Trustworthy Active Learning through Reputation and Weighted Voting Mechanisms

Anonymous Authors[1]

Abstract

Active Learning (AL) is a strategy designed to reduce annotation costs by allowing models to select the most informative samples from an unlabeled dataset, particularly in tasks like image classification. Integrated within the broader Human-in-the-Loop (HITL) framework, AL creates an interactive process where human annotators play a role in labeling data. However, this interaction introduces variability in annotation quality, since annotators may differ in domain knowledge, experience, and reliability. Traditional AL approaches often overlook these differences, assuming constant performance across all annotators. This assumption can lead to suboptimal model updates, especially when labels come from less reliable sources. To tackle this limitation, this work proposes a reputation-based framework that captures annotator performance over time and across domains. Reputation scores are computed based on past annotation accuracy and feedback from other annotators, and these scores are then used to weigh individual contributions through a voting mechanism. In addition, the study explores how annotator expertise and oracle size influence the effectiveness of this approach, evaluating their impact on model performance in controlled, simulated settings with imperfect annotators.

Keywords

Active Learning, Human-in-the-Loop, Trustworthy AI, Reputation, Weighted Voting, Imperfect Annotators

1. Introduction

The development of machine learning (ML) models often depends on the availability of large volumes of labeled data. Yet, in many real-world scenarios, collecting these labels is far from trivial, it's expensive, time-consuming, and often demands specialized human expertise. In response to these limitations, R. Munro [1] describes Human-in-the-Loop (HITL) learning as an alternative that allows human knowledge to be integrated into the model training process. One of the most widely adopted strategies within this paradigm is Active Learning (AL), where the model chooses which examples to label based on uncertainty, aiming to reduce annotation effort while still improving performance. In AL, the source of labels, typically a human or group of human annotators, is commonly referred to as an oracle. Its effectiveness can suffer in settings where the oracle is imperfect, which is, in fact, the norm rather than the exception in most practical situations. Many traditional approaches treat all members of the oracle as if they were equally reliable, ignoring the variability in human expertise. This can be problematic when some annotators are systematically wrong or inconsistent, and the model ends up learning from noisy or misleading inputs. In such cases, the challenge lies not only in selecting the right example to label, but also in deciding who within the oracle should be trusted with the task.

This work takes a closer look at how to make the annotation process in AL more robust in the face of these imperfections. To do so, its proposed a reputation-based mechanism that estimates how reliable each annotator is over time, and uses this information to guide both which examples are selected for annotation and how the final labels are aggregated. More specifically, this study is guided by three central questions: first, how can we define and integrate reputation into AL in a way that reflects real annotator behavior? Second, can such a reputation-aware strategy actually improve the quality of the models we train? And third, how do factors like annotator expertise and the number of annotators involved influence the overall performance of this method?

IAL 2025: 9th International Workshop on Interactive Adaptive Learning, Co-Located With ECML-PKDD 2025, September 15th or 19th, 2025

2. Related Work

2.1. Human-in-the-Loop Learning

The concept of HITL, as introduced by R. Munro [1], refers to methodologies that incorporate human input into the design, training, and operation of AI systems. The premise is that the collaboration between human and machine intelligence can achieve results unattainable by either entity alone. In this paradigm, human agents intervene when automated systems encounter ambiguous or unresolved problems, creating a continuous feedback loop through which the system can iteratively improve its performance [2, 3, 4].

As noted by Wu et al. [2] and Chai et al. [3], human involvement can be incorporated at multiple stages of the model development lifecycle. In the *data preprocessing* phase, which includes tasks such as data extraction, integration, and cleaning, human agents can define extraction heuristics, resolve entity ambiguities, and validate corrections to ensure the consistency of the data and integrity. Afterward, during dataset construction, labeling becomes an important and resource intensive step [4, 1]. By giving priority to the annotation of the most informative examples, human annotators enhance the model performance through a process of iterative refinements known as *data annotation* [2, 3]. During the *model training* and *inference* phases, humans can contribute by resolving uncertain classifications, validating predictions, and refining decision boundaries.

Nevertheless, as mentioned by Munro [1], erroneous human annotations can significantly degrade model accuracy and, large scale annotation tasks can be costly and time consuming, given the slowness and limited scalability of human labor relative to automated processes [4].

2.2. Active Learning

AL is a branch of ML and it is also known as *query learning*. This method is based on the principle that a learning algorithm can actively choose the data, from which it learns, as described by B. Settles [5] and R. Munro [1]. For a traditional ML system to operate effectively, it typically requires hundreds or even thousands of labeled data points. AL can minimize this requirement by iteratively selecting the most informative samples from a pool of unlabeled data, $\mathcal{U} = \{x_1, x_2, \ldots, x_{n_u}\}$, where each x_i is a data point from the feature space $\mathcal{X}$. The selection is performed according to a given scenario, $S = \{s_1, s_2, \ldots, s_n\}$, which defines how the selection and annotation process is carried out during training. The goal is to acquire the corresponding labels y_i for selected x_i, where $y_i \in \mathcal{Y}$ and $\mathcal{Y}$ denotes the label space.

The selection of samples to be labeled (i.e., assigned a target value y_i) is guided by query strategies, $Q = \{q_1, q_2, \ldots, q_n\}$, which estimate the utility of each instance based on criteria such as uncertainty, representativeness, or diversity. The selected instances are then submitted to a set of annotators, $A = \{a_1, a_2, \ldots, a_m\}$, which can consist of humans, machines, or a combination of both, and are responsible for assigning the appropriate y_i values. The labeled instances $\langle x_i, y_i \rangle$ are added to the labeled dataset, $\mathcal{L} = \{\langle x_1, y_1 \rangle, \ldots, \langle x_{n_l}, y_{n_l} \rangle\}$, which is used to train the predictive model, or *learner*, represented by a function h that learns patterns in $\mathcal{L}$.

As mentioned previously, one of the main advantages of AL is that it reduces the number of labeled instances required to train the learner h. This is achieved by selecting the most informative examples from the unlabeled pool $\mathcal{U}$, thereby avoiding redundant labeling and improving the efficiency of the labeled set $\mathcal{L}$ [5]. This process is particularly relevant in contexts where labeled data is limited or costly to acquire, such as in medical imaging and natural language processing [1]. Besides, AL can accelerate the convergence of h by enabling it to reach a decent performance with fewer training iterations [6]. When integrated into HITL systems, the selection process enables the incorporation of human expertise from the annotator set ($|A|$), contributing to the system's adaptability and stability [4].

Despite these benefits, one issue is its sensitivity to noise and outliers since selection mechanisms based on query strategies may give prioriority to mislabeled or unrepresentative instances, negatively affecting the quality of $\mathcal{L}$ and the performance of h [5]. Besides, the effectiveness of AL is dependent on the choice of $q_i \in Q$, which may not generalize across different tasks or datasets [1]. From a computational perspective, repeatedly retraining h and continuously evaluating utility over $\mathcal{U}$ can be costly. And lastly, instances labeled through this process may not transfer effectively to other models or learning scenarios [6].

2.2.1. Scenarios

According to B. Settles [5], there are three main scenarios in which an AL system can query instances from the $\mathcal{U}$: *pool-based*, *stream-based*, and *membership query synthesis*.

On the first hand, in the *pool-based* scenario, a set of the most informative instances is selected from $\mathcal{U}$ based on a model trained on $\mathcal{L}$. These selected instances are then sent to a group of annotators for labeling. Once labeled, they are added to $\mathcal{L}$, and the model is retrained to improve its performance. Although widely used, this approach can be computationally demanding due to the repeated evaluation of a large number of candidates [7]. On the other hand, in the *stream-based* scenario, B. Settles [5] states that obtaining an instance from $\mathcal{U}$ must be assumed to be free or inexpensive, allowing it to be drawn from the data distribution before h decides whether to request its label. This method is also known as *sequential AL* because instances from $\mathcal{U}$ are retrieved one at a time. Lastly, in the *membership query synthesis* scenario, the learner h can request labels for any instance in the input space, including synthetically generated queries rather than those sampled from the distribution. This method is effective for finite problem domains and does not require processing $\mathcal{U}$, as h can quickly generate new query instances. However, a major limitation is that, in some cases, it can create instances that are difficult or impossible for an annotator in A to understand and label correctly [7].

2.2.2. Query Strategies

Query strategies define how an AL system selects which instances to label, based on a utility function applied to the $\mathcal{U}$ [1, 5]. These strategies are generally divided into two categories: information-based, which prioritize uncertain instances near the model's decision boundary, and representation-based, which aim to select instances that best reflect the structure of the data distribution [7].

As for information-based [7], *Uncertainty Sampling* [1, 5] selects the samples for which the model h is least confident. This strategy includes three main methods: first, *Least Confidence*, which selects instances with the lowest predicted probability for the most likely class; second, *Margin Sampling*, which focuses on the smallest difference between the top two class probabilities; and third, *Entropy Sampling*, which considers the overall uncertainty across all class probabilities. Furthermore, *Query-by-Committee* [5] selects instances that lead to the highest disagreement among a group of models (the committee). In this case, disagreement is often quantified using vote entropy. Finally, *Expected Model Change* selects instances that would lead to the greatest change in the model if labeled. One example of this is the Expected Gradient Length, where the expected magnitude of the gradient update is used as a selection criterion [3, 8].

In contrast, representation-based strategies [7] focus on selecting instances that best reflect the structure of the input space. For example, *Density Sampling* [1] selects instances from high-density regions, using similarity measures such as the distance between feature vectors, with the goal of querying representative points. Additionally, *Diversity Sampling* gives priority to instances that maximize the coverage of the feature space, aiming to reduce redundancy by selecting diverse samples relative to the $\mathcal{L}$ set [1]. Lastly, *Cluster-Based* strategies partition the data into clusters and select the instances closest to the cluster centers [1].

Finally, it is also important to mention the *Random* query strategy, which selects unlabeled instances entirely at random, without considering information from the data distribution or model predictions. This strategy serves purely as a baseline for evaluating other query methods [1, 5].

2.3. Trust and Reputation

2.3.1. Trust

Trust (hereafter represented as $\mathcal{T}$), is a concept that has been studied extensively in various fields, leading to multiple definitions. S. Marsh [9], author of the *Marsh* $\mathcal{T}$ model, and M. Deutsch [10] describe $\mathcal{T}$ as a common social phenomenon. According to them, trusting behaviour arises when an individual faces an uncertain decision path, the outcome of which can be either positive or negative and depends on the actions of another individual. If the negative outcome is potentially more harmful than the positive outcome and the individual still chooses to proceed, this is considered a trusting decision. If not, the individual is said to be acting with distrust. Moreover, in groups composed of many individuals, situations often arise that require cooperation in order to achieve common goals. Such cooperation typically depends on $\mathcal{T}$, which can be established either explicitly, through agreements, or implicitly, through observation of behaviour over time [11, 12]. To complement this, Hoelz et al. [13] and J. Sabater et al. [11] add that $\mathcal{T}$ is crucial for meaningful social interactions within multi-agent systems. They define it as the quantified belief held by a *trustor* about the qualities of a *trustee*, such as competence, commitment, honesty, security, and dependability, evaluated in the context of the interaction.

A. Jøsang et al. [14], building on the ideas of S. Marsh [9], define $\mathcal{T}$ as a willingness to rely on another entity in a particular context, despite the potential for negative outcomes, driven by a sense of relative security. I. Pinyol et al. [15] see it as a process of practical reasoning that ultimately leads to a decision to engage with another entity.

In addition, a study by F. Mafızoğlu et al. [16] identifies and examines factors that influence $\mathcal{T}$ between entities, which can be humans, artificial agents, or both. They argue that $\mathcal{T}$ is inherently relational and applies to human-to-human, agent-to-agent, and human-to-agent interactions. In their research, they identified factors that are relevant to both humans and artificial agents. First, common factors include characteristics that the entities share, such as task performance, clarity of communication, reliability, and transparency of actions or decisions. Second, human-specific factors include characteristics that are unique to humans, such as age [17], personality [18], mood [19], and personal ideologies [20]. Finally, agent-specific factors are those that evaluate the trustworthiness of agents based on their hardware architecture, type of agent, technical precision, defection [21], cooperativeness, and deception [22], since artificial agents do not possess emotions.

2.3.2. Reputation

J. Sabater et al. [12, 11], T. Huynh [23], and A. Jøsang et al. [14] agree that reputation, $\mathcal{R}$, is constructed from observations about an individual's past behaviour. This means that, in order to build a measure of $\mathcal{R}$, a particular individual needs to consult other individuals in a community to gather their observations about the current individual being evaluated. These observations are usually in the form of ratings that individuals give to one another after an interaction. Once collected and aggregated, these ratings can be used to represent the $\mathcal{R}$ of an individual.

T. Huynh [23] further adds that $\mathcal{R}$ and $\mathcal{T}$ in each interaction have a close relationship, because the $\mathcal{T}$ that an individual A places in B after an interaction is reflected by the rating provided by A. Since the $\mathcal{R}$ of an individual is built based on the ratings given by others, it can be said that the $\mathcal{R}$ of an individual B is constructed from the interaction $\mathcal{T}$ it receives from others in the community,

even when those ratings may be imperfect. Furthermore, $\mathcal{R}$ serves as a decision-making mechanism when an individual A lacks sufficient knowledge or experience with B, using the $\mathcal{R}$ of B as a proxy for reliability, as discussed by I. Pinyol [15] and J. Sabater et al. [11]. Expanding on this concept, the FIRE model proposed by T. Huynh [23] adds more complexity to the idea of $\mathcal{R}$. For example, the author introduces the concept of *witness* $\mathcal{R}$, which is constructed from reports provided by third parties regarding an individual's behaviour, and the concept of *certified* $\mathcal{R}$, which is obtained from provable references presented by the individual itself.

2.3.3. Classification Dimensions

$\mathcal{T}$ and $\mathcal{R}$ are complex concepts, applied across diverse contexts, which makes their classification a difficult task. Therefore, as noted by J. Sabater et al. [11], it is essential to classify the dimensions that these concepts could extend to.

The authors [11] describe that $\mathcal{T}$ and $\mathcal{R}$ can be modeled in two distinct ways. As *cognitive models*, introduced by B. Esfandiari et al. [24] in 2001, which conceptualize $\mathcal{T}$ or $\mathcal{R}$ as merely beliefs, and understood as functions that represent a level of confidence in an entity or individual. Or, as *game-theoretical models*, which interpret them as probabilities that reflect one's expectations about another's actions, similar to Bayesian networks, as presented by D. Gambetta [25] in 1988.

Moreover, $\mathcal{T}$ and $\mathcal{R}$ are context-dependent variables [11], meaning that the level of $\mathcal{T}$ or $\mathcal{R}$ attributed to an individual can vary according to the situation. For example, trusting a doctor to recommend medication does not imply the same level of $\mathcal{T}$ when the same doctor suggests a bottle of wine. With this example, the authors distinguish between two types of context variables: *single-context*, which assign a single $\mathcal{T}$ or $\mathcal{R}$ value to each partner regardless of the situation, and *multi-context*, which maintain separate values for each context or, in other words, domain.

2.4. Imperfect Annotators

Traditional AL algorithms assume labeling oracles, or more precisely, annotators, to be infallible, always providing correct labels. However, this is an assumption that often fails in real-world scenarios, where annotators may offer labels of various types of qualities, as noted by S. Chakraborty [26]. One the first hand, the performance of human annotators depends on various factors, including expertise, experience, concentration, interest, and fatigue [27]. On the other hand, labels derived from simulations or test stands are subject to uncertainty due to imperfections, sensor noise, or transmission errors [28]. Additionally, the accuracy of the oracles answers can depend on how difficult the labeling task is since for simpler classification problems, oracles are more likely to give correct labels. As the task becomes more complex, it is more likely that oracles will be uncertain about their responses, and here, the term uncertainty is used as defined by Motro et al. [29], and refers to situations where the answer may be unlikely, unclear, unreliable, inconsistent, or hard to define precisely.

Urner et al., in their research [30], provide a good example of this topic since they discussed that certain datasets require labeling by top human experts, which can be both complex and costly. For instance, classifying brain tumors from CT scans as benign or malignant falls under this category. As a more cost-effective alternative, medical students could label these images, as they are more available and less expensive to employ. However, it's important to note that the labels provided by students may be inaccurate, especially for images that are difficult to classify.

Besides, Yan et al. [31] mention that when working with multiple annotators, the challenge is not just to pick the most informative sample to label, but also to choose the right annotator to label it. This adds a new dimension to AL because, some annotators can be more reliable than others, may behave maliciously, or have different levels of expertise depending on the domain.

Another important aspect raised by same authors [31] concerns the allocation of annotators when dealing with multiple unreliable sources. The authors propose an alternative method by simulating synthetic annotators, and rather than relying on human input, each synthetic annotator is modeled probabilistically through a confusion matrix. For instance, an annotator may correctly label class A 90% of the time while misclassifying it as class B in the remaining 10%. By drawing labels according to these distributions, the model is able to reproduce annotators with different levels of knowledge in each domain.

3. Methodology

In order to address the research questions stated, this study proposes a framework that takes into account evaluation metrics, paying particular attention to $\mathcal{R}$. The framework consists of a cycle of AL organised around five elements: the learner, which represents the model h; the data D, which is composed of $\mathcal{L}$ and $\mathcal{U}$; the query strategy; the scenario; and the oracle A. Building upon this structure, the study considers only a single learner, initially trained on a small portion of $\mathcal{L}$. The learner operates in a *pool-based* selective sampling scenario, where at each iteration it has access to the entire $\mathcal{U}$ and selects a small pool of the most informative instances to query from the oracle A. Here, informativeness is measured using the *least confidence* query strategy, which selects the instance for which h has the lowest predicted confidence in its most likely y.

Then, a weighted voting system is applied, where each annotator contributes to the final decision proportionally to their individual $\mathcal{R}$ within a specific domain. As a result, the oracle A, which consists of multiple simulated annotators, assigns a label y to each queried instance based on the aggregated votes. Once labeled, the instance is added to the $\mathcal{L}$, and the model h is updated accordingly. This cycle is repeated iteratively; that is, new samples are queried, labeled, and used to retrain h until a stopping condition is reached, such as a maximum number of queries. In what follows, the next sections provide an detailed explanation of each step involved in this process.

3.1. Synthetic Annotators

To simulate imperfect annotators, one could consider training them on existing datasets. However, this approach would significantly reduce the amount of data available for training, testing, and validating the model h. Even with access to large datasets containing diverse annotations, training a statistically significant number of annotators would be time-consuming. To overcome these limitations, its possible to simulate synthetic annotators using the strategy proposed by Yan et al. [31]. Following their method, each annotator $a_i \in A$ is modeled probabilistically by generating a confusion matrix π^{a_i}, where each row is sampled from a *dirichlet* distribution.

Consider a classification task with $K = 3$ classes, $\mathcal{Y} = \{0, 1, 2\}$. For annotator a_i, we define the confusion matrix $\pi^{a_i} \in \mathbb{R}^{K \times K}$, where each row $\pi_c^{a_i}$ corresponds to the true class $c \in \mathcal{Y}$ and represents a probability distribution over the predicted labels $k \in \mathcal{Y}$. Each row is generated as:

$$\pi_c^{a_i} \sim \text{Dirichlet}(\boldsymbol{\alpha}_c) \tag{1}$$

Here, $\boldsymbol{\alpha}_c = [\alpha_{c_0}, \alpha_{c_1}, \alpha_{c_2}]$ is the *dirichlet* concentration vector for true class c, where each α_{c_k} corresponds to the concentration parameter for predicting class k when the true class is c. Assigning a higher value to $\alpha_{c,c}$ (i.e., the diagonal element of the concentration vector) increases the probability that the annotator labels an instance of class c correctly. For instance, the resulting confusion matrix π^{a_i} might be:

$$\pi^{a_i} = \begin{bmatrix} 0.85 & 0.10 & 0.05 \\ 0.20 & 0.70 & 0.10 \\ 0.10 & 0.25 & 0.65 \end{bmatrix} \tag{2}$$

In this example, annotator a_i is particularly accurate in labeling instances of class 0 compared to classes 1 and 2, suggesting a stronger familiarity with that class. However, this does not necessarily imply domain expertise in a semantic sense. One could argue that an expert annotator should also recognize hierarchical subtypes (e.g., if the parent class is *tree*, the expert should identify subtypes like oak, palm, or pine).

Besides, by adjusting $\boldsymbol{\alpha}_c$, it is possible to simulate different levels of annotator domain knowledge. Following the ideas of Hannah et al. [32], to control the expected accuracy acc_t of an annotator in their expert class c, we rely on the property that the expected value of the Dirichlet-distributed probability for class k is:

$$\mathbb{E}[\pi_{c,k}^{a_i}] = \frac{\alpha_{c,k}}{\sum_{k'=0}^{K-1} \alpha_{c,k'}} \tag{3}$$

Given a total of K classes and a target expected accuracy acc_t for the expert class c, we set the concentration parameter for the correct label $k = c$ as:

$$\alpha_{c,c} = \frac{(K-1) \cdot acc_t}{1 - acc_t} \tag{4}$$

The remaining $K - 1$ classes receive a base concentration of 1.0. The resulting Dirichlet vector $\boldsymbol{\alpha}_c$ is then defined as:

$$\alpha_{c,k} = \begin{cases} \alpha_{c,c} & \text{if } k = c \\ 1.0 & \text{otherwise} \end{cases} \tag{5}$$

In addition to the expected accuracy, it is important to note that setting a desired expected value does not deterministically fix the sampled outcome. Due to the stochastic nature of the Dirichlet process, there exists a variance around the expected value, which reflects the uncertainty inherent to each sampled probability vector [33]. This variability is quantified by the variance of the Dirichlet distribution, which is given by:

$$\mathrm{Var}[p_c] = \frac{\alpha_{c,c}(\alpha_0 - \alpha_{c,c})}{\alpha_0^2(\alpha_0 + 1)} \tag{6}$$

where $\alpha_{c,c}$ is the concentration parameter associated with the correct class index c and $\alpha_0 = \sum_{j=0}^{K-1} \alpha_{c,j}$ is the total concentration for the Dirichlet distribution associated with class c.

3.2. Trust

Any definition of $\mathcal{T}$ stated in the last sections can be used in the context of AL, but because there are so many variables and conditions involved, some of them may add extra complexity. Therefore, without implementing *Marsh's* complete $\mathcal{T}$ model, this work accepts the general definition of $\mathcal{T}$ proposed by S. Marsh [9] and later extended by A. Jøsang et al. [14]. In the context of AL, $\mathcal{T}$ is relevant because the learner h relies on the oracle A to provide accurate labels for uncertain samples. For this reason, $\mathcal{T}$ is simply defined in this study as the willingness to rely on an entity despite potential consequences. Therefore, $\mathcal{T}$ is understood as the learner's belief that the oracle will provide trustworthy annotations, regardless of its errors. However, since annotators only

assign labels and evaluate each other's annotations, $\mathcal{T}$ is not extended to other components of the system.

Other perspectives that take social aspects into account when modelling $\mathcal{T}$ include those of J. Sabater [11] and F. Mafizoğlu et al. [16]. These may be relevant, but due to their complexity, they are beyond the scope of this study. In addition, since the primary goal is to examine $\mathcal{R}$, we will not categorise this variable according to its dimensions, which include granularity, visibility, and type of conceptual model.

3.3. Reputation

This work adopts the notion of $\mathcal{R}$ as defined by J. Sabater et al. [12, 11], and later extended by Huynh [23] and Jøsang et al. [14]. In their view, $\mathcal{R}$ is understood as a collective judgment of an agent $\mathcal{T}$, formed from the evaluations provided by other members within a community.

Building on this notion, each annotator a_j, in the context of AL, is assigned a reputation value $\mathcal{R}$ that is updated through peer evaluations. In order for $\mathcal{R}$ to exist in this setting, annotators need to rate each other after each labeling interaction. Specifically, these ratings are binary and converted into a score $\mathcal{S}$, being either good $(+1)$ or bad $(+0)$, depending on whether the annotators agreed on their predicted label. For example, if annotator a_1 evaluates a_2, then a_2 receives a good rating if both provided the same label for the queried instance; otherwise, the rating is bad.

Following the *multi-context scenario* [11], $\mathcal{R}$ can be treated as a class-wise property, recognizing that annotators can exhibit levels of expertises across multiple domains. Thus, a separate reputation value is maintained for each class. Given K classes, each annotator a_j has K reputation values, one per class $c \in \{1, 2, ..., K\}$. The score of annotator a_j for class c at interaction step t is computed as:

$$\mathcal{S}_j^{(c,t)} = \frac{1}{N} \sum_{i=1}^{N} r_i^{(c,t)} \tag{7}$$

where N is the number of ratings received at time t for class c, and $r \in \{0, 1\}$. However, scores $\mathcal{S}_j$ alone can be insufficient to fully capture $\mathcal{R}$, since they are based only on ratings. Therefore, it is important to include the accuracy, Acc_j, of the annotators, because it allows us to incorporate ground truth information. The reputation $\mathcal{R}_j$ of annotator a_j for class c at time t is thus defined as:

$$\mathcal{R}_j^{(c,t)} = \alpha \cdot \mathcal{S}_j^{(c,t)} + \beta \cdot Acc_j^{(c,t)} \tag{8}$$

Regarding visibility [11], $\mathcal{R}$ is modeled as a *global property*: all annotators evaluate every other peer in the oracle and have access to each other's reputation scores. Modeling $\mathcal{R}$ as a *subjective property* would involve simulating scenarios where each annotator only has access to a limited amount of information about the others.

3.4. Oracle Decision

According to Y. Qian et al. [34], individuals with higher reputations in a given domain are more likely to provide correct answers in real-world scenarios. Building on this principle, a component of the proposed framework involves defining the mechanism by which the oracle determines the final label to be assigned to each queried instance. Two strategies have been proposed and studied in the literature: majority voting and weighted voting.

Majority voting, where each annotator contributes equally to the final decision, is one common aggregation methods [35, 36, 37]. However, this method assumes that all annotators are equally competent which is an assumption that often does not hold in practical settings. In particular, as noted

by V. Sheng et al. [38] and Y. Zhang et al. [39], the presence of low-quality or noisy annotators can introduce significant label noise, negatively affecting the model's training and generalization performance.

To solve these limitations, this study adopts a *weighted voting* approach, in which each annotator's vote is scaled according to their reputation score. In doing so, this allows the oracle to give more influence to reliable annotators. Besides, weighted voting has been validated in several works that explicitly model annotator trustworthiness over time, including D. Zhou et at. [40], P. Jin et al. [41], and Y. Zhang et al. [42]. As a result, it is possible to improve the oracle's ability to handle annotation noise and operate more effectively in AL scenarios by integrating reputation-weighted voting, especially when annotation budgets are limited and each labeled instance carries a significant impact for the learner. In the case of ties, when two or more classes receive the same total reputation score, we adopt a random selection strategy among the tied classes. Although one alternative could be to choose the class with the most votes (i.e., majority count), this may inadvertently favor quantity over quality, especially in early iterations where reputation estimates are still stabilizing [38].

4. Experimental Setup

This study evaluates a reputation-based AL strategy under three main conditions: (i) using three different reputation configurations; (ii) varying levels of annotator expertise; and (iii) different oracle sizes. For the first 10 classes of the *MNIST* [43], *MNIST-Fashion* [44], *EMNIST-Letters* [45] and *CIFAR-10* [46] datasets, we explore all combinations of these three factors.

The interaction between the oracle and the learner h follows the procedure described in section 3.4, in which the oracle selects the class whose total $\mathcal{R}$, summed across all voting annotators, is the highest. To isolate the effect of $\mathcal{R}$, we control the contribution of its two components, $\mathcal{S}$ (peer ratings) and Acc (past accuracy), using weighting parameters α and β, respectively. Three configurations are considered: (a) when $\alpha = 1.0$ and $\beta = 0.0$, $\mathcal{R}$ is based only on peer ratings; (b) when both weights are set to 0.5, the $\mathcal{R}$ combines $\mathcal{S}$ and *Acc* equally; and (c) when $\alpha = 0.0$ and $\beta = 1.0$, only annotator *Acc* is used to compute $\mathcal{R}$.

Besides, each annotator a_i is randomly assigned a single class $c \in \mathcal{Y}$ in which they are considered an expert, a designation that remains fixed throughout the experiments. Their labeling behavior is simulated using a *dirichlet* distribution to generate class-conditional probability vectors, forming the confusion matrix π^{a_i}. The *dirichlet* vector $\boldsymbol{\alpha}_c$ is configured such that the expert class receives a higher concentration parameter α_{exp}, while the remaining $K - 1$ classes receive a base value of 2.5. The level of expertise, *Low*, *Medium*, or *High*, determines the value of α_{exp}, which is calculated using the formula described previously, corresponding approximately to 40%, 70%, or 98% expected accuracy, acc_t.

Configuration	α_{exp}	acc_t
High Expertise	441.0	98%
Medium Expertise	21.0	70%
Low Expertise	6.0	40%

Table 1
Values of α_{exp} and corresponding expected accuracy for expertise levels.

The resulting *dirichlet* concentration matrices $\boldsymbol{\alpha}^{a_i} \in \mathbb{R}^{K \times K}$ describe the behavior of each annotator. Each row corresponds to a true class c, and the elements specify the *dirichlet* concentration parameters used to sample a probability distribution over the predicted labels. The matrix is constructed such that one randomly selected class receives the expert-level concentration α_{exp} and the others follow a uniform pattern with base values.

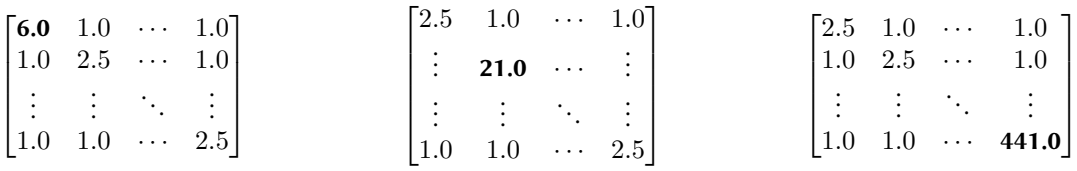

Figure 1: Concentration matrices ($K = 10$) for different expertises.

It is also relevant to study how the number of annotators (i.e., the oracle size) affects the performance of the reputation system. To this end, we include three oracle size configurations in the experiments: a small oracle composed of 5 annotators, a medium oracle with 15 annotators, and a large oracle consisting of 30 annotators. Lastly, the next table can resume all the variables and their configurations used in the research.

Variables	Configuration
Scenario	Pool-Based
Query Strategy	Uncertainty Sampling
Oracle Answer	Weighted Reputation Voting
Reputation Configuration	Only $\mathcal{S}$, Both $\mathcal{S}$ and *Acc*, Only *Acc*
Oracle Sizes	5, 15, 30
Expertise Levels	High (H), Medium (M), Low (L)

Table 2
Summary of the variables in the experiments.

4.1. Training Protocol

Training was conducted using the *Adam* optimizer due to its adoption in training CNNs such as *LeNet-5*, where it has been shown to achieve accurate results in image classification tasks [47, 48]. Regarding the learning rate, initial experiments tested the values 0.0001 and 0.01, which are commonly reported in the literature for *LeNet-5* training [47, 49, 50] and based on these results, a value of 0.002 was chosen. As for the batch size, preliminary tests revealed that smaller batches did not fully utilize the available GPU acceleration, resulting in data processing speeds comparable to CPU-bound training. Increasing the batch size allowed for better exploitation of the GPU's processing capabilities, with 256 as a value that ensured a better speed in training h.

Before the start of the AL cycles, the model h was pretrained on a small $labeledsubset$ corresponding to approximately 0.3% of the training data, which is a common practice in the literature, where initial labeled sets typically range from 0.1% to 2.5%, as reported by other authors [51, 52, 53]. Each AL cycle included exactly 10 training epochs, resulting in a total of up to 2.500 training epochs over the 250 iterations conducted throughout the experiments. Within each iteration, sample selection followed a pool-based strategy, in which 16 instances were selected from $\mathcal{U}$ based on the predictive uncertainty of h. The model h was trained using the *cross-entropy* loss function, which measures the discrepancy between the predicted probabilities and the true labels $y_i \in \mathcal{Y}$. This same loss was employed to evaluate h on the validation set after each iteration, and the final performance was calculated only at the end of the process using the test set.

Besides, each configuration was repeated 30 times, with the seed assigned according to the

run index, ranging from 0 to 29. Consequently, results are reported as the average and standard deviation computed across the 30 independent runs in which no early stopping was applied.

5. Results

5.1. MNIST-Digits

The next tables 3, 4, and 5 show the performance metrics obtained under the three different $\mathcal{R}$ configurations considering: only $\mathcal{S}$, *Acc* and $\mathcal{S}$, and both. As for table 6, it shows the averages of $\mathcal{R}$ for each configuration and oracle size.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Accuracy			
Baseline	0.57 ± 0.06	0.57 ± 0.06	0.57 ± 0.06
L	0.68 ± 0.04	0.88 ± 0.02	0.96 ± 0.01
M	0.70 ± 0.05	0.92 ± 0.02	0.98 ± 0.00
H	0.71 ± 0.04	0.93 ± 0.03	0.98 ± 0.00
Precision			
Baseline	0.57 ± 0.06	0.57 ± 0.06	0.57 ± 0.06
L	0.76 ± 0.03	0.89 ± 0.02	0.96 ± 0.01
M	0.77 ± 0.03	0.93 ± 0.02	0.98 ± 0.00
H	0.78 ± 0.03	0.94 ± 0.02	0.98 ± 0.00
F1-score			
Baseline	0.55 ± 0.06	0.55 ± 0.06	0.55 ± 0.06
L	0.68 ± 0.04	0.88 ± 0.02	0.96 ± 0.01
M	0.69 ± 0.06	0.92 ± 0.03	0.98 ± 0.00
H	0.70 ± 0.05	0.93 ± 0.03	0.98 ± 0.00

Table 3
Results for $\mathcal{R}$ using only $\mathcal{S}$ on *MNIST*.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Accuracy			
Baseline	0.57 ± 0.06	0.57 ± 0.06	0.57 ± 0.06
L	0.70 ± 0.05	0.89 ± 0.03	0.97 ± 0.01
M	0.71 ± 0.06	0.89 ± 0.03	0.97 ± 0.01
H	0.69 ± 0.05	0.85 ± 0.03	0.96 ± 0.02
Precision			
Baseline	0.57 ± 0.06	0.57 ± 0.06	0.57 ± 0.06
L	0.77 ± 0.03	0.90 ± 0.02	0.97 ± 0.01
M	0.79 ± 0.03	0.91 ± 0.02	0.97 ± 0.01
H	0.78 ± 0.03	0.88 ± 0.02	0.96 ± 0.02
F1-score			
Baseline	0.55 ± 0.06	0.55 ± 0.06	0.55 ± 0.06
L	0.69 ± 0.05	0.89 ± 0.03	0.97 ± 0.01
M	0.69 ± 0.06	0.89 ± 0.04	0.97 ± 0.01
H	0.67 ± 0.06	0.84 ± 0.03	0.96 ± 0.03

Table 4
Results for $\mathcal{R}$ using $\mathcal{S}$ and *Acc* on *MNIST*.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Accuracy			
Baseline	0.57 ± 0.06	0.57 ± 0.06	0.57 ± 0.06
L	0.70 ± 0.06	0.89 ± 0.03	0.96 ± 0.01
M	0.70 ± 0.05	0.88 ± 0.03	0.97 ± 0.02
H	0.69 ± 0.06	0.83 ± 0.08	0.95 ± 0.04
Precision			
Baseline	0.57 ± 0.06	0.57 ± 0.06	0.57 ± 0.06
L	0.77 ± 0.03	0.90 ± 0.02	0.96 ± 0.01
M	0.78 ± 0.03	0.90 ± 0.02	0.97 ± 0.01
H	0.78 ± 0.02	0.87 ± 0.04	0.95 ± 0.03
F1-score			
Baseline	0.55 ± 0.06	0.55 ± 0.06	0.55 ± 0.06
L	0.69 ± 0.07	0.88 ± 0.03	0.96 ± 0.01
M	0.68 ± 0.06	0.87 ± 0.04	0.97 ± 0.02
H	0.67 ± 0.06	0.81 ± 0.09	0.94 ± 0.04

Table 5
Results for $\mathcal{R}$ using only *Acc* on *MNIST*.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Only $\mathcal{S}$			
L	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
M	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
H	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
Both $\mathcal{S}$ and *Acc*			
L	0.12 ± 0.02	0.12 ± 0.02	0.12 ± 0.02
M	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
H	0.15 ± 0.02	0.15 ± 0.02	0.15 ± 0.02
Only *Acc*			
L	0.23 ± 0.04	0.23 ± 0.04	0.22 ± 0.04
M	0.26 ± 0.04	0.26 ± 0.04	0.25 ± 0.04
H	0.29 ± 0.03	0.28 ± 0.04	0.28 ± 0.04

Table 6
Reputation scores on *MNIST*.

5.2. MNIST-Fashion

For *MNIST-Fashion* dataset, the next tables 7, 8, and 9 present the performance metrics obtained under the three different $\mathcal{R}$ configurations. In contrast, Table 10 shows the average $\mathcal{R}$ values for each configuration and oracle size.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Accuracy			
Baseline	0.41 ± 0.12	0.41 ± 0.12	0.41 ± 0.12
L	0.56 ± 0.05	0.71 ± 0.02	0.79 ± 0.02
M	0.59 ± 0.04	0.75 ± 0.02	0.82 ± 0.01
H	0.59 ± 0.05	0.77 ± 0.03	0.84 ± 0.01
Precision			
Baseline	0.42 ± 0.12	0.42 ± 0.12	0.42 ± 0.12
L	0.64 ± 0.04	0.75 ± 0.02	0.82 ± 0.01
M	0.66 ± 0.03	0.78 ± 0.02	0.84 ± 0.01
H	0.66 ± 0.03	0.80 ± 0.02	0.85 ± 0.01
F1-score			
Baseline	0.38 ± 0.13	0.38 ± 0.13	0.38 ± 0.13
L	0.55 ± 0.05	0.71 ± 0.02	0.80 ± 0.01
M	0.57 ± 0.04	0.76 ± 0.02	0.83 ± 0.01
H	0.57 ± 0.06	0.78 ± 0.03	0.85 ± 0.01

Table 7
Results for $\mathcal{R}$ using $\mathcal{S}$ on *MNIST-Fashion*.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Accuracy			
Baseline	0.41 ± 0.12	0.41 ± 0.12	0.41 ± 0.12
L	0.60 ± 0.04	0.73 ± 0.04	0.82 ± 0.02
M	0.61 ± 0.05	0.75 ± 0.03	0.84 ± 0.02
H	0.60 ± 0.05	0.72 ± 0.03	0.83 ± 0.04
Precision			
Baseline	0.42 ± 0.12	0.42 ± 0.12	0.42 ± 0.12
L	0.65 ± 0.04	0.75 ± 0.03	0.82 ± 0.01
M	0.67 ± 0.04	0.76 ± 0.02	0.84 ± 0.02
H	0.68 ± 0.04	0.75 ± 0.02	0.84 ± 0.03
F1-score			
Baseline	0.38 ± 0.13	0.38 ± 0.13	0.38 ± 0.13
L	0.57 ± 0.05	0.73 ± 0.04	0.82 ± 0.02
M	0.58 ± 0.06	0.74 ± 0.04	0.83 ± 0.02
H	0.57 ± 0.06	0.70 ± 0.03	0.83 ± 0.04

Table 8
Results for $\mathcal{R}$ using $\mathcal{S}$ and Acc on *MNIST-Fashion*.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Accuracy			
Baseline	0.41 ± 0.12	0.41 ± 0.12	0.41 ± 0.12
L	0.59 ± 0.05	0.72 ± 0.05	0.81 ± 0.02
M	0.59 ± 0.06	0.74 ± 0.03	0.83 ± 0.02
H	0.58 ± 0.06	0.71 ± 0.03	0.82 ± 0.05
Precision			
Baseline	0.42 ± 0.12	0.42 ± 0.12	0.42 ± 0.12
L	0.66 ± 0.04	0.74 ± 0.03	0.81 ± 0.01
M	0.66 ± 0.06	0.76 ± 0.02	0.83 ± 0.02
H	0.66 ± 0.03	0.74 ± 0.02	0.82 ± 0.04
F1-score			
Baseline	0.38 ± 0.13	0.38 ± 0.13	0.38 ± 0.13
L	0.57 ± 0.05	0.71 ± 0.05	0.81 ± 0.02
M	0.57 ± 0.08	0.73 ± 0.04	0.83 ± 0.02
H	0.55 ± 0.07	0.69 ± 0.03	0.81 ± 0.05

Table 9
Results for $\mathcal{R}$ using only Acc on *MNIST-Fashion*.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Only $\mathcal{S}$			
L	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
M	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
H	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
Both $\mathcal{S}$ and *Acc*			
L	0.12 ± 0.02	0.12 ± 0.02	0.12 ± 0.02
M	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
H	0.15 ± 0.02	0.15 ± 0.02	0.15 ± 0.02
Only *Acc*			
L	0.22 ± 0.04	0.23 ± 0.04	0.23 ± 0.04
M	0.26 ± 0.04	0.26 ± 0.04	0.25 ± 0.04
H	0.28 ± 0.04	0.28 ± 0.04	0.28 ± 0.04

Table 10
Reputation scores on *MNIST-Fashion*.

5.3. EMNIST-Letters

The next tables 11, 12, and 13 display the performance metrics obtained under the three $\mathcal{R}$ configurations. Then, table 14 shows the average $\mathcal{R}$ values for each configuration and oracle size.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Accuracy			
Baseline	0.27 ± 0.11	0.27 ± 0.11	0.27 ± 0.11
L	0.49 ± 0.09	0.77 ± 0.05	0.90 ± 0.01
M	0.50 ± 0.14	0.86 ± 0.04	0.94 ± 0.01
H	0.52 ± 0.15	0.90 ± 0.04	0.96 ± 0.01
Precision			
Baseline	0.26 ± 0.15	0.26 ± 0.15	0.26 ± 0.15
L	0.58 ± 0.10	0.79 ± 0.04	0.91 ± 0.01
M	0.59 ± 0.17	0.87 ± 0.03	0.95 ± 0.01
H	0.60 ± 0.17	0.91 ± 0.03	0.96 ± 0.01
F1-score			
Baseline	0.22 ± 0.14	0.22 ± 0.14	0.22 ± 0.14
L	0.47 ± 0.11	0.77 ± 0.05	0.90 ± 0.01
M	0.47 ± 0.16	0.86 ± 0.04	0.94 ± 0.01
H	0.48 ± 0.17	0.90 ± 0.04	0.96 ± 0.01

Table 11
Results for $\mathcal{R}$ using only $\mathcal{S}$ on *EMNIST-Letters*.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Accuracy			
Baseline	0.27 ± 0.11	0.27 ± 0.11	0.27 ± 0.11
L	0.48 ± 0.17	0.78 ± 0.13	0.93 ± 0.01
M	0.54 ± 0.12	0.86 ± 0.05	0.95 ± 0.01
H	0.50 ± 0.13	0.86 ± 0.06	0.96 ± 0.02
Precision			
Baseline	0.26 ± 0.15	0.26 ± 0.15	0.26 ± 0.15
L	0.56 ± 0.20	0.79 ± 0.15	0.93 ± 0.01
M	0.64 ± 0.13	0.87 ± 0.04	0.95 ± 0.01
H	0.61 ± 0.18	0.87 ± 0.05	0.96 ± 0.02
F1-score			
Baseline	0.22 ± 0.14	0.22 ± 0.14	0.22 ± 0.14
L	0.45 ± 0.19	0.77 ± 0.15	0.93 ± 0.01
M	0.50 ± 0.13	0.86 ± 0.06	0.95 ± 0.01
H	0.46 ± 0.15	0.85 ± 0.07	0.96 ± 0.02

Table 12
Results for $\mathcal{R}$ using both $\mathcal{S}$ and Acc on *EMNIST-Letters*.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Accuracy			
Baseline	0.27 ± 0.11	0.27 ± 0.11	0.27 ± 0.11
L	0.50 ± 0.14	0.80 ± 0.04	0.93 ± 0.01
M	0.55 ± 0.11	0.85 ± 0.06	0.95 ± 0.02
H	0.53 ± 0.10	0.85 ± 0.06	0.95 ± 0.03
Precision			
Baseline	0.26 ± 0.15	0.26 ± 0.15	0.26 ± 0.15
L	0.59 ± 0.17	0.83 ± 0.02	0.93 ± 0.01
M	0.66 ± 0.13	0.86 ± 0.05	0.95 ± 0.02
H	0.65 ± 0.13	0.87 ± 0.04	0.95 ± 0.03
F1-score			
Baseline	0.22 ± 0.14	0.22 ± 0.14	0.22 ± 0.14
L	0.47 ± 0.16	0.80 ± 0.04	0.93 ± 0.01
M	0.52 ± 0.12	0.84 ± 0.07	0.95 ± 0.02
H	0.50 ± 0.12	0.84 ± 0.07	0.95 ± 0.03

Table 13
Results for $\mathcal{R}$ using only *Acc* on *EMNIST-Letters*.

	$\lvert A\rvert = 5$	$\lvert A\rvert = 15$	$\lvert A\rvert = 30$
Only $\mathcal{S}$			
L	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
M	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
H	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
Both $\mathcal{S}$ and *Acc*			
L	0.11 ± 0.02	0.11 ± 0.02	0.11 ± 0.02
M	0.12 ± 0.02	0.12 ± 0.02	0.12 ± 0.02
H	0.14 ± 0.02	0.14 ± 0.02	0.14 ± 0.02
Only *Acc*			
L	0.21 ± 0.03	0.21 ± 0.03	0.21 ± 0.04
M	0.24 ± 0.03	0.24 ± 0.03	0.23 ± 0.04
H	0.26 ± 0.03	0.26 ± 0.03	0.26 ± 0.04

Table 14
Reputation scores on *EMNIST-Letters*.

5.4. CIFAR-10

Tables 15, 16, and 17 show the performance obtained under the three $\mathcal{R}$ settings. Then, table 18 shows the average $\mathcal{R}$ values for each configuration and oracle size.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Accuracy			
Baseline	0.13 ± 0.01	0.13 ± 0.01	0.13 ± 0.01
L	0.18 ± 0.02	0.25 ± 0.02	0.31 ± 0.02
M	0.19 ± 0.03	0.27 ± 0.02	0.33 ± 0.02
H	0.19 ± 0.02	0.28 ± 0.02	0.34 ± 0.02
Precision			
Baseline	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
L	0.22 ± 0.03	0.28 ± 0.02	0.32 ± 0.01
M	0.23 ± 0.05	0.31 ± 0.02	0.36 ± 0.01
H	0.24 ± 0.05	0.33 ± 0.02	0.38 ± 0.01
F1-score			
Baseline	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
L	0.16 ± 0.02	0.24 ± 0.02	0.30 ± 0.02
M	0.16 ± 0.04	0.26 ± 0.02	0.33 ± 0.02
H	0.15 ± 0.03	0.27 ± 0.02	0.34 ± 0.02

Table 15
Results for $\mathcal{R}$ using only $\mathcal{S}$ on *CIFAR-10*.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Accuracy			
Baseline	0.13 ± 0.01	0.13 ± 0.01	0.13 ± 0.01
L	0.20 ± 0.02	0.27 ± 0.01	0.34 ± 0.01
M	0.20 ± 0.02	0.28 ± 0.02	0.35 ± 0.02
H	0.21 ± 0.02	0.29 ± 0.02	0.36 ± 0.03
Precision			
Baseline	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
L	0.24 ± 0.02	0.30 ± 0.01	0.35 ± 0.01
M	0.25 ± 0.02	0.32 ± 0.01	0.37 ± 0.01
H	0.26 ± 0.02	0.34 ± 0.02	0.38 ± 0.02
F1-score			
Baseline	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
L	0.18 ± 0.02	0.26 ± 0.01	0.33 ± 0.01
M	0.18 ± 0.02	0.28 ± 0.02	0.35 ± 0.02
H	0.18 ± 0.02	0.28 ± 0.02	0.36 ± 0.03

Table 16
Results for $\mathcal{R}$ using both $\mathcal{S}$ and Acc on *CIFAR-10*.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Accuracy			
Baseline	0.13 ± 0.01	0.13 ± 0.01	0.13 ± 0.01
L	0.20 ± 0.02	0.27 ± 0.01	0.34 ± 0.01
M	0.21 ± 0.02	0.28 ± 0.02	0.35 ± 0.01
H	0.21 ± 0.03	0.29 ± 0.02	0.36 ± 0.02
Precision			
Baseline	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
L	0.24 ± 0.02	0.30 ± 0.01	0.35 ± 0.01
M	0.24 ± 0.02	0.32 ± 0.02	0.37 ± 0.01
H	0.25 ± 0.03	0.34 ± 0.02	0.38 ± 0.02
F1-score			
Baseline	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
L	0.18 ± 0.02	0.27 ± 0.02	0.33 ± 0.01
M	0.18 ± 0.02	0.27 ± 0.02	0.35 ± 0.01
H	0.18 ± 0.03	0.28 ± 0.02	0.36 ± 0.03

Table 17
Results for $\mathcal{R}$ using only *Acc* on *CIFAR-10*.

	$\|A\| = 5$	$\|A\| = 15$	$\|A\| = 30$
Only $\mathcal{S}$			
L	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
M	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
H	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
$\mathcal{S}$ and *Acc*			
L	0.12 ± 0.02	0.12 ± 0.02	0.12 ± 0.02
M	0.13 ± 0.02	0.13 ± 0.02	0.13 ± 0.02
H	0.15 ± 0.02	0.15 ± 0.02	0.15 ± 0.02
Only *Acc*			
L	0.23 ± 0.03	0.23 ± 0.04	0.22 ± 0.04
M	0.26 ± 0.03	0.26 ± 0.04	0.25 ± 0.04
H	0.28 ± 0.03	0.28 ± 0.03	0.28 ± 0.04

Table 18
Reputation scores on *CIFAR-10*.

6. Discussion

In the *MNIST* dataset, the configuration that combines both $\mathcal{S}$ and *Acc* achieves the highest results across all metrics and $|A|$, particularly for annotators with L and M expertise. In comparison to the configuration that uses only *Acc*, the performance differences generally remain below 1.5%, and are more evident for L expertise, especially at $|A| = 15$ and $|A| = 30$. On the other hand, the configuration that considers only $\mathcal{S}$ shows lower results in all settings, with differences ranging from 1% to 2.5% relative to the other $\mathcal{R}$ configurations. Similarly, in *MNIST-Fashion*, the combination of $\mathcal{S}$ and *Acc* leads to the highest results, particularly for M and H expertise. Although the differences with the *Acc*-only configuration are generally below 2%, the $\mathcal{S}$-only configuration consistently produces lower values,

often with differences exceeding 3% to 5%. With regard to *EMNIST-Letters*, the use of *Acc* alone results in the best performance in most cases; however, the gap in comparison to the combined configuration remains below 1% in all scenarios. In contrast, using only $\mathcal{S}$ results in a reduced performance, especially for smaller $|A|$, where the differences frequently surpass 5%. Finally, in the *CIFAR-10* dataset, the configurations involving *Acc*, whether alone or in combination with $\mathcal{S}$, tend to outperform the $\mathcal{S}$-only approach. Nevertheless, the difference between the *Acc*-only and the combined configurations remains typically under 1%.

Across all reputation configurations, the oracle size ($|A|$) influences the final performance outcomes, as larger $|A|$ were systematically associated with better results across all experiments and datasets. The improvements observed when increasing $|A|$ from 5 to 30 ranged from approximately 12% to 18% in accuracy for *CIFAR-10*, 20% to 28% for *MNIST-Fashion*, 24% to 45% for *MNIST*, and up to 47% in *EMNIST-Letters*. The same trend is observed for expertise level, although the magnitude of improvement is less than the influence of $|A|$. In most cases, the increase from L to M falls within an interval of approximately 3-5%, while the improvement from M to H tends to be smaller and less significant, typically around 2%. Nonetheless, this progression is not always consistent since, in configurations that include *Acc*, a slight performance decrease, typically under 2%, is occasionally observed when moving from M to H expertise at $|A| = 5$, and sometimes at $|A| = 15$. This effect, however, is not present when $|A| = 30$.

In terms of reputation, the experiments showed that, in configurations including *Acc*, reputation values increase significantly with expertise level, by approximately 10%, but remain unaffected by changes in oracle size ($|A|$). On the contrary, in the configuration that considers only $\mathcal{S}$, reputation increases with oracle size by around 8%, but remains stable across different levels of expertise.

Finally, it is important to note that the results related to the annotators showed no variation, as they were generated in the same way using 30 different seeds, each corresponding to the index of the run being executed.

7. Conclusion

To conclude, this study began by simulating imperfect annotators using probabilistic confusion matrices sampled from *dirichlet* distributions. Each annotator was assigned a specific domain of expertise, modeled as a single class for which they had a higher likelihood of producing correct labels. To reflect different degrees of domain knowledge, three levels of expertise were defined, corresponding to expected accuracies of 40%, 70%, and 98%. Building on this setup, a reputation model was formulated to capture how reliable each annotator was in different domains. This model combined two components: the annotator's observed accuracy (*Acc*) for each class, and the ratings they received from others ($\mathcal{S}$), over time.

Whenever the learner queried a new instance, a group of annotators responded with their predictions and the final label was chosen through a weighted voting mechanism, where each vote was scaled according to the annotator's reputation for the queried class. In this way, the labeling process accounted not just for consensus but also for the reliability of each annotator, grounded in both their past performance and agreement with others.

Following this, results showed that combining $\mathcal{S}$ and *Acc* generally led to small increases in accuracy across all datasets when compared to using *Acc* alone. For instance, in the case of *MNIST*, the combined reputation outperformed *Acc* by approximately 0.7%, 0.9%, and 0.3% for oracle sizes $|A| = 5$, 15, and 30. In *MNIST-Fashion*, for $|A| = 5$ and 15, increases were about 0.9% and 1.3%, while the difference for $|A| = 30$ was smaller, roughly 0.8%. The *CIFAR-10* dataset showed improvements below 0.5% for all oracle sizes. In contrast, the *EMNIST-Letters* dataset exhibited benefits from the combined configuration, particularly with $|A| = 30$, where accuracy increased by nearly 1.5%. On

the other hand, relying only on $\mathcal{S}$ generally resulted in lower performance, especially when $|A| = 5$, accuracy dropped by around 1.5% in *MNIST*, 3% in *MNIST-Fashion*, more than 5% in *EMNIST-Letters*, and about 1.5% in *CIFAR-10*. These reductions tended to diminish as $|A|$ increased, but in most cases, performance still remained below that of *Acc* alone.

Moreover, results also show that both $|A|$ and annotator expertise influenced model performance. Specifically, increasing $|A|$ from 5 to 30 led to improvements ranging from 12% in *CIFAR-10* to as much as 47% in *EMNIST-Letters*, with the other datasets showing similar trends. Annotator expertise also had a positive impact: transitioning from L to M expertise resulted in accuracy gains of approximately 3% to 5%. However, moving from M to H expertise did not consistently lead to improvements. In scenarios with smaller oracles ($|A| = 5$ or 15), accuracy even declined by up to 2%. Nevertheless, this effect was no longer observed when $|A| = 30$.

For future work, there are many directions in which this study can be extended. One possibility is to explore different query strategies, such as margin sampling or entropy sampling, instead of using uncertainty sampling alone. Since, this would help to understand whether the reputation mechanism adapts well to different selection criteria. It would also be useful to test more levels of annotator expertise since, in this study, only three levels were used (Low, Medium, and High), but adding more levels (such as five instead of three) could reveal a more detailled pattern in how expertise influences performance. Another direction is to vary the oracle sizes beyond the three used here ($|A| = 5$, 15, 30), and to see whether the trends observed remain consistent. It may also be interesting to explore alternative ways of defining reputation itself. For example, instead of requiring that all annotators rate each other, one could study scenarios where only a subset of annotators provides ratings, or where a single designated expert is responsible for evaluating others.

Declaration on Generative AI

Generative AI tools, including Perplexity and Grammarly, were used in the preparation of this work.

References

[1] R. Monarch, R. Munro, Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI, Manning, 2021. URL: https://books.google.pt/books?id=LCh0zQEACAAJ.

[2] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, L. He, A survey of human-in-the-loop for machine learning, Future Generation Computer Systems 135 (2022) 364–381. URL: https://doi.org/10.1016/j.future.2022.05.014. doi:10.1016/j.future.2022.05.014.

[3] C. Chai, G. Li, Human-in-the-loop techniques in machine learning, IEEE Data Engineering Bulletin 43 (2020) 37–52.

[4] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, Ángel Fernández-Leal, Human-in-the-loop machine learning: A state of the art, Artificial Intelligence Review 56 (2023) 3005–3054. URL: https://doi.org/10.1007/s10462-022-10246-w. doi:10.1007/s10462-022-10246-w.

[5] B. Settles, Active Learning Literature Survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL: http://burrsettles.com/pub/settles.activelearning.pdf.

[6] B. Settles, From theories to queries: Active learning in practice, in: Proceedings of the Workshop on Active Learning and Experimental Design, volume 16 of *JMLR Workshop and Conference Proceedings*, JMLR, 2011, pp. 1–18. URL: http://proceedings.mlr.press/v16/settles11a.html.

[7] A. Tharwat, W. Schenck, A survey on active learning: State-of-the-art, practical challenges and research directions, Mathematics 11 (2023). URL: https://www.mdpi.com/2227-7390/11/4/820. doi:10.3390/math11040820.

[8] Y. Zhang, M. Lease, B. C. Wallace, Active discriminative text representation learning, 2016. URL: https://arxiv.org/abs/1606.04212. arXiv:1606.04212.

[9] S. P. Marsh, Formalising trust as a computational concept, 1994. PhD Thesis.

[10] M. Deutsch, Cooperation and trust: Some theoretical notes, in: M. R. Jones (Ed.), Nebraska Symposium on Motivation, University of Nebraska Press, Lincoln, NE, 1962, pp. 275–319.

[11] J. Sabater, C. Sierra, Review on computational trust and reputation models, Artificial Intelligence Review 24 (2005) 33–60. URL: https://link.springer.com/article/10.1007/s10462-004-0041-5. doi:10.1007/s10462-004-0041-5.

[12] J. Sabater, C. Sierra, Regret: A reputation model for gregarious societies, 2001. Technical Report.

[13] B. W. P. Hoelz, C. G. Ralha, Towards a cognitive meta-model for adaptive trust and reputation in open multi-agent systems, Autonomous Agents and Multi-Agent Systems 29 (2015) 1125–1156. URL: https://doi.org/10.1007/s10458-014-9278-9. doi:10.1007/s10458-014-9278-9.

[14] A. Jøsang, S. L. Presti, Analysing the relationship between risk and trust, ???? URL: https://www.josang.com/doc/JoLP2004-TrustRisk.pdf, available online. No publication details provided.

[15] I. Pinyol, J. Sabater-Mir, Computational trust and reputation models for open multi-agent systems: A review, Artificial Intelligence Review 40 (2013) 1–25. URL: https://doi.org/10.1007/s10462-011-9277-z. doi:10.1007/s10462-011-9277-z.

[16] F. M. Hafızoğlu, S. Sen, Reputation based trust in human-agent teamwork without explicit coordination, in: Proceedings of the 6th International Conference on Human-Agent Interaction (HAI 2018), Association for Computing Machinery, Inc, 2018, pp. 238–245. URL: https://doi.org/10.1145/3284432.3284454. doi:10.1145/3284432.3284454.

[17] G. Blank, W. H. Dutton, Age and trust in the internet: The centrality of experience and attitudes toward technology in britain, Social Science Computer Review 30 (2012) 135–151. URL: https://doi.org/10.1177/0894439310396186. doi:10.1177/0894439310396186.

[18] H. Du, M. Huhns, Determining the effect of personality types on human-agent interactions, in: Proceedings of the 2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2013), volume 2, 2013, pp. 239–244. URL: https://doi.org/10.1109/WI-IAT.2013.115. doi:10.1109/WI-IAT.2013.115.

[19] F. M. Hafızoğlu, S. Sen, Understanding the influences of past experience on trust in human-agent teamwork, ACM Transactions on Internet Technology 19 (2019) 1–22. URL: https://dl.acm.org/doi/10.1145/3324300. doi:10.1145/3324300.

[20] G. Haim, Y. K. Gal, M. Gelfand, S. Kraus, A cultural sensitive agent for human-computer negotiation, in: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1 (AAMAS '12), International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2012, pp. 451–458. URL: https://dl.acm.org/doi/10.5555/2343576.2343652.

[21] A. van Wissen, Y. Gal, B. A. Kamphorst, M. V. Dignum, Human–agent teamwork in dynamic environments, Computers in Human Behavior 28 (2012) 23–33. URL: https://www.sciencedirect.com/science/article/pii/S0747563211001610. doi:10.1016/j.chb.2011.08.006.

[22] A. Wissen, J. van Diggelen, V. Dignum, The effects of cooperative agent behavior on human cooperativeness, in: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), 2009, pp. 1179–1180. URL: https://doi.org/10.1145/1558109.1558200. doi:10.1145/1558109.1558200.

[23] T. D. Huynh, Trust and reputation in open multi-agent systems, 2006. PhD Thesis.

[24] B. Esfandiari, S. Chandrasekharan, On how agents make friends: Mechanisms for trust acquisition, 2001. URL: https://www.robocup.org.

[25] D. Gambetta, Can we trust trust?, in: D. Gambetta (Ed.), Trust: Making and Breaking Cooperative Relations, Blackwell, 1988, pp. 213–237.

[26] S. Chakraborty, Asking the right questions to the right users: Active learning with imperfect oracles, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 3365–3372. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5738. doi:10.1609/aaai.v34i04.5738.

[27] A. Calma, J. M. Leimeister, P. Lukowicz, S. Oeste-Reiß, T. Reitmaier, A. Schmidt, B. Sick, G. Stumme, K. Zweig, From active learning to dedicated collaborative interactive learning, in: Dagstuhl Manifestos, volume 5, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pp. 1–27. URL: https://drops.dagstuhl.de/opus/volltexte/2016/5552. doi:10.4230/DagMan.5.1.1.

[28] A. Calma, B. Sick, Simulation of annotators for active learning: Uncertain oracles, in: Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning @ ECML PKDD 2017, volume 1924 of *CEUR Workshop Proceedings*, 2017, pp. 49–58. URL: https://ceur-ws.org/Vol-1924/paper_6.pdf.

[29] A. Motro, P. Smets (Eds.), Uncertainty Management in Information Systems: From Needs to Solutions, Springer US, 1997. URL: https://link.springer.com/book/10.1007/978-1-4615-6245-0. doi:10.1007/978-1-4615-6245-0.

[30] R. Urner, S. Ben-David, O. Shamir, Learning from weak teachers, in: N. D. Lawrence, M. Girolami (Eds.), Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, volume 22 of *Proceedings of Machine Learning Research*, PMLR, La Palma, Canary Islands, 2012, pp. 1252–1260. URL: https://proceedings.mlr.press/v22/urner12.html.

[31] Y. Yan, R. Rosales, G. Fung, F. Farooq, B. Rao, J. Dy, Active learning from multiple knowledge sources, in: N. D. Lawrence, M. Girolami (Eds.), Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, volume 22 of *Proceedings of Machine Learning Research*, PMLR, La Palma, Canary Islands, 2012, pp. 1350–1357. URL: https://proceedings.mlr.press/v22/yan12.html.

[32] L. A. Hannah, D. M. Blei, W. B. Powell, Dirichlet process mixtures of generalized linear models, Journal of Machine Learning Research 12 (2011) 1923–1953.

[33] S. Kotz, N. Balakrishnan, N. L. Johnson, Continuous Multivariate Distributions, Volume 1: Models and Applications, 2nd ed., Wiley, 2000.

[34] Y. Qian, J. Tang, K. Wu, Weakly learning to match experts in online community, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 3841–3847. URL: https://doi.org/10.24963/ijcai.2018/534. doi:10.24963/ijcai.2018/534.

[35] A. P. Dawid, A. M. Skene, Maximum likelihood estimation of observer error-rates using the em algorithm, Journal of the Royal Statistical Society: Series C (Applied Statistics) 28 (1979) 20–28. doi:10.2307/2346806.

[36] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, in: J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems (NeurIPS), volume 23, Curran Associates, Inc., 2010, pp. 1297–1305. URL: https://proceedings.neurips.cc/paper/2010/hash/

0562f9c60cbe8f2f693713b6d8db8a0c-Abstract.html.

[37] J. Whitehill, T.-F. Wu, J. Bergsma, J. R. Movellan, P. Ruvolo, Whose vote should count more: Optimal integration of labels from labelers of unknown expertise, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems (NeurIPS), volume 22, Curran Associates, Inc., 2009, pp. 2035–2043. URL: https://proceedings.neurips.cc/paper/2009/hash/bb254fb1f5423896d5f9e3640212edbc-Abstract.html.

[38] V. S. Sheng, F. Provost, P. G. Ipeirotis, Get another label? improving data quality and data mining using multiple, noisy labelers, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2008, pp. 614–622. URL: https://doi.org/10.1145/1401890.1401965. doi:10.1145/1401890.1401965.

[39] Y. Zhang, D. Zhou, Q. Chen, M. I. Jordan, Spectral methods meet EM: A provably optimal algorithm for crowdsourcing, in: Advances in Neural Information Processing Systems (NeurIPS), volume 27, 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/53b6d32b5a3036c574b307e3e8912068-Paper.pdf.

[40] D. Zhou, J. C. Platt, S. Basu, Y. Mao, D. S. Modha, Learning from the wisdom of crowds by minimax entropy, in: Advances in Neural Information Processing Systems (NeurIPS), volume 25, 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/07176748b3e8a4f2b38c1641066c88ea-Paper.pdf.

[41] P. Jin, Y. Wang, Y. Liu, Z.-H. Zhou, Combining crowd and machine predictions for cost-effective learning, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), International Joint Conferences on Artificial Intelligence Organization, 2017, pp. 1634–1640. URL: https://doi.org/10.24963/ijcai.2017/226. doi:10.24963/ijcai.2017/226.

[42] Y. Zhang, Y. Zhang, Z.-H. Zhou, A survey of learning from crowds, ACM Computing Surveys (CSUR) 55 (2022) 1–36. URL: https://doi.org/10.1145/3520484. doi:10.1145/3520484.

[43] Y. LeCun, C. Cortes, C. J. C. Burges, Mnist handwritten digit database, http://yann.lecun.com/exdb/mnist, 2010. Accessed: 2024-12-11.

[44] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, CoRR abs/1708.07747 (2017). URL: http://arxiv.org/abs/1708.07747. arXiv:1708.07747.

[45] G. Cohen, S. Afshar, J. Tapson, A. Van Schaik, Emnist: Extending mnist to handwritten letters, 2017 International Joint Conference on Neural Networks (IJCNN) (2017). doi:10.1109/IJCNN.2017.7966217.

[46] A. Krizhevsky, V. Nair, G. Hinton, Cifar-10 (canadian institute for advanced research), http://www.cs.toronto.edu/~kriz/cifar.html, 2009. Accessed: 2024-12-11.

[47] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. El Ghaoui, M. I. Jordan, Theoretically principled trade-off between robustness and accuracy, in: Proceedings of the 36th International Conference on Machine Learning (ICML), volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 7472–7482. URL: https://proceedings.mlr.press/v97/zhang19p.html.

[48] T.-W. Weng, H. Zhang, P.-Y. Chen, X. Yi, L. Daniel, C.-J. Hsieh, H. Jin, Evaluating the robustness of neural networks: An extreme value theory approach, in: 6th International Conference on Learning Representations (ICLR), 2018. URL: https://openreview.net/forum?id=S17wWgW0-.

[49] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 39–57. URL: https://ieeexplore.ieee.org/document/7958570. doi:10.1109/SP.2017.49.

[50] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (ICLR) (2015). URL: https://arxiv.org/abs/1412.6980. arXiv:1412.6980.

[51] Y. Gal, R. Islam, Z. Ghahramani, Deep bayesian active learning with image data, in: D. Precup, Y. W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 1183–1192. URL: https://proceedings.mlr.press/v70/gal17a.html.

[52] O. Sener, S. Savarese, Active learning for convolutional neural networks: A core-set approach, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?

id=H1aIuk-RW.

[53] X. Zhan, X. Zhang, Y. Zhang, H. Xu, Z. Liu, A comparative study of uncertainty estimation methods in deep active learning for image classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 14674–14684. URL: https://doi.org/10.1109/CVPR52688.2022.01430. doi:10.1109/CVPR52688.2022.01430.