Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.064489

ARTICLE

An Optimized Unsupervised Defect Detection Approach via Federated Learning and Adaptive Embeddings Knowledge Distillation

Jinhai Wang[1], Junwei Xue[1], Hongyan Zhang[2], Hui Xiao[3,4], Huiling Wei[3,4], Mingyou Chen[3,4], Jiang Liao[2] and Lufeng Luo[3,4,*]

[1]School of Computer Science and Artificial Intelligence, Foshan University, Foshan, 528225, China
[2]Foshan Wision Furniture Manufacturing Co., Ltd., Foshan, 528200, China
[3]School of Mechatronic Engineering and Automation, Foshan University, Foshan, 528225, China
[4]Guangdong Provincial Key Laboratory of Industrial Intelligent Inspection Technology, Foshan University, Foshan, 528225, China

*Corresponding Author: Lufeng Luo. Email: luolufeng617@163.com

Received: 17 February 2025; Accepted: 17 April 2025; Published: 09 June 2025

ABSTRACT: Defect detection based on computer vision is a critical component in ensuring the quality of industrial products. However, existing detection methods encounter several challenges in practical applications, including the scarcity of labeled samples, limited adaptability of pre-trained models, and the data heterogeneity in distributed environments. To address these issues, this research proposes an unsupervised defect detection method, FLAME (Federated Learning with Adaptive Multi-Model Embeddings). The method comprises three stages: (1) Feature learning stage: this work proposes FADE (Feature-Adaptive Domain-Specific Embeddings), a framework employs Gaussian noise injection to simulate defective patterns and implements a feature discriminator for defect detection, thereby enhancing the pre-trained model's industrial imagery representation capabilities. (2) Knowledge distillation co-training stage: a multi-model feature knowledge distillation mechanism is introduced. Through feature-level knowledge transfer between the global model and historical local models, the current local model is guided to learn better feature representations from the global model. The approach prevents local models from converging to local optima and mitigates performance degradation caused by data heterogeneity. (3) Model parameter aggregation stage: participating clients utilize weighted averaging aggregation to synthesize an updated global model, facilitating efficient knowledge consolidation. Experimental results demonstrate that FADE improves the average image-level Area under the Receiver Operating Characteristic Curve (AUROC) by 7.34% compared to methods directly utilizing pre-trained models. In federated learning environments, FLAME's multi-model feature knowledge distillation mechanism outperforms the classic FedAvg algorithm by 2.34% in average image-level AUROC, while exhibiting superior convergence properties.

KEYWORDS: Federated learning; defect detection; knowledge distillation; unsupervised learning

1 Introduction

Automated monitoring systems are gradually replacing traditional manual inspection methods in modern manufacturing [1–4]. Defect detection based on computer vision has become an essential component of quality inspection in smart manufacturing processes. Supervised learning requires a large numbers of labeled samples to achieve optimal model performance. However, the high cost of collecting industrial defect samples presents a significant barrier to the acquisition of sufficient labeled data for defect detection models [5,6]. Moreover, variations across factories in equipment, environment, and production create inconsistent data distributions, causing non-identical data distribution (non-IID) issues that reduce model

performance [7]. Hence, effectively transferring knowledge among manufacturing sites while mitigating the impact of heterogeneous industrial data remains a key challenge.

Unsupervised defect detection has emerged as a widely adopted approach in industrial inspection systems. This method relies solely on normal samples for identifying defects. Contemporary unsupervised defect detection methods can be classified into three main categories: reconstruction-based, generation-based, and embedding-based approaches. Among various methods, embedding-based approaches have attracted increasing attention due to their promising performance. These approaches use pre-trained Convolutional Neural Networks (CNNs) from ImageNet to extract features, then apply statistical methods like multivariate Gaussian distributions [6], feature repositories [8], or normalizing flows [9] to model normal sample distributions. Industrial images have feature distributions that differ significantly from those in ImageNet. This difference limits the direct deployment of pre-trained CNNs in real-world industrial applications.

As a distributed learning framework, federated learning enables collaborative model training across different clients while preserving data privacy [10]. In the industrial defect detection context, privacy preservation is particularly critical. Manufacturing data often contains proprietary information about production processes, quality control standards, and competitive advantages. Federated learning addresses these ethical and privacy concerns. It enables model training without direct data sharing. This allows manufacturers to collaborate while protecting their intellectual property. It also helps them safeguard their competitive secrets. The classic federated learning algorithm FedAvg [11] achieves model optimization through iterative parameter aggregation. In each round, clients send updated model parameters to the central server. The server then aggregates these parameters to refresh the global model. However, traditional federated learning faces a key challenge. Models trained only on local datasets create gaps between local and global objectives. These gaps hinder model convergence. Knowledge distillation technology emerges as a promising solution to address this issue. FEDDFUSION [12] improves model performance via knowledge distillation on unlabeled data while maintaining privacy protection and computational efficiency. Building on knowledge distillation's success in federated learning, this research proposes a method that leverages multi-model embedding features.

In the intersection of federated learning and defect detection, IsIam et al. [13] implemented a federated framework for USB flash drive quality inspection, enabling distributed collaborative fine-tuning of a pre-trained Visual Geometry Group (VGG) model. Mehta and Shao [14] extended this paradigm to metal additive manufacturing, which improved defect detection over traditional single-machine methods. However, these methods mainly focus on supervised learning scenarios, limiting their practical applications due to the scarcity of labeled data. Zhang et al. [15] combined federated learning with unsupervised representation learning, but only explored conceptual integration without maximizing federated learning and unsupervised representation learning combined benefits.

This work proposes an unsupervised defect detection framework combining federated learning with knowledge distillation to address labeling limitations and enhance training efficiency in distributed industrial settings. The contributions of this work are summarized as follows:

(1) Introducing federated learning into unsupervised defect detection through FLAME (Federated Learning with Adaptive Multi-Model Embeddings), enabling collaborative model optimization while preserving data privacy.
(2) Proposing FADE (Feature-Adaptive Domain-Specific Embeddings), which generates synthetic defect embeddings for model optimization while eliminating defect sample dependencies and adapting to target domain distributions.

(3) Designing a multi-model feature knowledge distillation mechanism, which integrates local, global, and historical feature representations to optimize diverse knowledge and reduce data distribution gaps among clients.

2 Related Work

2.1 Defect Detection and Localization

A large body of work has employed supervised approaches for defect detection and localization [16–19]. Current unsupervised methods for defect detection and localization can be categorized into three fundamental approaches: reconstruction-based, generation-based and embedding-based techniques.

Reconstruction-based methods fundamentally leverage the autoencoder architecture. Abati et al. [20] minimize the differential entropy of latent distribution through joint optimization of reconstruction error and maximum likelihood estimation of latent representations. Although the reconstruction-based method exhibits strong expression and generalization capabilities, it may also accurately reconstructs defective samples, potentially leading to false detections [21]. Generation-based methods focus on synthesizing defective samples to establish decision boundaries between normal and defective data distributions. SimpleNet [22] addresses domain bias through the integration of a pre-trained feature extractor and lightweight feature adapter. Embedding-based methods rely on feature representations extracted from pre-trained CNNs for defect detection. PaDiM [6] leverages features extracted from pre-trained CNNs to construct the normal class distribution. Ma et al. [23] propose a Transformer-based generation, detection, and tracking network for drainage pipeline defect images, which enhances feature extraction capabilities through self-attention mechanisms and incorporates Generative Adversarial Network (GAN) for data augmentation. Jia et al. [24] propose a photovoltaic module defect detection method based on improved VarifocalNet, significantly enhancing both detection accuracy and speed.

2.2 Federated Learning

Following McMahan et al.'s [11] pioneering FedAvg algorithm, federated learning has emerged as a prominent research direction across industrial domains. Hsu et al. [7] study non-IID effects in federated visual tasks using Dirichlet distribution, generating datasets with controlled similarity levels to evaluate federated algorithms across different data distributions.

One promising strategy for addressing data heterogeneity lies in optimizing the local training procedure. For instance, FedProx, proposed by Li et al. [25], constrains the drift between local updates and the global model by adding a proximal term to the local objective function. Li et al. [26] tackle feature distribution shift in non-IID settings by applying local batch normalization, which reduced discrepancies and improved convergence. Wang et al. [27] introduce FedNova, which implements an adaptive normalization strategy to handle heterogeneous client update frequencies. Karimireddy et al. [28] introduce SCAFFOLD to correct client drift using control variables. These approaches mainly focus on basic classification tasks. However, their effectiveness for complex visual tasks, especially defect detection, remains largely unexplored.

2.3 Knowledge Distillation

Knowledge distillation [29] has gained prominence in recent years. Researchers now explore improved methods to combine knowledge distillation with federated learning. Data-efficient image Transformers (DeiT) [30] creates a novel distillation method using tokens that allow student models to extract teacher features via attention mechanisms. The Fast Knowledge Distillation (FKD) framework [31] reduces costs by precomputing region-level soft labels before training, avoiding repeated teacher network forward passes.

VanillaKD [32] enhances distillation efficacy through KL divergence loss optimization while generating labels for multiple cropped instances, preserving information fidelity comparable to conventional knowledge distillation.

Knowledge distillation has emerged as a critical technique within federated learning frameworks for addressing model heterogeneity. FedMD [33] integrates transfer learning with knowledge distillation to enable collaborative learning by sharing output scores on public datasets, preserving both architectural diversity and data privacy. FedGKD [34] uses past global models' knowledge to guide local training, regularizing features and maintaining performance.

3 Preliminaries and Definitions

This section presents the theoretical foundations and key definitions of federated learning, and the established frameworks are detailed in [27,28].

3.1 Generalized Update Rules of Federated Learning

Consider a federated learning framework with K $(K \geq 1)$ communication rounds across M clients. Each client has a private dataset D_i with size $D_i \triangleq |D_i|$, determining its aggregation weight $p_i = D_i/D$. During the kth round, client C_i performs $\tau_{(k,i)}$ $\left(\tau_{(k,i)} \geq 1\right)$ local optimization steps. The notation $w^{\lambda}_{(k,i)}$ represents local model parameters, where $\lambda = 0, 1, 2, \ldots, \tau_{(k,i)}$ indicates the local iteration index.

Each client constructs the local loss function $F_i\left(w^{\lambda}_{(k,i)}\right)$ derived from its respective dataset. The primary objective of the federated learning system is to minimize the global loss function $F\left(w_K\right)$. At the onset of each communication round $(\lambda = 0)$, clients initialize their local model parameters with the global parameters received from the server $\left(w^{0}_{(k,\mathrm{i})} = w_k\right)$. Throughout the local training iterations $\left(0 \leq \lambda \leq \tau_{(k,i)} - 1\right)$, C_i computes the local gradient $\nabla F_i\left(w^{\lambda}_{(k,i)}\right)$ based on its local loss function $F_i\left(w^{\lambda}_{(k,i)}\right)$ and current model parameters $w^{\lambda}_{(k,\mathrm{i})}$. The optimization process can be formalized as Eqs. (1)–(3):

$$F_i\left(w^{\lambda}_{(k,i)}\right) = \frac{1}{|\mathrm{D}_i|} \sum_{x_j \in \mathrm{D}_i} l\left(w^{\lambda}_{(k,i)}; x_j\right) \tag{1}$$

$$F\left(w_{k+1}\right) = \sum_{i=1}^{N} p_i F_i\left(w^{\lambda}_{(k,i)}\right) \tag{2}$$

$$w^{\lambda+1}_{(k,i)} = w^{\lambda}_{(k,i)} - \eta \nabla F_i\left(w^{\lambda}_{(k,i)}\right) \tag{3}$$

where $l\left(w^{\lambda}_{(k,i)}; x_j\right)$ represents the loss value of the local model $w^{\lambda}_{(k,i)}$ on sample x_j, with $\eta > 0$ denoting the learning rate of the optimizer. The client gradient update process is characterized by three essential parameters: the local normalized gradient $G_{(k,i)}$, the global gradient direction $d_k = \sum_{i=0}^{N} P_i G_{(k,i)}$, and the global step size $\tau_{\mathrm{k}} = \sum_{i=1}^{N} P_i \left\| a_i \right\|_1$, defined as Eq. (4):

$$G_{(k,i)} \triangleq \frac{1}{\left\| a_i \right\|_1} \sum_{\lambda=0}^{\tau_{(k,i)}-1} a_i^{\lambda} \nabla F_i\left(w^{\lambda}_{(k,i)}\right) \tag{4}$$

where $a_i \in \mathbb{R}^{\tau_{(k,i)}}$ defines a non-negative vector that determines the local accumulation of stochastic gradients, with its λth element denoted as a_i^{λ}. Upon completion of local optimization iterations, the parameter

server executes the global model parameter update. Given the above definitions, the global model update rule simplifies from Eqs. (5) and (6):

$$w_{k+1} = w_k - \eta \sum_{i=1}^{N} p_i \left\| a_i \right\|_1 G_{(k,i)} \tag{5}$$

$$w_{k+1} = w_k - \eta \tau_k d_k \tag{6}$$

3.2 FedAvg and FedNova Algorithms

3.2.1 FedAvg

In the FedAvg algorithm, clients communicate with the central server during the kth communication round. Each client C_i maintains a private dataset D_i. These datasets have a specified batch size B and local training occurs for E epochs. The number of local optimization iterations $\tau_{(k,i)} = \left\lfloor E \cdot \frac{D_i}{B} \right\rfloor$ is determined to help control the training process for each client. The cumulative vector a_i is defined as [1, 1, …, 1] with L1 norm $\left\| a_i \right\|_1 = \tau$. Consequently, the local normalized gradient $G_{(k,i)}$ and its corresponding model update rule are formulated as Eq. (7):

$$\begin{cases} G_{(k,i)} \triangleq \sum_{\lambda=0}^{\tau-1} \nabla F_i\left(w_{(k,i)}^{\lambda}\right) \\ w_{k+1} = w_k - \eta \sum_{i=1}^{N} p_i G_{(k,i)} \end{cases} \tag{7}$$

3.2.2 FedNova

The FedNova algorithm preserves FedAvg's cumulative vector definition $a_i = [1, 1, \ldots, 1]$ while introducing a novel local gradient normalization mechanism. This normalization of client contributions during global model aggregation effectively mitigates systematic biases stemming from heterogeneous local training iterations. The local normalized gradient $G_{(k,i)}$ and its corresponding model update rule are formulated as Eq. (8):

$$\begin{cases} G_{(k,i)} \triangleq \frac{1}{\tau_{(k,i)}} \sum_{\lambda=0}^{\tau_{(k,i)}-1} \nabla F_i\left(w_{(k,i)}^{\lambda}\right) \\ w_{k+1} = w_k - \eta \frac{\sum_{i=1}^{N} p_i \tau_{(k,i)} G_{(k,i)}}{\sum_{i=1}^{N} p_i \tau_{(k,i)}} \end{cases} \tag{8}$$

3.3 FedProx and SCAFFOLD Algorithms

3.3.1 FedProx

FedProx adds a regularization term based on the global model w_k to constrain local model divergence through scale control. The enhanced local optimization objective function is formulated as Eq. (9):

$$F_i\left(w_{(k,i)}^{\lambda}\right) = f_i\left(w_{(k,i)}^{\lambda}\right) + \frac{\mu}{2} \left\| w_{(k,i)}^{\lambda} - w_k \right\|^2 \tag{9}$$

where $f_i\left(w_{(k,\mathrm{i})}^{\lambda}\right)$ denotes the inherent loss function of C_i on its private dataset D_i, $\mu > 0$ acts as a regularization weight balancing between primary loss and regularization terms. It implements the following distinct rule during local optimization as Eq. (10):

$$w_{(k,i)}^{\lambda+1} = w_{(k,i)}^{\lambda} - \eta\left(\nabla F_i\left(w_{(k,i)}^{\lambda}\right) + \mu\left(w_{(k,i)}^{\lambda} - w_k\right)\right) \tag{10}$$

3.3.2 SCAFFOLD

SCAFFOLD mitigates model divergence which arises from heterogeneous data distributions through dual-level control variables, effectively minimizing gradient directional bias. The algorithm incorporates a control variable correction term in the client-side gradient computation, expressed as Eq. (11):

$$\nabla F_i\left(w_{(k,i)}^{\lambda}\right) = \nabla f_i\left(w_{(k,i)}^{\lambda}\right) - c_i + c \tag{11}$$

where $\nabla f_i\left(w_{(k,\mathrm{i})}^{\lambda}\right)$ represents C_i's original gradient computation, while c_i and c denote the client and server control variables, respectively. Following each communication round, the server updates and disseminates the global control variable c to guide subsequent training iterations. The local model parameters are updated using the corrected gradient according to Eq. (12):

$$w_{(k,i)}^{\lambda+1} = w_{(k,i)}^{\lambda} - \eta\nabla F_i\left(w_{(k,i)}^{\lambda}\right) \tag{12}$$

4 Method

4.1 Feature-Adaptive Domain-Specific Embeddings

Convolutional Neural Networks (CNNs) have emerged as the primary approach for image feature extraction in deep learning frameworks. However, pre-trained models' features are inherently biased toward their original training data. Adapting pre-trained models for domain-specific defect detection exhibits inadequate performance in practical scenarios. This study proposes a FADE (Feature-Adaptive Domain-Specific Embeddings) algorithm, which synthesizes defect embeddings to adaptively create realistic anomaly embeddings in specific domains. Unlike conventional anomaly detection algorithms, FADE synthesizes defect patterns in feature space rather than image space. By generating synthetic embeddings with Gaussian noise, it eliminates the need for real defect samples. FADE leverages domain knowledge from pretrained models to extract meaningful representations of normal patterns. It adopts a discriminative approach that directly identifies differences between normal and defective embeddings.

The pipeline of the FADE algorithm is illustrated in Fig. 1. The process begins with a set of N normal images as input. These images pass through a pretrained model that extracts activation vectors from different layers. The model concatenates these features to form N normal embeddings. A defect embeddings generator applies Gaussian noise to create synthetic defect representations. These are added to normal embeddings to produce defective embeddings. Both normal and synthetic defective embeddings are fed into a discriminator. The discriminator generates defect maps that highlight potential anomalous regions.

Let x_{train} and x_{test} denote the training and test sets respectively, where each image x_j is an element of $\mathbb{R}^{H\times W\times 3}$ representing an RGB input. The feature map is partitioned into non-overlapping patches, where spatial coordinates $(\mathrm{h},\mathrm{w}) \in [1,H]\times[1,W]$ define the dimensional bounds for embeddings generation. A hierarchical structure L is established. Each level $l \in L$ yields a feature activation map $\phi^{l,j} \sim \phi^{l}\left(x_j\right) \in \mathbb{R}^{H_l\times W_l\times C_l}$. The parameters H_l, W_l, and C_l denote spatial dimensions and channel depth. Feature subsets are integrated across hierarchical levels. This integration captures multi-scale semantic features. It produces embeddings ($E_{\mathrm{h,w}}^{j} \in \mathbb{R}^{c}$) that preserve semantic hierarchy and spatial characteristics, as shown in Eq. (13):

$$E_{\mathrm{h,w}}^{j} = \left[\phi_{\mathrm{h,w}}^{1}; \phi_{\mathrm{h,w}}^{2}; \ldots; \phi_{\mathrm{h,w}}^{n}\right] \tag{13}$$

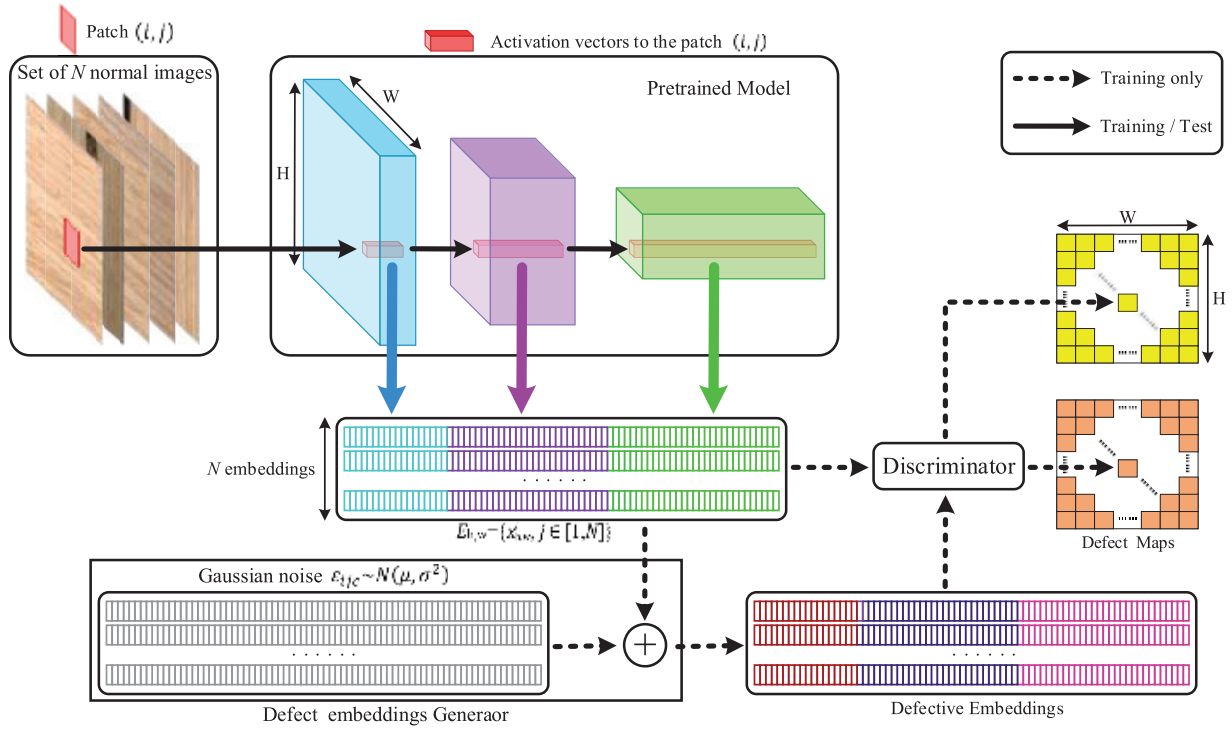


Figure 1: Overview of the proposed FADE algorithm. The diagram illustrates the complete workflow from normal image input through feature extraction, defect embeddings generation with Gaussian noise injection, to defect maps output

FADE synthesizes defect patch embeddings $E^{j-}_{\mathrm{h,w}} \in \mathbb{R}^c$ to approximate the distribution of defect feature patches. This synthesis occurs through the injection of Gaussian noise vectors $\varepsilon \in \mathbb{R}^c$ into normal patch embeddings. The process is formalized in Eq. (14):

$$E^{j-}_{\mathrm{h,w}} = E^{j}_{\mathrm{h,w}} + \varepsilon \tag{14}$$

The Gaussian noise vector ε is sampled from a multivariate normal distribution $\mathcal{N}(0, \sigma^2)$. This process creates synthetic defect features by controlled perturbation of normal embeddings. A smaller σ value produces subtle deviations that may simulate minor defects, while larger values create more obvious anomalies. Feature-space changes better preserve the underlying structure and statistical features of real defects.

Subsequently, both categories of patch embeddings are processed through the discriminator Dis_ψ to compute positional normality scores at coordinates (h, w). This study employs a two-layer multilayer perceptron (MLP) as the discriminator architecture. The normality evaluation is expressed as $Dis_\psi(\mathrm{h}, \mathrm{w}) \in \mathbb{R}^c$. The defect score of the discriminator is formulated as Eq. (15):

$$s^{j}_{\mathrm{h,w}} = -Dis_\psi\left(E^{j}_{\mathrm{h,w}}\right) \tag{15}$$

Employing truncation parameters $th^+ = 0.5$ and $th^- = -0.5$ to mitigate overfitting, the local unsupervised loss function is defined based on the discriminator response as Eq. (16):

$$l^{j}_{unsup}(\mathrm{h}, \mathrm{w}) = \max\left(0, th^+ - s^{j}_{\mathrm{h,w}}\right) + \max\left(0, -th^- - s^{j-}_{\mathrm{h,w}}\right) \tag{16}$$

Instead of generating defects in image space, FADE uses its enhanced capacity to create defective samples directly in feature space. This approach enables the distribution of defect features to better align with real-world patterns, leading to more precise decision boundaries in the model's discrimination process. Additionally, by extracting multi-scale features across the CNN, FADE captures both textural details and semantic information while adapting to domain-specific distributions that typically challenge pre-trained networks.

Following the removal of the noise generator and feature discriminator, N normal training images are processed through the feature extraction network to obtain patch embeddings $E_{\mathrm{h,w}} = \left\{x_{\mathrm{h,w}}^{j}, j \in [1, N]\right\}$. $E_{\mathrm{h,w}}$ is assumed to follow a multivariate Gaussian distribution $\mathcal{N}(\mu_{\mathrm{h,w}}, \Sigma_{\mathrm{h,w}})$. $\mu_{\mathrm{h,w}}$ represents the sample mean. The sample covariance matrix $\Sigma_{\mathrm{h,w}}$ is computed as Eq. (17):

$$\sum_{\mathrm{h,w}} = \frac{1}{N-1}\sum_{j=1}^{N}\left(E_{\mathrm{h,w}}^{j} - \mu_{\mathrm{h,w}}\right)\left(E_{\mathrm{h,w}}^{j} - \mu_{\mathrm{h,w}}\right)^{T} + \epsilon\, I \tag{17}$$

A regularization term εI is incorporated to ensure the full rank and invertibility of the sample covariance matrix $\Sigma_{\mathrm{h,w}}$. At each spatial position (h, w), the estimated multivariate Gaussian distribution $\mathcal{N}(\mu_{\mathrm{h,w}}, \Sigma_{\mathrm{h,w}})$ captures multi-level feature representations. The covariance matrix $\Sigma_{\mathrm{h,w}}$ encodes the inter-level feature correlations. During the defect detection phase, patches at position (h, w) in the test image are quantitatively evaluated using the Mahalanobis distance metric $d_{mahal}\left(E_{\mathrm{h,w}}^{j}\right)$. The distance quantifies the statistical disparity between the patch embeddings $E_{\mathrm{h,w}}^{j}$ of the test sample and the established normal distribution $\mathcal{N}(\mu_{\mathrm{h,w}}, \Sigma_{\mathrm{h,w}})$. The mathematical formulation of $d_{mahal}\left(E_{\mathrm{h,w}}^{j}\right)$ is expressed as Eq. (18):

$$d_{mahal}\left(E_{\mathrm{h,w}}^{j}\right) = \sqrt{\left(E_{\mathrm{h,w}}^{j} - \mu_{\mathrm{h,w}}\right)^{T}\sum_{\mathrm{h,w}}^{-1}\left(E_{\mathrm{h,w}}^{j} - \mu_{\mathrm{h,w}}\right)} \tag{18}$$

The resulting defect score map $S = (d_{mahal}(E_{\mathrm{h,w}}^{j}))_{1<\mathrm{h}<\mathrm{H},1<\mathrm{w}<\mathrm{W}}$ quantifies the deviation of each local region from the normal distribution. The score map magnitudes quantify defect occurrence probabilities, where elevated values indicate higher defect likelihood.

4.2 Federated Learning with Adaptive Multi-Model Embeddings

FLAME (Federated Learning with Adaptive Multi-Model Embeddings) is a federated learning framework that enables collaborative defect detection, using knowledge distillation with multi-model embeddings to address data heterogeneity across distributed clients. FLAME differs from traditional federated learning algorithms by using multi-model embeddings knowledge distillation instead of simple parameter averaging. It preserves historical model information for temporal consistency across learning rounds. FLAME operates at the embedding level for better feature alignment across diverse data distributions. Analysis in Section 4.1 reveals that embeddings encompass multi-level structured features. This results in global model embeddings that surpass local models in representational power and completeness.

As shown in Fig. 2, FLAME optimizes local training by analyzing knowledge similarities between models. The process begins with the central server distributing the global model to all participating clients. Each client then trains the model using their local private data. Clients enhance their models through knowledge distillation, incorporating information from both global and historical models. The updated local models are sent back to the server. The server then aggregates these models to create an improved global model. Each client maintains three parallel model branches. These include the historical local model, the current global model, and the current local model. Each model processes the same input data to generate embeddings. Knowledge transfer occurs through the Embeddings-KD loss. This loss measures the similarity

between feature representations. The framework combines this with unsupervised learning loss to guide local model training. The approach alleviates data heterogeneity challenges while preserving privacy.

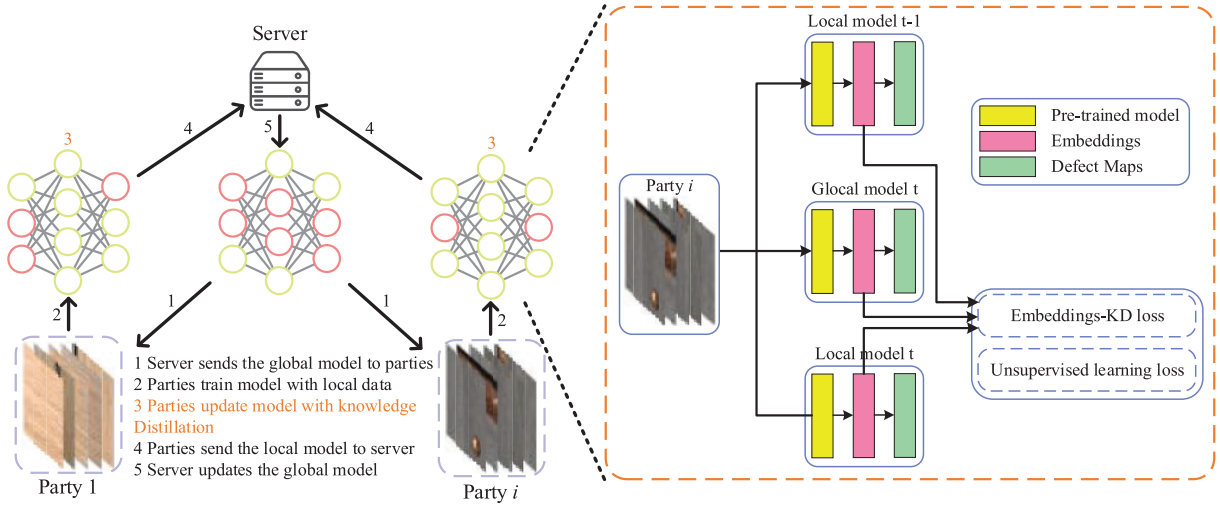


Figure 2: Overview of the proposed FLAME framework. Left: federated learning workflow between server and distributed clients with numbered communication steps. Right: client-side multi-model knowledge distillation mechanism integrating historical, global, and current local models

The composite loss function of FLAME incorporates two components: the unsupervised learning loss l_{unsup} from FADE and the distillation loss l_{kd} derived from multi-model embeddings. During C_i's local training process, w_{loc}^{k} denotes the current round's local model under training, w_{glo}^{k} represents the aggregated global model of the current round, and w_{loc}^{k-1} indicates the historical local model from the preceding round. These three models process the training dataset x_{train} in parallel, which can be formulated as Eq. (19):

$$E_{(\mathrm{h,w})}^{j,k} = R_{w^k}\left(x_j\right) \tag{19}$$

$E_{loc(\mathrm{h,w})}^{j,k}$, $E_{glo(\mathrm{h,w})}^{j,k}$ and $E_{loc(\mathrm{h,w})}^{j,k-1}(k \geq 1)$ represent the embeddings extracted from the training set by the current local model, global model, and historical local model, respectively. FLAME's optimization objective includes two aims: minimizing the distance between the current local embeddings $E_{loc(\mathrm{h,w})}^{j,k}$ and global embeddings $E_{glo(\mathrm{h,w})}^{j,k}$, maximizing the distance between the current local embeddings $E_{loc(\mathrm{h,w})}^{j,k}$ and its historical counterpart $E_{loc(\mathrm{h,w})}^{j,k-1}$ simultaneously. The embeddings distillation loss function is formulated as Eq. (20):

$$l_{kd} = -\log\left(\frac{\exp\left(\mathrm{KL}\left(E_{loc(\mathrm{h,w})}^{j,k}, E_{loc(\mathrm{h,w})}^{j,k-1}\right)/\tau\right)}{\exp\left(\mathrm{KL}\left(E_{loc(\mathrm{h,w})}^{j,k}, E_{glo(\mathrm{h,w})}^{j,k}\right)/\tau\right) + \exp\left(\mathrm{KL}\left(E_{loc(\mathrm{h,w})}^{j,k}, E_{loc(\mathrm{h,w})}^{j,k-1}\right)/\tau\right)}\right) \tag{20}$$

where τ denotes a temperature coefficient that regulates the magnitude of the embeddings distillation loss. When the knowledge representations between the local and global models achieve sufficient convergence such that $E_{loc(\mathrm{h,w})}^{j,k} = E_{glo(\mathrm{h,w})}^{j,k}$, the knowledge distillation loss converges to a fixed constant. Under these conditions, as model heterogeneity diminishes, FLAME becomes equivalent to the classical FedAvg algorithm,

which theoretically demonstrates the convergence stability of FLAME. The complete loss function for the input data x_{train} is formulated as Eq. (21):

$$l_{total} = l_{unsup}\left(w_{loc}^{k}; x_{train}\right) + \mu l_{kd}\left(w_{loc}^{k}; w_{loc}^{k-1}; w_{glo}^{k}; x_{train}\right) \tag{21}$$

where μ denotes a hyperparameter that regulates the distillation loss. The FLAME framework is illustrated in Algorithm 1. In each round, clients train locally through two phases. Firstly, they use FADE to extract features and generate defect embeddings from unlabeled data. Secondly, they perform knowledge distillation across current local, global, and historical local models. The dual optimization mechanism of FLAME serves two critical purposes: ensuring domain adaptability in feature extraction and mitigating the non-IID data challenge through knowledge distillation. The server aggregates these parameters based on each client's data volume. This weighting method gives proportional influence to clients with larger datasets. The framework repeats the training, aggregation, and distribution cycle for a fixed number of communication rounds. The final model achieves improved detection across diverse industrial environments.

Algorithm 1: The FLAME framework

Input: number of communication rounds K, number of parties N, local epochs E, temperature τ
Output: The final model w^K

Server execution:
Initialize w^0
for each round $k = 0, 1, \ldots, K-1$ **do**
  for each party $i = 1, 2, \ldots, N$ **in parallel do**
    send the global model w^k to P_i
    $w_i^k \leftarrow$ **PartyLocalTraining** $\left(i, w^k\right)$
    $w^{k+1} \leftarrow \sum_{i=1}^{N} \frac{|D^i|}{|D|} w_i^k$
return w^K

PartyLocalTraining $\left(i, w^k\right)$:
$w_i^k \leftarrow w^k$
for each epoch e = $1, 2, \ldots, E$ **do**
  for x **in** data_loader
  $loss_{unsup} \leftarrow$ loss_func$(Dis(q), Dis(q_-)).mean()$
  $loss_{kd} \leftarrow -\log\left(\frac{\exp\left(KL\left(E_{loc(\mathrm{h,w})}^{j,k}, E_{loc(\mathrm{h,w})}^{j,k-1}\right)/\tau\right)}{\exp\left(KL\left(E_{loc(\mathrm{h,w})}^{j,k}, E_{glo(\mathrm{h,w})}^{j,k}\right)/\tau\right)+\exp\left(KL\left(E_{loc(\mathrm{h,w})}^{j,k}, E_{loc(\mathrm{h,w})}^{j,k-1}\right)/\tau\right)}\right)$
  $loss_{total} \leftarrow loss_{unsup} + loss_{kd}$
return w_i^k to server

4.3 Computational Efficiency and Scalability Analysis

4.3.1 Computational Complexity of FADE

The computational costs of the FADE algorithm come from three main operations. Feature extraction requires $O(WHL)$ complexity for $H \times W \times 3$ images with L CNN layers. Defect embedding generation via Gaussian noise injection has $O(c)$ complexity. Discriminator operations using a two-layer MLP contribute $O(c^2)$ complexity. During inference, Mahalanobis distance calculations require $O(c^3)$ complexity per spatial

position, though some components can be computed offline. This results in $O(WHc)$ real-time inference complexity for detection.

4.3.2 Communication and Computation Efficiency of FLAME

For N clients each transmitting model parameters of size $|\theta|$ bytes per round, the total communication cost becomes $O(N|\theta|)$. FLAME's multi-model knowledge distillation requires each client to maintain three models in memory, adding $2|\theta|$ bytes of memory overhead per client beyond standard federated learning. The computational cost for knowledge distillation mainly comes from calculating KL divergence between feature embeddings, contributing $O(WHc)$ complexity per training batch.

4.3.3 Scalability Considerations

FLAME's scalability across varying client populations will be demonstrated in Section 5.5.5. The framework's communication complexity scales linearly with participating clients as $O(N|\theta|)$. When implementing client sampling strategies where only a fraction β of clients participate each round, this reduces to $O(\beta N|\theta|)$.

5 Experiment

5.1 Datasets

The heterogeneous data distribution inherent in real-world applications is simulated through the implementation of Dirichlet distribution for non-IID data partitioning. The process involves sampling probability vectors p_k from $Dir_N(\beta)$ [9], where β is the concentration parameter (default: 0.5). Input images undergo standard normalization based on the RGB channel means and standard deviations of the ImageNet dataset. For defective samples, annotation masks are spatially aligned through identical geometric transformations, while all-zero masks are generated by default for normal samples.

The Furniture Board Dataset contains five distinct board categories characterized by unique texture features: Warm White Grid (wwg), Straight Cloud Pattern (scp), Moon Shadow White Oak (mswo), Huixiang Warm Wood (hww), and Green Grey Wood (ggw). Each category incorporates five representative defect types: break corner (bc), break edge (be), edge indentation (ei), surface indentation (si), and stain (in). The dataset comprises 2250 images, which contains 1950 training data and 300 testing data and each image with a spatial resolution of 288 × 288 pixels.

The Magnetic Tile Defects (MT) dataset contains 1344 images of magnetic tiles with various surface defects commonly encountered in material manufacturing, including 952 normal samples and 392 defective samples [35]. This dataset includes five defect categories: blowhole, break, crack, fray, and uneven. All images were resized to 225 × 225 pixels.

5.2 Evaluation Metrics

The Image-level Area Under the Receiver Operating Characteristic curve (I-AUROC) is calculated using the defect detection score S_{AD} to evaluate image-level detection performance. Given that peak response points occur in defect regions of varying dimensions, the maximum value within the defect map is adopted as the image-wise defect detection score. The computational procedure is defined as Eq. (22):

$$S_{AD}\left(x_j\right) = \max_{(\mathrm{h},\mathrm{w})\in W\times H} d_{mahal}\left(E^{j}_{\mathrm{h},\mathrm{w}}\right) \tag{22}$$

The Pixel-level Area Under the Receiver Operating Characteristic curve (P-AUROC) is calculated using the generated anomaly localization map S_{AL} to evaluate pixel-level defect localization performance. The computational procedure is formulated as Eq. (23):

$$S_{AL}\left(x_j\right) = \left\{ d_{mahal}\left(E^{j}_{\mathrm{h,w}}\right) \mid (\mathrm{h}, \mathrm{w}) \in W \times H \right\} \tag{23}$$

5.3 Implementation Details

FADE is compared with three state-of-the-art approaches including PaDiM, Patchcore and CS-Flow. To test how different backbones impact performance, FADE and PaDiM are trained with different backbones: ResNet18, Wide ResNet-50-2 and EfficientNet-B5. For ResNet implementations, embeddings are extracted from the first three layers. For EfficientNet-B5 implementation, embeddings are extracted at layers 7, 20, and 26 to achieve multi-scale integration and high spatial resolution. During defect embedding generation, normal embeddings are disturbed with Gaussian noise $\mathcal{N}(0, \sigma^2)$, where σ is set to 0.015 by default.

FLAME is compared with five state-of-the-art approaches including FedAvg, FedProx, FedNova, SCAFFOLD and MOON. A baseline approach named SOLO is also evaluated, where each party trains a model with its local data without federated learning. SGD optimizer with a learning rate 0.01 is applied for all methods. SGD weight decay is set to 0.00001 and the SGD momentum is set to 0.09. The batch size is set to 32. The number of local epochs is set to 200 for SOLO and 10 for all federated learning approaches unless explicitly specified. For FLAME, the distillation parameter μ is set to 0.5 by default. MOON uses temperature parameter μ = 1.0. FedProx uses hyperparameter μ = 0.01. All federated models use ResNet18 backbone by default. The ablation studies are conducted following the methodology in federated learning work [36].

The experimental evaluations were conducted on a computing platform equipped with an RTX 4090 GPU (24 GB VRAM, Santa Clara, CA, USA), an AMD EPYC 7453 14-core processor (Santa Clara, CA, USA), and 64.4 GB RAM.

5.4 Experiment Result

5.4.1 Defect Detection of FADE

As illustrated in Table 1, FADE improved detection performance across all backbone networks for the five board categories. Compared to PaDiM which directly uses pre-trained models, FADE achieved improvements of 7.34% and 0.66% in average I-AUROC and P-AUROC metrics, respectively. EfficientNet-B5 achieved the best I-AUROC of 93.62% among all backbone architectures, indicating that deeper networks extract more effective features. Patchcore achieves an average I-AUROC of 89.78%, which is 3.1% lower than FADE with ResNet18. CS-Flow attains an average I-AUROC of 88.84%, falling 4.04% behind FADE's performance. These comparisons further validate FADE's effectiveness in capturing discriminative feature representations.

Table 1: Performance comparison of different defect detection methods on board defect detection, with metrics reported as I-AUROC%/P-AUROC%. The table compares FADE across three backbone networks against PaDiM, Patchcore, and CS-Flow

Model	ResNet18		Wide ResNet-50-2		EfficientNet-B5		Patchcore	CS-Flow
Type	FADE (Ours)	Padim	FADE (Ours)	Padim	FADE (Ours)	PaDiM		
wwg	85.8/98.7	81.1/97.7	87.2/98.9	85.6/97.3	87.5/98.6	85.8/97.5	86.3/98.1	84.7/97.8
scp	91.7/99.2	87.1/99.1	93.5/99.1	94.8/98.7	93.6/99.4	92.3/98.9	91.1/99.3	88.8/98.8
mswo	99.9/99.3	98.6/98.7	98.7/99.5	97.2/98.9	98.8/99.5	97.9/98.6	96.5/98.4	97.6/98.3
hww	88.7/99.2	79.8/98.9	89.4/98.7	90.1/99.2	90.9/99.3	88.6/98.8	82.2/96.6	86.9/97.5

(Continued)

Table 1 (continued)

Model	ResNet18		Wide ResNet-50-2		EfficientNet-B5		Patchcore	CS-Flow
Type	FADE (Ours)	Padim	FADE (Ours)	Padim	FADE (Ours)	PaDiM		
ggw	98.3/99.6	81.1/98.3	96.8/99.3	94.5/98.6	97.3/98.7	96.4/99.1	92.8/99.4	86.2/96.9
Average	92.88/99.20	85.54/98.54	93.12/99.10	92.44/98.54	93.62/99.10	90.20/98.58	89.78/98.36	88.84/97.86

5.4.2 Defect Localization of FADE

As shown in Fig. 3, FADE demonstrates excellent detection performance across diverse board categories and defect types. Experiment results show FADE accurately localizes defects of different types and sizes. For extensive surface defects such as 'ei', the generated heatmap achieves comprehensive coverage with precise boundary demarcation. Microscopic defects are precisely localized by FADE while keeping background response minimal. FADE's effective defect localization stems from two factors: adaptive feature mechanisms that improve feature distribution modeling for more discriminative representations, and feature-space defect synthesis that avoids image-space generation, enabling more accurate discriminant boundary learning.

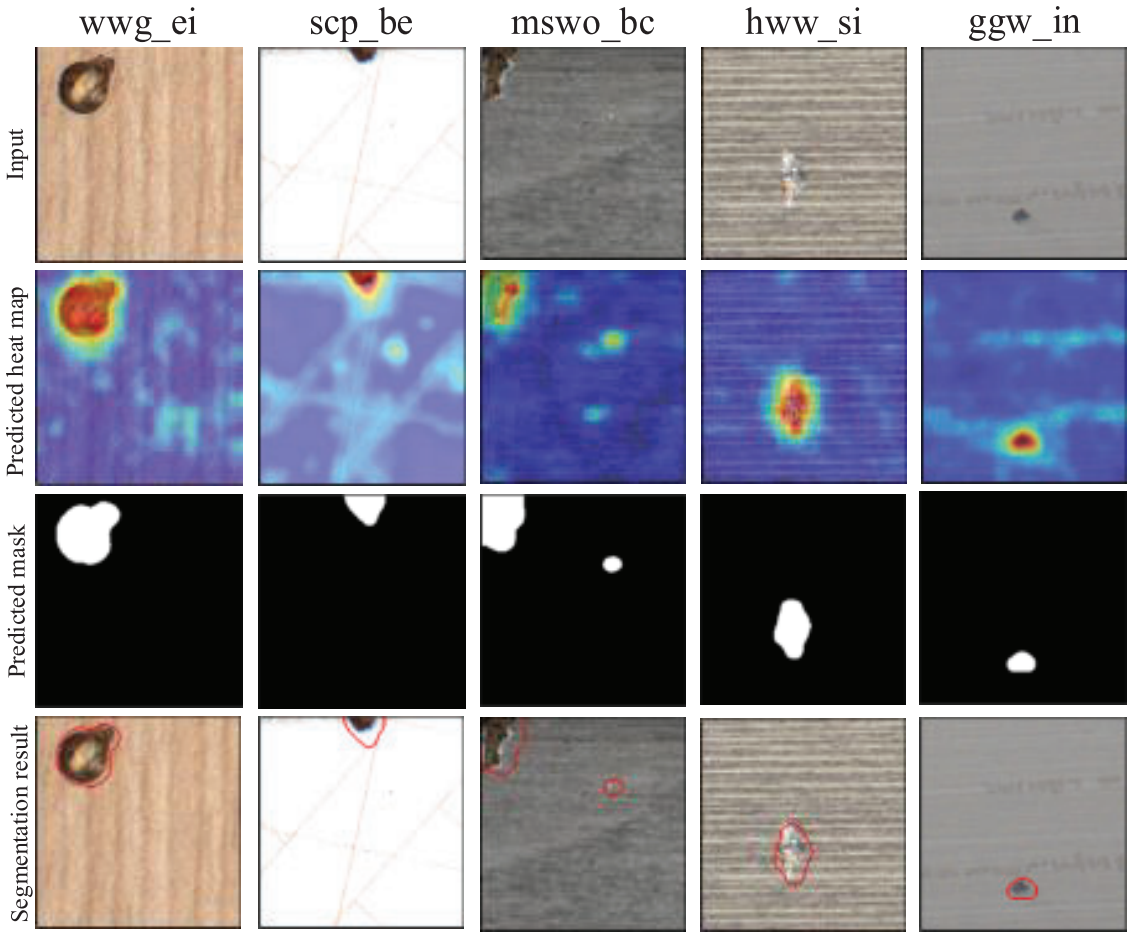


Figure 3: Detection visualization of five distinct defect categories in the board defect dataset. The nomenclature follows 'board-category_defect-type' format, where 'wwg_ei' denotes an 'ei' defect on 'wwg' textured board

5.4.3 Defect Detection of FLAME

The experimental results in Table 2 demonstrate a consistent performance improvement pattern as network complexity increases. As shown in the architectural progression of ResNet18, Wide ResNet-50-2, and EfficientNet-B5, a steady enhancement in detection capabilities becomes evident. With the baseline ResNet18 architecture, FLAME achieves an average I-AUROC of 89.16%, already outperforming other federated learning approaches. When using the wider and deeper Wide ResNet-50-2, FLAME shows a slight improvement with average I-AUROC increasing to 89.44%. However, the most significant performance gain is observed with EfficientNet-B5. Using this architecture, FLAME attains 91.42% average I-AUROC.

Table 2: Detection performance comparison between FLAME and other federated frameworks, with metrics expressed as I-AUROC%/P-AUROC%

Type	FLAME (Ours)	FedAvg	FedProx	FedNova	SCAFFOLD	MOON	SOLO
				ResNet18			
wwg	85.1/97.7	83.8/97.9	84.0/97.9	84.5/98.1	82.0/97.6	84.3/97.3	73.2 ± 0.25/85.9 ± 0.23
scp	91.2/99.0	90.3/99.1	90.0/99.2	90.8/99.2	88.5/98.8	90.6/99.1	75.8 ± 0.21/86.3 ± 0.24
mswo	98.8/98.9	98.7/98.9	98.6/98.9	98.6/99.0	98.4/98.8	98.7/98.9	82.4 ± 0.15/89.8 ± 0.22
hww	83.7/98.9	81.9/98.8	83.6/98.3	82.1/98.4	81.2/98.6	82.8/98.7	72.9 ± 0.28/85.5 ± 0.19
ggw	87.0/98.4	85.7/98.1	85.7/98.2	86.9/98.2	84.2/98.4	86.2/98.3	76.5 ± 0.32/87.2 ± 0.31
Average	89.16/98.58	88.08/98.56	88.38/98.50	88.58/98.58	86.86/98.44	88.52/98.46	76.16/86.94
				Wide ResNet-50-2			
wwg	85.1/98.2	83.7/97.9	84.4/98.1	83.0/97.8	83.6/96.9	84.2/98.0	74.6 ± 0.27/86.2 ± 0.21
scp	90.5/98.1	90.9/98.6	91.7/98.7	90.1/98.8	90.8/98.7	91.2/98.6	77.3 ± 0.24/87.5 ± 0.26
mswo	98.7/99.5	97.2/99.3	98.0/99.1	96.5/99.2	97.1/99.3	97.5/99.3	81.8 ± 0.18/88.9 ± 0.23
hww	86.3/98.3	84.7/98.0	84.5/98.2	83.9/97.9	84.6/98.1	86.1/97.4	75.9 ± 0.31/86.8 ± 0.28
ggw	86.6/98.3	85.3/98.4	88.1/98.3	84.5/98.3	87.7/98.4	87.3/98.2	73.5 ± 0.22/85.7 ± 0.25
Average	89.44/98.48	88.36/98.44	89.34/98.48	87.60/98.40	88.76/98.28	89.26/98.30	76.62/87.02
				EfficientNet-B5			
wwg	85.5/98.6	84.2/97.8	84.1/98.2	83.8/98.0	84.8/97.9	84.8/98.2	75.2 ± 0.23/86.5 ± 0.25
scp	91.6/98.9	89.5/98.9	91.8/98.7	90.4/98.8	87.2/98.4	91.0/98.8	78.4 ± 0.19/88.1 ± 0.22
mswo	98.8/99.5	96.3/99.1	95.2/99.2	96.5/99.0	96.6/99.2	96.8/99.0	82.5 ± 0.16/90.2 ± 0.20
hww	88.9/98.3	86.6/98.5	85.2/98.4	86.1/98.6	85.8/98.6	87.4/98.5	76.1 ± 0.29/87.2 ± 0.24
ggw	92.3/98.8	88.8/98.6	90.5/98.7	88.9/98.5	89.2/98.5	91.8/98.6	75.8 ± 0.25/88.4 ± 0.28
Average	91.42/98.82	89.08/98.58	89.36/98.64	89.14/98.58	88.72/98.52	90.36/98.62	78.20/88.08

As shown in Fig. 4, the performance gap between FLAME and other methods widens as architectural complexity increases. With EfficientNet-B5, FLAME surpasses FedAvg by 2.34%, compared to just 1.08% with ResNet18. This suggests that FLAME's multi-model embedding knowledge distillation mechanism better leverages the enhanced representational capacity of these networks. These findings highlight that FLAME maintains its effectiveness across all tested architectures. However, its full potential is best realized with more advanced backbone networks which can extract richer feature representations from industrial data.

5.4.4 Defect Localization of FLAME

Fig. 5 illustrates defect localization results across federated methods, showing progression from original images to predicted heatmaps, masks, and final segmentation. In the first row, FLAME generates heatmaps with strong defect highlights and suppressed backgrounds, facilitating effective segmentation. Other federated methods show excessive sensitivity to plate texture variations, incorrectly identifying defect-free regions as defective. The third row displays segmentation maps that highlight the detected defect regions.

MOON successfully detects defects at multiple locations simultaneously just like FLAME. However, in terms of boundary localization precision, MOON doesn't achieve the same level of accuracy as FLAME, whose segmentation results demonstrate superior alignment with the actual defect boundaries.

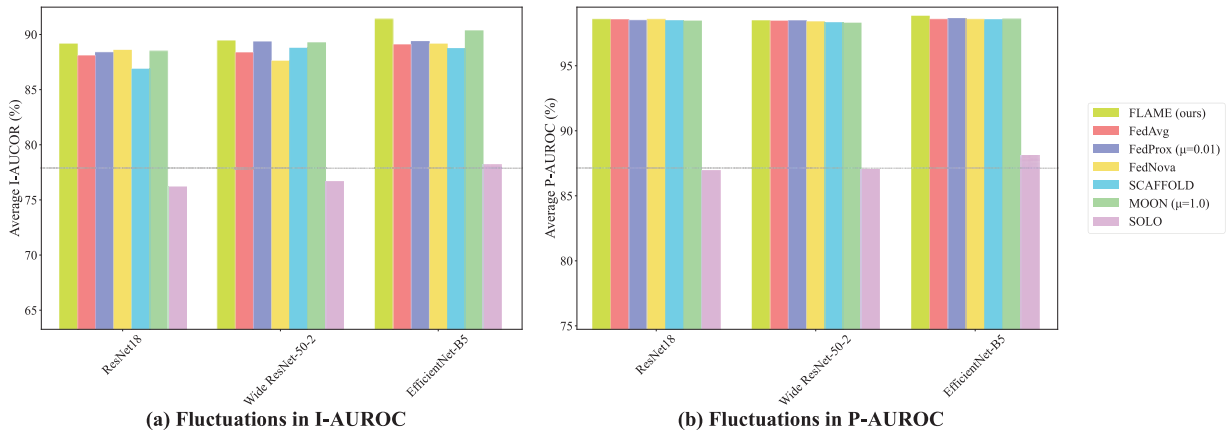


Figure 4: Comparative performance analysis of federated learning methods across different backbone architectures

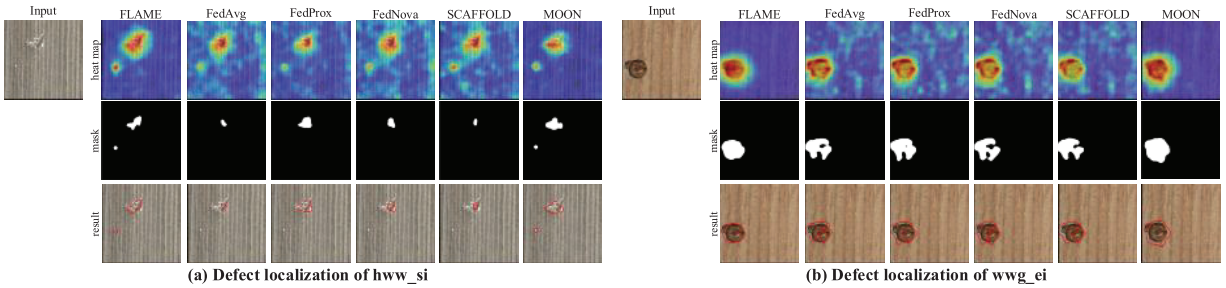
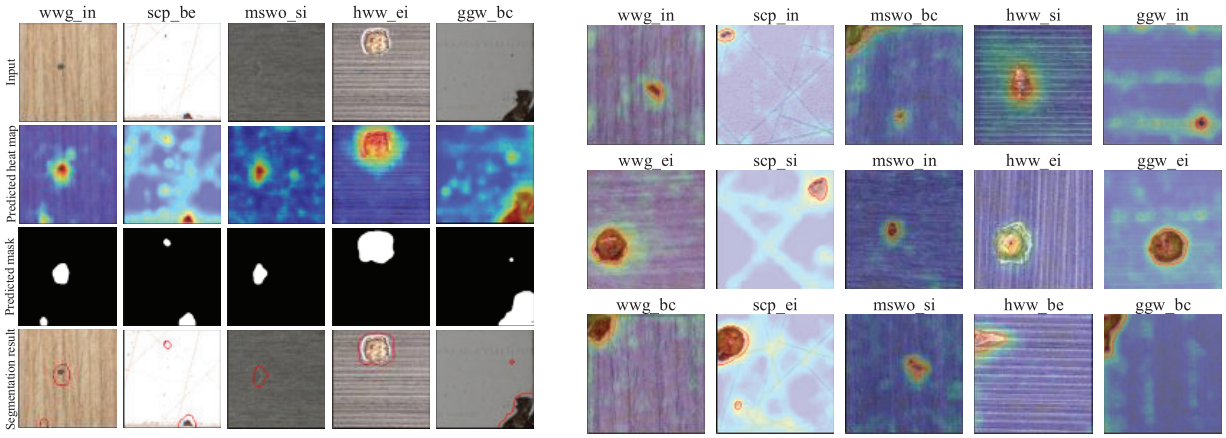


Figure 5: The defect localization performance of different federated learning methods. The nomenclature follows 'board-category_defect-type' format, where 'hww_si' denotes an 'si' defect on hww textured board

FLAME's defect localization performance across diverse board typologies is visualized in Fig. 6a. It can be observed that the method accurately identifies the locations of various defects regardless of their size, demonstrating detection capabilities for both microscopic anomalies and larger structural flaws. This superior performance stems primarily from its novel multi-model embeddings knowledge distillation mechanism. Through feature-level knowledge distillation from global, current local, and historical local models, FLAME achieves two key benefits: enhanced stability and discrimination in feature learning, while minimizing distribution shifts caused by non-IID data. Additional board defect detection diagrams are shown in Fig. 6b. These diagrams demonstrate the effect of superimposing the predicted heat maps and segmentation diagrams on the original images.



(a) The defect localization performance of FLAME **(b) The defect localization visualization results of FLAME**

Figure 6: The defect localization performance of FLAME. The nomenclature follows 'board-category_defect-type' format, where 'wwg_in' denotes an 'in' defect on 'wwg' textured board

5.4.5 Validation on Magnetic-Tiledefects-Datasets

As shown in Table 3, FADE demonstrates strong performance on the MT dataset across different backbone architectures. With EfficientNet-B5, FADE achieves 84.44% I-AUROC and 97.30% P-AUROC, showing improvements of 3.84% and 1.64% respectively compared to the ResNet18 implementation. In the federated learning setting, FLAME shows effective performance with EfficientNet-B5 reaching 79.36% I-AUROC and 87.44% P-AUROC. Due to the non-IID data distribution inherent in federated learning environments, performance metrics show some reduction compared to centralized training approaches. This phenomenon stems from the heterogeneity of magnetic tile defects across different clients, which complicates the knowledge aggregation process.

Table 3: Performance of FADE and FLAME on the MT dataset across different backbone networks, with metrics expressed as I-AUROC%/P-AUROC%

Type	FADE			FLAME		
	ResNet18	Wide ResNet-50-2	EfficientNet-B5	ResNet18	Wide ResNet-50-2	EfficientNet-B5
Blowhole	78.3/93.6	81.5/96.2	83.8/98.4	73.2/86.5	75.9/89.8	77.5/92.3
Break	83.5/97.9	85.2/99.3	85.7/99.5	75.6/80.4	79.8/83.2	82.3/85.7
Crack	79.2/94.7	82.1/97.0	82.3/98.9	74.1/87.9	76.4/80.6	78.2/83.1
Fray	85.1/99.6	86.8/91.2	88.6/92.3	76.4/81.6	80.5/84.3	82.0/86.9
Uneven	76.9/92.5	77.6/95.1	81.8/97.4	71.9/85.8	74.2/88.5	76.8/89.2
Average	80.60/95.66	83.64/95.76	84.44/97.30	74.24/84.44	77.36/85.28	79.36/87.44

The defect localization visualization results of FADE on the MT dataset are illustrated in Fig. 7. For compact defects like blowholes, FADE demonstrates sensitivity in detecting these subtle anomalies. Meanwhile, for extensive defects such as frays, the method maintains consistent detection accuracy across the entire affected region.

FLAME demonstrates superior detection capabilities for intricate defects in Fig. 8. It precisely delineates the contours of challenging linear structures like cracks. It also accurately captures irregular edge patterns in defects such as frays. This enhanced performance on boundary-complex defects can be attributed to the knowledge distillation mechanism that effectively integrates multi-perspective feature representations

across distributed clients. The performance on the MT dataset demonstrate that the approaches effectively generalize beyond the furniture manufacturing domain to other industrial contexts.

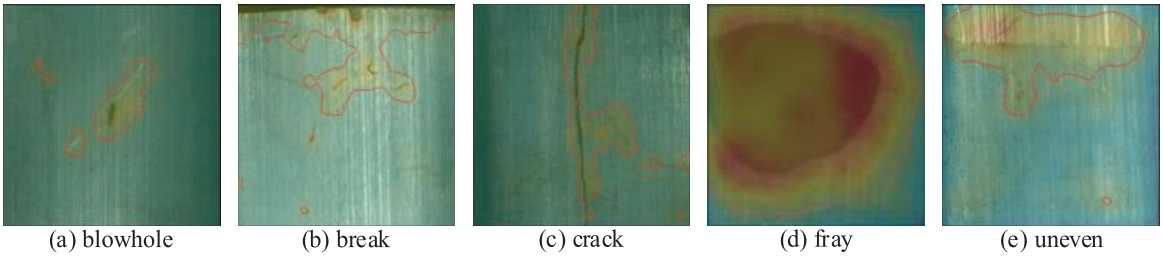


Figure 7: Defect localization visualization results of FADE on the MT dataset

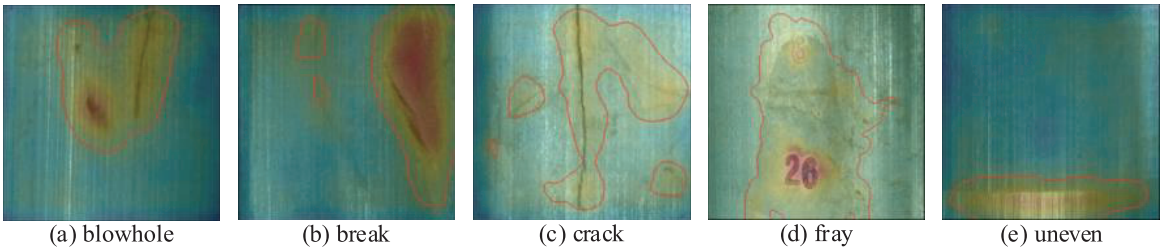
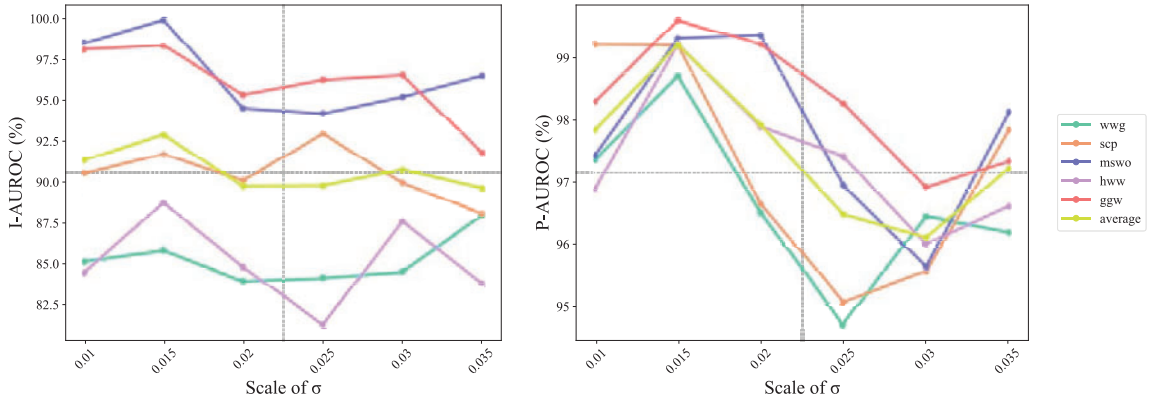


Figure 8: Defect localization visualization results of FLAME on the MT dataset

5.5 Ablation Study

5.5.1 Effects of Noise

As shown in Fig. 9, the variation of FADE's I-AUROC and P-AUROC metrics under different values of σ is illustrated. The optimal configuration at σ = 0.015 attains superior performance metrics with average I-AUROC and P-AUROC scores of 92.88% and 99.2%. These metrics represent substantial improvements of 3.28% and 3.09% compared to alternative noise configurations. Experimental results show that selecting appropriate noise can improve the detection performance and decision boundary learning ability.



(a) Fluctuations in I-AUROC **(b) Fluctuations in P-AUROC**

Figure 9: The variation of FADE's I-AUROC and P-AUROC under different values of σ

5.5.2 Effects of Knowledge Distillation Temperature μ

As demonstrated in Fig. 10, FLAME exhibits optimal convergence behavior at $\mu = 0.5$, reaching steady state after approximately 110 communication rounds with a minimal terminal loss of 0.66. At $\mu = 0.1$, the diminished knowledge distillation constraint leads to degraded convergence rates and elevated terminal loss values. Elevated distillation parameters ($\mu = 1.0$, 2.0, and 5.0) impose excessive regularization constraints, leading to convergence instability during model optimization. These findings show an optimal distillation parameter ($\mu = 0.5$) balances local optimization objectives and knowledge transfer, enabling efficient convergence and information propagation.

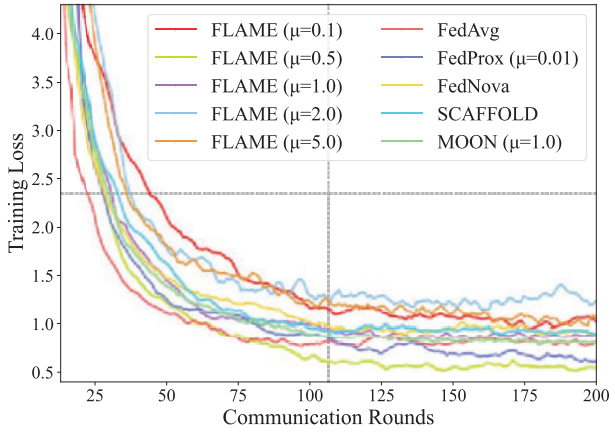


Figure 10: Loss convergence of FLAME across distillation temperature μ configurations

5.5.3 Effects of Heterogeneity

As shown in Fig. 11, Lower β values increase distributional skewness and cross-client heterogeneity. Under substantial data heterogeneity ($\beta = 0.1$), FLAME maintains robust performance through achieving convergence stability at approximately 90 communication rounds with a terminal loss value of 0.87. All methods show unstable behavior under high data heterogeneity, with SCAFFOLD and FedNova showing particularly poor convergence. At $\beta = 5.0$, the variations in convergence behavior among federated learning approaches become more apparent. FLAME demonstrates superior performance by achieving a training loss of 0.23 in contrast to SCAFFOLD's convergence at 1.12.

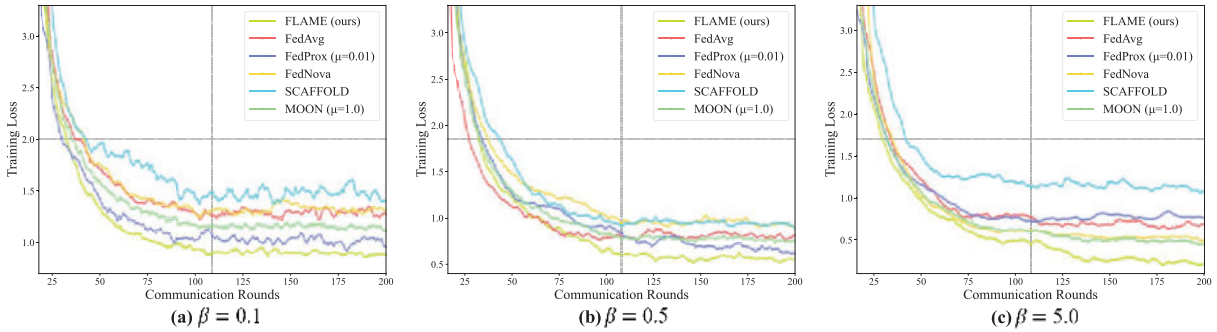


Figure 11: Loss convergence dynamics of FLAME under distinct heterogeneity configurations

As shown in Table 4, FLAME demonstrates superior performance across all three imbalanced data distributions compared to baseline methods. At β = 5.0, FLAME attains an average I-AUROC of 92.73%, surpassing the FedAvg algorithm by 2.73%. Compared to FedProx specialized in heterogeneity handling, FLAME achieves a 1.18% enhancement in average I-AUROC. Empirical results demonstrate FLAME's potential in mitigating data heterogeneity through its feature-level knowledge distillation mechanism, which reduces local-global model disparities.

Table 4: Performance of federated learning methods under different distributions of client data, with metrics expressed as average I-AUROC%/P-AUROC%

Method	β = 0.1	β = 0.5	β = 5.0
FLAME (ours)	83.19/97.75	89.23/98.72	92.73/99.12
FedAvg	78.71/97.23	87.89/97.78	92.25/98.76
FedProx	81.44/97.88	88.11/98.14	91.55/98.24
FedNova	79.12/96.98	87.13/97.69	91.75/98.54
SCAFFOLD	77.85/95.13	85.87/97.43	90.91/98.79
MOON	82.32/97.68	89.76/98.11	91.57/98.66
SOLO	69.34 ± 2.1/81.39 ± 1.7	76.62 ± 1.5/87.98 ± 1.3	81.77 ± 1.2/93.26 ± 1.1

5.5.4 Effects of Sampling Ratio

Training loss convergence was compared between different federated learning methods at sampling ratios of 1.0, 0.5, and 0.2. As shown in Fig. 12, a lower sampling rate results in higher convergence volatility due to reduced data participation per iteration. When a sampling ratio of 1.0 was used, FLAME exhibits a convergence behavior which reaches steady state after approximately 95 iterations where the terminal loss value stabilizes at 0.55.

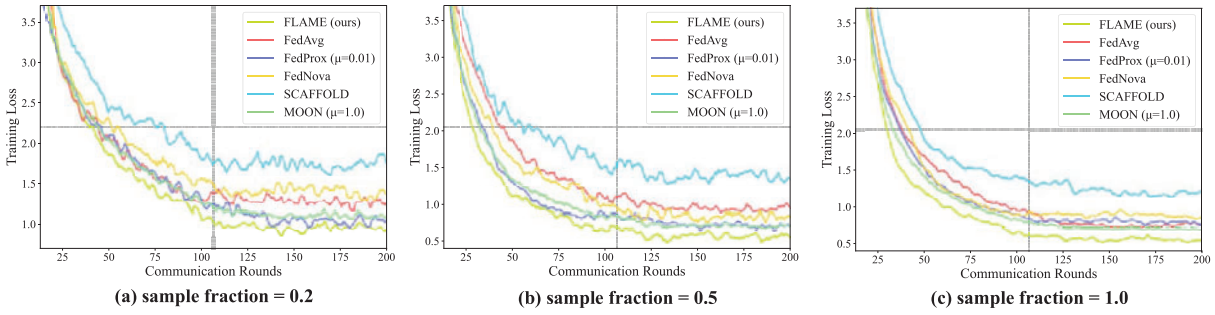


Figure 12: Loss convergence characteristics of FLAME under differential sampling rates

Even at a minimal sampling ratio of 0.2, FLAME exhibits superior convergence characteristics and achieves notably lower training loss values compared to baseline methods. In contrast, SCAFFOLD shows severe instability under low sampling ratios, while FedAvg and FedNova achieve convergence but with higher training loss values. These observations demonstrate FLAME's capacity to maintain robust performance across varying levels of participant engagement.

5.5.5 Effects of Scalability

To evaluate FLAME's scalability, experiments were conducted under two distinct configurations: (1) a distributed setup with 50 participants, each actively engaged in every communication round; (2) a larger-scale deployment with 100 participants, where 20% are randomly selected for participation in each round. As illustrated in Table 5, FLAME attained average I-AUROC values of 79.85% and 82.29% under configurations of 50 participants with 200 communication rounds and 100 participants with 500 communication rounds, respectively.

Table 5: Comparison of FLAME with different federated learning methods across client scales and rounds, with metrics expressed as average I-AUROC%/P-AUROC%

Method	Parties = 50		Parties = 100	
	100 Rounds	200 Rounds	300 Rounds	500 Rounds
FLAME (ours)	74.26/93.62	79.85/95.92	79.53/95.45	82.29/96.42
FedAvg	69.15/91.54	75.76/92.74	74.67/92.32	78.14/94.49
FedProx	71.42/92.92	77.98/94.96	77.95/92.73	79.47/91.86
FedNova	72.92/92.48	75.84/93.82	71.42/93.19	75.25/91.76
SCAFFOLD	66.62/90.46	72.95/91.63	70.81/88.28	74.57/92.58
MOON	72.01/92.54	77.41/93.88	76.41/91.82	79.58/95.62
SOLO	58.22 ± 6.32/86.98 ± 4.87		72.83 ± 5.92/83.45 ± 2.85	

Fig. 13 shows performance evolution across experimental configurations. Initial training shows slower convergence due to high embeddings distillation loss from heterogeneous knowledge distribution. As communication rounds increase, the model improves anomaly pattern recognition, ultimately surpassing baselines. FLAME consistently outperforms FedAvg and FedProx by 4.15% in average I-AUROC across both configurations.

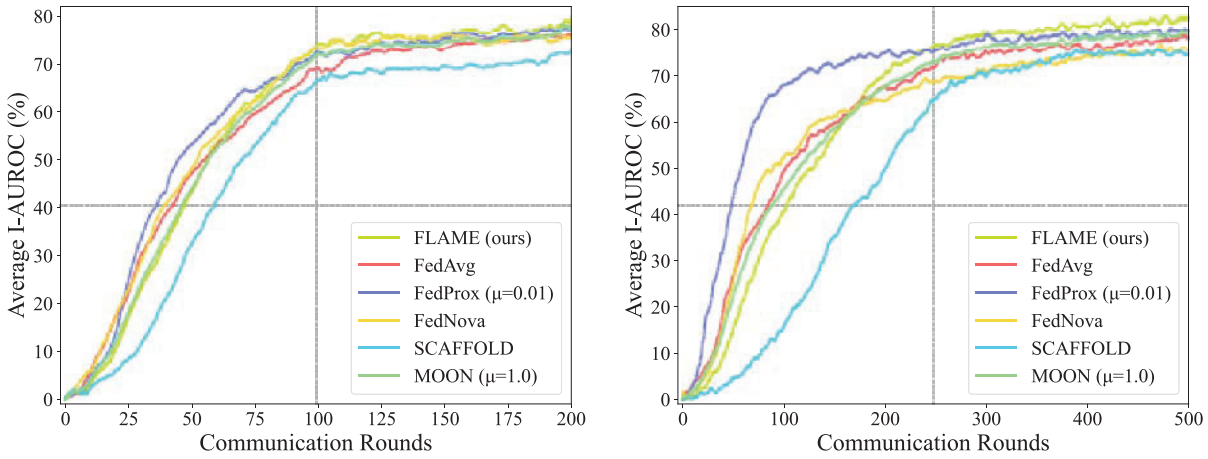


Figure 13: Average I-AUROC convergence curves across communication rounds for different federated learning methods under varying client scales

5.5.6 Effects of Local Epochs

As shown in Fig. 14. FLAME outperforms other federated learning approaches across various local training configurations. All federated learning methods show reduced performance with fewer local iterations due to insufficient use of private datasets during optimization. FLAME achieves peak performance with an I-AUROC of 89.16% when configured for 10 local training rounds, but performance declines beyond this point due to increasing divergence between local and global optimization paths. Despite this decline, FLAME still maintains superior performance with an I-AUROC of 86.87% compared to alternatives. These results demonstrate FLAME's effectiveness in reducing local overfitting by utilizing both global and historical model knowledge.

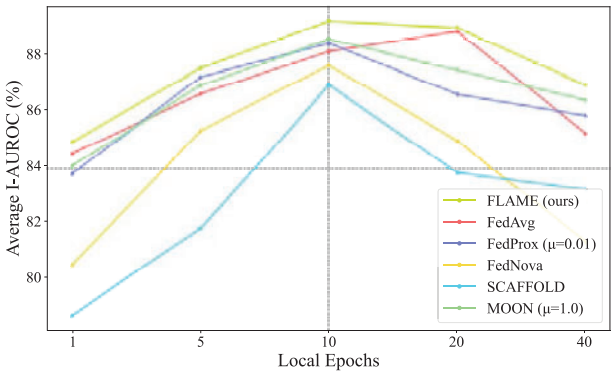


Figure 14: Effect of local epochs on average I-AUROC in federated learning

6 Conclusion

This research proposes an unsupervised learning method FADE and a distributed anomaly detection framework FLAME. FADE enhances feature extraction by synthesizing defect samples, eliminating the need for labeled data. FLAME employs adaptive embeddings distillation by leveraging feature-level knowledge from global and historical models to mitigate performance degradation caused by data heterogeneity. On the board defect detection dataset, experimental results demonstrate that FADE achieves an I-AUROC improvement of 7.34% compared to baseline pre-trained models. FLAME demonstrates superior detection and localization performance in federated settings, outperforming the classic FedAvg algorithm by 2.34% in average image-level AUROC, while also exhibiting better convergence stability. Future work will integrate the extraction-amplification-fusion mechanism to enhance FADE and FLAME. Key improvements include suppressing background noise, amplifying tiny defect features, and leveraging multi-scale fusion, thereby advancing FADE's synthesis capability and FLAME's accuracy in distributed anomaly detection.

Acknowledgement: The authors thank the financial support of the Open Subject Foundation of Ji-Hua Laboratory, China, the National Natural Science Foundation of China, the Guangdong Basic and Applied Basic Research Foundation, China; the Key Research Projects of Ordinary Universities in Guangdong Province, China. We also wish to thank the anonymous reviewers for their kind advice.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grants 32171909, 52205254, 32301704; the Guangdong Basic and Applied Basic Research Foundation under Grants 2023A1515011255, 2024A1515010199; the Scientific Research Projects of Universities in Guangdong Province under Grants 2024ZDZX1042, 2024ZDZX3057; the Ji-Hua Laboratory Open Project under Grant X220931UZ230.

Author Contributions: Conceptualization, Jinhai Wang, Lufeng Luo; methodology, Jinhai Wang, Hui Xiao, Huiling Wei, Mingyou Chen; software, Junwei Xue; validation, Junwei Xue; formal analysis, Jinhai Wang, Mingyou Chen; investigation, Hongyan Zhang, Mingyou Chen; resources, Hongyan Zhang, Jiang Liao; data curation, Hongyan Zhang, Jiang Liao; writing—original draft preparation, Junwei Xue; writing—review and editing, Jinhai Wang, Hui Xiao, Huiling Wei; visualization, Lufeng Luo; supervision, Huiling Wei, Lufeng Luo; project administration, Jinhai Wang, Lufeng Luo; funding acquisition, Lufeng Luo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The result data used to support the findings of this study are available from the corresponding author upon request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Wang H, He X, Zhang C, Liang X, Zhu P, Wang X, et al. Accelerating surface defect detection using normal data with an attention-guided feature distillation reconstruction network. Measurement. 2025;246(2):116702. doi:10.1016/j.measurement.2025.116702.
2. Li H, Wang C, Liu Y. Aircraft skin defect detection based on Fourier GAN data augmentation under limited samples. Measurement. 2025;245(12):116657. doi:10.1016/j.measurement.2025.116657.
3. Cao H, Peng X, Shi F, Tian Y, Kong L, Chen M, et al. Advances in subsurface defect detection techniques for fused silica optical components: a literature review. J Mater Res Technol. 2025;35(18):809–35. doi:10.1016/j.jmrt.2025.01.045.
4. Huang Z, Zhang C, Ge L, Chen Z, Lu K, Wu C. Joining spatial deformable convolution and a dense feature pyramid for surface defect detection. IEEE Trans Instrum Meas. 2024;73(5):1–14. doi:10.1109/TIM.2024.3370962.
5. Bergmann P, Fauser M, Sattlegger D, Steger C. MVTec AD—a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA. doi:10.1109/CVPR.2019.00982.
6. Defard T, Setkov A, Loesch A, Audigier R. Padim: a patch distribution modeling framework for anomaly detection and localization. In: Proceedings of the International Conference on Pattern Recognition 2021; 2021 Jan 10–11; Milan, Italy. doi:10.1007/978-3-030-68799-1_35.
7. Hsu TMH, Qi H, Brown M. Measuring the effects of non-identical data distribution for federated visual classification. arXiv:1909.06335. 2019.
8. Roth K, Pemula L, Zepeda J, Schölkopf B, Brox T, Gehler P. Towards total recall in industrial anomaly detection. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. doi:10.1109/cvpr52688.2022.01392.
9. Rudolph M, Wehrbein T, Rosenhahn B, Wandt B. Fully convolutional cross-scale-flows for image-based defect detection. In: Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2022 Jan 3–8; Waikoloa, HI, USA. doi:10.1109/wacv51458.2022.00189.
10. Liu G, Shen W, Gao L, Kusiak A. Active federated transfer algorithm based on broad learning for fault diagnosis. Measurement. 2023;208(7648):112452. doi:10.1016/j.measurement.2023.112452.
11. McMahan B, Moore E, Ramage D, Hampson S, Arcas BA. Communication-efficient learning of deep networks from decentralized data. Proc Mach Learn Res. 2017;54:1273–82.
12. Lin T, Kong L, Stich SU, Jaggi M. Ensemble distillation for robust model fusion in federated learning. Adv Neural Inf Process Syst. 2020;33:2351–63. doi:10.5555/3495724.3495922.
13. Islam F, Raihan AS, Ahmed I. Applications of federated learning in manufacturing: identifying the challenges and exploring the future directions with Industry 4.0 and 5.0 visions. arXiv:2302.13514. 2023.
14. Mehta M, Shao C. Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing. J Manuf Syst. 2022;64(8):197–210. doi:10.1016/j.jmsy.2022.06.010.

15. Zhang F, Kuang K, Chen L, You Z, Shen T, Xiao J, et al. Federated unsupervised representation learning. Front Inf Technol Electron Eng. 2023;24(8):1181–93. doi:10.1631/FITEE.2200268.
16. Sun X, Song K, Wen X, Wang Y, Yan Y. SDD-DETR: surface defect detection for no-service aero-engine blades with detection transformer. IEEE Trans Autom Sci Eng. 2024;22:6984–97. doi:10.1109/TASE.2024.3457829.
17. Chen H, Song K, Cui W, Zhang T, Yan Y, Li J. SRPCNet: self-reinforcing perception coordination network for seamless steel pipes internal surface defect detection. IEEE Trans Ind Inform. 2024;21(1):950–9. doi:10.1109/TII.2024.3470895.
18. Cui W, Song K, Zhang Y, Zhang Y, Lv G, Yan Y. Fine-grained tiny defect detection in spiral welds: a joint framework combining semantic discrimination and contrast transformation. IEEE Trans Instrum Meas. 2025;74:1–15. doi:10.1109/TIM.2025.3551901.
19. Song K, Feng H, Cao T, Cui W, Yan Y. MFANet: multifeature aggregation network for cross-granularity few-shot seamless steel tubes surface defect segmentation. IEEE Trans Ind Inform. 2024;20(7):9725–35. doi:10.1109/TII.2024.3383513.
20. Abati D, Porrello A, Calderara S, Cucchiara R. Latent space autoregression for novelty detection. In: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2019 Jun 15–20; Long Beach, CA, USA. doi:10.1109/CVPR.2019.00057.
21. Schlegl T, Seeböck P, Waldstein SM, Langs G, Schmidt-Erfurth U. f-AnoGAN: fast unsupervised anomaly detection with generative adversarial networks. Med Image Anal. 2019;54(3):30–44. doi:10.1016/j.media.2019.01.010.
22. Liu Z, Zhou Y, Xu Y, Wang Z. Simplenet: a simple network for image anomaly detection and localization. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. doi:10.1109/cvpr52729.2023.01954.
23. Ma D, Fang H, Wang N, Lu H, Matthews J, Zhang C. Transformer-optimized generation, detection, and tracking network for images with drainage pipeline defects. Comput Aided Civ Infrastruct Eng. 2023;38(15):2109–27. doi:10.1111/mice.12970.
24. Jia Y, Chen G, Zhao L. Defect detection of photovoltaic modules based on improved VarifocalNet. Sci Rep. 2024;14(1):15170. doi:10.1038/s41598-024-66234-3.
25. Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. Federated optimization in heterogeneous networks. Proc Mach Learn Syst. 2020;2:429–50.
26. Li X, Jiang M, Zhang X, Kamp M, Dou Q. Fedbn: federated learning on non-iid features via local batch normalization. arXiv:2102.07623. 2021.
27. Wang J, Liu Q, Liang H, Joshi G, Poor HV. Tackling the objective inconsistency problem in heterogeneous federated optimization. Adv Neural Inf Process Syst. 2020;33:7611–23. doi:10.5555/3495724.3496362.
28. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. Scaffold: stochastic controlled averaging for federated learning. Proc Mach Learn Res. 2020;119:5132–43. doi:10.5555/3524938.3525414.
29. Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv:1503.02531. 2015.
30. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H. Training data-efficient image transformers & distillation through attention. Proc Mach Learn Res. 2021;139:10347–57.
31. Shen Z, Xing E. A fast knowledge distillation framework for visual recognition. In: Proceedings of the European Conference on Computer Vision 2022; 2022 Oct 23–27; Tel Aviv, Israel. doi:10.1007/978-3-031-20053-3_39.
32. Hao Z, Guo J, Han K, Hu H, Xu C, Wang Y. Vanillakd: revisit the power of vanilla knowledge distillation from small scale to large scale. arXiv:2305.15781. 2023.
33. Li D, Wang J. Fedmd: heterogenous federated learning via model distillation. arXiv:1910.03581. 2019.
34. Yao D, Pan W, Dai Y, Wan Y, Ding X, Yu C, et al. FedGKD: toward heterogeneous federated learning via global knowledge distillation. IEEE Trans Comput. 2023;73(1):3–17. doi:10.1109/TC.2023.3315066.
35. Huang Y, Qiu C, Yuan K. Surface defect saliency of magnetic tile. Vis Comput. 2020;36(1):85–96. doi:10.1007/s00371-018-1588-5.
36. Li Q, He B, Song D. Model-contrastive federated learning. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. doi:10.1109/CVPR46437.2021.01057.