Robust Learning with Noisy Labels via Class-Subspace-Guided Contrastive Purification for Anomaly Detection in Industrial Data Analytics

Jiale Hong, Zidong Wang, Weibo Liu, Bo Shen, Jingzhong Fang, and Xiaohui Liu

***Abstract*—Deep learning models deployed in industry often rely on large-scale labeled data, where label noise is unavoidable due to imperfect sensing, human annotation errors, and complex operating environments. Such noisy labels can severely degrade model reliability and limit industrial applicability. To address this challenge, this paper proposes a contrastive-learning-based class-subspace-guided iterative purification (CLCSIP) framework for robust learning with noisy labels. Unlike conventional sample selection methods that depend on the small-loss assumption, the proposed approach exploits structural consistency between feature representations and class-specific subspaces as an alternative and more reliable criterion for identifying clean samples. A class-subspace-guided iterative purifier is developed to progressively refine class subspaces and filter noisy samples, thereby enhancing robustness under severe label corruption. Furthermore, a dual-loss supervised fine-tuning strategy, integrating classification and contrastive objectives, is introduced to improve discriminative representation learning. To avoid manual hyperparameter tuning and enhance adaptability across industrial scenarios, particle swarm optimization is employed to automatically balance the dual-loss components. Extensive experiments on CIFAR-10 and CIFAR-100 under both symmetric and asymmetric noise conditions demonstrate that the proposed framework consistently outperforms representative state-of-the-art methods, particularly at high noise levels. Moreover, the effectiveness and practical relevance of the proposed method are validated through an industrial anomaly detection task on real-world wire arc additive manufacturing data, where significant improvements in accuracy, precision, recall, and F1 score are achieved. These results indicate that the proposed framework provides a reliable and scalable solution for robust industrial intelligence in the presence of noisy labels.**

***Index Terms*—Learning with noisy labels, contrastive learning, sample purification, subspace-based classification, weakly supervised learning, particle swarm optimization.**

I. Introduction

Deep neural networks (DNNs) have achieved remarkable success in numerous classification tasks, largely owing to the availability of accurately labeled training data. However, in many practical applications, training data inevitably contain noisy labels, where a portion of the samples is assigned incorrect labels [4]. When DNNs are trained using data with noisy labels, the quality of learned feature representations is often compromised, which subsequently leads to degraded generalization performance of the resulting models [14], [27], [28].

To mitigate the adverse effects of noisy labels, a variety of approaches have been proposed in recent studies, including sample selection, label correction, and robust training strategies [20], [22], [26]. Sample selection aims to identify potentially clean samples for training DNNs [12], [19], whereas label correction attempts to amend incorrect labels in the training data. Robust training focuses on the design of loss functions or training strategies that are resilient to label noise. Among these approaches, sample selection has been widely adopted due to its reliable effectiveness and implementation simplicity. A commonly used strategy in sample selection methods is based on the small-loss assumption, which is motivated by the memorization behavior of DNNs, whereby clean samples tend to yield lower training losses than noisy labeled samples during training [8], [25], [29]. Representative algorithms based on this assumption, such as Co-teaching [8], Co-teaching plus [29], and JoCoR [25], have demonstrated satisfactory performance in handling noisy labels. In these methods, samples with relatively small training losses are collected to construct a potentially clean dataset for subsequent learning tasks.

Despite their widespread adoption in deep learning with noisy labels (DLNL), sample selection methods relying on the small-loss assumption suffer from several inherent limitations. First, the small-loss criterion is unreliable during the early stages of training, as its effectiveness only emerges after the model has been sufficiently trained [9]. Second, small-loss-based sample selection methods may exclude hard samples, namely correctly labeled samples associated with large training losses, which are crucial for learning complex decision boundaries [6]. These limitations indicate that reliance on the small-loss assumption alone is insufficient for achieving reliable sample selection in DLNL [10], [15].

To overcome the limitations of small-loss-based sample selection methods, an alternative criterion can be considered by examining the degree of alignment between sample representations and class-specific feature structures [5]. This idea is inspired by subspace-based classification, which assumes that

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) of the UK, the European Union's Horizon 2020 Research and Innovation Programme under Grant 820776 (INTEGRADDE), the National Natural Science Foundation of China under Grant 62273088, the Royal Society of the UK, and the Alexander von Humboldt Foundation of Germany.

Jiale Hong, Zidong Wang, Weibo Liu, Jingzhong Fang and Xiaohui Liu are with the Department of Computer Science, Brunel University of London, Uxbridge UB8 3PH, United Kingdom. (Email: Zidong.Wang@brunel.ac.uk).

Bo Shen is with the College of Information Science and Technology, Donghua University, Shanghai 201620, China and is also with the Engineering Research Center of Digitalized Textile and Fashion Technology, Ministry of Education, Shanghai 201620, China. (Email: bo.shen@dhu.edu.cn).

samples belonging to the same class lie in a low-dimensional subspace of the feature space, referred to as the class subspace [2]. Under this assumption, correctly labeled samples are expected to exhibit strong structural consistency with their corresponding class subspaces, whereas mislabeled samples are more likely to deviate from these structures. Accordingly, the structural consistency between a sample representation and its class subspace can be exploited as an alternative criterion for selecting potentially clean samples. Motivated by this insight, a novel sample selection method, termed the Class-Subspace-Guided Iterative Purifier (CSI-Purifier), is developed.

Building upon the proposed sample selection method, a contrastive learning-based class-subspace-guided iterative purification (CLCSIP) framework is further developed, in which the CSI-Purifier serves as a key component. The CLCSIP framework consists of four main modules: contrastive learning-based feature extraction, CSI-Purifier-based sample selection, dual-loss supervised fine-tuning, and particle swarm optimization (PSO)-based adaptive loss weighting. Contrastive learning-based feature extraction is employed to obtain discriminative feature representations without relying on annotated labels. The CSI-Purifier-based sample selection module is used to identify potentially clean samples. A dual-loss supervised fine-tuning strategy is adopted to jointly optimize classification and contrastive objectives, with the aim of improving robustness against noisy labels. Moreover, PSO-based adaptive loss weighting is introduced to automatically adjust the balance between classification and contrastive loss terms. The main contributions of this paper are summarized as follows.

1) A CSI-Purifier is proposed to select potentially clean samples by evaluating the structural consistency between sample representations and their corresponding class subspaces, thereby providing an alternative criterion for sample selection.
2) A dual-loss supervised fine-tuning strategy is developed to jointly optimize classification and contrastive objectives, aiming to enhance classification robustness against noisy labels.
3) PSO is employed to adaptively adjust the weight coefficient to properly balance the classification and contrastive loss terms, thereby mitigating the challenges caused by manual hyperparameter tuning.
4) The proposed CLCSIP framework is extensively validated through experiments on CIFAR-10 and CIFAR-100 datasets under various noise settings and is further applied to an anomaly detection task on real-world wire arc additive manufacturing (WAAM) data with noisy labels, demonstrating its effectiveness.

The remainder of this paper is organized as follows. The preliminaries are presented in Section II. The proposed CLCSIP framework is described in Section III. Experimental benchmark results are reported in Section IV. The application of the CLCSIP framework to WAAM anomaly detection is presented in Section V. Finally, concluding remarks are provided in Section VI.

II. PRELIMINARIES

A. Problem Formulation

Consider an M-class classification problem. The training data with noisy labels is denoted by $\widetilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{N}$. Each sample $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^{d_{\text{in}}}$ is associated with an observed label $\tilde{y}_i \in \mathcal{Y} = \{1, 2, \ldots, M\}$, which may be incorrect. Here, N is the number of training samples. The objective of this paper is to train a multi-class classifier $f_\theta : \mathcal{X} \to \mathbb{R}^M$, parameterized by θ, that is capable of accurately predicting ground-truth labels for unseen test samples in the presence of noisy labels in the training data.

B. Contrastive Learning

Contrastive learning has been widely adopted as an effective paradigm for learning discriminative feature representations [10], [15], [23]. SimCLR [1], [7], as a representative contrastive learning method, is designed to learn discriminative feature representations without relying on label information, making it well suited to scenarios involving noisy labels. SimCLR consists of an encoder network $f(\cdot)$ and a projection head $g(\cdot)$. Given a mini-batch of N samples, two different random augmentations are applied to each sample, resulting in $2N$ views. Each view is passed through the encoder to obtain a feature representation $\mathbf{z} = f(\mathbf{x}) \in \mathbb{R}^d$, which is subsequently mapped by the projection head to a lower-dimensional embedding $\mathbf{h} = g(\mathbf{z}) \in \mathbb{R}^{d'}$. In the resulting $2N$ views, each pair of augmented views originating from the same sample forms a positive pair, while all remaining view pairs within the batch are treated as negative pairs for a given view.

The objective of contrastive learning is to bring the embeddings of positive pairs closer together while pushing apart those of negative pairs in the embedding space. To achieve this objective, the contrastive loss employed in SimCLR [1] is defined as

$$\mathcal{L}_{\text{contrast}} = -\log \frac{\exp(\text{sim}(\mathbf{h}_i, \mathbf{h}_{i^+})/\tau)}{\sum_{k=1,\, k\neq i}^{2N} \exp(\text{sim}(\mathbf{h}_i, \mathbf{h}_k)/\tau)}, \quad (1)$$

where $\mathbf{h}_i$ and $\mathbf{h}_{i^+}$ are embeddings of a positive pair, $\mathbf{h}_k$ ($k \neq i$) represents all other embeddings in the batch, which are treated as negatives for $\mathbf{h}_i$. Here, τ is a temperature parameter and $\text{sim}(\cdot, \cdot)$ denotes cosine similarity.

C. Singular Value Decomposition

Singular value decomposition (SVD) is a fundamental matrix factorization technique that plays an important role in low-rank structure modeling and dimensionality reduction [11]. Given a real matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, the SVD of $\mathbf{A}$ is defined as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad (2)$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix whose non-negative diagonal entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_q > 0$ are the singular values of $\mathbf{A}$, with $q = \text{rank}(\mathbf{A})$.

Through SVD, a matrix can be decomposed into a sum of rank-one components, each corresponding to a singular value

and its associated singular vectors. By retaining only the top-r singular values and setting the remaining ones to zero, a low-rank approximation of the matrix $\mathbf{A}$ can be obtained as

$$\mathbf{A}_r = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top, \tag{3}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the ith columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. The approximation $\mathbf{A}_r$ is the rank-r matrix that minimizes the Frobenius norm $|\mathbf{A} - \mathbf{A}_r|_F$.

III. The proposed CLCSIP Framework

A. Framework

The developed CLCSIP framework is illustrated in Fig. 1 and is designed to train a neural-network-based classifier for supervised classification in the presence of noisy labels. The framework consists of four core modules: 1) contrastive learning-based feature extraction; 2) CSI-Purifier-based sample selection; 3) dual-loss supervised fine-tuning; and 4) PSO-based adaptive loss weighting.

The training procedure of CLCSIP begins with the contrastive learning-based feature extraction module, in which a feature extractor is trained using the SimCLR framework. The trained feature extractor is then applied to obtain feature representations for all training samples. These feature representations, together with their corresponding labels, are fed into the CSI-Purifier to perform sample selection, resulting in a set of potentially clean data. The selected data are subsequently used to jointly fine-tune the feature extractor and classifier under a dual-loss objective composed of cross-entropy and contrastive loss. Finally, PSO is employed to automatically determine the weighting coefficient used to balance the two loss components, with the objective of maximizing validation accuracy after fine-tuning.

B. Contrastive Learning-Based Feature Extraction

In the CLCSIP framework, the feature extractor is trained using the SimCLR framework introduced in Section II-B and is subsequently employed to extract feature representations $\mathbf{z}_i = f(\mathbf{x}_i) \in \mathbb{R}^d$ from all training samples, which are used as inputs for the subsequent modules (see Fig. 1).

C. CSI-Purifier-Based Sample Selection

The CSI-Purifier consists of three main phases: 1) class subspace construction; 2) pseudo-label assignment; and 3) clean sample selection. Let $\mathcal{Z} = \{(\mathbf{z}_i, \tilde{y}_i)\}_{i=1}^N$ denote the extracted features and their corresponding labels. For each class $c \in \{1, \ldots, M\}$, the set $\{\mathbf{z}_i \mid \tilde{y}_i = c\}$ of feature vectors with observed label c is arranged as row vectors to form

$$\mathbf{Z}_c = \left[\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)}, \cdots, \mathbf{z}_i^{(N_c)}\right]^\top \in \mathbb{R}^{N_c \times d}, \tag{4}$$

where each $\mathbf{z}_i^{(j)}$ corresponds to a sample with $\tilde{y}_i = c$, d is the dimensionality of the feature vector and N_c is the number of samples whose observed label is c.

In the first phase, a class subspace is constructed for each class. Specifically, for class c, the matrix $\mathbf{Z}_c$ is decomposed via SVD as

$$\mathbf{Z}_c = \mathbf{U}_c \mathbf{\Sigma}_c \mathbf{V}_c^\top, \tag{5}$$

where $\mathbf{U}_c \in \mathbb{R}^{N_c \times N_c}$, $\mathbf{\Sigma}_c \in \mathbb{R}^{N_c \times d}$ and $\mathbf{V}_c \in \mathbb{R}^{d \times d}$. By retaining the top-$r$ singular values in $\mathbf{\Sigma}_c$, along with the corresponding singular vectors in $\mathbf{U}_c$ and $\mathbf{V}_c$, a rank-r approximation of $\mathbf{Z}_c$ is obtained as

$$\mathbf{Z}_c \approx \mathbf{U}_c^{(r)} \mathbf{\Sigma}_c^{(r)} \mathbf{V}_c^{(r)\top}, \tag{6}$$

where $\mathbf{U}_c^{(r)} \in \mathbb{R}^{N_c \times r}$, $\mathbf{\Sigma}_c^{(r)} \in \mathbb{R}^{r \times r}$ and $\mathbf{V}_c^{(r)} \in \mathbb{R}^{d \times r}$ denote the truncated matrices formed by the top-r components. The class subspace for class c is thus given by

$$\mathcal{S}_c = \text{span}(\mathbf{V}_c^{(r)}), \tag{7}$$

where $\text{span}(\mathbf{V}_c^{(r)})$ denotes the linear subspace spanned by the columns of $\mathbf{V}_c^{(r)}$.

In the second phase, a pseudo-label is assigned to each sample. Specifically, for each feature vector $\mathbf{z}_i \in \mathbb{R}^d$, the projection onto the class subspace $\mathcal{S}_c$ is computed as

$$\hat{\mathbf{z}}_{i,c} = \mathbf{V}_c^{(r)} \mathbf{V}_c^{(r)\top} \mathbf{z}_i. \tag{8}$$

The similarity between $\mathbf{z}_i$ and $\hat{\mathbf{z}}_{i,c}$ is then computed as

$$s_{i,c} = \frac{\mathbf{z}_i^\top \hat{\mathbf{z}}_{i,c}}{\|\mathbf{z}_i\|_2 \cdot \|\hat{\mathbf{z}}_{i,c}\|_2}, \tag{9}$$

and the pseudo-label is assigned according to

$$\hat{y}_i = \underset{c \in \{1,\ldots,M\}}{\arg\max}\ s_{i,c}. \tag{10}$$

In the third phase, potentially clean samples are identified through an iterative refinement procedure. A sample is regarded as potentially clean if $\hat{y}_i = \tilde{y}_i$, indicating that the pseudo-label agrees with the observed label. Since the initial class subspaces are constructed from data containing noisy labels, the resulting pseudo-labels may be unreliable. To improve the reliability of sample selection, an iterative purification procedure is adopted. In each iteration, samples satisfying $\hat{y}_i = \tilde{y}_i$ are used to reconstruct class subspaces, and pseudo-labels are reassigned based on the updated subspaces. This process continues for a fixed number of rounds T or until convergence is achieved.

To illustrate the working mechanism of the CSI-Purifier, Fig. 2 presents a simplified example in a two-dimensional feature space, where the feature vector A of a sample is projected onto three class subspaces, denoted as S_{cat}, S_{dog}, and S_{car}. The similarity between A and each projection vector is then computed, resulting in three scores e_1, e_2 and e_3. The label corresponding to the class with the highest similarity score is assigned as the pseudo-label for the sample corresponding to feature vector A. Since e_1 corresponds to the car subspace and yields the highest similarity score, the pseudo-label assigned to this sample is the car class. More generally, a sample is considered potentially clean if the assigned pseudo-label matches the observed label. The complete implementation procedure of the CSI-Purifier is summarized in **Algorithm 1**.

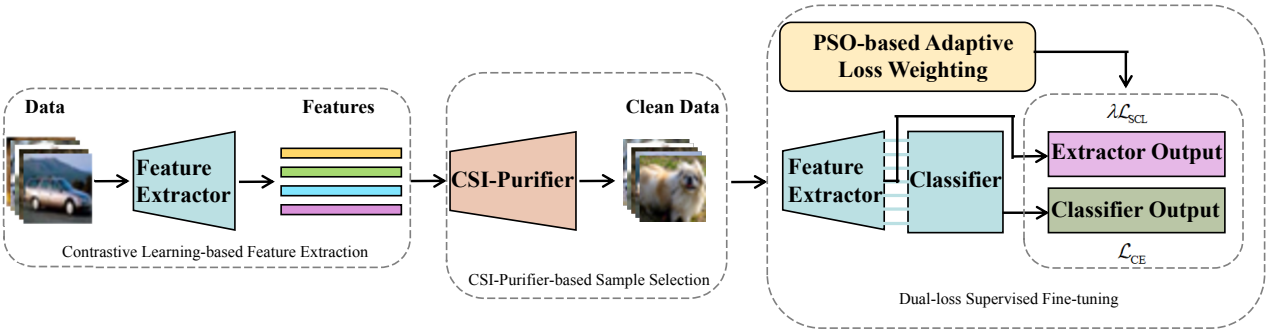


Fig. 1: Framework of CLCSIP.

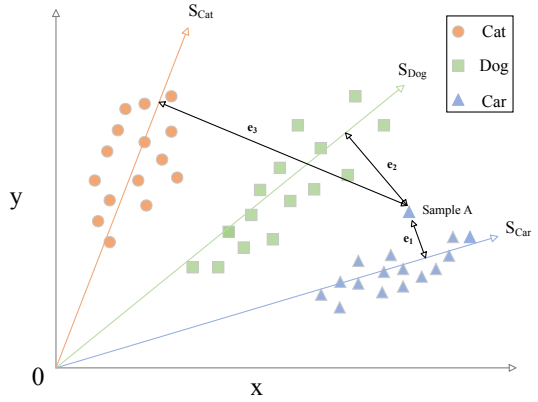


Fig. 2: Illustration of the CSI-Purifier. A represents the feature vector of a sample. The three arrows denoted as S_{cat}, S_{dog} and S_{car} represent the class subspaces constructed from the feature representations of samples labeled as cat, dog and car, respectively. The quantities e_1, e_2 and e_3 denote the similarity between feature vector A and its projection vectors onto the class subspaces S_{car}, S_{dog} and S_{cat}, respectively.

Remark 1: CSI-Purifier selects clean samples by evaluating the consistency between sample features and their corresponding class subspaces, thereby enabling effective identification of clean data in the presence of noisy labels. Two key advantages are provided. First, the class subspaces are iteratively refined during the purification process, which improves selection accuracy and ensures robustness even under high label noise rates. Second, since CSI-Purifier is decoupled from the training objective and operates solely on feature representations, it can be seamlessly integrated into a wide range of DLNL techniques.

D. Dual-Loss Supervised Fine-tuning

To facilitate the formulation of training losses, the input–output mapping of the network is specified. Given an input sample $\mathbf{x}_i \in \mathcal{X}$, the trained feature extractor $f : \mathcal{X} \to \mathbb{R}^d$ maps $\mathbf{x}_i$ to a feature representation $\mathbf{z}_i = f(\mathbf{x}_i)$, where $\mathbf{z}_i \in \mathbb{R}^d$. A classification head $g : \mathbb{R}^d \to \mathbb{R}^M$ is then applied to obtain the predicted probability vector $\hat{\mathbf{y}}_i = g(\mathbf{z}_i) \in \mathbb{R}^M$, where M denotes the number of classes. The resulting classifier is denoted by $f_\theta = g \circ f$.

The cross-entropy loss is defined as

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{M}\tilde{y}_{ij}\log(\hat{y}_{ij}), \tag{11}$$

Algorithm 1 Implementation Procedure of CSI-Purifier

Input: Noisy labeled data $\tilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^N$; trained feature extractor $f(\cdot)$; dimension r; iteration count T
Output: Clean subset $\tilde{\mathcal{D}}_{\text{clean}}$

Extract features $\mathbf{z}_i \leftarrow f(\mathbf{x}_i)$ and form $\mathcal{Z} = \{(\mathbf{z}_i, \tilde{y}_i)\}_{i=1}^N$
Initialize clean subset: $\mathcal{S} \leftarrow \tilde{\mathcal{D}}$
for $t = 1$ to T **do**
  `// Phase 1: Class subspace construction`
  for each class c **do**
    Collect $\{\mathbf{z}_i \mid (\mathbf{x}_i, \tilde{y}_i) \in \mathcal{S}, \tilde{y}_i = c\}$
    Compute $\mathbf{V}_c^{(r)}$ as in Equation (5) and Equation (6)
  end for
  `// Phase 2: Pseudo-label assignment`
  for $i = 1$ to N **do**
    for $c = 1$ to M **do**
      Compute $\hat{\mathbf{z}}_{i,c}$ using Equation (8)
      Compute similarity $s_{i,c}$ using Equation (9)
    end for
    Assign pseudo-label $\hat{y}_i$ using Equation (10)
  end for
  `// Phase 3: Clean sample selection`
  $\mathcal{S} \leftarrow \{(\mathbf{x}_i, \tilde{y}_i) \in \tilde{\mathcal{D}} \mid \hat{y}_i = \tilde{y}_i\}$
end for
return $\tilde{\mathcal{D}}_{\text{clean}} \leftarrow \mathcal{S}$

where B represents the number of samples in the current mini-batch, $\tilde{y}_{ij}$ denotes the one-hot encoding of the observed label for sample i and $\hat{y}_{ij}$ is the predicted probability of class j for sample i.

The contrastive loss [1] is defined as

$$\mathcal{L}_{\text{SCL}} = -\frac{1}{B}\sum_{i=1}^{B}\frac{1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(\mathbf{z}_i^\top\mathbf{z}_p/\tau)}{\sum_{j\neq i}\exp(\mathbf{z}_i^\top\mathbf{z}_j/\tau)}, \tag{12}$$

where B represents the number of samples in the current mini-batch, $\mathbf{z}_i = f(\mathbf{x}_i) \in \mathbb{R}^d$ is the feature representation of sample $\mathbf{x}_i$, $\mathbf{z}_p$ and $\mathbf{z}_j$ are the feature representations of samples $\mathbf{x}_p$ and $\mathbf{x}_j$ in the current mini-batch, $P(i)$ is the set of indices in the current mini-batch that share the same class label as i (excluding i itself), and τ is a temperature hyperparameter.

The overall training loss is defined as a weighted combination of the cross-entropy and contrastive losses:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{SCL}}, \tag{13}$$

Algorithm 2 Training Procedure of the CLCSIP Framework

Input: Noisy dataset $\tilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{N}$; pretraining epochs E_p; purification rounds T; classifier training epochs E; class subspace dimension r; batch size B; loss weight λ
Output: Trained classifier f_θ

`//Feature extraction`
Train encoder $f : \mathcal{X} \to \mathbb{R}^d$ using SimCLR on $\tilde{\mathcal{D}}$ for E_p epochs
`// Sample selection`
Extract features: $\mathbf{z}_i = f(\mathbf{x}_i)$ and form $\mathcal{Z} = \{(\mathbf{z}_i, \tilde{y}_i)\}_{i=1}^{N}$
Apply CSI-Purifier (Algorithm 1) to obtain clean data $\mathcal{S}$
`// Classifier fine-tuning`
Construct $f_\theta = g \circ f$, where g is a randomly initialized classification head
for $e \leftarrow 1$ to E **do**
  for each mini-batch $\{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{B} \subset \mathcal{S}$ **do**
    Compute total loss by (13)
    Update parameters via backpropagation
  end for
end for
return final trained classifier f_θ

where $\lambda > 0$ is a weighting coefficient that controls the relative contribution of the contrastive loss. The complete training procedure of the CLCSIP framework is summarized in **Algorithm 2**.

E. Particle Swarm Optimization-Based Adaptive Loss Weighting

The loss weighting coefficient λ plays a critical role in balancing the contributions of the cross-entropy and contrastive objectives. The total loss is formulated as a weighted sum of the two components, with $\lambda > 0$ controlling the relative strength of the contrastive loss. Manual tuning of the coefficient λ is inefficient and lacks generalizability across different data sets. To address this limitation, the determination of the optimal loss weighting coefficient is formulated as a black-box optimization problem and is solved using PSO [3], [16]. The objective is to identify the optimal coefficient λ^* that maximizes the classification performance of the CLCSIP framework when trained using the dual-loss defined in Equation (13).

The optimization procedure is summarized in **Algorithm 3**. Let $\mathcal{F}(\lambda)$ denote the validation accuracy obtained by training the CLCSIP framework for a fixed number of epochs with a given weighting coefficient λ. A population of P candidate coefficients $\{\lambda_j\}_{j=1}^{P}$ and their velocities are randomly initialized (line 1). Each λ_j is then evaluated by computing $\mathcal{F}(\lambda_j)$, based on which the initial personal bests and the global best are initialized (lines 2-7). Subsequently, during each of the T_s iterations, the velocity and position of each λ_j are updated using the standard PSO equations (line 10). The updated coefficients are then re-evaluated by computing $\mathcal{F}(\lambda_j)$, and the personal and global bests are updated accordingly (lines 11-17). Finally, the coefficient corresponding to the highest validation accuracy observed across all iterations is returned as the optimal solution (line 20).

Algorithm 3 Procedure of PSO-Based Adaptive Loss Weighting

Input: Number of particles P; number of PSO iterations T_s;
Output: Optimal loss weighting coefficient λ^*

1: Initialize particle positions $\{\lambda_j\}_{j=1}^{P}$ and velocities $\{v_j\}_{j=1}^{P}$ randomly
2: **for** each particle λ_j **do**
3: Train CLCSIP using λ_j for a fixed number of epochs
4: Compute validation accuracy $\mathcal{F}(\lambda_j)$
5: Set personal best $\text{pBest}_j \leftarrow \lambda_j$
6: **end for**
7: Identify global best $\text{gBest} \leftarrow \arg\max_{\lambda_j} \mathcal{F}(\lambda_j)$
8: **for** $t = 1$ to T_s **do**
9: **for** each particle λ_j **do**
10: Update velocity and position using standard PSO [16] rule
11: Evaluate $\mathcal{F}(\lambda_j)$ by training CLCSIP
12: **if** $\mathcal{F}(\lambda_j) > \mathcal{F}(\text{pBest}_j)$ **then**
13: Update personal best $\text{pBest}_j \leftarrow \lambda_j$
14: **end if**
15: **if** $\mathcal{F}(\lambda_j) > \mathcal{F}(\text{gBest})$ **then**
16: Update global best $\text{gBest} \leftarrow \lambda_j$
17: **end if**
18: **end for**
19: **end for**
20: **return** global best position $\lambda^* \leftarrow \text{gBest}$

IV. Benchmark Experiments and Results

This section presents a series of experiments designed to evaluate the effectiveness of the proposed CLCSIP framework. The experiments consist of four parts: 1) comparison of the CLCSIP framework with representative state-of-the-art algorithms; 2) analysis of the CSI-Purifier; 3) ablation study and 4) parameter sensitivity analysis.

A. Experimental Setup on CIFAR-10 and CIFAR-100

1) Data: CIFAR-10 and CIFAR-100 are standard image classification benchmarks, each consisting of 60,000 RGB images of size $32 \times 32 \times 3$, with 10 and 100 classes, respectively. Each dataset is split into a training set of 50,000 samples and a test set of 10,000 samples. In the training set, part of the original labels are replaced with incorrect ones to simulate noisy labels, while the test set remains unchanged.

2) Noise Settings: For CIFAR-10 and CIFAR-100, two types of noisy labels are considered: symmetric and asymmetric. In the symmetric setting, each label is randomly flipped to an incorrect class. In the asymmetric setting, class-dependent label transitions are applied. For CIFAR-10, specific flipping pairs are defined, including truck to automobile, bird to airplane, deer to horse, and cat and dog flipped to each other. For CIFAR-100, asymmetric label noise is introduced by replacing each label with another randomly selected class from the same superclass.

3) Implementation Details: For fair comparisons, all experiments are conducted on a Windows 11 enterprise machine equipped with an NVIDIA GeForce RTX 3060 Ti GPU (8 GB memory) and an 11th Gen Intel(R) Core(TM) $i9$-$11900K$ CPU. The software environment includes Python 3.10.16, PyTorch 2.5.1 with CUDA 12.1 and NVIDIA driver version 552.41 supporting CUDA 12.4.

The implementation settings for experiments on CIFAR-10 and CIFAR-100 are presented below. The feature extractor is a ResNet-18 trained using SimCLR for 1000 epochs. The input is a $32\times32\times3$ RGB image and the output is a 512-dimensional feature vector. The contrastive loss is used with a temperature of 0.5. The learning rate is 0.06. The batch size is 512.

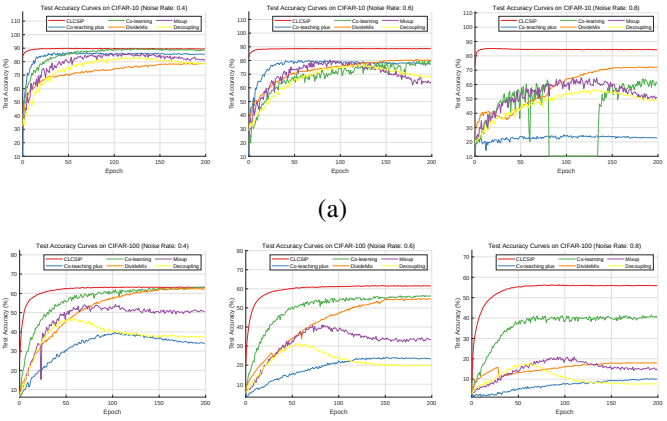
After contrastive training, a single-layer linear classification head is appended to the feature extractor. The output dimension of the classification head matches the number of classes. The contrastive loss uses a temperature of 0.07. The learning rate is 0.05. The batch size is 128 and the number of training epochs is 200. The CSI-Purifier is performed with 3 iterations. The dimension of the subspace is set to $r = 70$ for CIFAR-10 and $r = 10$ for CIFAR-100. The particle swarm optimization is configured with a population size of 5 and a total of 10 iterations. Each candidate coefficient is evaluated by training the CLCSIP framework for 10 epochs. The PSO uses commonly adopted parameter settings with inertia weight $w = 0.729$ and acceleration coefficients $c_1 = c_2 = 1.494$.

4) Baseline Methods: In this study, five state-of-the-art methods are employed as baselines, namely Co-teaching plus [29], Mixup [30], Decoupling [18], Co-learning [21] and DivideMix [13]. The selected methods cover several representative strategies for noisy label learning, including interpolation-based augmentation, collaborative training, small-loss-based sample selection and probabilistic modeling. For fair comparison, all methods adopt the same network architecture (ResNet-18), training epochs (200 epochs) and batch size (128), with implementations strictly following their official code repositories.

B. Comparison of the CLCSIP Framework with Representative State-of-the-Art Algorithms

The performance of the proposed CLCSIP framework is compared with five representative methods on CIFAR-10 and CIFAR-100, under various noise rates for both symmetric and asymmetric noisy labels. The symmetric noise rates are set to 40%, 50%, 60%, 70% and 80%, while the asymmetric noise rate is set to 40%. The reported results correspond to the mean classification accuracy over the last 10 training epochs, with standard deviation included. TABLE I and TABLE II summarize the results under both symmetric and asymmetric noisy labels on CIFAR-10 and CIFAR-100, respectively. The test accuracy curves under symmetric noise rates of 40%, 60% and 80% are shown in Fig. 3.

The CLCSIP framework achieves the highest test accuracy across all evaluated noise rates on both CIFAR-10 and CIFAR-100. As shown in TABLE I, CLCSIP outperforms all baselines under symmetric noisy labels. Specifically, CLCSIP achieves 88.66% at a 60% noise rate and maintains 84.34% at an 80% noise rate, while the best baseline, DivideMix, achieves only 72.09%. Similar trends are observed on CIFAR-100. As shown in TABLE II, CLCSIP consistently ranks first, with 63.16% at a 40% noise rate and 55.93% at an 80% noise rate, surpassing Co-learning and DivideMix by clear margins. Under asymmetric noisy labels with a 40% noise rate, CLCSIP achieves 89.25% on CIFAR-10 and 53.61% on CIFAR-100, both higher than the best-performing baselines. These results confirm the superiority of CLCSIP across datasets and noise configurations.



(a)

(b)

Fig. 3: Test accuracy curves on CIFAR-10 and CIFAR-100 under different noise rates.

The test accuracy of CLCSIP degrades much more slowly as the noise rate increases, indicating strong robustness against noisy labels. On CIFAR-10, as shown in TABLE I, the accuracy decreases from 89.44% at a 40% noise rate to 84.34% at an 80% noise rate, a margin of only 5.1 percentage points. In contrast, Co-teaching plus drops from 85.56% to 22.91% and Co-learning from 88.41% to 60.89%. A similar trend is observed on CIFAR-100. As shown in TABLE II, CLCSIP exhibits a drop of 7.2 percentage points, while Co-learning and DivideMix decline by 22.4 and 44.4 percentage points, respectively. The limited degradation of CLCSIP confirms its robustness under increasing noise rates.

The test accuracy of CLCSIP increases rapidly and remains stable throughout training. As shown in Fig. 3, CLCSIP reaches high accuracy within the first 40 epochs under all noise rates. The accuracy curves remain smooth and monotonic during the entire training process. DivideMix and Co-learning converge more slowly and exhibit fluctuations. Under a noise rate of 80%, the test accuracy of Co-teaching plus and Decoupling drops sharply after early-stage rise. On CIFAR-100, CLCSIP maintains the same trend, while baseline methods show degradation or plateauing. The results confirm that CLCSIP achieves stable and efficient convergence under noisy labels.

C. Evaluation of the Sample Selection Capability of CSI-Purifier

The effectiveness of CSI-Purifier is evaluated on CIFAR-10 and CIFAR-100 under symmetric noise rates from 0.4 to 0.8.

TABLE I: TEST ACCURACY (%) ON CIFAR-10 WITH SYMMETRIC AND ASYMMETRIC NOISY LABELS. RESULTS ARE AVERAGED OVER THE LAST 10 EPOCHS.

Noise Type	Symmetric Noise					Asymmetric Noise
Method	40%	50%	60%	70%	80%	40%
Mixup	81.59 ± 0.39	74.90 ± 0.45	64.25 ± 0.61	57.28 ± 0.66	51.14 ± 0.60	79.52 ± 0.42
Co-learning	88.41 ± 0.28	78.36 ± 2.23	78.31 ± 1.30	64.87 ± 3.47	60.89 ± 1.77	82.41 ± 0.45
Co-teaching plus	85.56 ± 0.09	81.77 ± 0.15	77.61 ± 0.23	62.79 ± 0.13	22.91 ± 0.08	80.52 ± 0.38
Decoupling	78.52 ± 0.15	74.18 ± 0.39	68.00 ± 0.26	59.83 ± 0.37	49.00 ± 0.40	79.06 ± 0.52
DivideMix	78.46 ± 0.15	80.26 ± 0.20	80.16 ± 0.18	79.48 ± 0.16	72.09 ± 0.19	60.77 ± 0.19
CLCSIP	$\mathbf{89.44 \pm 0.09}$	$\mathbf{89.18 \pm 0.07}$	$\mathbf{88.66 \pm 0.11}$	$\mathbf{87.64 \pm 0.07}$	$\mathbf{84.34 \pm 0.11}$	$\mathbf{89.25 \pm 0.24}$

TABLE II: TEST ACCURACY (%) ON CIFAR-100 WITH SYMMETRIC AND ASYMMETRIC NOISY LABELS. RESULTS ARE AVERAGED OVER THE LAST 10 EPOCHS.

Noise Type	Symmetric Noise					Asymmetric Noise
Method	40%	50%	60%	70%	80%	40%
Mixup	50.79 ± 0.29	43.44 ± 0.40	33.46 ± 0.50	23.54 ± 0.50	15.09 ± 0.39	41.18 ± 0.43
Co-learning	63.03 ± 0.17	60.22 ± 0.19	56.23 ± 0.19	49.92 ± 0.20	40.66 ± 0.33	51.12 ± 0.18
Co-teaching plus	34.20 ± 0.15	27.42 ± 0.12	23.49 ± 0.12	16.20 ± 0.12	10.08 ± 0.07	38.27 ± 0.24
Decoupling	37.57 ± 0.15	27.26 ± 0.21	19.87 ± 0.15	13.79 ± 0.11	7.72 ± 0.07	30.72 ± 0.19
DivideMix	62.36 ± 0.18	59.06 ± 0.14	54.67 ± 0.11	41.95 ± 0.20	18.01 ± 0.07	46.34 ± 0.16
CLCSIP	$\mathbf{63.16 \pm 0.09}$	$\mathbf{62.52 \pm 0.03}$	$\mathbf{61.57 \pm 0.08}$	$\mathbf{59.73 \pm 0.09}$	$\mathbf{55.93 \pm 0.10}$	$\mathbf{53.61 \pm 0.13}$

The evaluation metric is the *Clean Sample Ratio (CSR)*, which measures the proportion of correctly labeled samples among the selected ones and is computed as

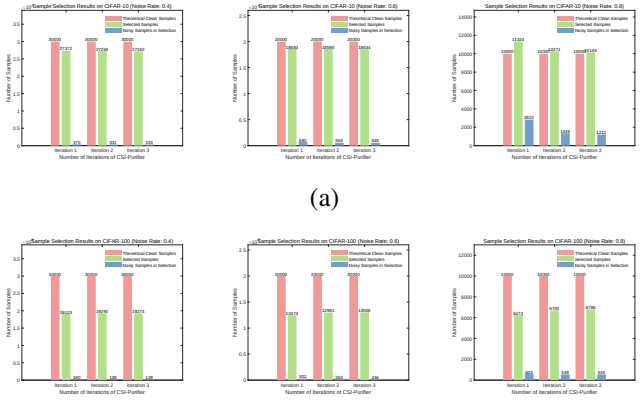
$$\mathrm{CSR} = \frac{|\mathcal{S}_{\mathrm{clean}}|}{|\mathcal{S}_{\mathrm{selected}}|}, \tag{14}$$

where $\mathcal{S}_{\mathrm{selected}}$ denotes the set of selected samples and $\mathcal{S}_{\mathrm{clean}}$ is its correctly labeled subset.

As shown in TABLE III, where the results are averaged over three independent runs, CSI-Purifier achieves consistently high CSR values across all noise rates. On CIFAR-10, the CSR remains above 95% for noise rates up to 0.7 and reaches 88.05% at a noise rate of 0.8. On CIFAR-100, the CSR exceeds 96% when the noise rate is up to 0.7 and maintains 91.97% at a noise rate of 0.8. Despite the increase in noise rate, the CSR remains high, indicating that CSI-Purifier maintains effective clean sample selection under different noise conditions.

Fig. 4 further illustrates the sample selection results under symmetric noise rates of 0.4, 0.6 and 0.8. In each subfigure, three bars represent the theoretical number of clean samples, the number of selected samples by CSI-Purifier and the number of noisy samples among the selected samples. For CIFAR-10, under a noise rate of 0.4, approximately 27700 samples are retained, closely matching the theoretical clean sample count. Similar patterns are observed under noise rates of 0.6 and 0.8, where about 18500 and 10000 samples are selected, respectively. On CIFAR-100, the retained samples are approximately 19200, 13000 and 6786 under the respective noise rates, indicating that CSI-Purifier maintains sufficient data volume across different noise levels.

A strong level of robustness to severe label noise is demonstrated by the proposed iterative purification mechanism through the progressive reduction of noisy samples involved in subspace construction. Even under a high noise rate of 0.8, a significant decrease in the number of noisy samples is observed across iterations. On CIFAR-10, the count is reduced from 2622 after the first iteration to 1212 after the third. On CIFAR-100, a similar reduction is observed from 605 to 545. Although the presence of noisy labels tends to disrupt the structure of class subspaces, such interference is effectively mitigated by the iterative design of the CSI-Purifier.



(a)

(b)

Fig. 4: Sample selection results on CIFAR-10 and CIFAR-100 under different noise rates.

TABLE III: CSR UNDER DIFFERENT SYMMETRIC NOISE RATES.

Data	Noise Rates	CSR (%)
CIFAR-10	0.4	98.78
	0.5	98.16
	0.6	97.04
	0.7	95.11
	0.8	88.05
CIFAR-100	0.4	99.28
	0.5	98.84
	0.6	98.11
	0.7	96.37
	0.8	91.97

D. Ablation Study

To investigate the contribution of each module in the CLCSIP framework, an ablation study is conducted on CIFAR-10 under symmetric noisy labels. The following three variants are considered for comparison: 1) CLCSIP-A removes the dual-loss supervised fine-tuning module and adopts a pure cross-entropy loss for classifier training; 2) CLCSIP-B disables the PSO-based adaptive loss weighting and instead assigns the loss weight λ as a random number sampled from a uniform distribution over $[0, 5]$ and 3) CLCSIP-C eliminates the CSI-Purifier-based sample selection module and uses the entire dataset for training. All experiments follow the same settings as described in Section IV-A, except that the training epoch is reduced to 50 to save computational resources. The adjustment described above is supported by the convergence observation in Section IV-B, where the test accuracy of all methods becomes stable after 20 epochs. The average and standard deviation of test accuracy over the last 10 epochs are reported in TABLE IV.

The results of the ablation study are reported in TABLE IV. According to the reported results, the following analyses are presented: 1) The inclusion of dual-loss supervised fine-tuning improves the generalization ability of the classifier, as CLCSIP outperforms CLCSIP-A across all noise rates. 2) PSO enables more effective loss weighting and enhances classification performance, as reflected by the consistent superiority of CLCSIP over CLCSIP-B. 3) CSI-Purifier-based sample selection is essential for mitigating the negative impact of noisy labels, as evidenced by the significant performance degradation of CLCSIP-C across all noise rates. The above analysis confirms the critical contribution of each component in the CLCSIP framework, with the best performance attained only when all modules are jointly integrated.

E. Parameter Sensitivity Analysis

The sensitivity of CLCSIP to three key hyperparameters is analyzed by varying one parameter at a time while keeping all others fixed. All settings follow those described in Section IV-A, except for the specific parameters being examined. The loss weighting coefficient is fixed at 0.5203, which is the optimal value obtained from the results in Section IV-B. The number of training epochs is set to 50, based on the convergence observation in Section IV-B, where the test accuracy of all methods becomes stable after 20 epochs. To provide a representative yet concise evaluation, all experiments are conducted on CIFAR-10 with a symmetric noise rate of 0.8. The average and standard deviation of test accuracy over the last 10 epochs are reported, along with the CSR achieved by the CSI-Purifier. These metrics are used to assess the impact of each parameter on both the quality of selected samples and the overall classification performance.

The impact of the class space dimension r on CSR and test accuracy is reported in TABLE V and illustrated in Fig. 5. As r increases from 1 to 20, both metrics steadily improve. The highest accuracy of 86.74% and CSR of 93.38% are obtained at $r = 10$. When r exceeds 40, both metrics decline consistently. These results confirm that the classification performance is affected by the choice of r and better results are achieved in the low-dimensional range.

The results under different iteration numbers are reported in TABLE VI and illustrated in Fig. 6. As T increases from 1 to 3, consistent improvements in CSR and accuracy are observed. When T exceeds 3, both metrics remain stable. The highest accuracy of 84.24% and CSR of 88.28% are reached at $T = 5$, with comparable results observed at $T = 3$. These results indicate that three iterations are sufficient to ensure stable performance.

The results under different values of the temperature coefficient τ are summarised in TABLE VII. As τ increases from 0.02 to 0.50, test accuracy shows a gradual improvement, reaching a peak of 85.21% at $\tau = 0.50$, followed by a slight decline at $\tau = 1.00$. While some variation in performance is observed across settings, the overall impact of τ remains limited within the tested range.

TABLE V: EFFECT OF CLASS SPACE DIMENSION R ON CSR AND TEST ACCURACY

r	CSR (%)	Accuracy (%)
1	92.06	83.76 ± 0.11
5	92.37	85.07 ± 0.07
10	93.38	86.74 ± 0.06
20	93.34	86.65 ± 0.05
40	92.11	85.90 ± 0.06
80	86.62	84.37 ± 0.08
120	77.81	80.36 ± 0.11
160	67.22	75.29 ± 0.18

Fig. 5: Effect of class space dimension r on CSR and test accuracy.

TABLE VI: EFFECT OF CSI-PURIFIER ITERATION NUMBER T ON CSR AND TEST ACCURACY

T	CSR (%)	Accuracy (%)
1	74.54	77.46 ± 0.11
2	86.52	83.57 ± 0.13
3	87.84	84.10 ± 0.05
4	88.12	84.17 ± 0.10
5	88.28	84.24 ± 0.10

V. Application of the CLCSIP Framework to WAAM Anomaly Detection

In this section, the CLCSIP framework is applied to the task of anomaly detection on WAAM data and its practical effec-

TABLE IV: ABLATION STUDY RESULTS (%) ON CIFAR-10 UNDER DIFFERENT SYMMETRIC NOISE RATES

Noise Type	Symmetric Noise				
Method	40%	50%	60%	70%	80%
CLCSIP-A	89.24 ± 0.06	88.79 ± 0.07	88.68 ± 0.08	87.92 ± 0.09	85.36 ± 0.09
CLCSIP-B	89.57 ± 0.06	89.10 ± 0.07	88.66 ± 0.09	87.85 ± 0.09	84.98 ± 0.10
CLCSIP-C	58.83 ± 0.06	49.33 ± 0.11	45.56 ± 0.04	30.05 ± 0.05	18.20 ± 0.06
CLCSIP	$\mathbf{89.72 \pm 0.07}$	$\mathbf{89.18 \pm 0.05}$	$\mathbf{88.81 \pm 0.06}$	$\mathbf{88.03 \pm 0.08}$	$\mathbf{85.57 \pm 0.09}$

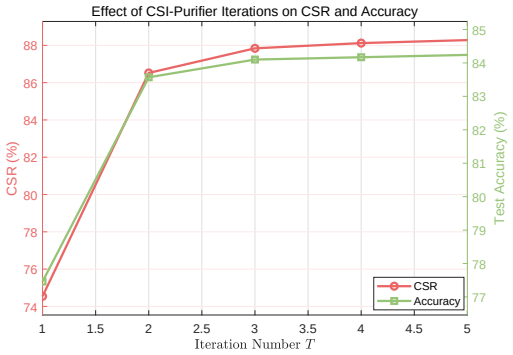


Fig. 6: Effect of CSI-Purifier iterations on CSR and test accuracy.

TABLE VII: EFFECT OF TEMPERATURE COEFFICIENT τ ON TEST ACCURACY

τ	Accuracy (%)
0.02	83.61 ± 0.09
0.03	83.87 ± 0.11
0.05	84.14 ± 0.08
0.07	84.45 ± 0.09
0.10	84.63 ± 0.09
0.20	85.08 ± 0.10
0.50	85.21 ± 0.10
1.00	85.17 ± 0.10

tiveness is demonstrated under realistic noisy label conditions.

The WAAM data used in this study are collected from a pilot metal additive manufacturing line deployed in Sweden. Five time series are extracted, each corresponding to a single production run and consisting of 98,000 records. Each record includes welding current and voltage measurements sampled at a 0.0002-second interval. The collected WAAM data are not publicly available and contain noisy labels that reflect practical industrial conditions. In the WAAM data, noisy labels arise naturally from imperfect human annotations.

The implementation settings for the WAAM anomaly detection task are presented as follows. The feature extractor is implemented as a one-dimensional convolutional network composed of two convolutional layers and one fully connected layer. The first convolutional layer takes a two-channel input and outputs 32 channels using a kernel size of 3 with padding 1. The second convolutional layer maps from 32 to 64 channels with the same kernel and padding settings. A max-pooling layer with kernel size 2 is applied after the second convolution. The output is flattened and passed through a fully connected layer to produce a 64-dimensional feature vector. The input is constructed from raw voltage and current measurements using a sliding window of length 100 and a stride of 10. Each input instance is a two-channel sequence of shape 2×100 and its label is assigned based on the center point of the corresponding window. The CSI-Purifier is performed with 3 iterations and the dimension of the subspace is set to $r = 10$. The classification head is a single-layer sigmoid classifier attached to the feature extractor and trained for 10 epochs without freezing the feature extractor. The optimizer is Adam with a learning rate of 0.001. The model is trained using the dual-loss. The batch size is 64 for both training and evaluation. The weighting coefficient for the dual-loss is optimized using the same PSO configuration as in the CIFAR experiments.

The performance is assessed using detection accuracy, precision, recall and $F1$ score. Each experiment is independently repeated three times and the results are averaged. To provide a comparison, a baseline model with the same network architecture and hyperparameter settings is trained directly using the conventional cross-entropy loss. All experimental results, including those of the CLCSIP framework and the baseline, are summarized in TABLE VIII.

Across the five time series data segments, the CLCSIP framework consistently outperforms the baseline in terms of accuracy, precision, recall and $F1$ score. For Data 2, CLCSIP achieves an accuracy of 96.80% and an $F1$ score of 95.02%, while the baseline only reaches 71.97% accuracy and 70.14% $F1$ score. For the remaining time series data segments, CLCSIP maintains accuracy above 95% and $F1$ scores exceeding 91%, whereas the baseline performance is notably lower, particularly in precision and $F1$ score. The comparative analysis confirms that CLCSIP consistently improves detection performance across different production runs. These results further demonstrate its practical applicability to industrial anomaly detection tasks with noisy labels.

TABLE VIII: ANOMALY DETECTION PERFORMANCE (%) OF CLCSIP AND THE BASELINE MODEL ON FIVE WAAM TIME SERIES DATA SEGMENTS.

Approach	Metric	Data 1	Data 2	Data 3	Data 4	Data 5
CLCSIP	Accuracy	97.51	96.80	95.53	96.93	96.86
	Precision	95.70	94.31	96.26	95.93	93.13
	Recall	95.51	95.78	88.95	92.71	95.30
	F1 Score	97.51	95.02	91.78	94.08	94.08
Baseline	Accuracy	93.26	71.97	92.68	93.19	93.05
	Precision	78.55	68.82	77.95	77.16	77.20
	Recall	81.44	80.53	82.10	80.99	79.66
	F1 Score	79.85	70.14	79.85	78.95	78.32

VI. Conclusion

In this paper, a CLCSIP framework has been proposed to address the challenge of learning with noisy labels. The

framework has integrated a CSI-Purifier-based sample selection module, a dual-loss supervised fine-tuning strategy, and a PSO-based adaptive loss weighting module. Through the CSI-Purifier, a novel sample selection criterion independent of training losses has been introduced by evaluating the consistency between sample features and their corresponding class representations. The dual-loss supervised fine-tuning strategy has jointly optimized classification and contrastive objectives, thereby improving robustness against noisy labels. In addition, PSO has been employed to adaptively determine the weighting between classification and contrastive losses, alleviating the need for manual hyperparameter tuning. Extensive experiments on CIFAR-10 and CIFAR-100 datasets under various noise settings have demonstrated that the proposed CLCSIP framework consistently outperforms representative methods across different noise levels. Furthermore, the CLCSIP framework has been applied to an anomaly detection task on real-world WAAM data with noisy labels, validating its practical effectiveness in industrial scenarios.

Future research directions can be summarized as follows: 1) developing dynamic sample selection mechanisms to refine the training set [17]; 2) extending the CLCSIP framework to semi-supervised learning using class subspaces; 3) adapting the framework to multimodal scenarios via joint embedding spaces; and 4) applying the proposed CLCSIP framework to additional industrial scenarios [24].

REFERENCES

[1] T. Chen, S. Kornblith, M. Norouzi and G. Hinton, A simple framework for contrastive learning of visual representations, *In: Proceedings of the 37th International Conference on Machine Learning*, vol. 119, Jul. 2020, pp. 1597-1607.

[2] E. Elhamifar and R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765-2781, Nov. 2013.

[3] J. Fang, Z. Wang, W. Liu, L. Chen and X. Liu, A new particle-swarm-optimization-assisted deep transfer learning framework with applications to outlier detection in additive manufacturing, *Engineering Applications of Artificial Intelligence*, vol. 131, May 2024.

[4] J. Fang, Z. Wang, W. Liu, N. Zeng, Y. He, Y. Cao, L. Chen and X. Liu, Learning with noisy labels for industrial time series outlier detection: A transformer-embedded contrastive learning framework, *IEEE Transactions on Industrial Informatics*, Oct. 2025.

[5] K. Fukui, Subspace methods, *In: Computer Vision: A Reference Guide*, Springer, 2021, pp. 1221-1224.

[6] M. Forouzesh and P. Thiran, Differences between hard and noisy-labeled samples: An empirical study, *In: Proceedings of the 2024 SIAM International Conference on Data Mining (SDM), Minneapolis, USA*, Apr. 2024, pp. 91-99.

[7] A. Ghosh and A. Lan, Contrastive learning improves model robustness under label noise, *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 2703-2708.

[8] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang and M. Sugiyama, Co-teaching: Robust training of deep neural networks with extremely noisy labels, *In: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, Canada*, Dec. 2018, vol. 31.

[9] B. Han, G. Niu, X. Yu, Q. Yao, M. Xu, I. Tsang and M. Sugiyama, SIGUA: Forgetting may make learning with noisy labels more robust, *In: Proceedings of the 37th International Conference on Machine Learning (ICML 2020), Online*, Jul. 2020, pp. 4006-4016.

[10] N.-R. Kim, J.-S. Lee and J.-H. Lee, Learning with structural labels for learning with noisy labels, *In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA*, Jun. 2024, pp. 27610-27620.

[11] V. C. Klema and A. J. Laub, The singular value decomposition: Its computation and some applications, *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164-176, Apr. 1980.

[12] Y. Li, H. Han, S. Shan and X. Chen, DISC: Learning from noisy labels via dynamic instance-specific selection and correction, *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 24070-24079.

[13] J. Li, R. Socher and S. C. H. Hoi, DivideMix: Learning with noisy labels as semi-supervised learning, *arXiv preprint arXiv:2002.07394*, 2020.

[14] S. Li, X. Xia, S. Ge and T. Liu, Selective-supervised contrastive learning with noisy labels, *In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, USA*, Jun. 2022, pp. 316-325.

[15] J. Li, C. Xiong and S. C. H. Hoi, Learning from noisy data with robust representation learning, *In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, Canada*, Oct. 2021, pp. 9485-9494.

[16] W. Liu, Z. Wang, X. Liu, N. Zeng and D. Bell, A novel particle swarm optimization approach for patient clustering from emergency departments, *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 632-644, Oct. 2018.

[17] S. Liu, S. Xue, J. Wu, C. Zhou, J. Yang, Z. Li and J. Cao, Online active learning for drifting data streams, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 1, pp. 186-200, Jan. 2023.

[18] E. Malach and S. Shalev-Shwartz, Decoupling when to update from how to update, *In: Advances in Neural Information Processing Systems 30*, Dec. 2017.

[19] D. Patel and P. S. Sastry, Adaptive sample selection for robust learning under label noise, *In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, USA*, Jan. 2023, pp. 3932-3942.

[20] H. Song, M. Kim, D. Park, Y. Shin and J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 11, pp. 8135-8153, Nov. 2023.

[21] C. Tan, J. Xia, L. Wu and S. Z. Li, Co-learning: Learning from noisy labels with self-supervision, *In: Proceedings of the 29th ACM International Conference on Multimedia (ACM MM), Chengdu, China*, Oct. 2021, pp. 1405-1413.

[22] Y. Tu, B. Zhang, Y. Li, L. Liu, J. Li, Y. Wang, C. Wang and C. R. Zhao, Learning from noisy labels with decoupled meta label purifier, *In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada*, Jun. 2023, pp. 19934-19943.

[23] Y. Tu, B. Zhang, Y. Li, L. Liu, J. Li, J. Zhang, Y. Wang, C. Wang and C. Zhao, Learning with noisy labels via self-supervised adversarial noisy masking, *In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, Canada*, Jun. 2023, pp. 16186-16195.

[24] C. Wang, Z. Wang, H. Liu, H. Dong and P. Lu, An optimal unsupervised domain adaptation approach with applications to pipeline fault diagnosis: Balancing invariance and variancek, *IEEE Transactions on Industrial Informatics*, vol. 20, no. 8, pp. 10019-10030, May 2024.

[25] H. Wei, L. Feng, X. Chen and B. An, Combating noisy labels by agreement: A joint training method with co-regularization, *In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA*, Jun. 2020, pp. 13723-13732.

[26] J. Xia, H. Lin, Y. Xu, C. Tan, L. Wu, S. Li and S. Z. Li, GNN Cleaner: Label cleaner for graph structured data, *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 2, pp. 640-651, Feb. 2024.

[27] X. Xia, P. Lu, C. Gong, B. Han, J. Yu and T. Liu, Regularly truncated M-estimators for learning with noisy labels, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 3522-3536, May 2024.

[28] L. Yi, S. Liu, Q. She, A. I. McLeod and B. Wang, On learning contrastive representations for learning with noisy labels, *In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16682-16691.

[29] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang and M. Sugiyama, How does disagreement help generalization against label corruption? *In: Proceedings of the 36th International Conference on Machine Learning, Long Beach, USA*, Jun. 2019, pp. 7164-7173.

[30] H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, Mixup: Beyond empirical risk minimization, *In: Proceedings of the 6th International Conference on Learning Representations (ICLR)*, May 2018.