Collaborative Multi-view Learning from Crowds

Shanshan Si[1] **Liangxiao Jiang**[*1] **Wenjun Zhang**[2] **Chaoqun Li**[3] **Liangjun Yu**[4]

[1] School of Computer Science, China University of Geosciences, Wuhan 430074, China.
[2] School of Information Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China.
[3] School of Mathematics and Physics, China University of Geosciences, Wuhan 430074, China.
[4] Hubei Key Laboratory of Inteligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, China.

Abstract

As an effective post-processing approach to improving the quality of integrated labels, noise correction has received extensive attention in recent years. Recent studies have shown that utilizing information from both the original attribute view and the multiple noisy label view can significantly improve the performance of noise correction. However, most existing algorithms often learn information independently on each view, making them difficult to fully exploit the complementary information between these two views. To address this limitation, we propose a noise correction algorithm called collaborative multi-view learning from crowds (CMLFC). CMLFC first constructs instance correlation graphs on the original attribute view and the multiple noisy label view, respectively, and designs dedicated encoders to obtain view-specific representations. To capture the deep complementary information between these two views, it designs a contrastive fusion encoder to distill common representations from these two view-specific representations, and then forms a common view. Next, it trains three view-specific classifiers on these three views, and designs a joint loss function to collaboratively train these encoders and classifiers, enabling information to flow bidirectionally among views, thereby improving the quality of representations in each view. Finally, it employs the trained view-specific encoders and classifiers to obtain each instance's representations and class probability distributions, respectively, which are then used to update its integrated label. Extensive experiments on real-world crowdsourced datasets validate the effectiveness, ablation, and sensitivity of CMLFC. Our codes and datasets are available at https://github.com/jiangliangxiao/CMLFC.

[*]Corresponding author, Liangxiao Jiang <ljiang@cug.edu.cn>

1 INTRODUCTION

Supervised learning is a powerful machine learning paradigm that has achieved remarkable success across diverse tasks, and its success heavily relies on large-scale labeled data [Jiang et al., 2019, Zhang et al., 2023]. However, the traditional methods for obtaining labeled data by hiring domain experts or well-trained workers are often expensive and time-consuming, making it difficult to meet the growing demand for labeled data required by supervised learning [Tanno et al., 2019, Zhang et al., 2025b]. Therefore, the primary challenge in supervised learning is how to efficiently obtain large-scale labeled data.

Fortunately, the emergence of crowdsourcing has addressed this challenge, providing a cost-effective and time-saving way to obtain large-scale labeled data by hiring crowd workers [Zhang et al., 2024]. However, due to the lack of professional expertise and specialized training, the labels collected from crowd workers often contain noise [Wu et al., 2023, Zhang et al., 2025a]. To solve this problem, a method called repeated labeling is usually used, in which each instance is repeatedly labeled by multiple crowd workers to obtain a multiple noisy label set [Sheng et al., 2008]. After that, label integration is used to infer each instance's integrated label from its multiple noisy label set.

Label integration has attracted significant attention in recent years, and a large number of label integration algorithms have been proposed from various perspectives [Sheng et al., 2008, Zhang, 2022, Li et al., 2024]. Although label integration algorithms are often effective, there inevitably remains a certain degree of noise in the integrated labels. Here, noise means that the integrated label of an instance is not consistent with its true label [Li et al., 2016], and such noise will have a negative impact on the performance of the model trained on the integrated labels. Therefore, it is essential to reduce the impact of noise by noise correction.

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:6310–6320.

Over the past few years, numerous noise correction algorithms have been proposed to further improve the quality of integrated labels [Su et al., 2025]. Among them, polishing labels (PL), self-training correction (STC), and cluster-based correction (CC) [Nicholson et al., 2016] are classic algorithms designed solely based on the original attribute view. As research has progressed, researchers have found that relying exclusively on the original attribute view limits the performance of noise correction. Accordingly, they have begun using information from the multiple noisy label view to design algorithms, such as adaptive voting noise correction (AVNC) [Zhang et al., 2018], cross-entropy-based noise correction (CENC) [Xu et al., 2021], and label distribution similarity-based noise correction (LDSNC) [Ren et al., 2024]. More recently, researchers have recognized the effectiveness of multi-view learning in noise correction and have subsequently proposed dual-view noise correction (DVNC) [Ji et al., 2023] and multi-view-based noise correction (MVNC) [Li et al., 2023].

Although some algorithms attempt to utilize the information from both the original attribute view and the multiple noisy label view, they usually learn information independently on each view, rather than collaboratively utilizing the complementary information from both views. To address this limitation, we propose a noise correction algorithm called collaborative multi-view learning from crowds (CMLFC). In CMLFC, to fully exploit the deep complementary information between these two views, we design a contrastive fusion encoder to distill common representations and form a common view. Then, to achieve the goal of enabling information flow among views and improving the quality of representations in each view, we propose a joint loss function to collaboratively train these view-specific encoders and classifiers. In summary, the contributions of this paper are as follows:

- We design a contrastive fusion encoder to distill the common representations from the original attribute view and the multiple noisy label view, and subsequently form a common view that captures the deep complementary information between these two views.
- We construct a joint loss function to enable the information flow among the original attribute view, the multiple noisy label view, and the common view, facilitating mutual enhancement of each view's representations.
- We propose a novel noise correction algorithm called collaborative multi-view learning from crowds (CMLFC) to fully utilize the information from all three views and improve the quality of integrated labels.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 introduces the proposed CMLFC in detail. Section 4 reports the experiments and results. Section 5 concludes this paper and outlines the future research direction.

2 RELATED WORK

In recent years, numerous noise correction algorithms have been proposed to further improve the quality of integrated labels. Among them, PL, STC, and CC [Nicholson et al., 2016] are classic noise correction algorithms designed solely based on the original attribute view. PL first divides the dataset into several subsets and trains classifiers on these subsets. It then uses these classifiers to predict labels for each instance. Finally, the majority vote of the predicted labels for each instance is regarded as its integrated label. STC first uses a filter algorithm to divide the dataset into a clean set and a noise set, and then trains a classifier on the clean set to correct noise instances and add them to the clean set. Finally, it repeats this process until a certain proportion of noise instances are corrected and added to the clean set. CC is a cluster-based noise correction algorithm. It first clusters the dataset several times. For each cluster, it then assigns the same weights to instances within the same cluster. Finally, CC sums the weights obtained from each cluster to correct each instance's integrated label.

As research progressed, some researchers have noticed that relying solely on the original attribute view limits the performance of noise correction. Therefore, they have attempted to use the information from the multiple noisy label view to design noise correction algorithms. For example, AVNC [Zhang et al., 2018] first uses the multiple noisy label view to evaluate the quality of each crowd worker. It then estimates the proportion of noise instances in the dataset and filters them to obtain a clean set and a noise set. Finally, it trains several classifiers on the clean set to correct instances in the noise set. CENC [Xu et al., 2021] first uses the multiple noisy label view to calculate each instance's entropy. It then uses the calculated entropy to identify and filter noise instances. Finally, it calculates the cross-entropy to correct each noise instance. LDSNC [Ren et al., 2024] first uses the multiple noisy label view to calculate each instance's multiple noisy label distribution. It then measures the similarity between this distribution and the predicted label distribution of each instance to divide the dataset into a clean set and a noise set. Finally, it trains a classifier on the clean set to correct instances in the noise set.

More recently, researchers have recognized the effectiveness of multi-view learning in noise correction and have subsequently proposed DVNC [Ji et al., 2023] and MVNC [Li et al., 2023]. DVNC first independently trains two classifiers on the original attribute view and the multiple noisy label view, and then employs these two classifiers to divide the dataset into a clean set and a noise set. Finally, it trains a new classifier on the clean set to correct instances in the noise set. MVNC first divides the dataset into a clean set and a noise set. It then independently trains two classifiers on the clean set, using the original attribute view and the multiple noisy label view, respectively, and subsequently

uses them to correct instances in the noise set.

As discussed above, researchers have gradually recognized the limitations of underutilizing information and have attempted to utilize the information from both the original attribute view and the multiple noisy label view to design noise correction algorithms. However, they still tend to utilize information from each view independently and cannot fully utilize the complementary information between these two views. This observation motivates us to propose a more comprehensive algorithm that can capture the deep complementary information between these two views to further improve the quality of integrated labels. For this purpose, we propose a novel noise correction algorithm called collaborative multi-view learning from crowds (CMLFC). More details of CMLFC are presented in Section 3.

3 THE PROPOSED CMLFC

Before introducing our proposed CMLFC in detail, we first define the basic notations in crowdsourcing. A crowdsourced dataset can be represented as $D = \{(\boldsymbol{X}_i, \boldsymbol{L}_i)\}_{i=1}^{N}$, where $\boldsymbol{X}_i$ is the i-th instance in the dataset D, $\boldsymbol{L}_i$ is the multiple noisy label set of $\boldsymbol{X}_i$, and N is the number of instances. $\boldsymbol{X}_i$ can be represented as $\{x_{i1}, ..., x_{im}, ..., x_{iM}\}$, where M is the dimension of attributes, and x_{im} represents the value of the m-th attribute a_m of $\boldsymbol{X}_i$. $\boldsymbol{L}_i$ can be represented as $\{l_{ir}\}_{r=1}^{R}$, where R is the number of crowd workers. l_{ir} is the label annotated by crowd worker u_r and takes a value from a fixed set $\{-1, c_1, ..., c_q, ..., c_Q\}$. Q is the number of classes, c_q is the q-th class, and -1 means that u_r does not annotate $\boldsymbol{X}_i$. Then, a label integration algorithm is employed to infer the integrated label $\hat{y}_i$ of $\boldsymbol{X}_i$. And noise correction starts with the dataset $D = \{(\boldsymbol{X}_i, \boldsymbol{L}_i, \hat{y}_i)\}_{i=1}^{N}$.

In this paper, we aim to exploit the deep complementary information between the original attribute view and the multiple noisy label view to design a noise correction algorithm, thereby improving the quality of integrated labels. For this purpose, we propose a novel noise correction algorithm called collaborative multi-view learning from crowds (CMLFC). Figure 1 illustrates the overall framework of CMLFC. As shown in Figure 1, CMLFC mainly consists of four components: instance correlation graph construction, contrastive fusion, collaborative learning, and label updating. We will introduce these four components in the following sections.

3.1 INSTANCE CORRELATION GRAPH CONSTRUCTION

The first problem we need to solve is how to effectively utilize information from the original attribute view and the multiple noisy label view. To address this problem, CMLFC constructs instance correlation graphs for these two views, which explicitly model the relationships among instances from different perspectives, and leverages graph neural networks to obtain view-specific representations, thereby effectively utilizing information from these two views.

For the original attribute view, the instance correlation graph based on *K*-nearest neighbors is constructed to represent the geometric structure of the original attribute space. Specifically, for each instance $\boldsymbol{X}_i$, we first calculate the distances between it and all other instances. These distances are then sorted in ascending order to find the *K*-nearest neighbor set $\mathcal{N}_K^X(i)$ of $\boldsymbol{X}_i$, where *K* can be calculated by Eq. (1):

$$K = \delta \cdot \frac{N}{Q}, \tag{1}$$

where $\frac{N}{Q}$ is used to estimate the number of instances in each class, and δ is a parameter used to scale $\frac{N}{Q}$.

In CMLFC, we take advantage of the widely used Euclidean distance to calculate d_{ij}^X between instances $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, which can be calculated by Eq. (2):

$$d_{ij}^X = \sqrt{\sum_{m=1}^{M} (x_{im} - x_{jm})^2}. \tag{2}$$

To ensure the information can flow bidirectionally in the graph and achieve the full aggregation of information, we first construct a symmetrical graph through the union symmetrization operation. We connect $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ with an edge if $\boldsymbol{X}_j \in \mathcal{N}_K^X(i)$ or $\boldsymbol{X}_i \in \mathcal{N}_K^X(j)$. Furthermore, to accurately depict the similarity among different neighbors and enable the graph neural network to focus on important edges while suppressing noisy edges, we assign a weight to each edge based on its distance. The closer two instances are to each other, the higher the similarity between them is, and the higher the edge weight will be. Formally, the edge weight w_{ij}^X between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ can be calculated by Eq. (3):

$$w_{ij}^X = \frac{1}{d_{ij}^X + \epsilon}, \tag{3}$$

where ϵ is set to 10^{-4} to prevent division by zero.

Accordingly, we can obtain a weighted symmetric adjacency matrix $\boldsymbol{A}^X$ for the original attribute view, which can be calculated by Eq. (4):

$$A_{ij}^X = A_{ji}^X = \begin{cases} w_{ij}^X & \text{if } j \in \mathcal{N}_K^X(i) \text{ or } i \in \mathcal{N}_K^X(j) \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

For the multiple noisy label view, the attributes of instances in this view are composed of the labels collected from different crowd workers. Since these labels are discrete and different workers have different reliability, it is difficult to accurately measure the similarity between instances in this view. In theory, the same worker tends to assign the same

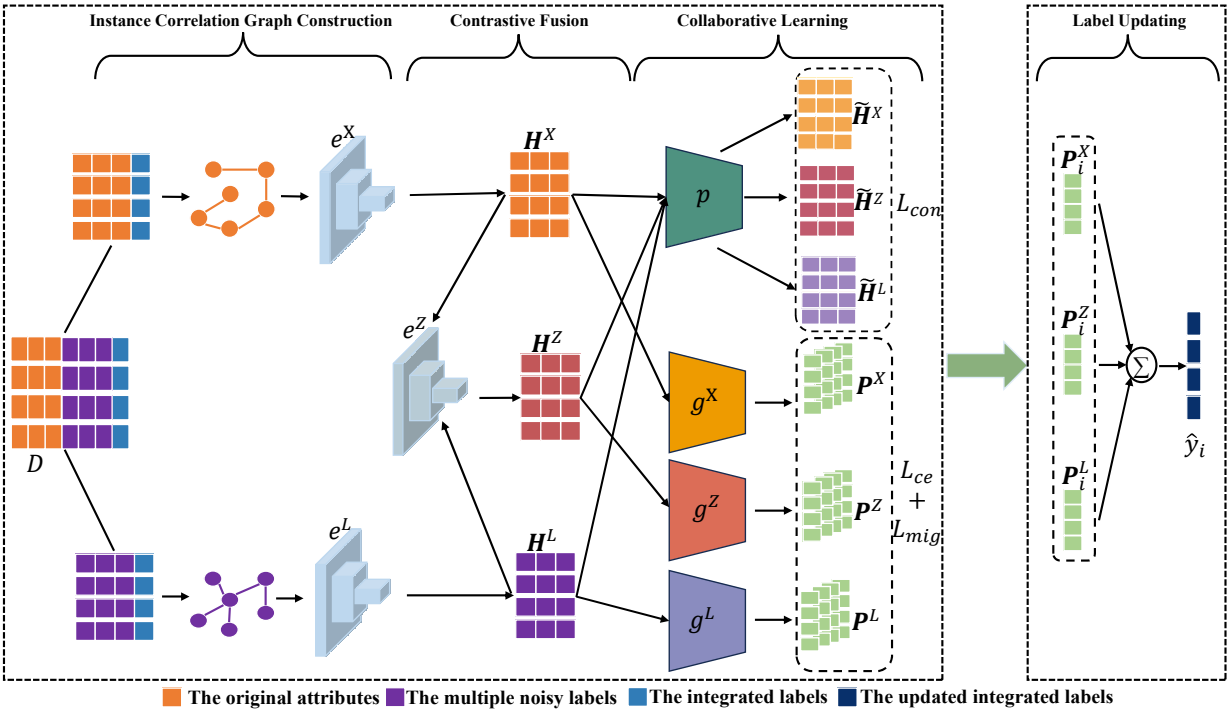


Figure 1: Overall framework of CMLFC.

labels to similar instances. If the majority of crowd workers assign the same labels to two instances, it indicates that these two instances have high similarity; conversely, if the majority of crowd workers assign different labels to them, it suggests that there are significant differences between these two instances. Therefore, we define worker quality-weighted label consistency to calculate the similarity between instances, and then construct an instance correlation graph on this view.

We first use the integrated label $\hat{y}_i$ to estimate the quality w_r of crowd worker u_r, which can be calculated by Eq. (5):

$$w_r = \frac{\sum_{i=1}^{N} I(l_{ir} = \hat{y}_i)}{\sum_{i=1}^{N} I(l_{ir} \neq -1)}, \tag{5}$$

where $I(\cdot)$ is an indicator function that returns 1 if its condition in parentheses is true, and returns 0 otherwise.

After obtaining the worker qualities, the similarity s_{ij}^L between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ can be calculated by the worker quality-weighted label consistency, which can be calculated bt Eq. (6):

$$s_{ij}^L = \begin{cases} 0 \text{ if } \sum_{r=1}^{R} (l_{ir} \neq -1 \wedge l_{jr} \neq -1) = 0 \\ \frac{\sum_{r=1}^{R} I(l_{ir}=l_{jr}) w_r}{\sum_{r=1}^{R} (l_{ir}\neq -1 \wedge l_{jr} \neq -1) w_r} \quad \text{otherwise} \end{cases}. \tag{6}$$

Similar to the original attribute view, the instance correlation graph based on K-nearest neighbors is constructed for the multiple noisy label view. Specifically, for each instance $\boldsymbol{X}_i$, we sort all other instances in descending order according to s_{ij}^L to find the K-nearest neighbor set $\mathcal{N}_K^L(i)$ for $\boldsymbol{X}_i$

Then, to ensure the information can flow bibdirectionally in the graph, we adopt the same union symmetrization operation as that used in the original attribute view. That is, an edge is connected between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ on the multiple noisy label view if $\boldsymbol{X}_j \in \mathcal{N}_K^L(i)$ or $\boldsymbol{X}_i \in \mathcal{N}_K^L(j)$. The weight of this edge is set to their worker quality-weighted label consistency s_{ij}^L. Accordingly, the weighted symmetric adjacency matrix $\boldsymbol{A}^L$ for the multiple noisy label view can be calculated by Eq. (7):

$$A_{ij}^L = A_{ji}^L = \begin{cases} s_{ij}^l & \text{if } j \in \mathcal{N}_K^L(i) \text{ or } i \in \mathcal{N}_K^L(j) \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

Since the labels collected from crowd workers are discrete, using them directly as initial representations is insufficient to capture the inherent relationships among labels. Therefore, these discrete labels are projected into a continuous space to capture the potential similarities among different labels, which can be calculated by Eq. (8):

$$\boldsymbol{L}_i' = \frac{\sum_{r=1}^{R} I(l_{ir} \neq -1) \boldsymbol{o}_{ir}}{\sum_{r=1}^{R} I(l_{ir} \neq -1)}, \tag{8}$$

where $\boldsymbol{o}_{ir}$ represents the one-hot encoding of l_{ir}, $\boldsymbol{L}_i'$ represents the initial representation of $\boldsymbol{X}_i$. The initial representations on this view can be denoted as $\boldsymbol{L}' = [\boldsymbol{L}_1^T, ..., \boldsymbol{L}_i^T, ..., \boldsymbol{L}_N^T]^T$.

Through the above steps, we can construct graphs $\mathcal{G}^X = (\boldsymbol{X}, \boldsymbol{A}^X)$ and $\mathcal{G}^L = (\boldsymbol{L}', \boldsymbol{A}^L)$ on the original attribute view

and the multiple noisy label view, respectively. Then, to enhance the representation ability of nodes, we need to convert the input vector into a high-dimensional representation. Specifically, we employ two-layer graph convolutional networks (GCNs) as view-specific encoders e^v to extract view-specific representations $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$ from the graphs $\mathcal{G}^X$ and $\mathcal{G}^L$, respectively, which can be represented as Eq. (9):

$$\boldsymbol{H}^v = e^v(\boldsymbol{A}^v, \boldsymbol{H}_0^v) = \sigma(\hat{\boldsymbol{A}}^v \sigma(\hat{\boldsymbol{A}}^v \boldsymbol{H}_0^v \boldsymbol{W}_1^v)\boldsymbol{W}_2^v), \quad (9)$$

where $\sigma(\cdot)$ is a ReLU function. For the v-th view, $\hat{\boldsymbol{A}}^v = (\tilde{\boldsymbol{D}}^v)^{-\frac{1}{2}} \tilde{\boldsymbol{A}}^v (\tilde{\boldsymbol{D}}^v)^{-\frac{1}{2}}$, $\tilde{\boldsymbol{A}}^v = \boldsymbol{A}^v + \boldsymbol{I}$ denotes the adjacency matrix augmented with self-loops, $\tilde{\boldsymbol{D}}^v$ is the degree matrix of $\tilde{\boldsymbol{A}}^v$, $\tilde{\boldsymbol{D}}_{ii}^v = \sum_j \tilde{\boldsymbol{A}}_{ij}^v$, $\boldsymbol{W}_1^v$ and $\boldsymbol{W}_2^v$ are the weight matrices of the first and second layers in e^v. The initial input $\boldsymbol{H}_0^v$ is set to $\boldsymbol{X}$ for the original attribute view and $\boldsymbol{L}'$ for the multiple noisy label view.

3.2 CONTRASTIVE FUSION

After resolving the problem of how to utilize information from the original attribute view and the multiple noisy label view, we now need to resolve how to effectively distill common representations from these two view-specific representations $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$, while preserving the discriminative power of each view-specific representations. A straightforward way to achieve this is to align $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$ into a shared representation space.

Inspired by contrastive learning [Ke et al., 2021, Liu et al., 2023], we design a contrastive fusion encoder to distill the common representations $\boldsymbol{H}^Z$ from $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$. The contrastive fusion encoder ensures that the common representations $\boldsymbol{H}^Z$ maintain semantic similarity with $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$. This not only establishes cross-view semantic alignment, but also ensures that $\boldsymbol{H}^Z$ comprehensively captures the critical information from both views. In CMLFC, we employ an MLP with two hidden layers as the contrastive fusion encoder e^Z, which can be represented as Eq. (10):

$$\boldsymbol{H}^Z = e^Z(\boldsymbol{H}_0^Z) = \boldsymbol{W}_2^Z \sigma(\boldsymbol{W}_1^Z \boldsymbol{H}_0^Z), \quad (10)$$

where $\boldsymbol{W}_1^Z$ and $\boldsymbol{W}_2^Z$ are the weight matrices of the first and second layer in e^Z, and $\boldsymbol{H}_0^Z$ is obtained by concatenating $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$, which can be expressed as $\boldsymbol{H}_0^Z = concat(\boldsymbol{H}^X, \boldsymbol{H}^L)$.

Following this, we employ a shared projection head p to map the view-specific representations $\boldsymbol{H}^X$, $\boldsymbol{H}^L$, and $\boldsymbol{H}^Z$ into a space where a contrastive loss is applied, which can be represented as Eq. (11):

$$\tilde{\boldsymbol{H}}^v = p(\boldsymbol{H}^v) = \boldsymbol{W}_2^p \sigma(\boldsymbol{W}_1^p \boldsymbol{H}^v), \quad (11)$$

where $\boldsymbol{W}_1^p$ and $\boldsymbol{W}_2^p$ are the weight matrices of the first and second layers in p. The input $\boldsymbol{H}^v$ is set to $\boldsymbol{H}^X$, $\boldsymbol{H}^L$, and $\boldsymbol{H}^Z$ for the original attribute view, the multiple noisy label view, and the common view, respectively.

3.3 COLLABORATIVE LEARNING

After distilling the common representations $\boldsymbol{H}^Z$ from the view-specific representations $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$, we design a joint loss function to enable information flow bidirectionally among these views, thereby improving the quality of representations in each view.

We extend the InfoNCE loss [Liu et al., 2022, Koromilas et al., 2024] to the crowdsourcing scenario and use it as the contrastive fusion objective, which can be calculated by Eq. (12):

$$\begin{aligned} L_{con} = &-\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(s\left(\tilde{\boldsymbol{H}}_i^Z, \tilde{\boldsymbol{H}}_i^X\right)/\tau\right)}{\sum_{j=1}^{N} \exp\left(s\left(\tilde{\boldsymbol{H}}_i^Z, \tilde{\boldsymbol{H}}_j^X\right)/\tau\right)} \\ &-\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\left(s\left(\tilde{\boldsymbol{H}}_i^Z, \tilde{\boldsymbol{H}}_i^L\right)/\tau\right)}{\sum_{j=1}^{N} \exp\left(s\left(\tilde{\boldsymbol{H}}_i^Z, \tilde{\boldsymbol{H}}_j^L\right)/\tau\right)} \end{aligned}, \quad (12)$$

where $s(\cdot,\cdot)$ is the cosine similarity function, $\tilde{\boldsymbol{H}}_i^X$, $\tilde{\boldsymbol{H}}_i^L$, and $\tilde{\boldsymbol{H}}_i^Z$ are the projected representations of instance $\boldsymbol{X}_i$ in the original attribute view, the multiple noisy label view, and the common view, respectively. τ is a temperature parameter, and we set τ to 0.1 in our experiments, as suggested in [Chen et al., 2020].

After obtaining these three view-specific representations, CMLFC trains three view-specific classifiers to obtain the class probability distributions of all instances. We use a single linear layer as view-specific classifiers g^v, which can be represented as Eq. (13):

$$\boldsymbol{P}^v = g^v(\boldsymbol{H}^v) = \text{softmax}(\boldsymbol{W}^{v,g}\boldsymbol{H}^v + \boldsymbol{b}^{v,g}), \quad (13)$$

where, for the v-th view, $\boldsymbol{W}^{v,g}$ and $\boldsymbol{b}^{v,g}$ denote the weight matrix and bias vector, $\boldsymbol{P}^v \in R^{N\times Q}$ is the class probability matrix, the i-th row $\boldsymbol{P}_i^v = \left[P_{i1}^v, ..., P_{iq}^v, ..., P_{iQ}^v\right]$ is the class probability distribution of $\boldsymbol{X}_i$. Then, to ensure that CMLFC obtains accurate integrated labels, we employ a cross-entropy loss L_{ce} to guide these view-specific classifiers, which can be defined as Eq. (14):

$$L_{ce} = -\frac{1}{3N}\sum_{i=1}^{N}\sum_{q=1}^{Q} \hat{y}_{iq}\left(\log(P_{iq}^X) + \log(P_{iq}^L) + \log(P_{iq}^Z)\right), \quad (14)$$

where P_{iq}^X, P_{iq}^L, and P_{iq}^Z are the probabilities that $\boldsymbol{X}_i$ is classified as class c_q by these three view-specific classifiers, respectively, $\hat{y}_{iq}$ is the one-hot encoding of $\hat{y}_i$ for class c_q. Specifically, if $\hat{y}_i = c_q$, then $\hat{y}_{iq} = 1$, otherwise $\hat{y}_{iq} = 0$.

To balance consistency and diversity in the prediction space, we introduce an f-mutual information gain (MIG^f) loss [Kong and Schoenebeck, 2018], which is defined as the difference between average consistency of predictions between views for the same instance and average consistency

between views for different instances. The f-mutual information gain MIG^f_{XL} between the original attribute view and multiple noisy label view can be calculated by Eq. (15):

$$MIG^f_{XL} = \frac{1}{N}\sum_{i=1}^{N}\partial f\left(\sum_{q=1}^{Q}\frac{P^X_{iq}\cdot P^L_{iq}}{p_q}\right) - \frac{1}{N(N-1)}\sum_{i\neq j} f^*\partial f\left(\sum_{q=1}^{Q}\frac{P^X_{iq}\cdot P^L_{jq}}{p_q}\right)^{'} \tag{15}$$

where p_q is the prior probability of class q, f is a convex function satisfying $f(1)=0$ and f^* is the Fenchel duality of f. Following Cao et al. [2019], we set $p_q=\frac{1}{Q}$ and $f(t)=t\log(t)$. Similarly, we can define the f-mutual information gain between the original attribute view and the common view, as well as that between the multiple noisy label view and the common view, denoted as MIG^f_{XZ} and MIG^f_{LZ}, respectively. Thus, the overall f-mutual information gain loss L_{mig} across all views can be defined as Eq. (16):

$$L_{mig} = -\frac{1}{3}\left(MIG^f_{XL}+MIG^f_{XZ}+MIG^f_{LZ}\right). \tag{16}$$

Finally, we can construct the final joint loss function L_{total} for CMLFC, which can be defined as Eq. (17):

$$L_{total} = L_{con}+L_{ce}+L_{mig}. \tag{17}$$

In CMLFC, the joint loss function L_{total} collaboratively trains these view-specific encoders and classifiers, enabling bidirectional information flow among the original attribute view, the multiple noisy label view, and the common view, thereby facilitating mutual enhancement of each view's representations.

3.4 LABEL UPDATING

After the collaborative learning, for each instance $\boldsymbol{X}_i$, CMLFC employs the trained view-specific encoders e^v to obtain its view-specific representations $\boldsymbol{H}^X_i$ and $\boldsymbol{H}^L_i$ from the instance correlation graphs $\mathcal{G}^X$ and $\mathcal{G}^L$, respectively. It then employs the trained contrastive fusion encoder e^Z to generate $\boldsymbol{H}^Z_i$ for $\boldsymbol{X}_i$. Next, it employs the trained view-specific classifiers g^v to obtain class probability distributions $\boldsymbol{P}^X_i$, $\boldsymbol{P}^L_i$, and $\boldsymbol{P}^Z_i$ for $\boldsymbol{X}_i$. Finally, these class probability distributions are used to update the integrated label of $\boldsymbol{X}_i$ by Eq. (18):

$$\hat{y}_i = \underset{c_q\in\{c_1,c_2,\ldots,c_Q\}}{\arg\max}\frac{1}{3}\left(P^X_{iq}+P^L_{iq}+P^Z_{iq}\right). \tag{18}$$

In summary, the entire learning procedure of CMLFC is described in **Algorithm 1**. In **Algorithm 1**, line 1 calculates the number of neighbors with a time complexity of $O(1)$. Lines 2 and 3 construct the instance correlation graph on the original attribute view with a time complexity of $O(N^2M+N^2logK+2NK)$. Lines 4-6 estimate the qualities of all crowd workers with a time complexity of $O(NR)$. Lines 7-9 construct the instance correlation graph on the multiple noisy label view with a time complexity of $O(NR+N^2R+N^2)$. Let the number of iterations be T and the representation dimension be d. Therefore, the time complexity of lines 10-20 is $O(T(N^2(d+Q)+N(M+Q)d+ND^2+NdQ))$. Lines 21-26 update the integrated label of each instance with a time complexity of $O(N^2d+N(M+Q)d+Nd^2+NdQ+NQ)$. In summary, the overall time complexity of CMLFC is $O(N^2M+N^2logK+2NK+NR+N^2R+N^2+T(N^2(d+Q)+N(M+Q)d+Nd^2+NdQ))$. If only the highest-order terms are considered, the total time complexity of CMLFC is $O(N^2(M+R)+TN^2(d+Q))$.

Algorithm 1: CMLFC

Input: $D=\{(\boldsymbol{X}_i,\boldsymbol{L}_i,\hat{y}_i)\}_{i=1}^N$ - a crowdsourced dataset with integrated labels, δ - a predefined parameter
Output: $\{\hat{y}_i\}_{i=1}^N$ - the updated integrated labels

1 Use δ to calculate K by Eq. (1);
2 Calculate $\boldsymbol{A}^X$ by Eqs. (2)-(4);
3 Construct graph $\mathcal{G}^X$ for the original attribute view;
4 **for** $r=1$ *to* R **do**
5 Calculate the weight w_r for worker u_r by Eq. (5);
6 **end**
7 Calculate $\boldsymbol{A}^L$ by Eqs. (6)-(7);
8 Calculate $\boldsymbol{L}'$ by Eq. (8);
9 Construct graph $\mathcal{G}^L$ for the multiple noisy label view;
10 **repeat**
11 Calculate $\boldsymbol{H}^X$ and $\boldsymbol{H}^L$ by Eq. (9);
12 Calculate $\boldsymbol{H}^Z$ by Eq. (10);
13 Calculate $\tilde{\boldsymbol{H}}^X$, $\tilde{\boldsymbol{H}}^L$, and $\tilde{\boldsymbol{H}}^Z$ by Eq. (11);
14 Calculate L_{con} by Eq. (12);
15 Calculate $\boldsymbol{P}^X$, $\boldsymbol{P}^L$, and $\boldsymbol{P}^Z$ by Eq. (13);
16 Calculate L_{ce} by Eq. (14);
17 Calculate L_{mig} by Eqs. (15)-(16);
18 Calculate L_{total} by Eq. (17);
19 Apply backpropagation on L_{total} to update parameters;
20 **until** L_{total} *no longer changes*;
21 **for** $i=1$ *to* N **do**
22 Calculate $\boldsymbol{H}^X_i$ and $\boldsymbol{H}^L_i$ for $\boldsymbol{X}_i$ by Eq. (9);
23 Calculate $\boldsymbol{H}^Z_i$ for $\boldsymbol{X}_i$ by Eq. (10);
24 Calculate $\boldsymbol{P}^X_i$, $\boldsymbol{P}^L_i$, and $\boldsymbol{P}^Z_i$ for $\boldsymbol{X}_i$ by Eq.(13);
25 Update $\hat{y}_i$ for $\boldsymbol{X}_i$ by Eq. (18);
26 **end**
27 Return $\{\hat{y}_i\}_{i=1}^N$;

Table 1: Descriptions of four real-world datasets.

Dataset	Instances	Attributes	Classes	Workers	Labels
Leaves	384	64	6	83	3840
Income	600	10	2	67	6000
Music	700	124	10	44	2946
Labelme	1000	512	8	59	2547

4 EXPERIMENTS AND RESULTS

4.1 EXPERIMENTAL SETUP

To validate the effectiveness of the proposed CMLFC, we compare it with eight noise correction algorithms, including PL, STC, CC [Nicholson et al., 2016], AVNC [Zhang et al., 2018], CENC [Xu et al., 2021], DVNC [Ji et al., 2023], MVNC [Li et al., 2023], and LDSNC [Ren et al., 2024] in terms of noise ratio. Here, the noise ratio is defined as the percentage of instances in the dataset whose integrated labels do not match their true labels. For these comparison algorithms, we use the existing implementations of PL, CC, STC, and AVNC provided by the Crowd Environment and its Knowledge Analysis (CEKA) [Zhang et al., 2015] platform, as well as the implementations of CENC, DVNC, MVNC, and LDSNC provided by their authors. We use the simple but effective label integration algorithm, majority voting (MV) [Sheng et al., 2008] as the benchmark algorithm to obtain the initial integrated label of each instance. The parameters of all these comparison algorithms are consistent with those specified in their original papers. For CMLFC, the parameter δ is set to 0.6.

Our experiments are conducted on four real-world crowdsourced datasets, including "Leaves", "Income", "Music", and "Labelme". These datasets are all collected from Amazon Mechanical Turk (AMT) platform, and they represent different crowdsourcing scenarios and requirements. The details of these four datasets are shown in Table 1. To facilitate constructing the instance correlation graph for the original attribute view, we first use the mode and the mean value to fill in the missing values of the nominal attributes and the numeric attributes, respectively. Each nominal attribute is transformed into a one-hot encodered vector and concatenated with the numeric attributes to obtain the processed attributes, and then all attributes are standardized.

4.2 EFFECTIVENESS ANALYSIS

To reduce the impact of randomness, each experiment is independently repeated 10 times, and the average noise ratio of each algorithm is used as the final result. The detailed experimental results on these four real-world datasets in terms of noise ratio are shown in Figure 2. Based on these experimental results, we can observe that CMLFC consistently outperforms the other noise correction algorithms on these four real-world datasets. Specifically, the noise ratios of CMLFC on "Leaves", "Income", "Music", and "Labelme"

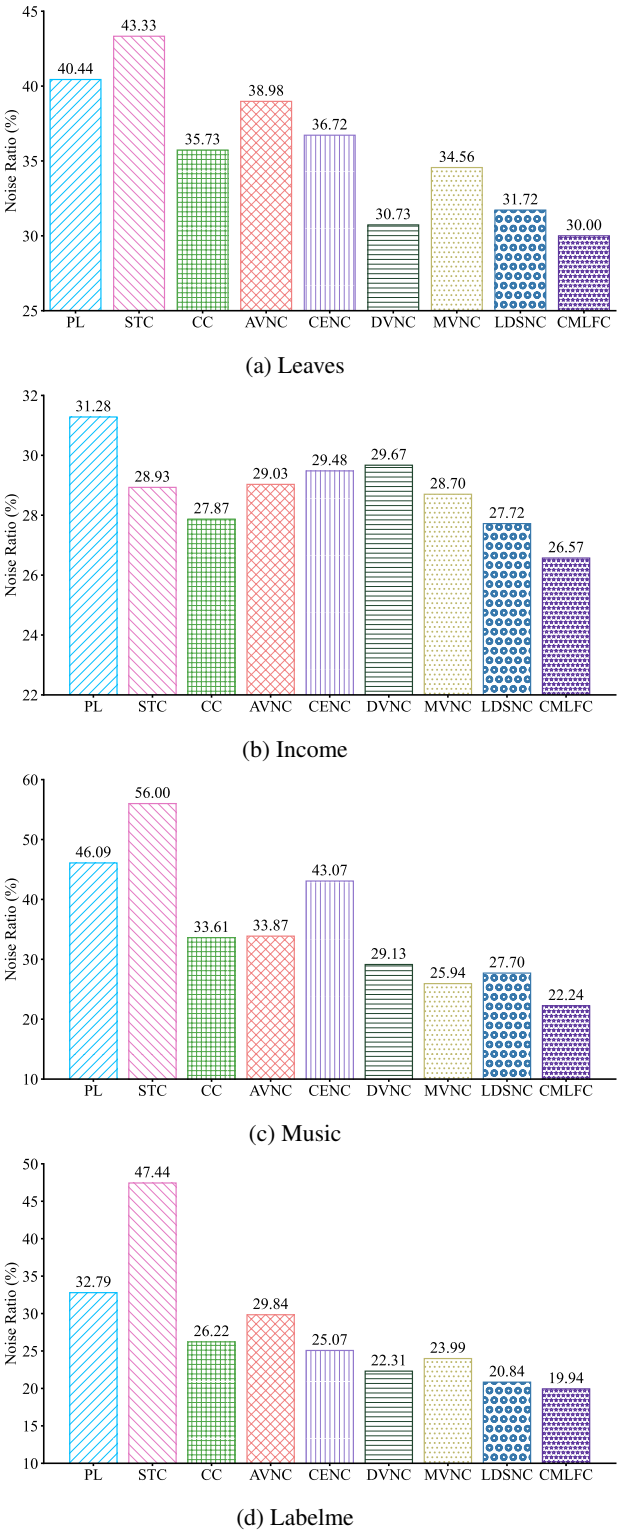
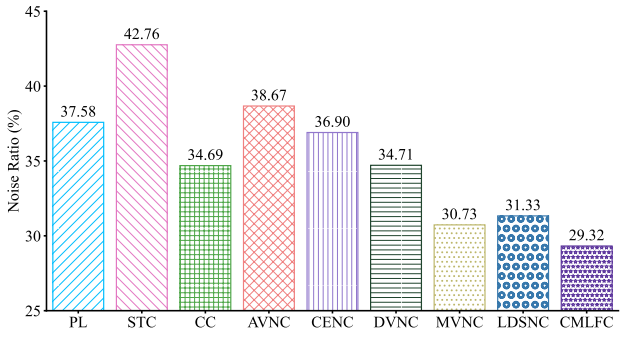
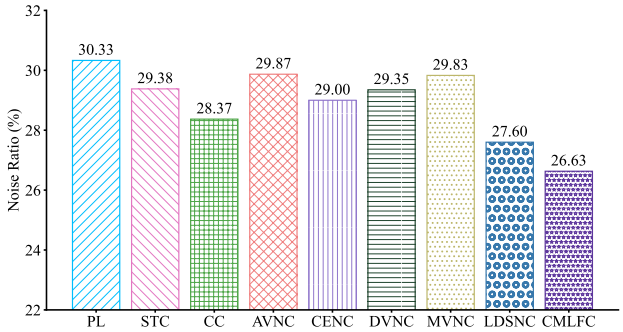


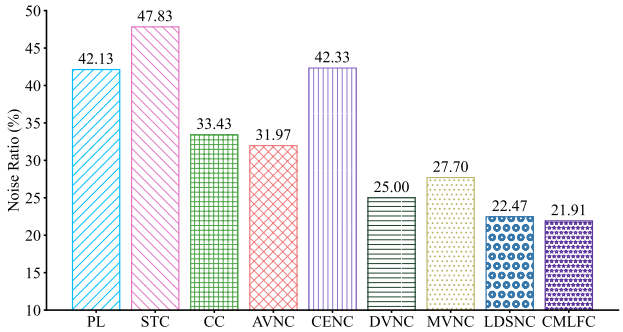
Figure 2: Noise ratio (%) comparisons based on MV for CMLFC versus PL, STC, CC, AVNC, CENC, DVNC, MVNC, and LDSNC on four real-world datasets.



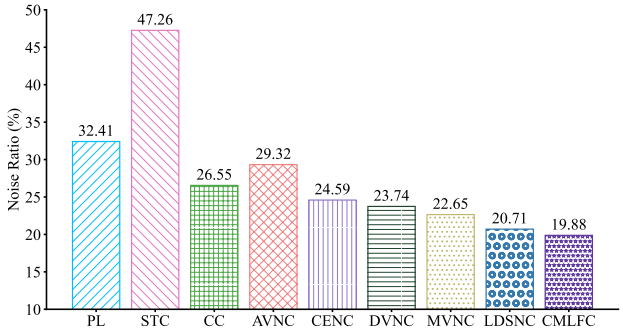
(a) Leaves



(b) Income



(c) Music



(d) Labelme

Figure 3: Noise ratio (%) comparisons based on ZC for CMLFC versus PL, STC, CC, AVNC, CENC, DVNC, MVNC, and LDSNC on four real-world datasets.

are 30.00%, 26.57%, 22.24%, and 19.94%, respectively, which are significantly lower than those of PL, STC, CC, AVNC, CENC, DVNC, MVNC, and LDSNC. These results indicate that CMLFC is able to effectively exploit the deep complementary information from the original attribute view and the multiple noisy label view to improve the quality of integrated labels.

To further validate the effectiveness of CMLFC, we conduct another series of experiments using another label integration algorithm. As one of the classic label integration algorithms, ZenCrowd (ZC) [Demartini et al., 2012] possesses a mature theoretical foundation and stable performance. Therefore, in the new experiments, ZC is used as the benchmark algorithm to obtain the initial integrated label of each instance. The detailed experimental results on these four real-world datasets are shown in Figure 3. As shown in Figure 3, we can observe that the noise ratios of CMLFC on "Leaves", "Income", "Music", and "Labelme" are 29.32%, 26.63%, 21.91%, and 19.88%, respectively, which are significantly lower than those of PL, STC, CC, AVNC, CENC, DVNC, MVNC, and LDSNC. Based on these experimental results, we can draw the same conclusion as when using MV as the benchmark algorithm, indicating that CMLFC is not sensitive to the label integration algorithms.

4.3 ABLATION ANALYSIS

In the above experiments, we have only validated the effectiveness of the whole CMLFC. To further verify the effectiveness of each component in CMLFC, we compare CMLFC with its two variants in terms of noise ratio. For simplicity and representativeness, we conduct this experiment on the multi-class dataset "Leaves" and the binary-class dataset "Income". The two variants are denoted as CMLFC-CF and CMLFC-CL, which remove the contrastive fusion and the collaborative learning from CMLFC, respectively. Figure 4 shows the detailed experimental results on the "Leaves" and "Income" datasets. As shown in Figure 4, we can observe that the performance becomes worse when any component is taken away. These results powerfully validate the rationality and effectiveness of each component in the proposed CMLFC.

To further analyze the effectiveness of the joint loss function in CMLFC, we compare CMLFC with its three variants in terms of noise ratio on the "Leaves" and "Income" datasets. The three variants of CMLFC are denoted as CMLFC-L_{con}, CMLFC-L_{mig}, and CMLFC-L_{ce}, which remove the constrastive loss L_{con}, the f-mutual information gain loss L_{mig}, and the cross entropy loss L_{ce} from the joint loss L_{total}, respectively. Figure 5 shows the experimental results on the "Leaves" and "Income" datasets. It can be observed that all these three variants perform worse than CMLFC, indicating that each part of the joint loss contributes positively to effectiveness of CMLFC. These results demonstrate the rationality and effectiveness of the joint loss function.

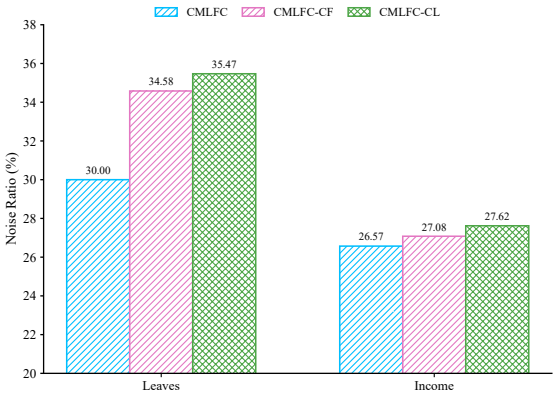


Figure 4: Noise ratio (%) comparisons for CMLFC versus CMLFC-CF and CMLFC-CL on the "Leaves" and "Income" datasets.

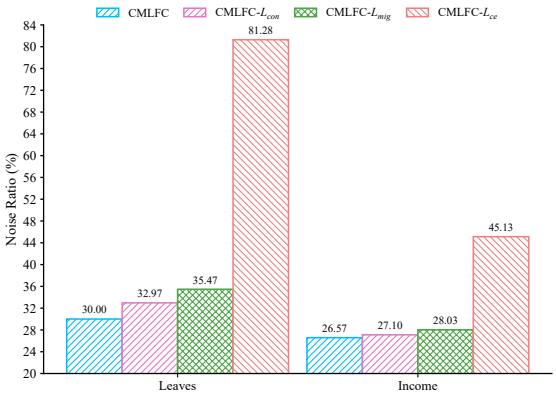


Figure 5: Noise ratio (%) comparisons for CMLFC versus CMLFC-L_{con}, CMLFC-L_{mig}, and CMLFC-L_{ce} on the "Leaves" and "Income" datasets.

4.4 SENSITIVITY ANALYSIS

In CMLFC, the value of δ determines the number of neighbors K. Theoretically, too large or too small δ will both harm the performance of CMLFC. If the δ is too large, edges may be created between irrelevant instances during the graph construction, resulting in the propagation of noisy information. If δ is too small, some important neighbors will be lost during the graph construction, making the graph structure incomplete. Therefore, we conduct a set of experiments to observe how the noise ratio of CMLFC changes as δ increases from 0.1 to 1.0 with a step size of 0.1 on the "Leaves" and "Income" datasets. Figure 6 shows the detailed experimental results. As shown in it, on the "Leaves" dataset, the noise ratio of CMLFC generally decreases as δ increases from 0.1 to 0.6 and reaches the lowest value of 30.00% at δ = 0.6. As δ increases from 0.6 to 1.0, the noise ratio of CMLFC increases, indicating that an excessively large value of δ may introduce more unreliable edges and thus degrade the performance of CMLFC. In contrast, on the "Income"

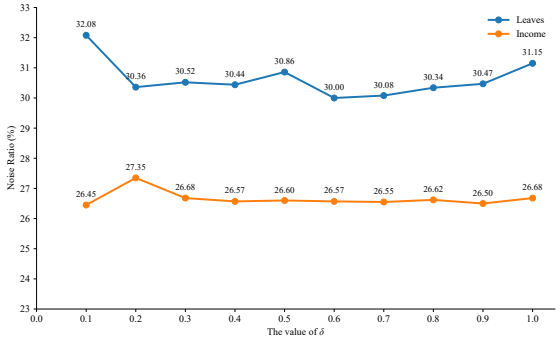


Figure 6: Noise ratio (%) comparisons for CMLFC when δ varies from 0.1 to 1.0 on the "Leaves" and "Income" datasets.

dataset, the noise ratio of CMLFC remains relatively stable across different values of δ. When the value of δ is 0.6, the noise ratio of CMLFC is relatively low. Therefore, by comprehensively considering the performance of CMLFC on both datasets, we set δ to 0.6 in all previous experiments.

5 CONCLUSION AND FUTURE WORK

This paper reveals that existing noise correction algorithms usually learn information separately from the original attribute view and the multiple noisy label view, which fail to fully exploit the deep complementary information between these two views. To address this limitation, we propose a novel noise correction algorithm called collaborative multi-view learning from crowds (CMLFC) to capture the deep complementary information between the original attribute view and the multiple noisy label view to improve the quality of integrated labels. The experimental results on real-world crowdsourced datasets strongly validate the effectiveness, rationality, and sensitivity of CMLFC.

However, CMLFC neglects the natural graphical structure between workers and instances in the crowdsourcing scenario. In future research, we aim to leverage techniques such as deep graph structure learning to fully utilize the correlation information between workers and instances, thereby further improving the quality of integrated labels.

Acknowledgements

The work was partially supported by the National Natural Science Foundation of China (62276241), the Open Project Program of Yunnan Key Laboratory of Intelligent Systems and Computing (ISC25Z01) and the Joint Open Fund of the Research Platforms of School of Computer Science, China University of Geosciences (Wuhan) (PTLH2025-B-07).

References

Peng Cao, Yilun Xu, Yuqing Kong, and Yizhou Wang. Maxmig: an information theoretic approach for joint learning from crowds. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In Alain Mille, Fabien Gandon, Jacques Misselis, Michael Rabinovich, and Steffen Staab, editors, *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 469–478. ACM, 2012.

Qiang Ji, Liangxiao Jiang, and Wenjun Zhang. Dual-view noise correction for crowdsourcing. *IEEE Internet Things J.*, 10(13):11804–11812, 2023.

Liangxiao Jiang, Lungan Zhang, Chaoqun Li, and Jia Wu. A correlation-based feature weighting filter for naive bayes. *IEEE Trans. Knowl. Data Eng.*, 31(2):201–213, 2019.

Guanzhou Ke, Zhiyong Hong, Zhiqiang Zeng, Zeyi Liu, Yangjie Sun, and Yannan Xie. CONAN: contrastive fusion networks for multi-view clustering. In Yixin Chen, Heiko Ludwig, Yicheng Tu, Usama M. Fayyad, Xingquan Zhu, Xiaohua Hu, Suren Byna, Xiong Liu, Jianping Zhang, Shirui Pan, Vagelis Papalexakis, Jianwu Wang, Alfredo Cuzzocrea, and Carlos Ordonez, editors, *2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, December 15-18, 2021*, pages 653–660, 2021.

Yuqing Kong and Grant Schoenebeck. Water from two rocks: Maximizing the mutual information. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 177–194. ACM, 2018.

Panagiotis Koromilas, Giorgos Bouritsas, Theodoros Giannakopoulos, Mihalis Nicolaou, and Yannis Panagakis. Bridging mini-batch and asymptotic analysis in contrastive learning: From infonce to kernel-based losses. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, 2024.

Chaoqun Li, Victor S. Sheng, Liangxiao Jiang, and Hongwei Li. Noise filtering to improve data and model quality for crowdsourcing. *Knowl. Based Syst.*, 107:96–103, 2016.

Jiao Li, Liangxiao Jiang, Xue Wu, and Wenjun Zhang. Learning from crowds with dual-view k-nearest neighbor. In Negar Kiyavash and Joris M. Mooij, editors, *Uncertainty in Artificial Intelligence, 15-19 July 2024, Universitat Pompeu Fabra, Barcelona, Spain*, volume 244 of *Proceedings of Machine Learning Research*, pages 2238–2249. PMLR, 2024.

Xinyang Li, Chaoqun Li, and Liangxiao Jiang. A multi-view-based noise correction algorithm for crowdsourcing learning. *Inf. Fusion*, 91:529–541, 2023.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.*, 35(1):857–876, 2023.

Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. Towards unsupervised deep graph structure learning. In Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 1392–1403. ACM, 2022.

Bryce Nicholson, Victor S. Sheng, and Jing Zhang. Label noise correction and application in crowdsourcing. *Expert Syst. Appl.*, 66:149–162, 2016.

Lijuan Ren, Liangxiao Jiang, Wenjun Zhang, and Chaoqun Li. Label distribution similarity-based noise correction for crowdsourcing. *Frontiers Comput. Sci.*, 18(5):185323, 2024.

Victor S. Sheng, Foster J. Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 614–622. ACM, 2008.

Bingrui Su, Liangxiao Jiang, and Shanshan Si. Confident learning-based noise correction for crowdsourcing. *Pattern Recognit.*, 169:111962, 2025.

Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C. Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11244–11253. Computer Vision Foundation / IEEE, 2019.

Gongqing Wu, Liangzhu Zhou, Jiazhu Xia, Lei Li, Xianyu Bao, and Xindong Wu. Crowdsourcing truth inference based on label confidence clustering. *ACM Trans. Knowl. Discov. Data*, 17(4):46:1–46:20, 2023.

Wenqiang Xu, Liangxiao Jiang, and Chaoqun Li. Improving data and model quality in crowdsourcing using cross-entropy-based noise correction. *Inf. Sci.*, 546:803–814, 2021.

Huan Zhang, Liangxiao Jiang, Wenjun Zhang, and Chaoqun Li. Multi-view attribute weighted naive bayes. *IEEE Trans. Knowl. Data Eng.*, 35(7):7291–7302, 2023.

Jing Zhang. Knowledge learning with crowdsourcing: A brief review and systematic perspective. *IEEE CAA J. Autom. Sinica*, 9(5):749–762, 2022.

Jing Zhang, Victor S. Sheng, Bryce Nicholson, and Xindong Wu. CEKA: a tool for mining the wisdom of crowds. *J. Mach. Learn. Res.*, 16:2853–2858, 2015.

Jing Zhang, Victor S. Sheng, Tao Li, and Xindong Wu. Improving crowdsourced label quality using noise correction. *IEEE Trans. Neural Networks Learn. Syst.*, 29(5): 1675–1688, 2018.

Jing Zhang, Ming Wu, Zeyi Sun, and Cangqi Zhou. Learning from crowds using graph neural networks with attention mechanism. *IEEE Trans. Big Data*, 11(1):86–98, 2025a.

Wenjun Zhang, Liangxiao Jiang, Ziqi Chen, and Chaoqun Li. FNNWV: farthest-nearest neighbor-based weighted voting for class-imbalanced crowdsourcing. *Sci. China Inf. Sci.*, 67(10), 2024.

Wenjun Zhang, Liangxiao Jiang, and Chaoqun Li. ELDP: enhanced label distribution propagation for crowdsourcing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(3):1850–1862, 2025b.