MMM: A Unified Weakly-supervised Anomaly Detection Framework for Multi-distributional Data

Xu Tan, *Student Member, IEEE,* Junqi Chen, *Student Member, IEEE,* Jiawei Yang, *Senior Member, IEEE,* Jie Chen, *Senior Member, IEEE,* and Susanto Rahardja, *Fellow, IEEE*

***Abstract*—Weakly-Supervised Anomaly Detection (WSAD) has garnered increasing research interest in recent years, as it enables superior detection performance while demanding only a small fraction of labeled data. However, existing WSAD methods face two major limitations. From the data aspect, they struggle to detect anomalies between normal clusters or collective anomalies due to overlooking the multi-distribution and complex manifolds of real-world data. From the label aspect, they fall short of detecting unknown anomalies because of the label-insufficiency and anomaly contamination. To address these issues, we propose MMM, a unified WSAD framework for multi-distributional data. The framework consists of three components: a Multi-distribution data modeler captures latent representations of complex data distributions, followed by a Multiform feature extractor that extracts multiple underlying features from the modeler, highlighting the characteristics of potential anomalies. Finally, a Multi-strategy anomaly score estimator converts these features into anomaly scores, with the aid of a novel training approach with three strategies that maximize the utility of both data and labels. Experimental results showed that MMM achieved superior performance and robustness compared to state-of-the-art WSAD methods, while providing interpretable results that facilitate practical anomaly analysis.**

***Index Terms*—Anomaly detection, Weakly-supervised, Multi-distribution, Adjacency graph, Variational mixture model, Label augmentation.**

I. Introduction

Anomalies, also known as outliers, are defined as the instances that deviate markedly from the general members in a database [1]. These instances typically reflect underlying issues or phenomena worthy of attention, making the identification of such instances a highly meaningful and crucial task, which is known as anomaly detection. Anomaly detection plays essential roles in many applications, such as fraud detection [2], network traffic monitoring [3], medical diagnosis [4], and transportation analysis [5].

Conventionally, anomaly detection methods adopt an unsupervised manner [6], since manually labeling data as anomalies in practical applications is often time-consuming and labor-intensive. However, as unsupervised anomaly detection methods have no prior knowledge of anomaly information, they are prone to false alarms or missed detection, thus limiting their ultimate detection performance [7], [8]. Additionally, in some applications, similar anomalies tend to form clusters [9], which contradicts the density-based assumptions underlying most unsupervised anomaly detection methods, leading to degraded detection effectiveness. To address this issue, Weakly-Supervised Anomaly Detection (WSAD) methods have emerged. These methods assume the availability of a small set of labeled anomalies and utilize these labeled anomalies along with a large amount of unlabeled data for anomaly detection [10]. This assumption is actually more practical in real-world applications, as it reflects the accumulated experience across various application domains, such as diagnosed medical cases or previously discovered and defended cyber attacks. By leveraging this label information, WSAD methods can better distinguish between normal and anomaly data based on application characteristics, thereby achieving improved performance [8]. It is worth noting that, WSAD here is different from weakly-supervised video anomaly detection [11], [12], which defined the inexact video-level label as the weak supervision of frame-level anomaly detection. In this work we focus on WSAD for tabular data.

However, existing WSAD methods typically face two issues. First, from the data aspect, most WSAD methods categorize data into two classes with simply aggregated distributions:

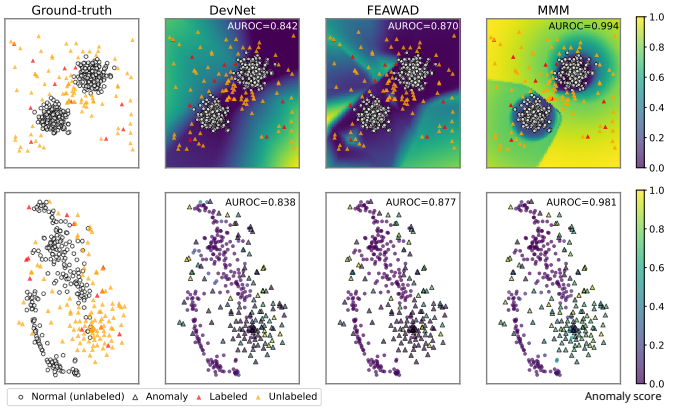

Fig. 1. The upper panel shows synthetic bimodal Gaussian data with anomalies between clusters. The lower panel displays the UCI *Ionosphere* dataset (with dimensional-reduction via t-SNE), where normal data forms complex clusters and anomalies appear grouped nearby. Standard WSAD methods like DevNet [13] and FEAWAD [14], assuming a single distribution, fail to detect such inter-cluster or collective anomalies. In contrast, our proposed MMM effectively detects most anomalies by modeling clusters separately, achieving significantly higher AUROC.

Xu Tan, Junqi Chen, and Jie Chen are with the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an Shaanxi, 710072, P.R.China. E-mails: xutan@ieee.org, jqchen@ieee.org, dr.jie.chen@ieee.org.

Jiawei Yang is with the Department of Computing, University of Turku, 20014, Turku, Finland. E-mail: jiaweiyang@ieee.org.

Susanto Rahardja is with the College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou Zhejiang, 310058, P.R.China. E-mail: susantorahardja@ieee.org.

This work has been co-funded by the European Union's Horizon Europe research and innovation programme for research and innovation 2021-2027 under Marie Skłodowska-Curie grant agreement n° 101126611.

(Corresponding Author: Susanto Rahardja and Jiawei Yang)

normal and anomalous. They overlook the fact that data in real-world scenarios often follows multiple distributions, where normal data frequently contains multiple clusters [15], and anomalies can also comprise multiple types [16]. Without enough labels, these WSAD methods fail to capture and differentiate the characteristics of different data distributions, leading to inaccurate data modeling and missed detection of local anomalies, such as anomalies between two adjacent normal data clusters or collective anomalies [17], as illustrated in Fig. 1. In addition, some applications have complex data distributions with non-convex manifolds [18], which further increase the difficulty of data modeling and anomalies recognition. Second, from the label aspect, existing WSAD methods typically treat all unlabeled data as normal data during training [14], [19], [20]. However, since the quantity of labeled anomalies is significantly smaller than unlabeled data (i.e., label-insufficiency) and unlabeled data contain mixed anomalies (anomaly contamination), this approach leads to false negatives in detecting unlabeled anomalies. As the anomaly label ratio decreases, the performance of these methods deteriorates dramatically [10].

To address these issues, we propose a unified WSAD framework called MMM, which consists of three components: a Multi-distribution data modeler, a Multiform feature extractor, and a Multi-strategy anomaly score estimator. The data modeler, based on a deep variational mixture model using adjacency graphs, reconstructs and clusters unlabeled data while estimating latent distributions of normal and anomaly clusters. By incorporating consistency constraints between adjacent data, it effectively models complex non-convex distributions. The feature extractor derives various underlying features from the modeler that leverage latent information from unlabeled data to characterize different data types. These extracted features refine information that highlights anomalies, facilitating efficient detection by the following estimator. The anomaly score estimator maps these features to anomaly scores with the aid of a small set of labeled anomalies. To better train the estimator, we propose a novel training approach combining self-supervised learning, self-ensemble, and self-smoothing strategies to mitigate label-insufficiency and anomaly contamination, improve model robustness, and enhance anomaly label effectiveness.

In summary, the main contributions of this paper are as follows:

- We propose MMM, a novel WSAD framework for multi-distributional data with complex manifolds. It can identify different types of anomalies by explicitly modeling diverse non-convex distributions within the dataset, and shows superior performance and stability compared with existing WSAD methods.
- A novel training approach with three strategies: self-supervised learning, self-ensemble, and self-smoothing, is proposed to enhance information from abundant unlabeled data and limited labeled anomalies, mitigating the label-insufficiency and anomaly contamination. MMM shows less performance decline with fewer labeled anomalies compared to existing WSAD methods.
- MMM offers a novel perspective on anomaly detection by explicitly considering data structure, anomaly types, and result interpretability, rather than focusing solely on detection accuracy, which is an aspect lacking in other WSAD methods. This can help practitioners to better analyze and understand the location of anomalies, their specific characteristics, and the underlying causes of their occurrence.

II. Related Works

The landscape of WSAD techniques for tabular data has evolved along two primary methodological streams. The first stream focused on feature extraction techniques that leveraged available label information to distinguish normal features from anomalies. Several typical approaches emerged in this direction: DSAD[1] [19] and ESAD [21] both operated by manipulating latent feature spaces to amplify the discrepancy between unlabeled instances and known anomalies. DSAD employed hypersphere-based constraints on the latent space, while ESAD utilized a dual-encoder architecture to optimize reconstruction and feature objectives. REPEN [22] constructed data pairs with different labels through pseudo-labeling and sampling, and employed contrastive learning to differentiate features between normal and anomalous data. Additionally, AA-BiGAN [23] introduced a novel label-aware adversarial training mechanism that steered the generated distribution toward normal patterns while maintaining distance from known anomalies. CAD [24] proposed a two-stage denoising-aware contrastive training framework to learn distinctive representations. Generally, these methods demonstrated better generalization capability as they learned the underlying data characteristics, but their final accuracy was slightly weaker.

The second methodological stream focused on direct anomaly score estimation. Early works proposed by Barbora [25] and Zhao [26] integrated the anomaly scores output by multiple unsupervised detectors, and then employed supervised classification using anomaly labels. DevNet [13] learned to predict Gaussian-distributed anomaly scores, treating scores around the mean value as unlabeled instance scores and scores in the right tail as anomaly instance scores. PReNet [27] created three pairwise data sets, and mapped data pairs to distinct targets based on their label relationships. Typically, these methods achieved better detection performance on known anomaly types through explicit mapping learning of anomaly scores, while their generalization to unknown anomalies remained limited.

Moreover, some methods attempted to combine both streams to benefit from their respective strengths. FEAWAD [14] extracted latent features with an autoencoder, then used the features and reconstruction errors to estimate the anomaly scores. LEDGM [20] employed a Gaussian mixture variational autoencoder to model the latent distributions of normal and anomaly data respectively, and estimated the belonging probabilities as the anomaly scores. RoSAS [28] leveraged mass interpolation to generate synthetic instances and their labels from known anomalies. It employed two networks: one for learning discriminative feature representations by maximizing

[1]All algorithm acronyms follow their original publications.

TABLE I
FEATURES OF THE TYPICAL WSAD METHODS AND MMM

Methods	Feature Learning	Multi-distribution Modeling	Non-convex Manifold	Score Learning	Label Augmentation
DSAD [19]	✓				
ESAD [21]	✓				
REPEN [22]	✓				✓
AA-BiGAN [23]	✓				✓
CAD [24]	✓				✓
DevNet [13]				✓	
PReNet [27]				✓	
FEAWAD [14]	✓			✓	
LEDGM [20]	✓			✓	
RoSAS [28]	✓			✓	✓
WVAD [17]	✓	✓		✓	
TFN [29]	✓	✓		✓	✓
AnoOnly [30]	✓			✓	
MMM*	✓	✓	✓	✓	✓

* denotes the proposed method, which remains consistent throughout.

the separation between anomaly and unlabeled instances, and another for score estimation. WVAD [17] pioneered to model the distributions of different data clusters, and extracted multiple features to estimate the anomaly scores. TFN [29] leveraged fuzzy C-means clustering to establish multiple data prototypes, then employed twin fuzzy neural networks to extract rule-based features and generate pairwise anomaly scores. AnoOnly [30] proposed only to optimize the loss of anomalies while batch-normalizing the features of data, to address the class-imbalance issue.

In this paper, the proposed MMM framework also adopts a combination of both streams. To strengthen existing solutions, we employ more sophisticated modeling techniques to handle multi-distribution data with complex manifolds, and differentiate characteristics of different data types through multiple forms of underlying features. Meanwhile, we introduce three strategies to augment the labels for training the anomaly score estimator, enabling it to better exploit unlabeled data and leverage labeled anomalies for enhanced performance and robustness. Incidentally, some researchers processed multi-distribution data by learning feature prototypes [31], [32]. However, they require full normal/anomaly labels to ensure representative prototypes, which does not conform to weakly-supervised settings. The features of the typical WSAD methods and proposed MMM are listed in TABLE I.

III. METHODOLOGY

A. Problem Statement

The problem of WSAD can be described as follows. Given an unlabeled dataset $\mathcal{X}_{\mathrm{U}} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ and a labeled anomaly dataset $\mathcal{X}_{\mathrm{A}} = \{\mathbf{x}_{N+1}, \cdots, \mathbf{x}_{N+M}\}$, where N and M denote the number of the unlabeled instances and labeled anomalies, respectively. Each data instance in the datasets is a D-dimensional vector. Assuming the number of the unlabeled anomalies is N_{A} in $\mathcal{X}_{\mathrm{U}}$, M satisfies $M \ll N_{\mathrm{A}}$, as the labeled anomalies only account for a small portion of all anomalies. The final training set $\mathcal{X} = \mathcal{X}_{\mathrm{U}} \cup \mathcal{X}_{\mathrm{A}}$. The target is to compute the anomaly score s for each instance in $\mathcal{X}_{\mathrm{U}}$.

B. Overview of MMM

The key idea of the proposed MMM framework is to learn distinguishable features for different data instances according to their latent distributions and reconstruction results, then use these features to detect anomalies with the aid of minor labeled anomalies. As illustrated in Fig. 2, MMM consists of three components: a multi-distribution data modeler, a multiform feature extractor, and a multi-strategy anomaly score estimator. The functions of these components are as follows:

(a) The multi-distribution data modeler aims to individually model the intrinsic data distributions existing in the dataset by simultaneously implementing reconstruction and clustering.
(b) The multiform feature extractor aims to extract various underlying features according to the latent information provided by the multi-distribution data modeler.
(c) The multi-strategy anomaly score estimator aims to learn the mapping mechanism from the extracted features to anomaly scores using limited label information.

The detailed implementations of the three components are described below.

C. Multi-distribution Data Modeler

To synchronously implement reconstruction, clustering, and modeling the distribution for each cluster, the deep variational mixture model is adopted for its intuitive structure [17]. It encodes data to low-dimensional latent distributions, predicts the probabilities that the data belong to different clusters, and then reconstructs data from the latent distributions. However, one main challenge of the general deep variational mixture model is the inability to handle the complex data structures (e.g., non-convex manifolds), thus leading to unreliable features for the subsequent anomaly detection. To address this problem, local-consistency is introduced to ensure that adjacent data have similar latent distributions and cluster assignments [33]. This can be achieved by combing the adjacency graph with the mixture model.

1) Construction of the Digraph: To integrate the local-consistency into the modeler, we first construct the digraph of the original dataset. In view of avoiding the interference of the scattered anomalies from $\mathcal{X}_{\mathrm{A}}$, we use only $\mathcal{X}_{\mathrm{U}}$ to construct the digraph and train the following mixture model. During the digraph construction, we treat each instance in $\mathcal{X}_{\mathrm{U}}$ as a vertex, and connect it to its k-nearest-neighbors under the Euclidean measurement. Consequently, we obtain a digraph with N vertexes and its adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is defined as

$$A_{ij} = \begin{cases} \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right), & \text{if } \mathbf{x}_j \in \mathbf{\Phi}_i, \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where A_{ij} denotes the element in the i-th row and j-th column of $\mathbf{A}$, $\mathbf{\Phi}_i$ denotes the k-nearest-neighbors set of $\mathbf{x}_i$. The edge weight is obtained by applying the Gaussian kernel function to the Euclidean distance between the data pair, resulting in substantially higher weights for adjacent data instances.

2) Construction of the Modeler: The model consists of three modules: an encoder network transforming original data to latent distributions, a classifier outputting the probabilities of assigning data to each cluster, and a decoder reconstructing data from the latent distributions. The encoder and classifier make up the inference module, and then the decoder serves as the generation module.

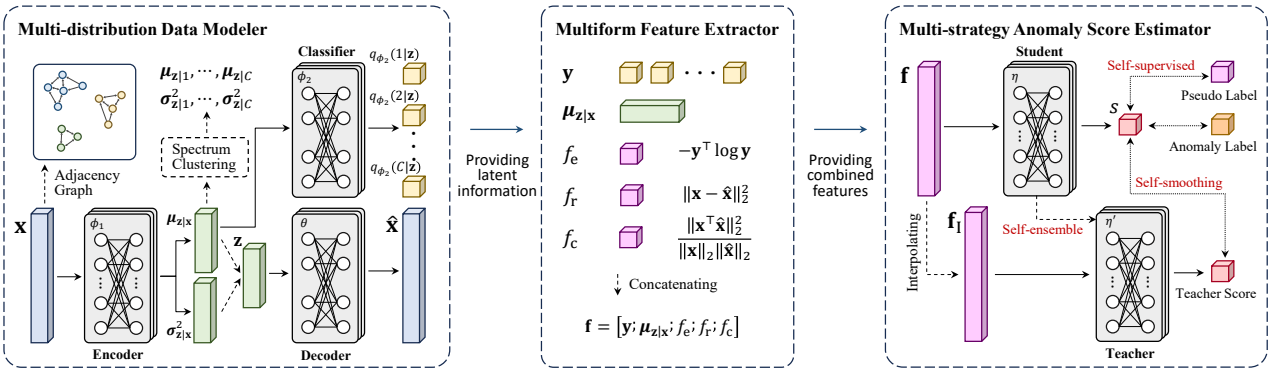

Fig. 2. This diagram illustrates the MMM framework architecture, comprising three components: (1) A multi-distribution data modeler that jointly reconstructs and clusters data while constrained by an adjacency graph for local consistency; (2) A multiform feature extractor that leverages the modeler's latent information to extract and combine diverse underlying features describing data instances; (3) A multi-strategy anomaly score estimator that maps the combined features to final scores, enhanced by unique training strategies maximizing information from all data and labels.

Supposing that the data instances in $\mathcal{X}_{\mathrm{U}}$ come from C different clusters. We assume that each data instance $\mathbf{x}$ is generated from a latent Gaussian mixture model, and the generative process can be described as:

$$\begin{aligned} p_\theta(\mathbf{x}, y, \mathbf{z}) &= p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}|y)p(y), \\ p(y) &\sim \mathrm{Cat}\left(\boldsymbol{\pi}\right), \\ p(\mathbf{z}|y) &\sim \mathcal{N}\big(\boldsymbol{\mu}_{\mathbf{z}|\mathbf{y}}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\mathbf{z}|\mathbf{y}})\big), \end{aligned} \tag{2}$$

where y is a discrete variable representing the cluster assignment, and $\mathbf{z}$ is a continuous variable sampled from the latent distribution of the corresponding cluster. $p_\theta(\mathbf{x}|\mathbf{z})$ denotes the likelihood probability of $\mathbf{x}$, using the decoder with parameters θ to estimate its generative distribution. $p(y)$ follows the prior distribution that follows the categorical distribution with $\boldsymbol{\pi} \in \mathbb{R}^C$ as the parameters. $p(\mathbf{z}|y)$ follows the prior latent distribution of cluster y, which is assumed to be an isotropic Gaussian distribution.

Given that the computation of the true posterior probability $p(y, \mathbf{z}|\mathbf{x})$ of latent variables is intractable, we use the variational inference [34] to approximate the true posterior probability with $q_\phi(y, \mathbf{z}|\mathbf{x})$, which can be formulated as:

$$\begin{aligned} q_\phi(y, \mathbf{z}|\mathbf{x}) &= q_{\phi_2}(y|\mathbf{z})q_{\phi_1}(\mathbf{z}|\mathbf{x}), \\ q_{\phi_1}(\mathbf{z}|\mathbf{x}) &\sim \mathcal{N}\big(\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\mathbf{z}|\mathbf{x}})\big), \\ q_{\phi_2}(y|\mathbf{z}) &\sim \mathrm{Cat}(\widetilde{\boldsymbol{\pi}}|\mathbf{z}), \end{aligned} \tag{3}$$

where $\widetilde{\boldsymbol{\pi}}$ denotes the approximated parameter of the posterior distribution of the cluster assignment. ϕ_1 and ϕ_2 denote the parameters of the encoder and classifier, respectively.

3) Optimization with Local-consistency Regularization: Generally, for each training instance $\mathbf{x}_i$, the original optimization target of the modeler is to minimize the divergence between $q_\phi(y, \mathbf{z}|\mathbf{x}_i)$ and $p(y, \mathbf{z}|\mathbf{x}_i)$, which can be measured by the Kullback-Leibler (KL) divergence:

$$\mathcal{J}_1(\mathbf{x}_i) = \mathrm{KL}[q_\phi(y, \mathbf{z}|\mathbf{x}_i)||p(y, \mathbf{z}|\mathbf{x}_i)]. \tag{4}$$

However, this target neglects the spatial relationships between different instances, causing the model to prioritize global optimization at the expense of local data structures. As a result, when adjacent non-convex data manifolds exist, models biased toward convex cluster shapes induce partitioning errors [33].

To mitigate this limitation, we seek to enforce local-consistency by ensuring that predictions become increasingly similar for proximally located data points. To this end, we imposed the following constraints on latent distributions and cluster assignments based on the adjacency matrix $\mathbf{A}$.

$$\begin{aligned} \mathcal{J}_2(\mathbf{x}_i) &= \textstyle\sum_{j=1}^N W_{ij}\mathrm{KL}[q_\phi(y, \mathbf{z}|\mathbf{x}_j)||p(y, \mathbf{z}|\mathbf{x}_i)], \\ W_{ij} &= A_{ij}/\textstyle\sum_{l=1}^N A_{il}, \end{aligned} \tag{5}$$

where W_{ij} denotes the normalized edge weights for vertex $\mathbf{x}_i$. Consequently, the objective function can be formulated as:

$$\begin{aligned} \mathcal{J}(\mathbf{x}_i) &= \mathcal{J}_1(\mathbf{x}_i) + \mathcal{J}_2(\mathbf{x}_i) \\ &= 2\log p(\mathbf{x}_i) + \mathrm{KL}[q_\phi(y, \mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{x}_i, y, \mathbf{z})] \\ &\quad + \textstyle\sum_{j=1}^N W_{ij}\mathrm{KL}[q_\phi(y, \mathbf{z}|\mathbf{x}_j)||p_\theta(\mathbf{x}_i, y, \mathbf{z})], \end{aligned} \tag{6}$$

where $\log p(\mathbf{x}_i)$ denotes the nature (true) generative probability of $\mathbf{x}_i$, which can be treated as a fixed term and discarded from the objective function. Finally, the objective function can be decomposed as:

$$\begin{aligned} &\mathcal{J}_{\theta,\phi}(\mathbf{x}_i) = -\mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|y, \mathbf{z})] + \mathrm{KL}[q_\phi(y|\mathbf{x}_i)||p(y)] \\ &+ \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z}|y)] - \textstyle\sum_{j=1}^N W_{ij}\mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{x}_j)}[\log p_\theta(\mathbf{x}_j|y, \mathbf{z})] \\ &+ \textstyle\sum_{j=1}^N W_{ij}\big(\mathrm{KL}[q_\phi(y|\mathbf{x}_j)||p(y)] + \mathrm{KL}[q_\phi(\mathbf{z}|\mathbf{x}_j)||p_\theta(\mathbf{z}|y)]\big). \end{aligned} \tag{7}$$

For the prior clustering probability $p(y)$ in this function, instead of setting a shared value for all instances, we use the fused clustering probabilities from the neighborhoods of each instance to further enhance the local-consistency:

$$p(y) = \frac{1}{2}q_{\phi_2}(y|\mathbf{z}_i) + \frac{1}{2}\textstyle\sum_{j=1}^N W_{ij}q_{\phi_2}(y|\mathbf{z}_j). \tag{8}$$

D. Multiform Feature Extractor

Based on the modeler we built, diverse forms of underlying features can be extracted or computed to facilitate AD. In this work, we use five types of features: the latent cluster assignment probability $\mathbf{y}$, the mean of latent distributions $\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}$, the clustering entropy f_{e}, the reconstruction error f_{r}, and the cosine similarity f_{c}. These features contain crucial information relevant to anomaly detection for each instance:

(a) The latent cluster assignment probability $\mathbf{y}$ is a C-dimensional vector, where the c-th element denotes the probability $q_{\phi_2}(y = c|\mathbf{z})$ output by the classifier.

(b) The mean vector $\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}$ of the latent distributions is directly extracted from the output of the encoder $\phi_1(\mathbf{x})$. Vectors belonging to different clusters are relatively far apart from each other.

(c) The clustering entropy f_{e} is computed based on $\mathbf{y}$ as follows:

$$f_{\mathrm{e}} = -\mathbf{y}^{\top} \log \mathbf{y}. \tag{9}$$

This feature quantifies the model's confidence in assigning a data instance to a specific cluster. A high f_{e} value suggests that the instance poorly fits any cluster, indicating a high likelihood of being an anomaly.

(d) The reconstruction error f_{r} is defined as the mean squared error between $\mathbf{x}$ and the decoder's output $\hat{\mathbf{x}}$:

$$f_{\mathrm{r}} = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2. \tag{10}$$

It reflects how accurately the model captures the data's latent distributions and underlying generation mechanisms. A higher f_{r} value indicates an increased probability that $\mathbf{x}$ is an anomaly.

(e) The cosine similarity f_{c} is also computed based on $\mathbf{x}$ and $\hat{\mathbf{x}}$ as follows:

$$f_{\mathrm{c}} = ||\mathbf{x}^{\top}\hat{\mathbf{x}}||_2^2 / \big(||\mathbf{x}||_2 ||\hat{\mathbf{x}}||_2\big). \tag{11}$$

It is a scale-invariant feature that measures the difference between the model's input and output. It helps mitigate the impact of scale variations across different data instances on reconstruction errors. The lower the f_{c}, the more likely that $\mathbf{x}$ is an anomaly.

After extraction and computation, these five features are concatenated, forming a new F-dimensional feature vector

$$\mathbf{f} = [\mathbf{y}; \boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}; f_{\mathrm{e}}; f_{\mathrm{r}}; f_{\mathrm{c}}], \tag{12}$$

and imported to the following anomaly score estimator. Compared to original data, the extracted features refine the information that contributes to highlighting the anomalies, making the following estimator detect them more efficiently. It is worth noting that there are more than five features that are useful for AD, such as the divergence between the posterior and prior latent distributions, and the sample energy [35] of each instance. These features can be beneficial to some specific applications according to the dataset property.

E. Multi-strategy Anomaly Score Estimator

The network architecture of the estimator is a fully-connected neural network whose output is one single score scaled between 0 and 1. A larger score implies a higher probability that the data instance is predicted to be an anomaly.

Given a large unlabeled dataset and a tiny labeled anomaly set, and considering that the anomalies are relatively rare in the unlabeled set, the basic way of training is treating the labeled anomalies as 1-score instances and unlabeled data as 0-score instances. However, this approach leads to three problems: (1) The numbers of two training classes are highly imbalanced, thus the estimator will tend to be biased toward the majority class (i.e., the normal class), and scarifies the predictive accuracy on the anomaly class; (2) The anomalies mixed in the unlabeled set are also treated as normal, which will impede the estimator from learning true normal patterns, and cause more missing alarms; (3) The number of labeled anomalies is small, making the estimator easily overfit these anomalies and lose the generalization to other anomalies.

To solve these problems, we propose a novel training approach with three strategies: self-supervised learning with pseudo scores, self-ensemble with mean-teacher, and self-smoothing with interpolation consistency, to enhance the weakly-supervised learning progress.

1) Self-supervised Learning with Pseudo Scores: Although the ground-truth labels are rare under the scenario of weakly-supervised learning, our modeler is a smart learner that can explore the intrinsic regularity within the unlabeled data. Therefore, we propose a novel strategy that can exploit the information extracted from the modeler to find relatively reliable instances and create pseudo labels for estimator training.

Specifically, we compute the pseudo score $\hat{s}$ for each unlabeled instance $\mathbf{x}$ as follows:

$$\begin{aligned} \hat{s}_{err} &= ||\mathbf{x} - \hat{\mathbf{x}}||_2^2 / a_1, \\ \hat{s}_{ent} &= -\mathbf{y}^{\top} \log \mathbf{y} / a_2, \\ \hat{s} &= (\hat{s}_{err} + \hat{s}_{ent})/2, \end{aligned} \tag{13}$$

where a_1 and a_2 are two scaling factors that scale the corresponding scores between 0 and 1 for all instances in $\mathcal{X}_{\mathrm{U}}$. Apparently, $\hat{s}_{err}$ and $\hat{s}_{ent}$ are equivalent to the reconstruction error and clustering entropy described in Sec. III-D. They both indicate a high likelihood of being an anomaly when their values are large. Therefore, the instance with a smaller $\hat{s}$ is more reasonable to be treated as normal while training the estimator.

By using the approach above, before each estimator training epoch, we form a pseudo-labeling set $\mathcal{X}_{\mathrm{P}}$ by selecting unlabeled instances with the lowest pseudo scores, with these scores as training targets. The pseudo-labeling set is then merged with the labeled anomaly set $\mathcal{X}_{\mathrm{A}}$ to create the training set. Moreover, the size of $\mathcal{X}_{\mathrm{P}}$ progressively increases as the training proceeds, which can be determined as follows:

$$N_P = \ m + \left\lfloor \frac{S \cdot \min(e - e_0,\, E - e_0)}{E - e_0} \right\rfloor \cdot \left\lfloor \frac{N - m}{S - 1} \right\rfloor, \tag{14}$$

where m denotes the minimum amount, S denotes the number of increasing steps, e denotes the current epoch number, e_0 denotes the initial epoch number, and E denotes the maximum epoch number. Then the loss function can be expressed as:

$$\mathcal{J}_{\eta}(\mathbf{x}) = \begin{cases} BCE(\eta(\mathbf{f}|\mathbf{x}), \hat{s}), & \mathbf{x} \in \mathcal{X}_{\mathrm{P}}, \\ BCE(\eta(\mathbf{f}|\mathbf{x}), 1), & \mathbf{x} \in \mathcal{X}_{\mathrm{A}}. \end{cases} \tag{15}$$

Consequently, instances more likely to be normal receive higher training intensity, which enhances the estimator's accuracy in learning normal patterns and alleviates the influence of potential wrong pseudo-labeled anomalies. Meanwhile, by removing the constraint of setting training targets to 0 for unlabeled data, the estimator is prevented from being biased toward the normal class.

2) Self-ensemble with Mean-teacher: As the pseudo-labeling set evolves during training, the estimator exhibits more intense parameter and output perturbations compared to training with a fixed dataset. To alleviate the effect of perturbations and obtain a more robust estimator, we introduce mean-teacher [36] into the model construction.

Mean-teacher serves as a self-ensemble strategy that integrates current and historical network parameters. It treats the original network as the student model, and introduces a new

network with the same architecture as the teacher model. The parameters of the teacher model are the exponential moving average of the student model's parameters:

$$\eta'_s = \alpha\eta'_{s-1} + (1-\alpha)\eta_s, \tag{16}$$

where η'_s and η'_{s-1} denote the parameter states of the teacher model at the current and last training step, respectively. η_s denotes the parameter state of the student model at the current step (after updating). α is a decay factor that controls how much information is reserved from the history. By setting α with a high value, the teacher model only receives limited updates from the current training set, thus reducing the perturbations and improving the robustness.

In addition, a consistency regularization between the output of the teacher and student model is added to the loss function, to increase the stability of the student model:

$$\mathcal{R}_{\eta',\eta}(\mathbf{x}) = \mathrm{BCE}(\eta'(\mathbf{f}|\mathbf{x}), \eta(\mathbf{f}|\mathbf{x})), \tag{17}$$

where $\mathrm{BCE}(\cdot)$ denotes the binary cross-entropy function. After training, only the teacher model is kept for testing.

3) Self-smoothing with Interpolation Consistency: Since the number of anomalies is generally small in AD applications, and the labeled anomalies are even rarer under the WSAD setting, the model can easily overfit the known anomalies and lose the generalization to unknown anomalies. To solve this problem, we utilize the interpolation consistency [37] to train the estimator.

Specifically, we select two instances from the training feature set and combine them linearly to interpolate a new feature:

$$\mathbf{f}_I = \lambda\mathbf{f}_p + (1-\lambda)\mathbf{f}_q, \tag{18}$$

where $\mathbf{f}_p$ and $\mathbf{f}_q$ denote the original features, $\mathbf{f}_I$ denotes the interpolated feature. λ is the interpolation factor that follows the Beta distribution $\mathrm{Beta}(\kappa,\kappa)$. Thereafter, a regularization term is added to encourage consistency between the estimated score of the interpolated feature and the interpolated score of the original features:

$$\begin{aligned} s_I &= \lambda\cdot\eta(\mathbf{f}_p) + (1-\lambda)\cdot\eta(\mathbf{f}_q), \\ \mathcal{R}'_{\eta',\eta}(\mathbf{f}_I) &= \mathrm{BCE}(\eta'(\mathbf{f}_I), s_I). \end{aligned} \tag{19}$$

It is worth noting that, this operation is merged into the regularization term of mean-teacher, which can reduce the computational cost. The interpolation consistency term smooths the regions surrounding known data points with continuous anomaly scores, thereby enhancing the estimator's robustness and generalization capability [38].

F. Training Procedure

In this section, we will introduce the detailed procedure for training the proposed MMM framework.

1) Pretraining and Initialization: Since the likelihood term ($\mathbb{E}[\cdot]$) in $\mathcal{J}(\mathbf{x}_i)$ is weak at the early stage of training, the encoder and decoder network may not correctly approximate the true posterior latent distribution and true generative distribution, leading to undesirable local optima which is hard to escape [39]. Therefore, pretraining and initialization are needed to ensure good reconstruction and clustering results.

In this work, the complete initialization progress consists of three steps. Firstly, the modeler is treated as a vanilla autoencoder (discarding the latent variance estimation and the classifier) and pretrained with the mean squared error function for e_{p0} epochs.

Algorithm 1 Pretraining and Initialization

Input: Unlabeled data and their quantity - $\mathcal{X}_\mathrm{U}$, N
Output: Pretrained data modeler - $\Psi\langle\phi_1,\phi_2,\theta\rangle$

```
Initialize data modeler Ψ⟨φ1, φ2, θ⟩.
for epoch = 1 to e_p0 do
    repeat
        Sample one batch of instances B from X_U.
        Optimize φ1 and φ2 with standard MSE loss.
    until N instances have been sampled.
end for
Obtain latent vectors {μ̂_{z|x}}_{x∈X_U} from the pretrained encoder φ1(X_U).
Compute the adjacency matrix of {μ̂_{z|x}}_{x∈X_U} using (20).
Apply spectral clustering on the adjacency matrix, obtain the clustering
    results {ŷ|x}_{x∈X_U} and prior distributions N(μ_{z|y}, diag(σ²_{z|y})).
for epoch = 1 to e_p1 do
    repeat
        Sample a batch B' from {{μ̂_{z|x}}_{x∈X_U}, {ŷ|x}_{x∈X_U}}.
        Optimize θ with standard cross-entropy loss.
    until N pairs have been sampled.
end for
Initialize (2) with N(μ_{z|y}, diag(σ²_{z|y})).
```

Secondly, spectral clustering [40] is applied to the pretrained latent vectors $\{\hat{\boldsymbol{\mu}}_{\mathbf{z}|\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}_\mathrm{U}}$ to get the initial clustering assignments. Spectral clustering is a graph-based method that can adapt to complex data manifolds. It partitions the entire graph into several subgraphs, aiming to maximize the sum of edge weights within each subgraph while minimizing the sum of edge weights between different subgraphs. The edge weight between each data pair is calculated as follows:

$$e(\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\mu}}_j) = \exp - \|\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j\|_2^2 / 2s_{ij}^2. \tag{20}$$

Since in practical AD datasets, data in different clusters may have different densities, setting a fixed value of s_{ij} for all instances is not a proper choice. Thus in this work, we introduce an adaptive way to determine s_{ij}, motivated by the concept of perplexity in information theory [41]. Specifically, we solve the following system of equations:

$$\begin{cases} a_{ij} = \exp\left(-\|\hat{\boldsymbol{\mu}}_i - \hat{\boldsymbol{\mu}}_j\|_2^2 / 2s_i^2\right), \\ b_{ij} = a_{ij} / \sum_{l=1}^{N} a_{il}, \\ \log_2 h = -\sum_{j=1}^{N} b_{ij}\log_2 b_{ij}, \\ i \neq j, \end{cases} \tag{21}$$

where h denotes the complexity factor, which is a predefined constant; s_i is the unknown to be solved. The system is solved using the Quasi-Newton method, and s_j is solved in the same way. Finally, the value of s_{ij} is determined as follows:

$$s_{ij} = (s_i + s_j)/2. \tag{22}$$

Lastly, the clustering results $\{\hat{y}|\mathbf{x}\}_{\mathbf{x}\in\mathcal{X}_\mathrm{U}}$ are used to pretrain the classifier for e_{p1} epochs. Specifically, the classifier takes $\{\hat{\boldsymbol{\mu}}_{\mathbf{z}|\mathbf{x}}\}_{\mathbf{x}\in\mathcal{X}_\mathrm{U}}$ as the input and $\{\hat{y}|\mathbf{x}\}_{\mathbf{x}\in\mathcal{X}_\mathrm{U}}$ as the targets, with binary cross-entropy as the loss function. In addition, the mean and variance values of each cluster are used as the initial values of the prior distributions $\mathcal{N}\big(\boldsymbol{\mu}_{\mathbf{z}|\mathbf{y}}, \mathrm{diag}(\boldsymbol{\sigma}^2_{\mathbf{z}|\mathbf{y}})\big)$. The complete progress of pretraining and initialization is demonstrated in Algorithm 1.

Algorithm 2 Warming-up and Joint-training

Input: Unlabeled data and their quantity - $\mathcal{X}_{\mathrm{U}}$, N
Labeled anomalies and their quantity - $\mathcal{X}_{\mathrm{A}}$, M
Output: Trained data modeler - $\Psi\langle\phi_1, \phi_2, \theta\rangle$
Trained anomaly score estimator - η'

```
Initialize student anomaly score estimator η.
Initialize teacher anomaly score estimator η'.
Compute the adjacency matrix of X_U using (1).
for epoch = 1 to E do
    if epoch > e_0 then
        Create pseudo-labeled set X_P according to (13), (14).
    else
        Set X_P = X_U with ŝ = 0.
    end if
    repeat
        Sample half a batch of instances B_P from X_P.
        Sample half a batch of instances B_A from X_A.
        Obtain the feature set F from Ψ([B_P, B_A]) using (12).
        Create the interpolation set B_I from F using (18).
        Optimize Ψ and η using loss function (23).
        Update η' using (16).
    until N_P + M instances have been sampled.
end for
```

2) Warming-up and Joint-training: After the pretraining and initialization, three components are combined and trained jointly. The loss function is the combination of (7), (15), and (19):

$$\begin{aligned}\mathcal{J}(\mathbf{x}) = &\frac{1}{B_P}\sum_{i=1}^{B_P}\mathcal{J}_{\theta,\phi}(\mathbf{x}_i) \\ &+\beta\big(\frac{1}{B_{P+A}}\sum_{i=1}^{B_{P+A}}\mathcal{J}_{\eta}(\mathbf{x}_i)+\tfrac{1}{B_I}\sum_{i=1}^{B_I}\mathcal{R}'_{\eta',\eta}(\mathbf{f}_i)\big),\end{aligned} \tag{23}$$

where B_P denotes the number of pseudo-labeled data in a batch, B_{P+A} denotes the summed number of pseudo-labeled data and labeled anomalies in a batch, B_I denotes the number of interpolated features in a batch. β is a weighting parameter that balances the losses from the modeler and estimator.

Given that both the estimator's input and the self-supervised learning process rely on features extracted from the modeler, a stable and well-trained modeler is essential for effective estimator training. We require the model to focus on training the modeler at the early stage, which can be viewed as a warming-up phase. During this phase, β is set to 0.01, and the whole $\mathcal{X}_{\mathrm{U}}$ is treated as $\mathcal{X}_{\mathrm{P}}$ with $\hat{s}=0$ in (15). This phase will last for e_0 epochs, which is equal to that in (14). After that, β is set to 1 and the self-supervised learning part gets up to work normally. The complete progress of warming-up and joint-training is demonstrated in Algorithm 2.

IV. Experiments

In this section, we presented a comprehensive evaluation of MMM on the WSAD task. We evaluated its effectiveness using 11 real-world anomaly detection datasets, each containing diverse manifolds and anomaly types. We compared MMM against 10 state-of-the-art (SOTA) WSAD methods, which we reproduced as baselines. To demonstrate MMM's robustness, we examined its performance across varying anomaly label ratios. Furthermore, we provided visualized examples with systematic analysis to enhance interpretability. We also conducted an ablation study to validate the effectiveness of each training strategy employed in our multi-strategy anomaly score estimator. Finally, we tested the sensitivity of three hyperparameters and scalability of MMM.

TABLE II
Information of 11 anomaly detection datasets

Name	Sample	Dim.	Ratio	Cluster
Letter	1,600	32	6.3%	3
Ionosphere	351	32	35.9%	2
Satellite	6,435	36	31.6%	6
Pima	768	8	34.9%	3
PageBlocks	5,473	10	10.2%	2
Mammography	7,854	6	3.2%	3
Thyroid	7,200	21	7.4%	4
Fraud	10,436	29	4.5%	5
Vowels	1,456	12	3.4%	3
CelebA	9,195	39	5.1%	2
Cardiotography	2,126	21	22.2%	2

A. Datasets

We used 11 public tabular anomaly detection datasets from DAMI[2], ODDS[3], and ADRepository[4] repositories. *Letter* contains synthetic anomalies, while the other datasets include real-world anomalies from practical applications. All datasets feature multiple manifolds and diverse types of anomalies, making them well-suited for evaluating WSAD methods under complex conditions. Additionally, *Mammography*, *Fraud*, and *CelebA* were preprocessed by removing duplicate instances with conflicting labels. Additionally, *Fraud* and *CelebA* were sampled to increase the anomaly ratio to around 5% to introduce greater challenge into the evaluation. The detailed information of these datasets is shown in TABLE II.

B. Baselines and Settings

We adopted 11 SOTA WSAD methods as our baselines. They are DSAD [19], DevNet [13], FEAWAD [14], LEDGM [20], PReNet [27], RoSAS [28], WVAD [17], TFN [29], MTGFLOW [42], AnoOnly [30], and CAD [24]. These methods have been introduced in Sec. II except MTGFLOW, which is originally an unsupervised time series anomaly detection method. We chose it as our baseline since it also utilized clustering to process multi-distributional data, and we modified it slightly to make it adapt WSAD on tabular data. For a fair comparison, we used consistent network architectures for similar functional components across different methods. Specifically, for the encoder component, a network with $[D,\ D/2,\ D/4,\ D/8]$ units in each layer was adopted for *Letter*, *Ionosphere*, *Satellite*, *Thyroid*, *Fraud*, and *Vowels*; a network with $[D,\ 128,\ 128,\ D/2]$ units in each layer was adopted for *Pima*, *PageBlocks*, *Mammography*, *CelebA*, and *Cardiotocography*. Since a stronger network was needed for learning the inherent regularity in these three datasets. For the score estimation component, a network with $[F,\ F\times 2,\ F\times 2,\ 1]$ units in each layer was adopted. Other hyperparameters of the baselines were set to the values recommended by their original papers. The cluster count C used in MMM was determined based on two criteria: (1) 2-dimensional t-SNE [43] visualizations of both original data and their latent representations generated by MMM, and (2) decision graphs derived from a heuristic density-peak-based approach. The decision graphs will be described in detail in Sec. IV-C5. For implementations, we reproduced DSAD, LEDGM, and WVAD

[2] www.dbs.ifi.lmu.de/research/outlier-evaluation/DAMI
[3] odds.cs.stonybrook.edu
[4] github.com/GuansongPang/ADRepository-Anomaly-detection-datasets

TABLE III
AUROC COMPARISONS OF ALL METHODS UNDER AN ANOMALY LABEL RATIO OF 0.1

Methods	Datasets											AVG.
	Letter	Ionosphere	Satellite	Pima	Pageblocks	Mammo.	Thyroid	Fraud	Vowels	CelebA	Cardio.	
DSAD	0.8275±0.0194	0.7753±0.0358	0.7810±0.0583	0.6868±0.0216	0.9595±0.0078	0.8824±0.0145	0.8284±0.0662	0.9339±0.0157	0.9120±0.0302	0.9089±0.0177	0.9131±0.0227	0.8553±0.0282
DevNet	0.7442±0.0273	0.8193±0.0472	0.8243±0.0145	0.7396±0.0405	0.9458±0.0079	0.9209±0.0163	0.8096±0.0408	0.9598±0.0054	0.9609±0.0134	0.9480±0.0032	0.9476±0.0116	0.8745±0.0207
FEAWAD	0.8882±0.0264	0.7832±0.0843	0.8883±0.0128	0.6329±0.0507	0.9532±0.0150	0.8697±0.0268	0.8657±0.0318	0.9561±0.0063	0.9114±0.0552	0.8425±0.0197	0.7406±0.0504	0.8484±0.0345
LEDGM	0.6136±0.0614	0.8203±0.0961	0.8383±0.0901	0.7580±0.0059	0.9615±0.0026	0.8925±0.0189	0.8088±0.0371	0.9623±0.0036	0.7937±0.0687	0.8736±0.0115	0.9359±0.0069	0.8417±0.0366
PReNet	0.6249±0.0539	0.5508±0.0570	0.8045±0.0264	0.5913±0.0861	0.7934±0.0142	0.7686±0.0181	0.9493±0.0141	0.8392±0.0393	0.7448±0.0560	0.5402±0.0361	0.5821±0.0419	0.7081±0.0403
RoSAS	0.7349±0.0225	0.5177±0.0660	0.8796±0.0125	0.6374±0.0190	0.8863±0.0289	0.8971±0.0147	0.9519±0.0177	0.9243±0.0171	0.8580±0.0938	0.8517±0.0221	0.8432±0.0214	0.8165±0.0305
WVAD	0.8819±0.0278	0.9254±0.0223	0.9340±0.0056	0.7296±0.0197	0.9379±0.0092	0.8834±0.0116	0.8784±0.0424	0.9461±0.0098	0.9598±0.0151	0.9164±0.0104	0.9271±0.0078	0.9018±0.0165
TFN	0.7419±0.0042	0.6329±0.0082	0.8734±0.0028	0.7676±0.0020	0.9394±0.0026	0.9252±0.0012	0.8466±0.0015	0.9659±0.0011	0.9685±0.0021	**0.9498±0.0014**	0.9460±0.0009	0.8688±0.0025
MTGFLOW	0.7650±0.0296	0.9426±0.0153	0.3402±0.0690	0.4633±0.1048	0.3413±0.1017	0.4752±0.1551	0.5079±0.1819	0.6611±0.1231	0.2117±0.1224	0.7312±0.0282	0.6117±0.0824	0.5501±0.0921
AnoOnly	0.5606±0.0359	0.8142±0.0379	0.5576±0.0835	0.6707±0.0400	0.9191±0.0122	0.8940±0.0252	0.7887±0.0390	0.9529±0.0079	0.8178±0.0627	0.8952±0.0074	0.8965±0.0138	0.7970±0.0332
CAD	0.7841±0.0269	0.7169±0.0659	0.8629±0.0063	0.6078±0.1599	0.9002±0.0317	0.9060±0.0065	0.8826±0.0072	**0.9675±0.0008**	0.9353±0.0487	0.9415±0.0012	**0.9530±0.0045**	0.8598±0.0327
MMM*	**0.9030±0.0112**	**0.9666±0.0069**	**0.9371±0.0025**	**0.7798±0.0118**	**0.9662±0.0024**	**0.9318±0.0047**	**0.9670±0.0108**	0.9540±0.009	**0.9756±0.0093**	0.9391±0.0028	0.9480±0.0046	**0.9335±0.0069**

Bold: best results; Underlined: second-best results.

TABLE IV
AUPRC COMPARISONS OF ALL METHODS UNDER AN ANOMALY LABEL RATIO OF 0.1

Methods	Datasets											AVG.
	Letter	Ionosphere	Satellite	Pima	Pageblocks	Mammo.	Thyroid	Fraud	Vowels	CelebA	Cardio.	
DSAD	0.4593±0.0369	0.6491±0.0438	0.7013±0.0603	0.5678±0.0184	0.8125±0.0095	0.5360±0.0378	0.4563±0.1065	0.7567±0.0663	0.6303±0.0529	0.3871±0.0533	0.7461±0.0655	0.6093±0.0501
DevNet	0.3403±0.0351	0.7915±0.0392	0.7710±0.0193	0.5737±0.0347	0.7541±0.0245	0.5737±0.0094	0.3434±0.0756	0.8124±0.0543	0.6900±0.0851	0.4520±0.0188	0.8485±0.0196	0.6319±0.0378
FEAWAD	0.5165±0.0537	0.7564±0.0808	0.8484±0.017	0.5395±0.0360	**0.8362±0.0190**	0.5598±0.0400	0.3899±0.1845	0.8441±0.0354	0.6063±0.1601	0.3749±0.0176	0.5962±0.0527	0.6244±0.0633
LEDGM	0.1417±0.0604	0.7469±0.1644	0.7861±0.0830	0.5588±0.0089	0.7560±0.0165	0.5440±0.0454	0.3298±0.0506	0.8313±0.0115	0.1929±0.0898	0.4157±0.0076	0.8384±0.0106	0.5583±0.0499
PReNet	0.2799±0.0443	0.4993±0.0340	0.7805±0.0304	0.5133±0.0582	0.6811±0.0131	0.5232±0.0127	0.8320±0.0111	0.7232±0.0521	0.4964±0.1047	0.2302±0.0119	0.4783±0.0294	0.5488±0.0365
RoSAS	0.4071±0.0346	0.5008±0.0460	0.8284±0.0146	0.5253±0.0187	0.7547±0.0193	0.5785±0.0391	0.8325±0.0312	0.7547±0.0670	0.4843±0.0936	0.3891±0.0177	0.6879±0.0320	0.6130±0.0376
WVAD	0.5649±0.0328	0.8703±0.0223	0.8832±0.0089	0.5700±0.0141	0.7913±0.0128	0.5767±0.0349	0.5420±0.1106	0.8501±0.0105	0.5771±0.1043	0.4268±0.0173	0.8118±0.0141	0.6786±0.0348
TFN	0.3039±0.0057	0.6516±0.0068	0.8181±0.0044	0.5736±0.0032	0.8035±0.0033	0.6431±0.0091	0.4808±0.0048	0.8477±0.0161	0.7196±0.0084	**0.4528±0.0062**	0.8514±0.0027	0.6496±0.0064
MTGFLOW	0.2325±0.0480	0.9313±0.0217	0.2386±0.0240	0.3400±0.0654	0.0740±0.0153	0.0338±0.0120	0.0915±0.0435	0.0759±0.0371	0.0207±0.0025	0.0961±0.0125	0.3184±0.0691	0.2230±0.0319
AnoOnly	0.1742±0.0055	0.7305±0.0442	0.4780±0.1292	0.4915±0.0481	0.6174±0.0611	0.3049±0.0526	0.3193±0.1144	0.6778±0.0309	0.2956±0.1515	0.3180±0.0120	0.6995±0.0433	0.4642±0.0630
CAD	0.4246±0.0476	0.6751±0.0736	0.8169±0.0073	0.4608±0.1167	0.7032±0.0765	0.5724±0.0165	0.5582±0.0157	0.8466±0.0022	0.5923±0.1893	0.4205±0.0059	0.8651±0.0141	0.6305±0.0514
MMM*	**0.6220±0.0202**	**0.9510±0.0115**	**0.8932±0.0053**	**0.6430±0.0219**	0.8134±0.0120	**0.6754±0.0337**	**0.8381±0.0235**	**0.8518±0.0055**	**0.7654±0.0860**	0.4229±0.0135	**0.8652±0.0087**	**0.7583±0.0220**

in PyTorch, while TFN, MTGFLOW, AnoOnly, and CAD were implemented using their official codes. The remaining baselines were implemented using the DeepOD toolkit [44]. The source code of MMM is available on GitHub[5] to ensure reproducibility and facilitate further research.

As for training, we used the Adam [45] optimizer. The number of training epochs E was set to 500, and the batch size was set to $\lfloor (N+M)/10 \rfloor$ for all the methods. To further alleviate the label-insufficiency, we employed balanced sampling to create the batch, which resulted in approximately equal numbers of unlabeled data and labeled anomalies in each batch. For the pretraining, e_{p0} and e_{p1} were set to 100 and 20, respectively. For the data modeler, the neighbor count k was fixed to 10. Then for self-supervised learning, we set $m = \lfloor 0.3N \rfloor$, $S = 8$, and $e_0 = 100$. For self-ensemble, α was fixed to 0.99 for the first e_0 epochs and 0.999 for the remaining epochs. For self-smoothing, κ was fixed to 0.5, and the number of interpolated data was set to be equal the the batch size. β in (23) was fixed to 0.01 for the first e_0 epochs and 1 for the remaining epochs.

We used two most widely used evaluation metrics, the Area Under the Receiver Operating characteristic Curve (AUROC) and the Area Under the Precision-Recall Curve (AUPRC), as our performance metrics. Both of them range from 0 to 1, where 1 indicates the best performance. The relative improvement using in the following sections was calculated by the following equation:

$$\mathrm{RI} = \frac{\mathrm{AUC}_h - \mathrm{AUC}_l}{1 - \mathrm{AUC}_l}, \tag{24}$$

where AUC_h and AUC_l denote the higher and lower AUC metrics, respectively.

[5]https://github.com/Ra1demmmm/MMM

C. Experimental Results

1) Overall Performance: We evaluated the performance of the proposed MMM against 11 baseline methods across 11 datasets, maintaining a fixed anomaly label ratio of 0.1 (defined as the proportion of labeled anomalies during training). All experiments were conducted with 10 independent runs using different random seeds. The mean and standard deviation values for both AUROC and AUPRC metrics were computed across these 10 runs, with results presented in TABLE III and TABLE IV, respectively.

MMM clearly outperformed all baselines, setting a new SOTA for WSAD. It achieved the highest AUROC on 8 of 11 datasets, with comparable results to the top baseline on the remaining three. It also achieved a superior average AUROC of 0.9335, demonstrating a significant relative improvement of 32.3% over the second-best baseline (WVAD), and 47.0% over the third-best baseline (DevNet). Regarding AUPRC, MMM achieved top performance on 9 datasets and ranked second on one, achieving the highest overall average AUPRC of 0.7583, a 24.8% and 31.0% relative improvement over WVAD and TFN, respectively. Although performance on *CelebA* was limited due to its violation of MMM's multi-distributional assumption, the overall results clearly establish MMM as the leading method among current WSAD approaches.

Furthermore, MMM exhibited remarkable stability, as evidenced by its consistently low standard deviations. Among all baselines except TFN, MMM achieved the smallest average standard deviations in both AUROC (0.0069) and AUPRC (0.0220). While TFN exhibited the least fluctuation overall due to its fuzzy-neural-network architecture, which was insensitive to initial states and optimized in closed form, MMM demonstrated superior stability compared to all other conventional neural-network-based WSAD methods.

In summary, MMM demonstrated significant advantages

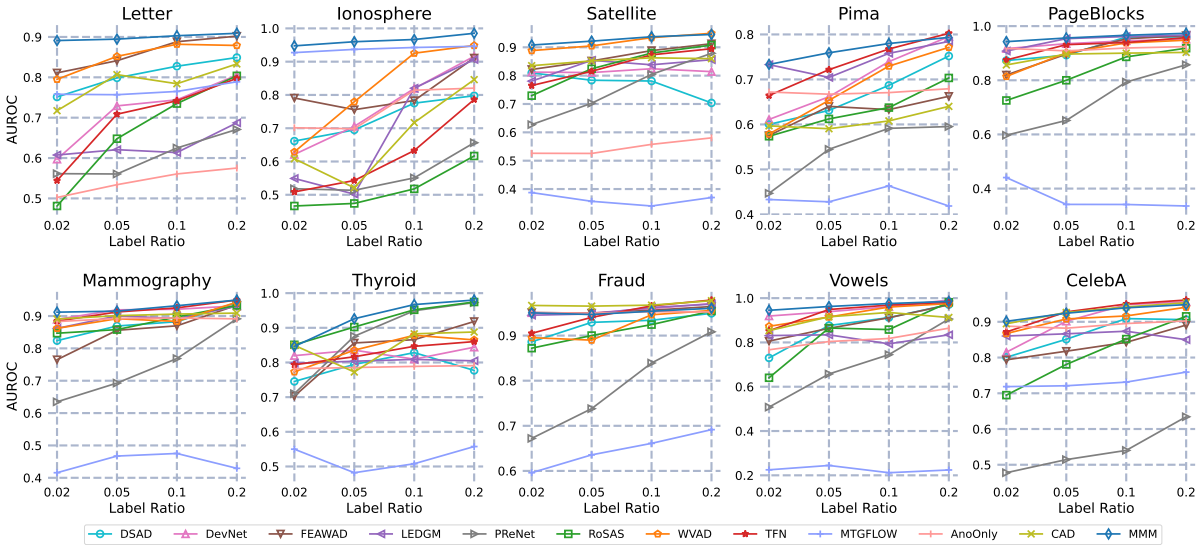

Fig. 3. The AUROC performance of different WSAD methods under the anomaly label ratio of 0.02, 0.05, 0.1, and 0.2, respectively.

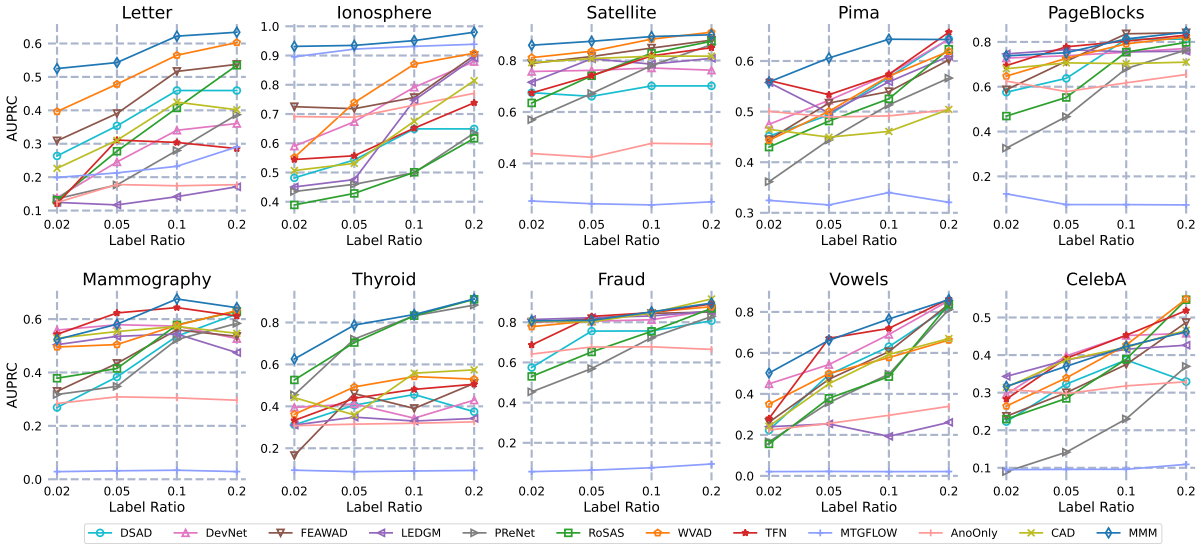

Fig. 4. The AUPRC performance of different WSAD methods under the anomaly label ratio of 0.02, 0.05, 0.1, and 0.2, respectively.

in both performance and stability, establishing itself as an effective, robust, and reliable solution for WSAD applications.

2) Evaluation with Different Anomaly Label Ratios: For WSAD methods, the anomaly label ratio is a crucial factor significantly influencing performance. Therefore, we evaluated all methods across four different anomaly label ratios: 0.02, 0.05, 0.1, and 0.2. Results were illustrated in Fig. 3 and 4.

The figures revealed that the proposed MMM demonstrated superior or comparable performance across all tested anomaly label ratios and datasets, especially on datasets *Letter*, *Ionosphere*, *Satellite*, *Pima*, *Thyroid*, and *Vowels*. Moreover, MMM exhibited significantly less performance degradation as the anomaly label ratio decreased, compared to other baselines. It can be observed that, MMM's AUPRC performance was suboptimal on *Mammography* and *CelebA*. This was attributed to the fact that, within these datasets, certain labeled anomalies exhibited a high degree of similarity to normal data. As a result, the model assigned elevated anomaly scores to some normal data points that were proximate to the anomalies, thereby contributing to a reduction in the AUPRC metric. Nevertheless, MMM remained the strongest method when comprehensively evaluated across all datasets and anomaly label ratios.

Furthermore, we analyzed the average performance across the tested 10 datasets for each method under different anomaly label ratios, as shown in TABLE V. MMM achieved the highest average AUROC and AUPRC scores across all anomaly label ratios. Moreover, as indicated in the "RI" rows, the relative improvement compared to the best baseline increased as the anomaly label ratio decreased, demonstrating MMM's robustness and efficiency in utilizing limited label information.

3) Interpretation on Visualized Examples: As introduced in Sec. III, the proposed method is constructed under the multi-distributional condition, and implemented by joint clustering and scoring. Therefore, MMM is highly interpretive for applications where the data have multiple manifolds, and can handle different types of anomalies. To demonstrate the interpretability, we give some visualized examples here to analyze MMM's behaviors for different situations.

As shown in Fig. 5, datasets *Letter*, *Ionosphere*, and *Page-*

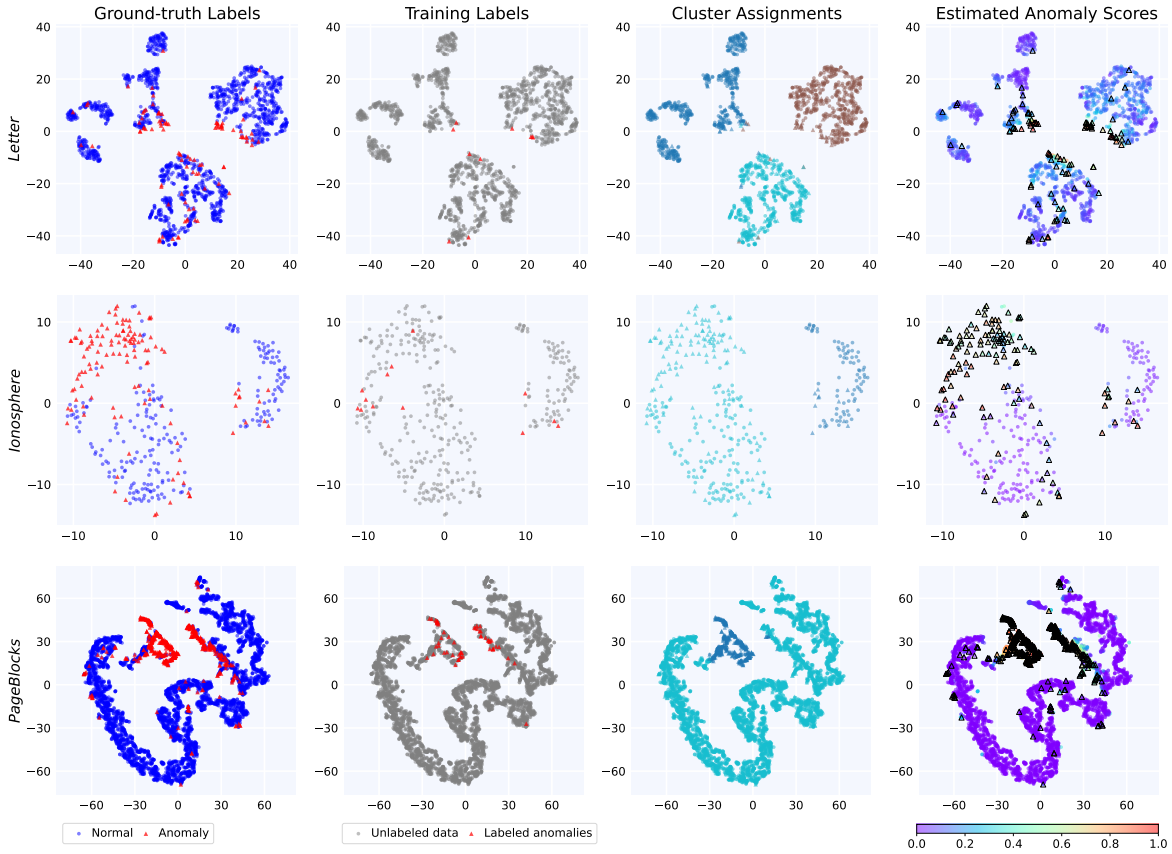

Fig. 5. Visualized examples for interpreting the behaviors of MMM. Three datasets: *Letter*, *Ionosphere*, and *PageBlocks* were visualized in the form of the 2-dimensional scatter plot after dimensional-reduction. From left to right, the figures illustrated the ground-truth labels of data, the labeled anomalies for training, the clustering assignments output by MMM, and the estimated anomaly scores given by MMM, respectively. MMM demonstrated excellent modeling capability for multiple data distributions, along with interpretable estimation of anomaly score distributions.

TABLE V
AVERAGE PERFORMANCE COMPARISONS OF ALL METHODS WITH VARYING ANOMALY LABEL RATIOS ON TESTED DATASETS

Methods	**AVG. AUROC**				**AVG. AUPRC**			
	0.02	0.05	0.10	0.20	0.02	0.05	0.10	0.20
DSAD	0.7681	0.8121	0.8496	0.8607	0.4049	0.5050	0.5956	0.6219
DevNet	0.7953	0.8387	0.8672	0.8976	0.5188	0.5678	0.6102	0.6545
FEAWAD	0.7837	0.8329	0.8591	0.9030	0.4648	0.5638	0.6272	0.6980
LEDGM	0.7874	0.7987	0.8323	0.8590	0.4806	0.5009	0.5303	0.5598
PReNet	0.5753	0.6451	0.7207	0.7971	0.3311	0.4356	0.5559	0.6691
RoSAS	0.6881	0.7664	0.8139	0.8708	0.3875	0.4915	0.6056	0.7241
WVAD	0.7975	0.8531	0.8993	0.9176	0.5099	0.5925	0.6653	0.7113
TFN	0.7664	0.8266	0.8611	0.8969	0.4720	0.5875	0.6295	0.6750
MTGFLOW	0.5451	0.5371	0.5440	0.5521	0.2091	0.2069	0.2134	0.2221
AnoOnly	0.7590	0.7647	0.7871	0.7981	0.4145	0.4112	0.4307	0.4437
CAD	0.8069	0.8147	0.8505	0.8722	0.5001	0.5356	0.6071	0.6420
MMM*	**0.8977**	**0.9166**	**0.9320**	**0.9429**	**0.6387**	**0.6919**	**0.7476**	**0.7771**
RI	47.0%	43.2%	32.5%	30.7%	24.9%	24.4%	24.6%	19.2%

Blocks were visualized in the form of the 2-dimensional scatter plot. Specifically, data were delivered to the encoder to extract the latent distributions, and the mean values were dimension-reduced to 2-dimension using the t-SNE algorithm. The anomaly label ratio was fixed as 0.1. In Fig. 5, the points in the plots were marked with different colors and shapes according to their characteristics. In the first column, the normal data were marked as black dots while anomalies were marked as red triangles, according to the ground-truth labels. In the second column, only the labeled anomalies in the training set were marked as red triangles, while other unlabeled data were marked as gray dots. In the third column, points with different cluster assignments given by MMM were marked in different colors, and all the anomalies were shaped in triangles. In the last column, points were colored according to the anomaly scores estimated by MMM. The higher the score the warmer the color, and the true anomalies were circled with black triangle boxes.

From the figures, we can observe that in dataset *Letter*, the data form three distributions (corresponding to three letters). Anomalies were scattered randomly across these distributions, primarily manifesting as point anomalies. Analysis of MMM's cluster assignments showed that the majority of data points were correctly clustered, with misclassified points consistently being anomalies. The anomaly scores output by the estimator indicated that most anomalies were successfully detected. Notably, a significant portion of these anomalies were located at the intersection of the three clusters, suggesting uncertain cluster assignments by the modeler. This aligned with our expectations, as for point anomalies, we desired either incorrect or uncertain cluster assignments to distinguish them from normal data.

In dataset *Ionosphere*, two distinct distributions were evident. Anomalies exist in both clusters but were predominantly concentrated at one end of one cluster, exhibiting both collective and point anomalies. The anomaly score visualization shows the successful detection of most anomalies, with a clear separation between normal and anomalous data scores. Although the collective anomalies region had few labels, they were successfully detected due to their significant reconstruc-

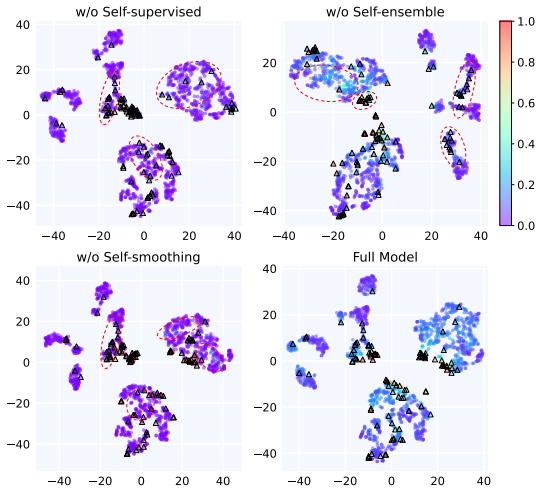

Fig. 6. Visualized ablation study of training strategies on the dataset *Letter*, with a fixed anomaly label ratio of 0.1. The figures illustrated the estimated anomaly scores given by different models. From top-left to bottom-right, the corresponding AUROC scores were 0.8226, 0.8469, 0.8828, and 0.9212, respectively.

TABLE VI
ABLATION STUDY ON EXTRACTED FEATURES ACROSS 11 DATASETS

Models	AVG. AUROC	AVG. AUPRC
w/o cluster probability $\mathbf{y}$	0.9308 ± 0.0079	0.7521 ± 0.0206
w/o mean latent $\boldsymbol{\mu}_{\mathbf{z}\|\mathbf{x}}$	0.8765 ± 0.0262	0.6613 ± 0.0480
w/o clustering entropy f_{e}	0.9306 ± 0.0080	0.7538 ± 0.0185
w/o reconstruction error f_{r}	0.9264 ± 0.0106	0.7549 ± 0.0243
w/o cosine similarity f_{c}	0.9256 ± 0.0091	0.7451 ± 0.0202
full features	**0.9335 ± 0.0069**	**0.7583 ± 0.0220**

tion errors and clustering uncertainty resulting from their deviation from normal patterns.

In dataset *PageBlocks*, despite its complex manifold structure, most anomalies were distributed along one manifold, with some attached to the extremity of the normal manifold and a small portion scattered within it. MMM's results showed that clustered collective anomalies were grouped into a single cluster and assigned high anomaly scores. Anomalies at the normal manifold's extremity were largely detected without significantly affecting nearby normal data. The scattered anomalies within the normal manifold showed lower detection rates due to their limited quantity, sparse labeling, and close similarity to normal data.

In summary, MMM demonstrated excellent performance in modeling, clustering, and anomaly detection for multi-distributional data. Through feature fusion, information propagation, and joint training, the model effectively detected both collective and point anomalies. Additionally, the model's output of latent distribution and anomaly scores clearly revealed the distribution of different manifolds and anomalies in the dataset, helping practitioners better analyze, understand, and interpret detected anomalies.

4) Ablation Study: Firstly, to validate the effectiveness of the five features $\mathbf{y}, \boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}, f_{\mathrm{e}}, f_{\mathrm{r}}, f_{\mathrm{c}}$ extracted from the modeler and delivered to the anomaly score estimator, we conducted ablation experiments by removing each feature individually and evaluating the average performance across all 11 datasets with a fixed anomaly label ratio of 0.1. The results were presented in TABLE VI. We observed that both AUROC and AUPRC metrics degraded after removing any single feature, with the most significant impact occurring when removing the mean vector of the latent distributions $\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}$, which contained compressed and clustered data information from the modeler. Notably, except lowest performing ablated model "w/o $\boldsymbol{\mu}_{\mathbf{z}|\mathbf{x}}$", other models all outperformed the top baseline WVAD, demonstrating the superiority of MMM.

TABLE VII
ABLATION STUDY ON TRAINING STRATEGIES ACROSS 11 DATASETS

Models	AVG. AUROC	AVG. AUPRC
w/o self-supervised	0.9024 ± 0.0161	0.7142 ± 0.0306
w/o self-ensemble	0.9196 ± 0.0096	0.7365 ± 0.0244
w/o self-smoothing	0.8895 ± 0.0158	0.6761 ± 0.0327
full strategies	**0.9335 ± 0.0069**	**0.7583 ± 0.0220**

Secondly, to validate the effectiveness of three strategies used in training the estimator: self-supervised learning, self-ensemble, and self-smoothing, we conducted ablation studies across all 11 datasets with a fixed anomaly label ratio of 0.1. We first demonstrated the visualization results using dataset *Letter* as an example, following the same visualization approach as the previous section. As shown in Fig. 6, the four anomaly score plots represented the outputs from models with self-supervised learning removed, self-ensemble removed, self-smoothing removed, and the full model, achieving AUROC scores of 0.8226, 0.8469, 0.8828, and 0.9212, respectively. The following observations can be made:

(a) Without self-supervised learning, the estimator's training became biased toward the normal class, tending to assign very low anomaly scores to most data points, resulting in numerous undetected anomalies, as shown in the three red circles in the figure.
(b) Without self-ensemble, the anomaly score distribution became chaotic, with significant variations among normal or anomalous points in the same region (left two red circles), due to unstable estimator training. Additionally, multiple undetected anomalies appeared in the two red circles on the right.
(c) Without self-smoothing, the estimator's output scores predominantly polarized toward extremes: either very close to 0 or 1. While this highlighted certain anomalies, it made uncertain anomalies more difficult to detect, as shown by numerous undetected anomalies in the red circles on the left and bottom. Furthermore, some regions exhibited abrupt score transitions, leading to false alarms, particularly in the two red circles on the right.
(d) With all strategies implemented, the estimator produced uniformly distributed and smoothly transitioning anomaly scores that accurately reflected most anomalies. This demonstrated the successful and effective combination of these three strategies.

Finally, TABLE VII presented the average performance of the four model variants across all 11 datasets. The model incorporating all three strategies achieved the highest average AUROC and AUPRC while maintaining the lowest standard deviations, confirming the performance and stability improvement brought by the proposed training approach.

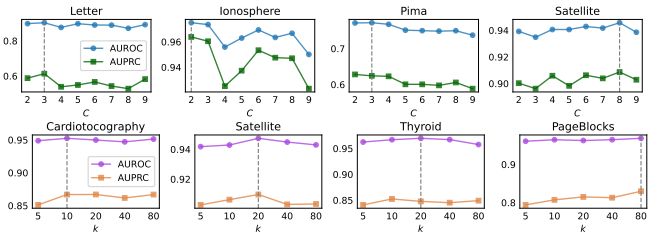

Fig. 7. The AUROC and AUPRC performance of MMM with different cluster count C (top row) and the neighbor count k (bottom row). The dashed line indicates the value of C or k at which the AUROC peaks.

TABLE VIII
AVERAGE AUROC AND AUPRC OF MMM WITH VARYING INTERPOLATION PARAMETER κ ON 11 DATASETS

Metric	Interpolation parameter κ			
	0.1	0.2	0.5	1.0
AVG. AUROC	0.9291	0.9317	**0.9335**	0.9307
AVG. AUPRC	0.7469	0.7532	**0.7583**	0.7457

5) Sensitivity Test: Firstly, we tested the cluster count C (from 2 to 9) on four datasets: *Letter*, *Ionosphere*, *Pima*, and *Satellite*. As this hyperparameter was highly application-dependent, these datasets were selected based on their discernible cluster structures confirmed by original descriptions or t-SNE visualizations. Their predefined cluster counts C' were 3, 2, 3, and 6, respectively. Fig. 7 showed that the optimal AUROC was achieved when $C = C'$ for the first three datasets. For *Satellite*, peak performance occurred at $C = 8$ rather than $C' = 6$, since its six-class data exhibited highly irregular distributions in high-dimensional space, requiring more clusters for effective modeling. Nevertheless, $C = 6$ still yielded competitive results. Overall, the performance did not degrade much when C was suboptimal, and though we did not dedicate to tune C in the main experiments, MMM still exhibited superior performance, demonstrating MMM's robustness and potential to perform even better.

In addition, to facilitate the search for optimal C, we also introduced a heuristic density-peak-based approach [46]. It first identified dense cores by finding local representatives for each point based on maximum density within their natural neighborhoods, applying representative competition rules, and retaining only core points with density above a threshold to filter out noise. It then computed geodesic distances between dense cores using common neighborhood-based distance measures and Dijkstra's algorithm. Finally, we constructed a decision graph plotting the product of geodesic distance and local density (γ) against the index of dense cores sorted by γ, where the cluster count was determined by finding the cores with anomalously large γ values, as illustrated in Fig. 8.

Secondly, we tested the neighbor count k (within $\{5, 10, 20, 40, 80\}$) on four datasets: *Cardiotocography*, *Satellite*, *Thyroid*, and *PageBlocks*. This hyperparameter was related to the distributional structure of the datasets. As illustrated in Fig. 7, *PageBlocks* preferred a large k since its clusters were concentrated and dense; conversely, *Cardiotocography* preferred a small k since its data were scattered with no obvious clusters. However, the performance remained stable as k varied. Given that a smaller k results in less computational overhead, we used $k = 10$ in all other experiments in this work.

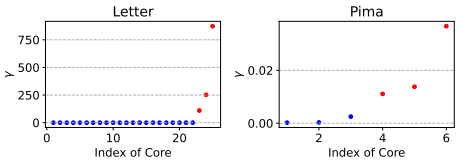

Fig. 8. Decision graph of the density-peak-based cluster count searching method on *Letter* and *Pima*. The density cores with anomalously large γ values (red dots) can be treated as the potential cluster centers.

TABLE IX
RUNTIME OF MMM WITH $D = 50$ AND VARYING N

Stage	Runtime (s)				
	N=1000 D=50	N=2000 D=50	N=5000 D=50	N=10000 D=50	N=20000 D=50
Pre-clustering	2.643	9.143	32.386	119.716	442.671
Network training	140.508	149.817	176.193	249.653	347.722
Total training	143.151	158.960	208.578	369.369	790.394
Total Inference	0.003	0.003	0.003	0.004	0.004
Fast pre-clustering	2.322	3.523	10.528	35.510	77.761

Finally, we tested the interpolation parameter κ used in distribution $\mathrm{Beta}(\kappa, \kappa)$. The values of the interpolation factor λ sampling from $\mathrm{Beta}(\kappa, \kappa)$ lay between 0 and 1. When $\kappa \to 0$, λ was almost 0 or 1, so the interpolated point collapses to one original point. When $\kappa \to 1$, λ was nearly uniform, placing the interpolation anywhere between the two points. Thus, too small κ gave sparse local coverage and weak self-smoothing, while too large κ may misalign the interpolated point with the interpolated label, leading to over-smoothing. Following [37], we tested the values of κ in $\{0.1, 0.2, 0.5, 1.0\}$, and showed the results in TABLE VIII. It can be observed that the best performance corresponded to $\kappa = 0.5$, while other settings also achieved competitive performance.

6) Scalability Test: To evaluate the scalability of MMM, we tested the runtime efficiency and parameter size of MMM with different scale of data. The test data were synthesized containing three Gaussian clusters. The units numbers were set to [D, 128, 128, $D/4$] for the modeler encoder, and [F, $F \times 2$, $F \times 2$, 1] for the estimator. The experimental platform was an Intel Xeon E5-2696 v4 CPU with a NVIDIA Tesla P40 GPU.

The total training time of MMM consists of two main parts: the pre-clustering time and network training time. Thus, we tested the runtime of these two parts respectively and added them up.

Firstly, we fixed the dimensionality $D = 50$ and increased the dataset size N from 1000 to 20000. As shown in TABLE IX, when N was small, network training time dominated, while when N was large, pre-clustering time dominated. This was because the time complexity of spectral clustering grew polynomially with dataset size, while network training time grew linearly. This implied that spectral clustering became a computational bottleneck when dealing with large-scale datasets. This issue can be alleviated by employing more efficient clustering methods, such as fast spectral clustering (FSC). As demonstrated in TABLE VIII, we tested an FSC approach [47], which largely reduced the runtime when N was large. The exploration of other computationally efficient clustering methods remains one of our future research directions.

TABLE X
RUNTIME OF MMM WITH $N = 5000$ VARYING D

Stage	Runtime (s)				
	N=5000 D=10	N=5000 D=20	N=5000 D=50	N=5000 D=100	N=5000 D=200
Pre-clustering	71.249	41.646	32.386	34.698	34.110
Network training	183.476	185.935	176.193	180.612	197.411
Total training	254.725	227.580	208.578	215.310	231.521
Total Inference	0.003	0.003	0.003	0.004	0.005

TABLE XI
PARAMETER SIZE OF MMM WITH $D = 50$

Component	Parameter size (KB)				
	D=10	D=20	D=50	D=100	D=200
Data modeler	210.57	226.63	270.8	347.1	497.68
Student estimator	1.69	3.1	8.02	23.25	74.82
Teacher estimator	1.69	3.1	8.02	23.25	74.82
Total	213.95	232.83	286.84	393.6	647.32

Secondly, we fixed the dataset size $N = 5000$ and increased the dimensionality D from 10 to 200. As shown in TABLE X, the network training time did not change much due to its parallel processing capability. A counterintuitive observation was that pre-clustering time increased when dimensionality was low. This occured because runtime largely depended on whether the data possessed a clear cluster structure. In this experiment, the cluster structure was disrupted when the latent dimensionality was too low, thus increasing clustering time.

It is worth noting that, the computational cost of pre-clustering is needed only in the training stage. During inference, we only need to deliver data into neural networks to obtain the anomaly scores, making the total inference time very short (around 0.004s) for all conditions.

Finally, we tested the parameter size of MMM with different D. Since all the networks were fully-connected-networks with two hidden layers, they had small parameter sizes (on the order of KB), enabling their ready deployment on memory-constrained devices. Additionally, the student estimator was discarded during inference, further reducing the parameter count.

V. CONCLUSIONS

In this work, we proposed MMM, a unified WSAD framework for multi-distributional data. MMM performs data reconstruction and clustering across different distributions while leveraging adjacency graphs to effectively handle complex data manifolds. In this way, both collective anomalies and anomalies positioned between different normal data clusters can be detected effectively. In addition, with the help of our proposed training approach, the anomaly score estimator of MMM became more effective and robust through augmented training labels and stabilized network parameters, helping mitigate the label-insufficiency and anomaly contamination. The experimental results demonstrated that the proposed MMM outperformed existing SOTA WSAD baselines in both performance and stability. Notably, MMM exhibited superior resilience to performance degradation as anomaly label ratios decrease. Furthermore, it provided interpretable results that enable practitioners to gain deeper insights into data structures and anomaly patterns.

In summary, this work highlights the importance of carefully considering data structures and fully leveraging both data information and label information in WSAD. Building on this foundation, the proposed MMM framework significantly enhances both performance and interpretability in practical WSAD applications. MMM has the potential for further evolution through the enhancement of its individual components. Future research directions may include the integration of advanced clustering and modeling techniques, the design of application-specific feature selection and label augmentation strategies, and the reduction of computational complexity.

REFERENCES

[1] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969. DOI: 10.1080/00401706.1969.10490657

[2] W. Hilal, S. A. Gadsden, and J. Yawney, "Financial fraud: A review of anomaly detection techniques and recent advances," *Expert Syst. Appl.*, vol. 193, p. 116429, 2022. DOI: 10.1016/j.eswa.2021.116429

[3] M. Odiathevar, W. K. Seah, M. Frean, and A. Valera, "An online offline framework for anomaly scoring and detecting new traffic in network streams," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5166–5181, 2022. DOI: 10.1109/TKDE.2021.3050400

[4] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Deep learning for medical anomaly detection–a survey," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–37, 2022. DOI: 10.1145/3464423

[5] J. Yang, X. Tan, and S. Rahardja, "Mipo: How to detect trajectory outliers with tabular outlier detectors," *Remote Sens. (Basel)*, vol. 14, no. 21, p. 5394, 2022. DOI: 10.3390/rs14215394

[6] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, et al., "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Min. Knowl. Discov.*, vol. 30, no. 4, pp. 891–927, 2016. DOI: 10.1007/s10618-015-0444-8

[7] B. Liu, Y. Xiao, S. Y. Philip, Z. Hao, and L. Cao, "An efficient approach for outlier detection with imperfect data labels," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1602–1616, 2014. DOI: 10.1109/TKDE.2013.108

[8] S. Han, X. Hu, H. Huang, M. Jiang, and Y. Zhao, "Adbench: Anomaly detection benchmark," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 32 142–32 159, 2022.

[9] Z. Li, C. Sun, C. Liu, X. Chen, M. Wang, and Y. Liu, "Dual-mgan: An efficient approach for semi-supervised outlier detection with few identified anomalies," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 6, pp. 1–30, 2022. DOI: 10.1145/3530990

[10] M. Jiang, C. Hou, A. Zheng, X. Hu, S. Han, H. Huang, et al., "Weakly supervised anomaly detection: A survey," arXiv preprint *arXiv:*, 2023.

[11] C. Huang, C. Liu, J. Wen, L. Wu, Y. Xu, Q. Jiang, et al., "Weakly supervised video anomaly detection via self-guided temporal discriminative transformer," *IEEE Trans. Cybern.*, vol. 54, no. 5, pp. 3197–3210, 2024. DOI: 10.1109/TCYB.2022.3227044

[12] C. Huang, W. Huang, Q. Jiang, W. Wang, J. Wen, and B. Zhang, "Multimodal Evidential Learning for Open-World Weakly-Supervised Video Anomaly Detection," *IEEE Trans. Multimed.*, vol. 27, pp. 3132–3143, 2025. DOI: 10.1109/TMM.2025.3557682

[13] G. Pang, C. Shen, and A. Van Den Hengel, "Deep anomaly detection with deviation networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 353–362, 2019. DOI: 10.1145/3292500.3330871

[14] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, and L. Liu, "Feature encoding with autoencoders for weakly supervised anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2454–2465, 2022. DOI: 10.1109/TNNLS.2021.3086137

[15] D. Fu, Z. Zhang, and J. Fan, "Dense projection for anomaly detection," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 8, pp. 8398–8408, 2024. DOI: 10.1609/aaai.v38i8.28682

[16] D. Nkashama, A. Soltani, J.-C. Verdier, M. Frappier, P.-M. Tardif, and F. Kabanza, "Robustness evaluation of deep unsupervised learning algorithms for intrusion detection systems," arXiv preprint *arXiv:*, 2022.

[17] X. Tan, J. Chen, S. Rahardja, J. Yang, and S. Rahardja, "Weakly-supervised anomaly detection for multimodal data distributions," in *Proc. IEEE Int. Conf. Signal Process. Commun. Comput. (ICSPCC)*, pp. 1–5, 2024. DOI: 10.1109/ICSPCC62635.2024.10770302

[18] A. Khan and P. Maji, "Approximate graph laplacians for multimodal data clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 798–813, 2021. DOI: 10.1109/TPAMI.2019.2945574

[19] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, et al., "Deep semi-supervised anomaly detection," arXiv preprint *arXiv:*, 2019.
[20] Q. Xie, P. Zhang, B. Yu, and J. Choi, "Semisupervised training of deep generative models for high-dimensional anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 6, pp. 2444–2453, 2022. DOI: 10.1109/TNNLS.2021.3095150
[21] C. Huang, F. Ye, P. Zhao, Y. Zhang, Y.-F. Wang, and Q. Tian, "Esad: End-to-end deep semi-supervised anomaly detection," arXiv preprint *arXiv:*, 2020.
[22] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 2041–2050, 2018. DOI: 10.1145/3219819.3220042
[23] B. Tian, Q. Su, and J. Yin, "Anomaly detection by leveraging incomplete anomalous knowledge with anomaly-aware bidirectional gans," arXiv preprint *arXiv:*, 2022. DOI: 10.24963/ijcai.2022/313
[24] J. Gao, C. Tao, Z. Sun, et al., "Semi-Supervised Anomaly Detection through Denoising-Aware Contrastive Distance Learning," *Proc. ACM Web Conf.*, pp. 2111–2119, 2025. DOI: 10.1145/3696410.3714626
[25] B. Micenková, B. McWilliams, and I. Assent, "Learning outlier ensembles: The best of both worlds–supervised and unsupervised," in *Proc. ACM SIGKDD Workshop Outlier Detect. Descr. Data Divers. (ODD2)*. New York, NY, USA. Citeseer, 2014, pp. 51–54.
[26] Y. Zhao and M. K. Hryniewicki, "Xgbod: improving supervised outlier detection with unsupervised representation learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, pp. 1–8, 2018. DOI: 10.1109/IJCNN.2018.8489605
[27] G. Pang, C. Shen, H. Jin, and A. van den Hengel, "Deep weakly-supervised anomaly detection," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, pp. 1795–1807, 2023. DOI: 10.1145/3580305.3599302
[28] H. Xu, Y. Wang, G. Pang, S. Jian, N. Liu, and Y. Wang, "Rosas: Deep semi-supervised anomaly detection with contamination-resilient continuous supervision," *Inf. Process. Manage.*, vol. 60, no. 5, p. 103459, 2023. DOI: 10.1016/j.ipm.2023.103459
[29] Z. Cao, Y. Shi, Y.-C. Chang, X. Yao, and C.-T. Lin, "Twin fuzzy networks with interpolation consistency regularization for weakly-supervised anomaly detection," *IEEE Trans. Fuzzy Syst.*, vol. 32, no. 9, pp. 5086–5097, 2024. DOI: 10.1109/TFUZZ.2024.3412435
[30] Y. Zhou, P. Yang, Y. Qu, X. Xu, Z. Sun, and A. Cichocki, "AnoOnly: Semi-supervised anomaly detection with the only loss on anomalies," *Expert Syst. Appl.*, vol. 262, p. 125597, 2025. DOI: 10.1016/j.eswa.2024.125597
[31] D. Gong, L. Liu, V. Le, et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 1705–1714, 2019. DOI: 10.1109/ICCV.2019.00179
[32] C. Huang, C. Liu, Z. Zhang, et al., "Pixel-level anomaly detection via uncertainty-aware prototypical transformer," *Proc. 30th ACM Int. Conf. Multimed.*, pp. 521–530, 2022. DOI: 10.1145/3503161.3548082
[33] L. Yang, N.-M. Cheung, J. Li, and J. Fang, "Deep clustering by gaussian mixture variational autoencoders with graph embedding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 6439–6449, 2019. DOI: 10.1109/ICCV.2019.00654
[34] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint *arXiv:*, 2013.
[35] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, et al., "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018.
[36] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 1195–1204, 2017. DOI: 10.5555/3294771.3294885
[37] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, A. Solin, Y. Bengio, et al., "Interpolation consistency training for semi-supervised learning," *Neural Netw.*, vol. 145, pp. 90–106, 2022. DOI: 10.1016/j.neunet.2021.10.008
[38] J. Yang, S. Rahardja, and P. Fränti, "Smoothing outlier scores is all you need to improve outlier detectors," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 7044–7057, 2024. DOI: 10.1109/TKDE.2023.3332757.
[39] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: an unsupervised and generative approach to clustering," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, pp. 1965–1972, 2017. DOI: 10.24963/ijcai.2017/273
[40] U. von Luxburg, "A tutorial on spectral clustering," *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007. DOI: 10.1007/s11222-007-9033-z
[41] G. E. Hinton and S. Roweis, "Stochastic neighbor embedding," *Adv. Neural Inf. Process. Syst.*, vol. 15, pp. 857–864, 2002. DOI: 10.5555/2968618.2968725
[42] Q. Zhou, S. He, H. Liu, J. Chen, and W. Meng, "Label-free multivariate time series anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3166–3179, 2024. DOI: 10.1109/TKDE.2024.3349613
[43] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[44] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, and G. Pang, "Calibrated one-class classification for unsupervised time series anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 5723–5736, 2024. DOI: 10.1109/TKDE.2024.3393996
[45] D. P. Kingma, "Adam: A method for stochastic optimization," arXiv preprint *arXiv:*, 2014.
[46] D. Cheng, J. Huang, S. Zhang, X. Zhang, and X. Luo, "A novel approximate spectral clustering algorithm with dense cores and density peaks," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 4, pp. 2348–2360, 2021. DOI: 10.1109/TSMC.2021.3049490
[47] R. Zhang, S. Hang, Z. Sun, F. Nie, R. Wang, and X. Li, "Anchor-based fast spectral ensemble clustering," *Inf. Fusion*, vol. 113, p. 102587, 2025. DOI: 10.1016/j.inffus.2024.102587

Xu Tan (Student Member, IEEE) received his B.S. degree in detection guidance and control technology in 2020, from the School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China, where he is currently pursuing his Ph.D. degree in information and communication engineering.

Junqi Chen (Student Member, IEEE) received his B.S. degree in detection guidance and control technology in 2020, and his M.S. degree in signal and information processing in 2023, both from Northwestern Polytechnical University, Xi'an, China, where he is currently pursuing his Ph.D. degree in information and communication engineering.

Jiawei Yang (Senior Member, IEEE) received his B.Eng. degree in Automation from Beihang University, China, in 2013, and his M.Sc. and Ph.D. degrees in Computer Science from the University of Eastern Finland, in 2019 and 2020, respectively. He is currently a senior researcher at the University of Turku, Finland.

Jie Chen (Senior Member, IEEE) received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2006, the Dipl.-Ing. and M.S. degrees in information and telecommunication engineering from the University of Technology of Troyes (UTT), Troyes, France, and Xi'an Jiaotong University, respectively, in 2009, and the Ph.D. degree in systems optimization and security from UTT, in 2013. He is currently a Professor with Northwestern Polytechnical University, Xi'an, China.

Susanto Rahardja (Fellow, IEEE) received his B.Eng. degree from National University of Singapore and M.Eng. and Ph.D. from Nanyang Technological University, all in electronic engineering. He is currently a Professor at the College of Information Science & Electronic Engineering, Zhejiang University. He is a Fellow of IEEE and a Fellow of the Academy Engineering, Singapore.