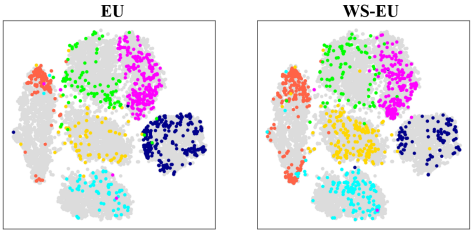

Figure 3: Visualized tSNE results of EU and WS-EU in MNIST with FPS ($q = 0.5$). The colored samples hilights the selected samples.

Table 1: 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 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	93.16 / 93.42	74.76 / 76.18 ↑	21.66 / 22.14	21.75 / 22.58
MMU	95.18 / 96.37 ↑	74.22 / 76.44 ↑	20.60 / 21.65 ↑	20.03 / 21.83 ↑
MCM	93.75 / 94.68	64.59 / 65.75 ↑	18.48 / 19.88 ↑	23.37 / 24.77 ↑
EU	90.83 / 91.28	64.58 / 65.16	21.16 / 22.17 ↑	22.16 / 23.45 ↑
Coreset	86.05 / 87.65 ↑	53.14 / 61.62 ↑	20.70 / 20.81	20.69 / 22.73 ↑
BALD	94.08 / 95.11 ↑	70.95 / 72.95 ↑	20.18 / 20.90 ↑	19.02 / 21.26 ↑
BADGE	96.01 / 96.49	76.75 / 77.10	19.03 / 21.08 ↑	24.04 / 24.68
LL4AL	81.91 / 82.75	60.91 / 61.62	18.79 / 19.02	17.88 / 18.87 ↑
VAAL	90.68 / 91.08	75.18 / 75.44	19.17 / 19.76	19.88 / 20.18
TA-VAAL	90.93 / 91.26	75.21 / 75.90	20.18 / 20.97	19.86 / 20.96 ↑
WS-MMU	95.74 / **96.66**	77.08 / **77.75**	20.46 / 20.96	21.36 / 23.06 ↑
WS-MCM	94.96 / 95.17	68.36 / 69.77 ↑	19.42 / 20.16	20.77 / 21.44
WS-EU	93.90 / 94.80	66.01 / 67.75 ↑	21.30 / **22.21**	24.03 / **25.02**

Table 2: Test performance of the proposed WorseNet modules and other methods on five real-world datasets. -/+ WP denotes whether the predictor is helped by WorseNet. The underline points out improved accuracy by WP. ↑ indicates the improved accuracy is beyond 3%. The best results among all methods with the same backbone are marked in **bold**. Here the standard deviation is ignored.

Methods (-/+ WP)	Lost	MSRCV2	BirdSong	SoccerPlayer	Yahoo!News
RS	22.32 / 25.18 ↑	22.16 / 25.05 ↑	41.68 / 47.20 ↑	47.45 / 49.15	24.44 / 27.32 ↑
MMU	21.38 / 23.82	19.32 / 22.39	43.04 / 47.80 ↑	47.33 / 49.65	21.65 / 25.80 ↑
MCM	20.71 / 22.68	19.55 / 22.73 ↑	31.08 / 33.92	46.94 / 48.98	20.86 / 24.98 ↑
EU	22.89 / 25.54	19.43 / 22.39	30.84 / 34.48 ↑	42.71 / 48.30 ↑	21.69 / 23.90
Coreset	22.32 / 23.75	19.43 / 20.91 ↑	39.72 / 47.64 ↑	48.70 / 49.44	21.03 / 22.89
BALD	20.36 / 23.04	21.02 / 26.59	43.44 / 47.48 ↑	47.03 / 49.07	24.15 / **27.94** ↑
BADGE	23.04 / 26.25 ↑	21.14 / 26.27 ↑	43.32 / **51.12** ↑	48.04 / 49.19	22.35 / 26.32 ↑
WS-MMU	21.43 / 24.64 ↑	21.93 / 26.02 ↑	44.18 / 48.60 ↑	46.34 / 48.37	22.20 / 26.62 ↑
WS-MCM	22.14 / **26.61** ↑	20.91 / **27.95** ↑	31.84 / 35.88 ↑	47.35 / **49.58** ↑	20.97 / 24.59 ↑
WS-EU	20.36 / 25.00 ↑	21.70 / 24.89 ↑	42.40 / 45.40 ↑	47.78 / 48.84	23.60 / 27.13 ↑

5.2 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 (Cour et al., 2011), MSRCv2 (Liu & Dietterich, 2012), BirdSong (Briggs et al., 2012a), Soccer Player (Zeng et al., 2013) and Yahoo!News (Guillaumin et al., 2010). Note that all five of these real-world datasets are annotated with the given candidate label sets, so we simply use them as the oracle annotation. For these five datasets, we adopt the same 3-layer MLP used in Section 5.1 as the sole backbone since these real-world datasets are simple vector inputs, which also follows conventions in (Feng & An, 2019a;b; Feng et al., 2020; Lv et al., 2020; Wen et al., 2021; Wang et al., 2022; Zhang et al., 2022).

Compared methods and training settings. As mentioned above, we only select 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 deep models. Please refer to Appendix C.3 for the detailed training and ALPL settings for these five datasets. 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 5.1.

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-MCM, and WS-EU) in the selector also show competitive

performance compared to the baselines, and they moreover achieve the state-of-the-art performance in Lost and MSRCv2 datasets with WP, validating the benefit of WorseNet to both the selector and predictor in addressing ALPL.

5.3 NUMBER OF SELECTED SAMPLES ON WORSENET

In this part, we demonstrate that WorseNet could deliver sustainable improvements during the whole query process. 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 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 compared to the ten AL-based methods. These results validate the long-lasting benefits of our proposed WorseNet to ALPL. More relevant results can be found in Appendix C.6.

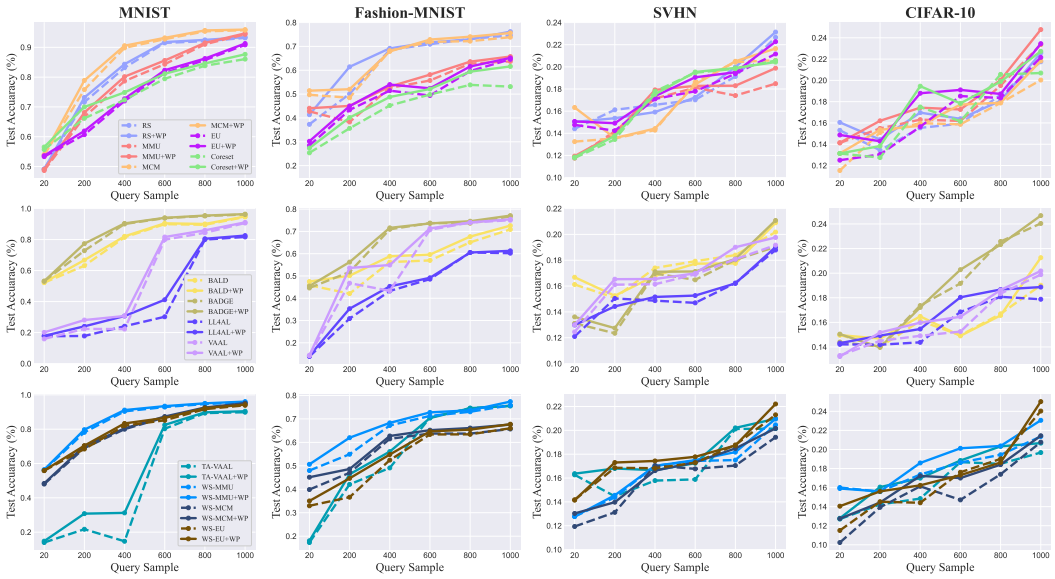

Figure 4: The average test accuracy over the different number of query samples on four benchmark datasets during the training time. Note that here the settings are the same with Table 1.

6 CONCLUSION

We have proposed and investigated a new setting based on pool-based *Active Learning* (AL), i.e., *active learning with partial labels* (ALPL), where the oracle is asked to provide a partial label to the selected samples during the query process. To address ALPL, we firstly adopted RC loss, one of the state-of-the-art methods in PLL, on different AL frameworks to form a strong and effective baseline. Motivated by the salutary effects of *counter examples* (CEs) in human reasoning, we designed CEs in ALPL, which essentially are the inverse version of the original partially labeled examples. Based on the designed CEs, we proposed WorseNet to directly learn from them using the proposed Worse loss. Worse loss is comprised of IRC loss and a *Kullback-Leibler divergency* (KLD)-based regularizer, explicitly regularizing WorseNet to learn in a way beneficial to the predictor. Taking advantage of the probability gap between the predictor and WorseNet, the proposed WorseNet could not only enhance the accuracy of the predictor during the inference, but also improve the selector to select samples in a more exact way during the query process. Comprehensive experimental results on various datasets and AL frameworks demonstrate that our WorseNet achieves state-of-the-art performance in ALPL.

Due page restrictions, please refer to Appendix A for Related Work, which contains a comprehensive introduction about pool-based active learning, active learning with imperfect oracle and partial label learning.

REFERENCES

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations (ICLR)*, 2020.

William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9368–9377, 2018.

Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. Rank-loss support instance machines for miml instance annotation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 534–542, 2012a.

Forrest Briggs, Balaji Lakshminarayanan, Lawrence Neal, Xiaoli Z Fern, Raviv Raich, Sarah JK Hadley, Adam S Hadley, and Matthew G Betts. 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, 2012b.

Lile Cai, Xun Xu, Jun Hao Liew, and Chuan Sheng Foo. 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.

Shayok Chakraborty. Asking the right questions to the right users: Active learning with imperfect oracles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3365–3372, 2020.

Jinying Chen, Andrew Schein, Lyle Ungar, and Martha Palmer. 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.

Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. Learning from ambiguously labeled images. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 919–926. IEEE, 2009.

Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *The Journal of Machine Learning Research (JMLR)*, 12:1501–1536, 2011.

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

Pinar Donmez and Jaime G Carbonell. 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.

Jun Du and Charles X Ling. Active learning with human-like noisy oracle. In *2010 IEEE International Conference on Data Mining*, pp. 797–802. IEEE, 2010.

Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Shankar Sasrty. A convex optimization framework for active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 209–216, 2013.

Meng Fang and Xingquan Zhu. I don't know the label: Active learning with blind knowledge. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 2238–2241. IEEE, 2012.

Lei Feng and Bo An. Leveraging latent label distributions for partial label learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2107–2113, 2018.

Lei Feng and Bo An. Partial label learning by semantic difference maximization. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2294–2300, 2019a.

Lei Feng and Bo An. Partial label learning with self-guided retraining. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3542–3549, 2019b.

Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In *European conference on computer vision*, pp. 562–577. Springer, 2014.

Kenji Fukumizu. Statistical active learning in multilayer perceptrons. *IEEE Transactions on Neural Networks*, 11(1):17–26, 2000.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning (ICML)*, pp. 1183–1192. PMLR, 2017.

Chen Gong, Tongliang Liu, Yuanyan Tang, Jian Yang, Jie Yang, and Dacheng Tao. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics*, 48(3): 967–978, 2017.

Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision (ECCV)*, pp. 634–647. Springer, 2010.

Yuhong Guo. Active instance sampling via matrix partition. *Advances in Neural Information Processing Systems*, 23, 2010.

Elmar Haussmann, Michele Fenzi, Kashyap Chitta, Jan Ivanecky, Hanson Xu, Donna Roy, Akshita Mittel, Nicolas Koumchatzky, Clement Farabet, and Jose M Alvarez. Scalable active learning for object detection. In *2020 IEEE intelligent vehicles symposium (iv)*, pp. 1430–1435. IEEE, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active learning of a kernel machine ensemble for recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1209–1216, 2013.

Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 2, pp. 897–904. Citeseer, 2002.

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

Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 ieee conference on computer vision and pattern recognition*, pp. 2372–2379. IEEE, 2009.

Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8166–8175, 2021a.

Yoon-Yeong Kim, Kyungwoo Song, JoonHo Jang, and Il-Chul Moon. Lada: Look-ahead data acquisition via augmentation for deep active learning. *Advances in Neural Information Processing Systems (NIPS)*, 34:22919–22930, 2021b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems (NIPS)*, 32, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine learning proceedings 1994*, pp. 148–156. Elsevier, 1994.

David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pp. 3–12. Springer, 1994.

Liping Liu and Thomas Dietterich. Learnability of the superset label learning problem. In *International Conference on Machine Learning (ICML)*, pp. 1629–1637. PMLR, 2014.

Liping Liu and Thomas G Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 548–556. Citeseer, 2012.

Jie Luo and Francesco Orabona. Learning from candidate labeling sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

Wenjie Luo, Alex Schwing, and Raquel Urtasun. Latent structured active learning. *Advances in Neural Information Processing Systems*, 26, 2013.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning (ICML)*, pp. 6500–6510. PMLR, 2020.

Jiaqi Lv, Lei Feng, Miao Xu, Bo An, Gang Niu, Xin Geng, and Masashi Sugiyama. On the robustness of average losses for partial-label learning. *arXiv preprint arXiv:2106.06152*, 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 79, 2004.

Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pp. 551–559, 2008.

Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Gholamreza Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12237–12246, 2022.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Dan Roth and Kevin Small. Margin-based active learning for structured output spaces. In *European Conference on Machine Learning*, pp. 413–424. Springer, 2006.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

Burr Settles. Active learning literature survey. 2009.

Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pp. 1070–1079, 2008.

Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. *Advances in neural information processing systems*, 20, 2007.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pp. 1308–1318. PMLR, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5972–5981, 2019.

Toan Tran, Thanh-Toan Do, Ian Reid, and Gustavo Carneiro. Bayesian generative active deep learning. In *International Conference on Machine Learning*, pp. 6295–6304. PMLR, 2019.

Niki Verschueren, Walter Schaeken, and Gery d'Ydewalle. Everyday conditional reasoning: A working memory—dependent tradeoff between counterexample and likelihood use. *Memory & Cognition*, 33(1):107–119, 2005.

Deng-Bao Wang, Li Li, and Min-Ling Zhang. Adaptive graph guided disambiguation for partial label learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 83–91, 2019.

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 11091–11100. PMLR, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Ning Xu, Jiaqi Lv, and Xin Geng. Partial label learning via label enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pp. 5557–5564, 2019.

Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active learning from imperfect labelers. *Advances in Neural Information Processing Systems*, 29, 2016.

Yan Yan and Yuhong Guo. Partial label learning with batch label correction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 6575–6582, 2020.

Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015.

Yao Yao, Jiehui Deng, Xiuhua Chen, Chen Gong, Jianxin Wu, and Jian Yang. Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pp. 12669–12676, 2020.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 93–102, 2019.

Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *Asian Conference on Machine Learning (ACML)*, pp. 96–111. PMLR, 2016.

Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 708–715, 2013.

Chicheng Zhang and Kamalika Chaudhuri. Active learning from weak and strong labelers. *Advances in Neural Information Processing Systems*, 28, 2015.

Fei Zhang, Lei Feng, Bo Han, Tongliang Liu, Gang Niu, Tao Qin, and Masashi Sugiyama. Exploiting class activation value for partial-label learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *Twenty-fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1335–1344, 2016.

APPENDIX

A RELATED WORK

A.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 (Settles, 2009). 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 (Lewis & Catlett, 1994; Lewis & Gale, 1994), the margin between posterior probabilities of a predicted class and the secondly predicted class (Roth & Small, 2006; Joshi et al., 2009), or the entropy (Settles & Craven, 2008; Joshi et al., 2009; Luo et al., 2013). 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 (Gal et al., 2017; Beluch et al., 2018; Kirsch et al., 2019). Some methods also modeled the impacts of the selected sample on the current model through Fisher information (Fukumizu, 2000; Settles et al., 2007), mutual information (Gal et al., 2017; Kirsch et al., 2019), or expected gradient length (Ash et al., 2020). Specifically, Ash et al. (2020) 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 (Guo, 2010; Elhamifar et al., 2013; Yang et al., 2015) focused on sample subset selection while Nguyen & Smeulders (2004) 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 (Freytag et al., 2014), or the distribution distance between the unlabeled pool and the selected samples (Sener & Savarese, 2018; Shui et al., 2020).

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 (Sinha et al., 2019; Tran et al., 2019; Kim et al., 2021a). Moreover, some methods turned to adopting or designing data augmentation to help the selector better learn the input space (Kim et al., 2021b; Parvaneh et al., 2022). Yoo & Kweon (2019) introduced an auxiliary deep network, predicting the "loss" of the unlabeled samples, to select the samples with large "loss" to help the query process.

A.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 (Donmez & Carbonell, 2008; Du & Ling, 2010; Hua et al., 2013; Zhang & Chaudhuri, 2015; Yan et al., 2016; Chakraborty, 2020). Early works (Donmez & Carbonell, 2008; Zhang & Chaudhuri, 2015) 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. Du & Ling (2010) modeled a human-like oracle that would provide noisy labels for the samples with low confidence from the predictor. Yan et al. (2016) studied a case where the oracle could choose to return incorrect labels or abstain from labeling. Some works (Hua et al., 2013; Chakraborty, 2020) 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 sample with a partial label.

A.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).

ABS treats all candidate labels equally and then averages the model outputs of all candidate labels for evaluation. Some non-parametric methods (Hüllermeier & Beringer, 2006; Gong et al., 2017) focused on predicting the label by using the outputs of its neighbors. Moreover, some approaches (Cour et al., 2009; Zhang et al., 2016; Yao et al., 2020) concentrated on leveraging the labels outside the candidate set to discriminate the potential true label. Yao et al. (2020) proposed an entropy-based regularizer to minimize the entropy of each label, maximizing the margin between the potential true label and the impossible labels. Some recent works (Feng et al., 2020; Lv et al., 2020; Wen et al., 2021; Lv et al., 2021) focused on the data generation process and proposed a classifier-consistent method based on a transition matrix. Wen et al. (2021) 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 (Jin & Ghahramani, 2002; Liu & Dietterich, 2014) or the maximum margin criterion (Nguyen & Caruana, 2008; Yu & Zhang, 2016). Later, many researchers have engaged in leveraging the representation information of the feature space to generate the score for each candidate label (Zhang & Yu, 2015; Zhang et al., 2016; Feng & An, 2018; Wang et al., 2019; 2022; Zhang et al., 2022). Xu et al. (2019) proposed to model the generalized label distribution by using the topological information of the feature space. Wang et al. (2022) turned to a contrastive learning framework to eliminate the label disambiguation and reinforce the feature representation learning. Zhang et al. (2022) proposed to take advantage of the class activation map, discriminating the learning pattern of the classifier, to distinguish the potential true label from the candidate label set.

B Generation of Candidate Labels

In Section 2.2 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.

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

$$\widetilde{P}(\boldsymbol{x}, S) = \sum_{i=1}^{k} P(S|y=i)P(\boldsymbol{x}, y=i), P(S|y=i) = \begin{cases} \frac{1}{2^{k-1}-1} & i \in S, \\ 0 & i \notin S. \end{cases} \tag{12}$$

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. (12) illustrates that the candidate label set for each instance is uniformly sampled.

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 label.

$T_{ij} = q, i \neq j$ holds for other elements. The matrix representation of T is expressed as:

$$\begin{bmatrix} 1 & q & q & q & q & q & q & q & q & q \\ q & 1 & q & q & q & q & q & q & q & q \\ q & q & 1 & q & q & q & q & q & q & q \\ q & q & q & 1 & q & q & q & q & q & q \\ q & q & q & q & 1 & q & q & q & q & q \\ q & q & q & q & q & 1 & q & q & q & q \\ q & q & q & q & q & q & 1 & q & q & q \\ q & q & q & q & q & q & q & 1 & q & q \\ q & q & q & q & q & q & q & q & 1 & q \\ q & q & q & q & q & q & q & q & q & 1 \end{bmatrix}$$

C Detailed Supplementary for Experiments

C.1 Benchmark Datasets

In Section 5.1, we use four widely-used benchmark datasets, i.e., MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), and CIFAR-10 (Krizhevsky et al., 2009). Table 3 lists the characteristics of these datasets. We respectively describe these datasets as follows.

- MNIST: It is a 10-class dataset of handwritten digits. Each data is a 28 × 28 grayscale image.
- Fashion-MNIST: 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: Each instance is a 32 × 32 × 3 colored image in RGB format. It is a 10-class dataset of digits.
- CIFAR-10: Each instance is a 32 × 32 × 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 3: Characteristics of benchmark datasets

Datasets	#Train	#Test	#Features	#Classes
MNIST	60,000	10,000	784	10
Fashion-MNIST	60,000	10,000	784	10
SVHN	73,257	26,032	3,072	10
CIFAR-10	50,000	10,000	3,072	10

C.2 Training and ALPL settings for Benchmark Datasets

In section 5.1, we compare ten methods in ALPL. For the seven model-driven methods, we adopt the Adam optimizer (Kingma & Ba, 2015) 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 (Yoo & Kweon, 2019; Sinha et al., 2019; Kim et al., 2021a). Besides, we only 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 from $\mathbb{U}$ in each query round, following prior works (Gal et al., 2017; Kirsch et al., 2019; Kim et al., 2021b). We repeat the query process 10 times such that the overall budget size $B = 1000$.

C.3 Real Datasets

In Section 5.2, we select five real-world datasets including Lost (Cour et al., 2011), MSRCv2 (Liu & Dietterich, 2012), BirdSong (Briggs et al., 2012a), Soccer Player (Zeng et al., 2013), and Yahoo!News (Guillaumin et al., 2010). 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 4: Characteristics of the real-world datasets.

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

Table 5: The explicit query size b and budget size B on five real-world datasets in ALPL.

Parameters	Lost	MSRCV2	BirdSong	SoccerPlayer	Yahoo!News
Query size (b)	4	6	20	60	90
Query budget (B)	20 (1.7%)	30 (1.7%)	100 (2%)	300 (1.7%)	450 (1.9%)

In Section 5.2, it is mentioned that the real-world datasets have their own partial labels, so we directly adopt the corresponding labels as the oracle annotation. Specifically, we set the size of the initial labeled set $\mathbb{L}$ to 5, and repeat the query process 5 times. Based on the different data quantities among these five datasets, we set the budget size B to account for about 2% of all unlabeled samples. we regulate the total number of annotated samples to be less than 2% of the overall training samples. Table 5 lists the detailed settings of our ALPL settings in five real-world datasets. Therefore, as shown in Table 5, we set the total query budget B to 20, 30, 100, 300, and 450 for Lost, MSRCV2, BirdSong, Soccer Player, and Yahoo!News.

C.4 Compared Methods

In this section we will briefly introduce ten compared methods used in Section 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 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.
3) Minimum confidence uncertainty (MCM): 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.
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 (Sener & Savarese, 2018): 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 (Kirsch et al., 2019): It is developed based on (Gal et al., 2017). The original version (Gal et al., 2017) 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 (Ash et al., 2020): 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 (Yoo & Kweon, 2019): 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 (Sinha et al., 2019): 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 (Kim et al., 2021a): Building upon VAAL, it further exploits the space difference between the labeled data and the unlabeled data, and incorporate the "learning loss" (Yoo & Kweon, 2019) module to select better representative samples in each query round.

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 6-10. 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-MCM, 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 10. However, it is admitted that WS sometimes degrades the original selection strategies. As shown in Table 9, we can see that WS-MCM are inferior (about 1% accuracy decline) to MCM in Fashion-MINST. More similar phenomenon inordinately appears in different situations in Tables 6-10. Figure 5 shows visualized selected samples of two uncertainty-based methods and their improved versions by WS. It is intuitively seen that our proposed WS could help the selector select more representative and distinct samples during the query process.

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 7 (Figure 6) 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 6: 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	19.09 / 19.65	20.97 / 22.61 ↑
MMU	85.91 / 87.97 ↑	59.64 / 61.60 ↑	19.96 / 20.57	21.45 / 21.99
MCM	92.66 / 93.90 ↑	74.80 / 76.19 ↑	20.34 / 21.63 ↑	20.78 / **23.51** ↑
EU	85.33 / 86.59 ↑	62.36 / 64.73 ↑	20.27 / 20.77	22.25 / 23.36 ↑
Coreset	84.33 / 86.10 ↑	64.34 / 65.79 ↑	19.73 / 20.95 ↑	22.13 / 22.75
BALD	93.50 / 93.90	69.55 / 72.68 ↑	20.58 / 21.39	21.79 / 22.07
BADGE	95.00 / 95.25	74.82 / 75.75	22.09 / **22.37**	22.18 / 22.58
LL4AL	82.74 / 83.31	59.10 / 59.65	19.63 / 19.93	21.11 / 22.87 ↑
VAAL	90.98 / 91.21 ↑	73.12 / 73.83	19.01 / 19.60	19.71 / 20.12
TA-VAAL	90.85 / 91.13	71.94 / 72.45	18.97 / 19.34	22.14 / 22.73
WS-MMU	95.21 / **95.54**	78.55 / **78.80**	21.06 / 21.77	19.38 / 21.47 ↑
WS-MCM	92.44 / 92.90	70.52 / 71.50	19.10 / 19.39	19.97 / 21.88
WS-EU	93.56 / 94.03	65.62 / 67.99 ↑	20.87 / 21.07	20.42 / 21.60 ↑

Table 7: 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	23.52 / 24.19	25.83 / 28.46 ↑
MMU	96.65 / 96.76	72.16 / 72.35	22.01 / 22.47	25.84 / 26.31
MCM	96.99 / 97.21	79.35 / 79.44	22.00 / 22.51	24.96 / 25.85
EU	94.84 / 95.41	68.99 / 70.51 ↑	24.40 / 24.79	25.54 / 28.16 ↑
Coreset	89.71 / 90.76	64.98 / 68.26 ↑	22.60 / 23.65 ↑	25.02 / 25.87
BALD	96.61 / 96.74	75.59 / 75.84	21.82 / 22.85 ↑	24.02 / 25.25 ↑
BADGE	97.08 / **97.37**	77.86 / 78.30	23.98 / 24.60	27.87 / **29.68**
LL4AL	92.85 / 93.11	75.09 / 75.58	22.71 / 23.05	21.44 / 22.69 ↑
VAAL	93.36 / 93.61	77.72 / 77.98	23.83 / 24.19	24.15 / 25.16 ↑
TA-VAAL	93.07 / 93.30	76.94 / 77.44	26.11 / **26.74**	25.69 / 26.18
WS-MMU	97.11 / 97.35	79.47 / **79.80**	23.66 / 24.49	27.46 / 28.07
WS-MCM	96.15 / 96.41	74.96 / 75.28	23.08 / 23.81	25.98 / 26.32
WS-EU	96.10 / 96.33	74.01 / 74.51	22.08 / 22.91	26.16 / 27.69 ↑

Table 8: 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	23.15 / 24.35 ↑	26.64 / 27.19
MMU	77.77 / 78.23	68.53 / 69.04	21.31 / 23.01 ↑	25.45 / 25.66
MCM	85.76 / 86.03	77.77 / 78.25	21.70 / 25.40 ↑	26.89 / 27.17
EU	78.36 / 78.81	63.70 / 64.52	23.38 / 25.88 ↑	25.53 / 25.97
Coreset	70.73 / 71.58	67.18 / 67.86	27.57 / 28.25	24.66 / 24.98
BALD	67.18 / 67.56	73.52 / 74.25	28.07 / **29.37** ↑	25.88 / 26.22
BADGE	86.37 / 86.90	76.82 / 77.36	27.18 / 27.89	27.59 / **28.05**
WS-MMU	87.94 / **88.03**	78.45 / **78.97**	26.98 / 27.80 ↑	26.67 / 27.00
WS-MCM	82.38 / 82.67	72.61 / 73.12	26.17 / 27.22 ↑	25.84 / 26.04
WS-EU	83.18 / 83.40	67.85 / 68.23	26.45 / 27.29	25.42 / 25.92

Table 9: 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 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 / 87.96	79.13 / 79.63	33.91 / 34.93 ↑	31.63 / 33.98 ↑
MMU	88.60 / 89.12	73.66 / 74.12	29.34 / 30.37	31.73 / 32.14
MCM	91.44 / **91.91**	80.14 / **80.83**	38.06 / 39.36 ↑	33.19 / 34.49 ↑
EU	85.64 / 86.41	66.79 / 67.53	29.82 / 30.14	28.69 / 29.08
Coreset	73.47 / 74.24	65.75 / 66.21	35.86 / 37.01 ↑	30.93 / 31.33
BALD	90.19 / 90.80	79.10 / 79.68	38.55 / **40.35** ↑	30.61 / 31.91 ↑
BADGE	91.09 / 91.41	78.09 / 79.91 ↑	36.94 / 38.46 ↑	33.97 / 34.43 ↑
WS-MMU	91.08 / 91.48	78.84 / 79.42	34.11 / 34.74	34.45 / **34.73**
WS-MCM	89.14 / 89.98	72.47 / 73.12	33.20 / 34.07	31.73 / 31.91
WS-EU	88.11 / 88.98	74.07 / 74.59	33.93 / 34.30	33.15 / 33.94

Table 10: 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 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	85.45 / 86.17	77.18 / 77.85	26.35 / 29.07 ↑	22.91 / 26.21
MMU	89.14 / **89.48**	78.14 / **78.89**	24.86 / 25.95 ↑	28.04 / 28.17
MCM	82.13 / 82.94	59.16 / 59.69	25.15 / 25.48	24.11 / 24.91
EU	79.74 / 80.06	64.97 / 65.02	24.25 / 24.53	21.33 / 21.71
Coreset	72.18 / 72.59	63.52 / 64.85 ↑	28.66 / 30.09 ↑	23.08 / 24.09 ↑
BALD	87.40 / 87.67	74.67 / 75.58	27.68 / 28.91 ↑	25.79 / 26.47
BADGE	88.91 / 89.12	76.97 / 77.19	28.14 / 29.87 ↑	27.54 / **28.50**
WS-MMU	88.45 / 88.56 ↑	77.34 / 78.19	28.53 / 29.07	27.95 / 28.14
WS-MCM	85.10 / 85.40	70.13 / 71.69 ↑	30.69 / **31.45**	26.73 / 27.06
WS-EU	82.80 / 83.91 ↑	66.36 / 66.73	27.18 / 27.58	25.34 / 25.97

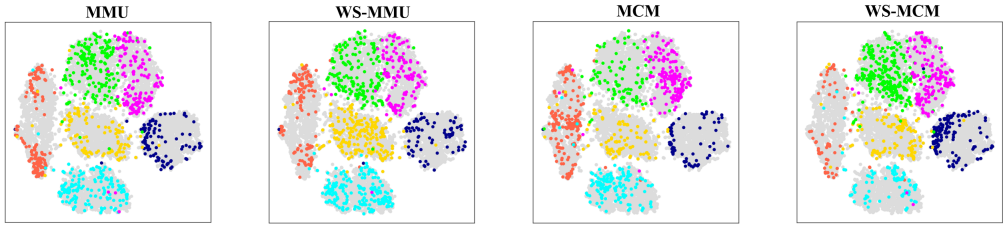

Figure 5: The average test accuracy over the different number of query samples on four benchmark datasets during the training time. Note that here settings are corresponding to Table 1 (FPS with $q = 0.5$).

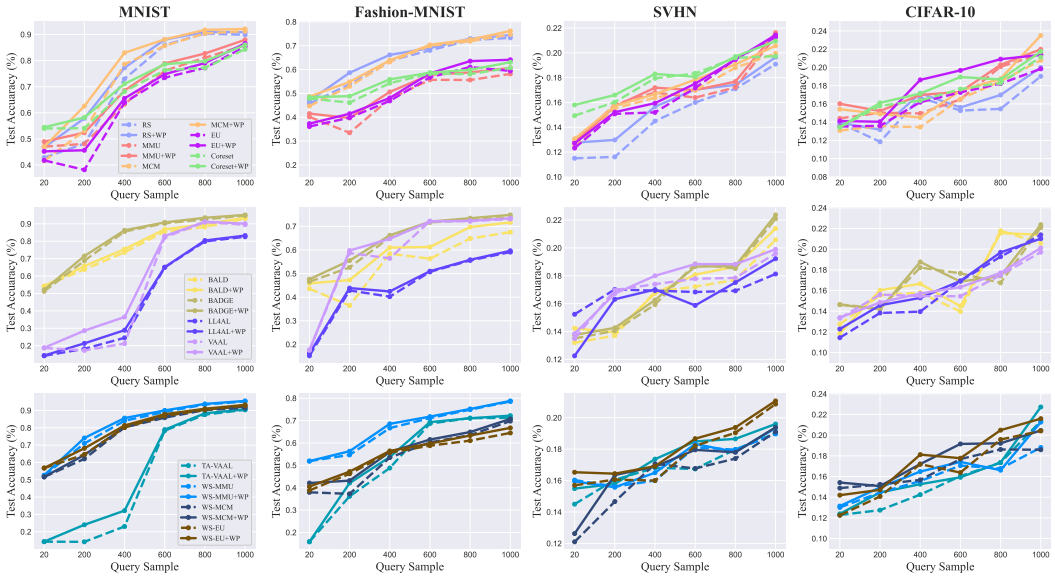

Figure 6: The average test accuracy over the different number of query samples on four benchmark datasets during the training time. Note that here settings are corresponding to Table 6 (USS).

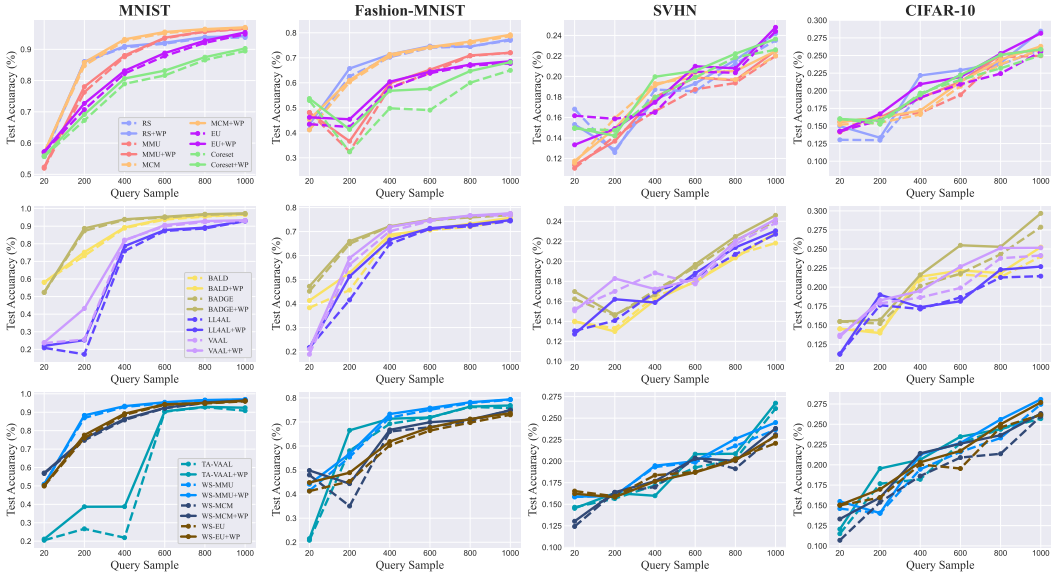

Figure 7: The average test accuracy over the different number of query samples on four benchmark datasets during the training time. Note that here settings are corresponding to Table 7 (FPS with $q = 0.3$).