Dual Data-centric Separation with Circular Mixup for Noise-resistant Time Series Learning

Yuhang Pei*
Northeastern University
Shenyang, China
peiyh@mails.neu.edu.cn

Fanchun Meng*
Northeastern University
Shenyang, China
2110495@stu.neu.edu.cn

Qinghua Ran
Peking University
Beijing, China
2301213075@stu.pku.edu.cn

Tao Ren
Northeastern University
Shenyang, China
rent@swc.neu.edu.cn

Yifan Wang†
University of International Business and Economics
Beijing, China
yifanwang@uibe.edu.cn

Wei Ju
Peking University
Beijing, China
juwei@pku.edu.cn

Zimo Wang
Beijing Bayi School
Beijing, China
zimo.wang@bayims.cn

Xian-Sheng Hua
Tongji University
Shanghai, China
huaxiansheng@gmail.com

Xiao Luo
University of Wisconsin–Madison
Madison, Wisconsin, USA
xiao.luo@wisc.edu

Abstract

Deep neural networks (DNNs) have achieved extensive progress in time series learning. However, they could suffer from performance degradation when it comes to label noise in the real world. Towards this end, this paper studies an underexplored yet realistic problem of noise-resistant time series learning and proposes a novel data-centric approach named Dual Data-centric Separation with Circular Mixup (DREAM) for this problem. The core of our DREAM is to explore and exploit the noisy data from dual data-centric views for reduced overfitting. On the one hand, we assume that samples with similar features share similar labels and infer the pseudo label of each sample using its affinity graph to capture the corresponding pseudo margin. On the other hand, we monitor the optimization status by simulating the mislabeled data to generate flexible criteria for accurate separation of clean and noisy samples. In addition, we leverage circular Mixup to interpolate between clean and noisy samples in the embedding space. These mixed samples are incorporated into a discrepancy-aware consistency learning framework to ensure robust time series representations of all the separated samples. Experimental results on a wide range of publicly accessible datasets reveal the effectiveness of our DREAM.

CCS Concepts

• **General and reference** → **Reliability**; • **Computing methodologies** → **Neural networks**.

*Both authors contributed equally to this research.
†Corresponding author.

KDD '26, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2258-5/2026/08
https://doi.org/10.1145/3770854.3780265

Keywords

Time-Series Classification, Label-Noise Learning, Data Augmentation, Deep Neural Networks

ACM Reference Format:
Yuhang Pei, Fanchun Meng, Qinghua Ran, Tao Ren, Yifan Wang, Wei Ju, Zimo Wang, Xian-Sheng Hua, and Xiao Luo. 2026. Dual Data-centric Separation with Circular Mixup for Noise-resistant Time Series Learning. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770854.3780265

Resource Availability:
The source code of this paper has been made publicly available at https://doi.org/10.5281/zenodo.18042700.

1 Introduction

Time series data are characterized by their sequential nature and temporal dependencies, holding essential information about the dynamics of various systems. In light of its importance, time series classification has received increasing attention in numerous real-world applications, such as human activity recognition [12, 53], traffic state forecasting [9, 22], climate estimation [39, 45] and economic analysis [2, 35]. Thanks to the powerful feature extraction capabilities of deep learning, its emerging development has significantly enhanced performance in time series classification. However, the process acquires abundant labeled data, which inevitably results in noisy labels due to sensor errors or manual labeling mistakes. As a result, models can easily overfit these noisy instances, severely impairing their generalization performance, as shown in Fig. 1.

Label-Noise Learning (LNL) endeavors to enhance model robustness by identifying and mitigating the impact of noisy labels during training. Indeed, there are a handful of LNL works, and existing techniques can be categorized into three dominant groups. Noise-tolerant methods [16, 63] aim to design alternative loss functions instead of relying on the cross-entropy to handle label noise. Label correction approaches [10, 30, 41], on the other hand, adjust label

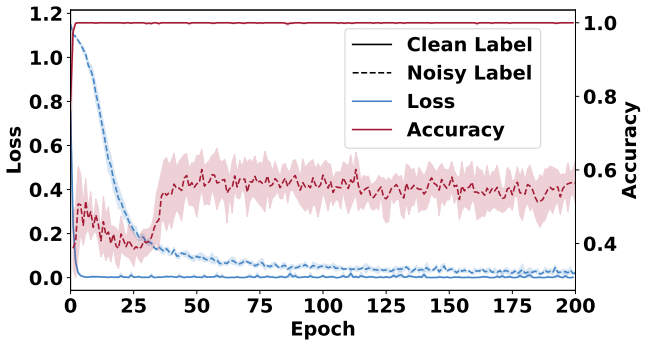


Figure 1: Training loss and test accuracy of a time series classification model under clean and noisy label settings. Noisy labels and model overfitting jointly cause a decline in test performance.

noise by leveraging an estimated noise transition matrix. In contrast, sample separation approaches [21, 31, 57] directly distinguish clean samples from noisy ones and filter out the latter to improve training quality. Recently, several pioneer studies have begun to study time series LNL. CTW [36] enhances the representation of confident instances via Time-Warping augmentation in time series LNL. Scale-teaching [34] designs a multi-scale embedding mechanism to correct noisy time series samples. RTS [64] separates noisy samples and performs purified oversampling learning.

Despite the encouraging performance of these time series LNL methods, there are still significant challenges that remain in the field, primarily due to the following two reasons: ❶ *Heavily relies on the static model confidence to distinguish clean samples.* When constructing the confident set, most existing approaches are based solely on the small-loss criterion at the current iteration, paying no attention to the similar neighbors' predictions from earlier iterations, which could be valuable for improving the quality of the set. ❷ *Fail to fully harness the potential of noisy data.* Most existing works, particularly sample separation approaches, treat clean and noisy samples independently, with each type assigned to different loss functions for more targeted processing. However, the effective design of representations and loss functions that leverage the interaction between clean and noisy samples remains underexplored.

MixUp [61] is a widely used data-augmentation routine that effectively constructs virtual training data by linearly interpolating continuous values of random samples to improve the generalization and robustness of the model. Recent processes of MixUp training have demonstrated its effectiveness in semi-supervised learning. For example, MixMatch [5] and follow-up work ReMixMatch [4] generate pseudo labels for unlabeled data and mix labeled and unlabeled samples for training. Manifold MixUp [48] further extends the MixUp idea by applying linear interpolation to the extracted embedding vectors in the embedding space. In the case of LNL, applying the mixup on the clean and noisy samples could smooth the data manifold and dilute the noise effects.

Towards this end, we propose **DREAM**, a novel **D**ual data-cent**R**ic s**E**paration with circul**A**r **M**ixup for time series LNL, which separates clean and noisy samples while facilitating their interaction in both representations and losses to enhance model performance. Specifically, given time series data with noisy labels, we first apply data augmentation to each sample and construct an affinity graph based on the learned intermediate representation to identify potential neighbors. Instead of relying solely on the current loss criterion to distinguish clean samples, we assume that similar samples share similar labels. Consequently, we then utilize these neighbors to infer pseudo labels and introduce pseudo margins, which capture the discrepancy between the output logit corresponding to the observed label class and others, serving as a confidence measure of the sample. Next, by monitoring the model's training dynamics, we learn a flexible confidence threshold to identify clean samples. Finally, we introduce a circular embedding MixUp, which interpolates between clean and noisy samples, along with a new consistency loss to fully leverage noisy data for time series LNL.

To summarize, we make the following contributions:

- *Conceptual:* We study the underexplored problem of noise-resistant time series learning and are the first to propose a data-centric framework for this problem.
- *Methodological:* We identify clean data from both neighborhood affinity and optimization status, and incorporate circular MixUp with discrepancy-aware consistency learning for robust hidden time series representations.
- *Experimental:* We evaluate DREAM on multiple public datasets through extensive experiments. Experimental results verify that the proposed framework consistently outperforms extensive baseline methods.

2 Related Work

2.1 Learning with Noisy Labels

Learning with Noisy Labels poses a significant challenge in tasks like image classification/segmentation [27, 40], cross-modal retrieval [23], and graph learning [7, 56, 57]. Recent efforts about LNL can be divided into three main types: 1) Noise-tolerant methods [16, 63] aim to design robust loss functions that are resilient to label noise. For example, TCE [16] approximates the cross-entropy loss using a Taylor series for training models with label noise. More recent developments include LogitClip [49], which limits the scale of logits to prevent overconfidence on noisy instances, and Active Negative Loss [55], which introduces Normalized Negative Loss Functions to better handle noisy labels and improve convergence, particularly under complex noise settings. 2) Label correction approaches [10, 30, 41] combat label noise by either adjusting the loss or correcting erroneous labels during training. A representative work [41] learns a noise transition matrix to correct label noise throughout training. Dynamic Loss Adjustment [26] and spatial-aware correction strategies [54] introduce learnable modulators or incorporate structural priors to enhance robustness. Despite these advances, label correction methods remain limited by their dependency on accurate noise estimation and often suffer from poor scalability and reduced effectiveness in high-noise or large-class scenarios, highlighting the need for more flexible and adaptive solutions. 3) Sample separation methods [21, 31, 57] choose to pick up clean samples and directly eliminate the noisy samples for the training process. Among them, Co-teaching and its extensions [21, 58]employ dual-network training, where each model selects presumably clean samples to guide the other, thereby reducing error accumulation. Building upon these ideas, ITEM [50]

adopts multi-expert networks, and CSS [37] leverages large-scale pre-trained models to enhance noise filtering via collaborative or contrastive learning strategies. Additionally, semi-supervised techniques [44, 46, 60] can also be employed to filter clean samples. However, these sample separation approaches are primarily tailored for computer vision. In the context of time series, Scale-teaching [34] designs a multi-scale embedding mechanism for time series label correction. TS-CoT [62] generates different views of the time series and enhances the robustness of the model through a collaboratively trained comparative learning method. CTW [36] and RTS [64] separate noisy samples and expand the distribution of confident instances. However, we argue that the threshold for confident clean sample collection is inflexible, and the disconnect between clean and noisy sets limits the potential to better utilize noisy data.

2.2 Data Augmentation for Generalization

Data augmentation has been effectively integrated with consistency regularization within semi-supervised learning frameworks to enhance generalization for LNL [24]. DivideMix and UNICON [27, 31] first identify and group samples into distinct labeled clean and unlabeled noisy sets. They then perform consistency regularization using MixUp augmentation. Previous work [15] demonstrates that enforcing prediction consistency across augmentations improves robustness against label noise. ProMix [11] focuses on hard noisy samples, filtering them out in a two-stage process before applying MixUp augmentation. NoiseMix [29] strengthens classification robustness by mixing clean and noisy samples. AdaWAC [14] is an adaptive weighting scheme for data augmentation consistency to enhance robustness under concept shift. Manifold DivideMix [17] splits label noise into in-distribution and out-of-distribution types, leveraging the clean and in-distribution sets for data augmentation. However, most of these approaches inadvertently disrupt the temporal structure of time series. For time series data, several augmentation techniques [28, 47], including GaussNoise, Convolve, Drift, and Crop, have been introduced. Among these methods, Time-Warping [36] simulates sampling from different temporal locations, but it may also introduce additional noisy instances. DTW-based averaging [51] earlier demonstrated the importance of preserving temporal alignment when synthesizing new sequences. TENOR [38] further models temporally correlated label noise via adaptive loss functions, highlighting the unique challenges of LNL in sequential domains. CTW [36] improves robustness by applying warping only to confident samples, mitigating noise amplification. In addition, recent empirical studies [19] systematically validate that mix-based augmentations, including MixUp and its variants, consistently improve classification performance across diverse physiological time series datasets, without requiring expert heuristics or heavy tuning. In this paper, we incorporate Time-Warping augmentation with an effective data MixUp method to better utilize noisy data for time series LNL.

3 The Proposed DREAM

The core concept of our DREAM is constructing a time series LNL model capable of effectively utilizing noisy data. Fig. 2 outlines the proposed framework, which consists of two key elements: Dual Data-centric Exploration for Sample Separation and Circular MixUp for Data Exploitation.

3.1 Problem Definition

Consider an ideal time series dataset $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1}^N$ consisting of N time series, where $\mathcal{X} = \{\boldsymbol{x}_i\} \in \mathbb{R}^{N\times m}$ is the input sequence of length m, and $\mathcal{Y} = \{y_i\}$ contains the corresponding labels across C classes, with each y_i in one-hot format $\boldsymbol{Y}_i \in \{0,1\}^C$. However, since the data collection process may introduce labeling errors, the resulting noisy dataset is represented as $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^N$, in which $\tilde{y}_i$ represents the observed label of $\boldsymbol{x}_i$. The objective is to train a robust classifier $g(\cdot;\Theta) : \mathcal{X} \to \mathcal{Y}$ parameterized by $\Theta = \{\theta, \psi\}$ with a feature extractor $f(\cdot;\theta)$ and a classification head $h(\cdot;\psi)$, such that $g(\boldsymbol{x}_i;\Theta) = y_i$ as accurately as possible.

3.2 Dual Data-centric Sample Separation

When training with noisy labels, deep neural networks (DNNs) initially learn general patterns but eventually overfit to the noise, leading to poor performance on the clean evaluation set [46]. Instead of relying on the commonly used small-loss criterion at the current iteration to select clean samples, we utilize flexible thresholds from both neighbor and model perspectives for sample separation.

Affinity Graph for Pseudo Margin. Guided by the assumption that samples with similar features share similar labels [6], we learn a deep neural network to extract features of each sample, i.e., $\boldsymbol{z}_i = f(\boldsymbol{x}_i) \in \mathbb{R}^d$ with dimension d, and define the radius $r_k(\boldsymbol{x}_i)$ of affinity graph for sample $\boldsymbol{x}_i$ as:

$$\begin{aligned} r_k(\boldsymbol{x}_i) &:= \inf\{r : |B(\boldsymbol{x}_i, r) \cap \mathcal{X}| \geq k\}, \\ B(\boldsymbol{x}_i, r) &:= \{\boldsymbol{x}_j \in \mathcal{X} : \mathrm{Dis}(\boldsymbol{z}_i, \boldsymbol{z}_j) \leq r\}, \end{aligned} \tag{1}$$

where $\mathrm{Dis}(\cdot,\cdot)$ denotes the feature distance. In practice, we use the reciprocal of cosine similarity as the distance function to select top-k similar neighbors. Then, for each sample $\boldsymbol{x}_i$, the affinity graph is defined as:

$$N_k(\boldsymbol{x}_i) := B(\boldsymbol{x}_i, r_k(\boldsymbol{x}_i)) \cap \mathcal{X}. \tag{2}$$

We normalize the cosine similarity as w_{ij} and weighted sum their observed label as the affinity graph based classifier's output to infer the neigh-aware pseudo label, defined as:

$$\boldsymbol{\eta}(\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_j \in N_k(\boldsymbol{x}_i)} w_{ij} \widetilde{Y}_j. \tag{3}$$

We utilize the pseudo-margin of $\boldsymbol{x}_i$ w.r.t. pseudo-label $\eta(\boldsymbol{x}_i)$ at iteration t as follows:

$$\mathrm{PM}_c^t(\boldsymbol{x}_i) = \boldsymbol{\eta}_c - \max_{c\neq c'}(\boldsymbol{\eta}_{c'}), \tag{4}$$

where $\boldsymbol{\eta}_c$ is the logit of class c corresponding to the observed label $\tilde{y}_i$ and $\max_{c\neq i}(\boldsymbol{\eta}_i)$ is the largest logit of any *other* class c' that is different from c [33, 43, 46]. Meanwhile, to monitor the neighbor's prediction performance on x_i relative to the observed label class c throughout the training process, we compute the average pseudo-margin (APM) w.r.t. c from the start-up to iteration t, written as:

$$\mathrm{APM}_c^t(\boldsymbol{x}_i) = \mathrm{PM}_c^t(\boldsymbol{x}_i) * \frac{\tau}{1+t} + \mathrm{APM}_c^{t-1}(\boldsymbol{x}_i) * (1 - \frac{\tau}{1+t}), \tag{5}$$

where $\mathrm{APM}_c^0(\boldsymbol{x}_i)$ is initialized as $\mathrm{PM}_c^0(\boldsymbol{x}_i)$ and τ is the smoothing parameter [42, 46]. Intuitively, if the predicted pseudo-label at iteration t is c, namely, $c = \arg\max \boldsymbol{\eta}(\boldsymbol{x}_i)$, then PM_c^t w.r.t. class c will

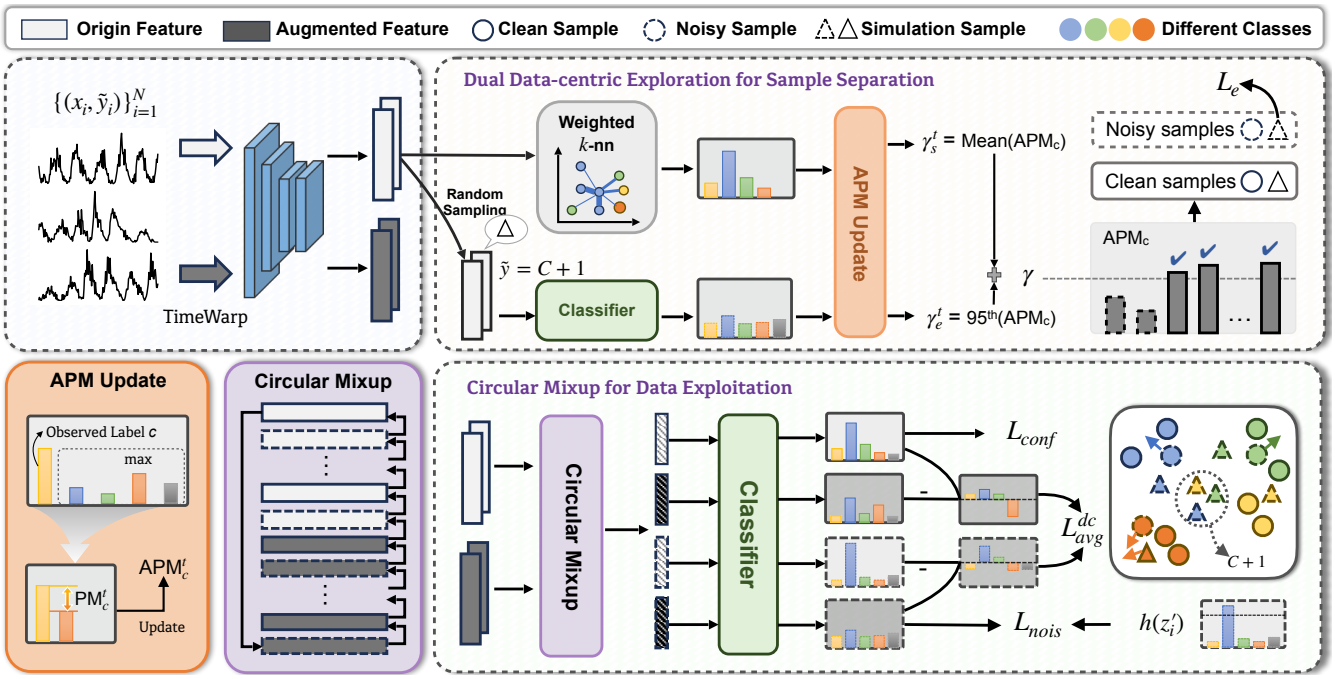


Figure 2: The overall framework of the DREAM. We apply Time-Warping for time series data augmentation and identify confident, clean samples via the dual data-centric sample separation. Then, a circular embedding MixUp, along with a data consistency loss combined with the standard loss of the clean/noisy sets, is employed to fully leverage the noisy data.

be positive at iteration t, otherwise, it will be negative. Therefore, if throughout the iterations, the inferred pseudo-labels from similar neighbors frequently disagree with the observed label c, the APM for class c will be low, likely negative, indicating high uncertainty regarding the label for x_i. Building on this insight, we leverage mean value of APM as the threshold, defined as:

$$\gamma_s^t = \frac{1}{N} \sum_{\boldsymbol{x_i} \in \mathcal{X}} \text{APM}_c^t(\boldsymbol{x_i}). \tag{6}$$

In this way, we can select clean samples at iteration t based on similar neighbors from the data perspective.

Data Simulation for Flexible Criteria. To further estimate the threshold from the model perspective, we analyze the training dynamics of a specific class within the observed data. Specifically, we randomly select a subset of samples $\mathcal{D}_e$ from $\tilde{\mathcal{D}}$, re-assign them to a non-existent (virtual) class $C+1$ at the beginning of the training process. Since all these samples originate from one of the C original classes, assigning them to the non-existent class directly makes them mislabeled. Thus, these mislabeled samples can simulate the training dynamics of noisy samples.

At iteration t, we compute the APM_{C+1}^t as a proxy to estimate the threshold. In practice, we set the model-aware threshold γ_e^t as the APM of the 95^{th} percentile selected mislabeled samples. Then, by integrating the thresholds from both perspectives, the final threshold for selecting confident clean samples can be:

$$\gamma^t = \alpha \gamma_s^t + (1-\alpha)\gamma_e^t, \tag{7}$$

where α is the trade-off parameter. We consider all samples with APMc^t above the threshold γ^t as the confident clean set $\mathcal{D}_{conf}$, while those below the threshold are classified as the noisy set $\mathcal{D}_{nois}$. The mislabeled sample loss becomes:

$$\mathcal{L}_e = \sum_{(\boldsymbol{x}_i, \cdot) \in \mathcal{D}_e - \mathcal{D}_{conf}} H(C+1, g(\text{TimeWarp}(\boldsymbol{x}_i))), \tag{8}$$

where $H(\cdot,\cdot)$ denotes the cross-entropy loss. Here we employ a time series augmentation method, namely Time-Warping [36], which adjusts the time axis by interpolating and re-sampling to create temporally distorted versions of the original sequence.

3.3 Circular MixUp for Data Exploitation

After eliminating the noisy instances from training, we further interpolate between clean and noisy samples using embedding MixUp to mitigate the effects of noise. Additionally, we introduce a delta-based consistency regularization to better leverage noisy data for time series LNL.

Circular MixUp for Virtual Data Generation. Building on the findings that perturbing the embedding space contributes to enhanced performance across various learning tasks [25], we apply embedding mixup to encourage interaction between clean and noisy samples, thereby improving their representation learning. Given a batch comprising B clean samples and $\mu \cdot B$ noisy samples, we extract the feature of the original and its augmented data via $f(\cdot)$ to the embedding array, namely $\boldsymbol{Z} \in \mathbb{R}^{2(1+\mu)B \times d}$. The general embedding MixUp framework can be:

$$\boldsymbol{Z}' = (\boldsymbol{I} + \boldsymbol{A})\boldsymbol{Z}, \tag{9}$$

where $\boldsymbol{A}$ and $\boldsymbol{I}$ denote the MixUp and identity matrix, respectively. Inspired by the previous work [59], we impose three constraints

on the MixUp matrix A:

$$\begin{aligned} &(i)\ \text{rank}(\boldsymbol{I}+\boldsymbol{A}) = (1+\mu)B, \\ &(ii)\ [\boldsymbol{I}+\boldsymbol{A}]_{ii} > [\boldsymbol{I}+\boldsymbol{A}]_{ij}, i \neq j, \\ &(iii)\ \|[\boldsymbol{I}+\boldsymbol{Q}]_i\|_1 = 1, \end{aligned} \tag{10}$$

where the first full rank constraint ensures that the MixUp process preserves information from all original embeddings, the second constraint guarantees that the trained classifier receives sufficient information about the original embedding to accurately predict its label, and the final constraint enforces the normalization of the mixed embeddings.

To fulfill the above constraints, we introduce a specific and straightforward construction for the MixUp matrix A, termed the *circular shift* [59]. In this construction, each embedding $\boldsymbol{z}_i$ is softly mixed by its subsequent neighbor $\boldsymbol{z}_{i+1}$. Formally, the MixUp matrix $\boldsymbol{A}$ can be implemented as $\boldsymbol{A} = \beta * \boldsymbol{U} - \beta * \boldsymbol{I}$, where $\boldsymbol{U}_{i,j} = \sigma_{i+1,j}$ with $\sigma_{i+1,j}$ representing the Kronecker delta indicator [18], utilizing wraparound (or "circular") indexing. Thus, the MixUp matrix can be:

$$\boldsymbol{A} = \begin{bmatrix} -\beta & \beta & 0 & 0 & 0 \\ 0 & -\beta & \beta & 0 & 0 \\ & & \ddots & \ddots & 0 \\ 0 & 0 & & & \beta \\ \beta & 0 & 0 & 0 & -\beta \end{bmatrix} \tag{11}$$

where β is the MixUp strength parameter. And each row can be simply expressed as $\boldsymbol{z}'_i = (1-\beta)\boldsymbol{z}_i + \beta \boldsymbol{z}_{i+1}$.

3.3.1 Discrepancy-aware Consistency Learning. As noted in the previous work [25], the change in predicted probabilities due to data augmentation should exhibit consistency across various sample types. Therefore, to better make use of the noisy data, we propose a discrepancy-aware consistency learning strategy that explicitly quantifies the differences in predicted probabilities between clean and noisy samples during the data augmentation process:

$$\begin{aligned} \Delta_i^{conf} &= h(\boldsymbol{z}_i) - h(\boldsymbol{z}'_i), \forall(\boldsymbol{x}_i, \cdot) \in \mathcal{D}_{conf}, \\ \Delta_i^{nois} &= \frac{1}{\mu}\sum_{i=1}^{\mu} h(\boldsymbol{z}_i) - h(\boldsymbol{z}'_i), \forall(\boldsymbol{x}_i, \cdot) \in \mathcal{D}_{nois}. \end{aligned} \tag{12}$$

Here Δ_i^{conf} and Δ_i^{nois} denote the average change in predicted probabilities across clean and noisy instances under the same augmentations within a minibatch. The objective is to encourage Δ_i^{conf} closely resemble the change vector Δ_i^{nois}. Specifically, we optimize by reducing the mean squared Euclidean distance of the vector pairs, defined as:

$$\mathcal{L}_{avg}^{dc} = \frac{1}{B}\sum_{i=1}^{B} \|\Delta_i^{conf} - \Delta_i^{nois}\|_2^2. \tag{13}$$

3.4 Overall Training Objective

Our proposed DREAM consists of four parts: 1) Classification loss on confident set. Given the confident clean samples with MixUp embeddings, classification learning loss can be:

$$\mathcal{L}_{conf} = \sum_{(\boldsymbol{x}_i, \tilde{y}_i) \in \mathcal{D}_{conf}} H(\tilde{y}_i, h(\boldsymbol{z}'_i)). \tag{14}$$

2) Instance-wise consistency loss on noisy set. Given the noisy samples with MixUp embeddings, we assume that classes with fewer examples present greater learning challenges and employ the flexible FlexMatch [60] approach to promote the inclusion of more training examples from these classes. Formally, the consistency loss on noisy set is:

$$\mathcal{L}_{nois} = \sum_{(\boldsymbol{x}_i, \cdot) \in \mathcal{D}_{nois}} \mathbb{1}(\max(h(\boldsymbol{z}'_i)) > \xi_{\boldsymbol{z}'_i}^t) * H(h(\boldsymbol{z}'_i), h(\hat{\boldsymbol{z}}'_i)), \tag{15}$$

where $\hat{\boldsymbol{z}}'_i$ denotes the mixed embedding for the augmented noise data. $\xi_{\boldsymbol{z}'_i}^t$ is the flexible threshold estimated as in FlexMatch. 3) Delta-based consistency loss $\mathcal{L}_{avg}^{dc}$. And the overall loss function can be defined as

$$\mathcal{L} = \mathcal{L}_{conf} + \lambda_{nois}(\mathcal{L}_{nois} + \mathcal{L}_e) + \lambda_{dc}\mathcal{L}_{avg}^{dc}, \tag{16}$$

where λ_{nois} and λ_{dc} weight the respective components of the total loss.

3.5 Theoretical Analysis

We provide a theoretical analysis to show how DREAM works. First, we show that leveraging similar neighbors of each sample with pseudo-margins is a more effective way to distinguish the clean set. Then, we prove that the variance of MixUp embedding is smaller than that of the original embedding, leading to a more robust prediction.

Why Similar Neighbors and Pseudo Margins. Firstly, we believe that noise mainly appears on the labels and ignores the noise on $\mathcal{X}$. Therefore, if two samples exhibit similar input features, it is reasonable to assume that their corresponding true labels are likely to be consistent. Based on this assumption, we perform neighbor-based label aggregation by computing a weighted average of the predicted labels from samples in the local neighborhood of $\mathcal{X}$. This strategy is designed to mitigate the influence of label noise. Furthermore, if the predicted label of a sample significantly deviates from the aggregated pseudo-label, the resulting confidence (threshold) will be low, and the sample will consequently be identified as a noisy sample.

Next, we will discuss the role of pseudo margins. The noise set selected by most methods is $\mathcal{D}_{noise} = \{(\boldsymbol{x}_i, y_i) | \arg\max(\boldsymbol{\eta}(\boldsymbol{x}_i)) \neq c\}$, where c is corresponding to the observed label $\overline{y}_i$. But they overlook a situation that the predicted label and observed label are the same but there is another category c' with a logit value that is not significantly different from the observed label logit value. This means that it is difficult to determine whether the true label of the sample is c or c'. For a simple example, if $\boldsymbol{x}_i|y_i = 1 \sim N(0,1)$ and $\boldsymbol{x}_i|y_i = 2 \sim N(10,1)$, it is hard to say whether sample $\boldsymbol{x}_i = 5$ belongs to label 1 or label 2, and it is even more likely to belong to an unknown label. We assume that the distribution centers corresponding to known labels are far apart. So if a sample has a similar probability of belonging to two labels, it indicates that its features are far from the centers of the two distributions. Therefore, it is reasonable to consider it as a noisy sample.

Definition 3.1. We assume that the input data of samples belonging to label c comes from distribution D_c, that is $\boldsymbol{x}|y = c \sim D_c(\mu_c)$, where $\mathbb{E}(\boldsymbol{x}) = \boldsymbol{\mu}_c$. And given a sample $\boldsymbol{x_i}$, we define the probability that it belongs to label c as $p(\boldsymbol{x}_i, c) = \mathbb{P}(\|\boldsymbol{x} - \boldsymbol{\mu}_c\|_2 > \|\boldsymbol{x}_i - \boldsymbol{\mu}_c\|_2)$.

Theorem 3.1. *Suppose we have two labels c and c', and the probabilities of sample x_i belonging to label c is $p(\boldsymbol{x}_i, c) = p$ and that belonging to label c' is $p(\boldsymbol{x}_i, c') = p - \delta$, then the probability of sample i get a noise label is*

$$p_{noise} = 1 - 2p + \delta + 2p(p - \delta), \tag{17}$$

and if we only consider the case where the real label is c or c', the probability is

$$p_{noise} = \frac{2p(p-\delta)}{(2p-\delta)^2}, \tag{18}$$

and it is a decreasing function of $\delta, \delta \geq 0$.

Proof. First, the probability of a sample having noisy labels can be decomposed into two situations where the true labels and observed labels are not the same:

$$\begin{aligned} p_{noise} =& p(y_i = c, \widetilde{y_i} = c' | y_i \in \{c, c'\}) + \\ & p(y_i = c', \widetilde{y_i} = c | y_i \in \{c, c'\}). \end{aligned} \tag{19}$$

Then, by utilizing the properties of conditional distributions, we can obtain:

$$\begin{aligned} & p(y_i = c, \widetilde{y_i} = c' | y_i \in \{c, c'\}) \\ &= p(\widetilde{y_i} = c' | y_i \in \{c, c'\}) p(y_i = c | y_i \in \{c, c'\}) \\ &= \frac{p(y_i = c, y_i \in \{c, c'\})}{p(y_i \in \{c, c'\})} \frac{p(\widetilde{y_i} = c', y_i \in \{c, c'\})}{p(y_i \in \{c, c'\})} \\ &= \frac{p}{2p-\delta} \frac{p-\delta}{2p-\delta} = \frac{p(p-\delta)}{(2p-\delta)^2}. \end{aligned} \tag{20}$$

We can also obtain the same:

$$p(y_i = c', \widetilde{y_i} = c | y_i \in \{c, c'\}) = \frac{p(p-\delta)}{(2p-\delta)^2}. \tag{21}$$

So, $p_{noise} = \frac{2p(p-\delta)}{(2p-\delta)^2}$.

In the above derivation, we made the implicit assumption that $p(\widetilde{y_i} = c' | y_i \in \{c, c'\}) = p(y_i = c' | y_i \in \{c, c'\})$, which means that the probabilities of the observed label and the true label are equal. However, in reality, the observed label can be influenced by other factors, such as incorrect sample labeling. We made this assumption because when PM is close to a small positive value, it is usually due to the sample lying within the overlapping region of the two class distributions. In such cases, the classifier tends to assign the sample fairly evenly to both classes. Therefore, if an observed label error occurs, it is primarily due to the overlap between the two class distributions, rather than other external factors. As a result, we can ignore the influence of external factors on the misclassification probability in the derivation.

Next, we differentiate p_{noise} with respect to δ:

$$\begin{aligned} \frac{\mathrm{d}p_{noise}}{\mathrm{d}\delta} &= \frac{-p}{(2p-\delta)^2} + 2\frac{p(p-\delta)}{(2p-\delta)^3} \\ &= \frac{-\delta p}{(2p-\delta)^3}. \end{aligned} \tag{22}$$

Note that $2p - \delta > 0$, and we can confirm $\frac{\mathrm{d}p_{noise}}{\mathrm{d}\delta} \leq 0$, so it is a decreasing function of $\delta, \delta \geq 0$. Obviously, when δ approaches 0, p_{noise} reaches its maximum value of 0.5, so it is necessary to consider samples with smaller PM as noise samples. □

Remark. In Eqn. 18, we can find that as p decreases, the probability of noise increases. However, considering that the labels c and c' are the two labels with the highest probability, if p is too small, it may be due to too many types of labels. In this case, we believe that although the sample is far from the distribution center of the true label, it is also far from the centers of other labels, so useful information can still be learned from it. So we use Eqn. 18 to avoid the influence of p on the judgment of noisy samples.

Variance of MixUp Embedding. For labeled data, we can assume that in the absence of noise, samples under the same label come from the same distribution, while the difference between samples with different labels comes mainly from the difference in their distribution centers (such as mean). In other words, samples closer to the distribution center are easier to identify. For time series data, the correlation between contexts is inevitable, and the MixUp operation can be seen as summing up two correlated random variables from the same distribution. For clean samples, through MixUp operation, we can reduce the variance of embeddings and make them closer to the distribution center. For noisy samples, we can move their centers away from the distribution center of their observed labels, making them easier to identify as noisy samples.

Definition 3.2. We assume that the samples with the same label in the clean set have embeddings from the same distribution and that there is a correlation between adjacent samples. So for a given label, let $\boldsymbol{z_i}$ be the embedding of sample i, and $z_{i,j}$ be its j-th dimension. And $\boldsymbol{z_i} \sim D(\boldsymbol{\mu}, \Sigma)$, where D represents the share distribution cross the samples and $\mathbb{E}(\boldsymbol{z_i}) = \boldsymbol{\mu}, Cov(\boldsymbol{z_i}) = (\sigma_{jj'}) \in \mathbb{R}^{d \times d}$. And the covariance between $z_{i,j}$ and $z_{i+1,j'}$ is defined as $Cov(z_{i,j}, z_{i+1,j'}) = \rho_{i,jj'}\sigma_{jj'}, 0 \leq \rho_{i,jj'} \leq 1$.

Theorem 3.2. *Suppose the MixUp embedding is $\boldsymbol{z}'_i = (1-\beta)\boldsymbol{z}_i + \beta \boldsymbol{z}_{i+1}$, and the j-th dimension of the embedding is $z'_{i,j} = (1-\beta)z_{i,j} + \beta z_{i+1,j}$, where $\beta \in (0, 1)$ is the MixUp strength. Then for each j, we have $Var(z'_{i,j}) \leq Var(z_{i,j})$.*

Proof. We start with the variance of $z'_{i,j}$:

$$\begin{aligned} Var(z'_{i,j}) &= Var((1-\beta)z_{i,j} + \beta z_{i+1,j}) \\ &= (1-\beta)^2 Var(z_{i,j}) + \beta^2 Var(z_{i+1,j}) \\ &+ 2\beta(1-\beta) Cov(z_{i,j}, z_{i+1,j}) \\ &= \sigma_j^2((1-\beta)^2 + \beta^2 + 2\rho_{ij}\beta(1-\beta)) \\ &\leq \sigma_j^2((1-\beta)^2 + \beta^2 + 2\beta(1-\beta)) \\ &= \sigma_j^2 = Var(z_{i,j}) \end{aligned} \tag{23}$$

So we have proven Theorem 2. □

Remark. The requirement that the samples come from the same distribution is a strong condition. In fact, the above theorem only requires that the samples have the same mean and variance distribution.

4 Experiments

4.1 Experimental Setup

Datasets and Baseline. We conduct experiments on a diverse set of time-series classification datasets from the UCR and UEA archives, comprising 13 benchmarks in total, including 8 UCR datasets and

Table 1: Performance comparison across different methods and noise settings. The best and runner-up results are highlighted in bold and underline, respectively. The detailed results on 13 datasets are provided in Appendix C.1.

Methods	Sym				Asym				IDN	
	15%	30%	45%	60%	10%	20%	30%	40%	30%	40%
Vanilla	0.768(0.039)	0.667(0.047)	0.543(0.055)	0.398(0.048)	0.793(0.031)	0.730(0.049)	0.651(0.052)	0.547(0.053)	0.646(0.039)	0.550(0.05)
SIGUA	0.777(0.046)	0.709(0.054)	0.589(0.069)	0.444(0.072)	0.795(0.035)	0.751(0.047)	0.691(0.050)	0.584(0.076)	0.657(0.061)	0.597(0.058)
Co-teaching	0.809(0.031)	0.749(0.041)	0.673(0.051)	0.509(0.060)	0.814(0.027)	0.779(0.034)	0.738(0.035)	0.635(0.045)	0.722(0.052)	0.653(0.051)
MixUp-BMM	0.762(0.044)	0.718(0.043)	0.616(0.057)	0.494(0.057)	0.761(0.051)	0.743(0.038)	0.706(0.048)	0.611(0.061)	0.681(0.056)	0.611(0.057)
DivideMix	0.413(0.084)	0.420(0.082)	0.412(0.077)	0.345(0.075)	0.440(0.089)	0.414(0.066)	0.430(0.080)	0.422(0.093)	0.423(0.081)	0.378(0.104)
Sel-CL	0.708(0.053)	0.700(0.048)	0.645(0.053)	0.580(0.063)	0.728(0.033)	0.705(0.045)	0.687(0.053)	0.623(0.070)	0.685(0.043)	0.659(0.052)
SREA	0.802(0.024)	0.747(0.040)	0.638(0.048)	0.495(0.058)	0.803(0.022)	0.764(0.038)	0.712(0.048)	0.610(0.043)	0.708(0.037)	0.647(0.042)
CTW	**0.827**(0.020)	0.786(0.027)	0.690(0.054)	0.522(0.066)	0.836(0.017)	0.819(0.022)	0.771(0.038)	0.692(0.041)	0.758(0.043)	0.677(0.059)
DREAM	**0.827(0.017)**	**0.801(0.026)**	**0.755(0.038)**	**0.644**(0.065)	**0.837**(0.024)	**0.820**(0.027)	**0.793(0.038)**	**0.734**(0.056)	**0.794(0.027)**	**0.732**(0.063)

5 UEA datasets [3, 13]. To assess the model's performance under varying noise conditions, we consider three noise settings: Symmetric Noise (Sym), Asymmetric Noise (Asym), and Instance-Dependent Noise (IDN). We compare our model with several state-of-the-art methods for noisy time-series classification, including Vanilla, SIGUA [20], Co-teaching [21], Mixup-BMM [1], DivideMix [31], Sel-CL [32], SREA [8], and CTW [36]. Vanilla serves as a conventional supervised learning approach that does not incorporate any label-noise mitigation techniques for time series classification. The setting details are in Appendix A.

Implementation. Our framework adopts a convolutional encoder based on the structure outlined in SREA [8] detailed in Appendix B. Specifically, the model is optimized using Adam for 300 training epochs with a weight decay of 10^{-4}. The hyperparameters are set as $k = 10$, $\alpha = 0.95$, $\beta = 0.1$, $\lambda_{\text{nois}} = 1$, and $\lambda_{\text{dc}} = 0.5$. We set the Adam parameters to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, starting with a learning rate of 10^{-3} that undergoes a 50% reduction every 60 epochs. The batch size is set to $\min\left(\frac{1}{10} \times \text{dataset size}, 128\right)$. To comprehensively evaluate performance, we report the mean F1-score over all datasets under each noise type and ratio. For each setting, the results are averaged from five runs with different seeds.

4.2 Performance on Time Series Classification

We evaluate the performance of our model against the baselines. Table 1 presents the average F1-scores of each method across 13 benchmark datasets, under symmetric noise (15%, 30%, 45%, and 60%), asymmetric noise (10%, 20%, 30%, and 40%), and instance-dependent noise (30% and 40%) settings. Based on these results, we can draw the following conclusions:

- Both CTW and DREAM leverage data augmentation to enhance confident sample selection and improve representation learning, which enables them to outperform traditional baselines. This highlights the importance of augmentation in time-series LNL.
- DREAM delivers state-of-the-art results under all noise settings, with particularly impressive gains in high-noise regimes (e.g., more than 30% noise), outperforming other methods by notable margins. This underscores the effectiveness of our proposed margin threshold and circular MixUp in handling label corruption.
- In low-noise scenarios, DREAM and CTW attain comparable performance, as conventional classification losses tend to dominate the optimization process when most labels remain reliable. The results validate our framework's adaptability to both mild and extreme label noise.

Table 2: Performance comparison on 128 UCR datasets. The optimal and second-best outcomes are highlighted in bold and underline, respectively. *Avg_F1* denotes the mean F1-score and *#Best* indicates the count of best results.

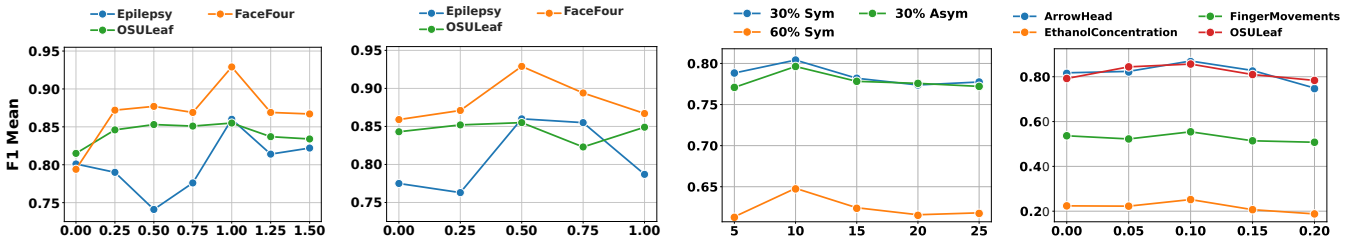
Methods	Sym 30%		Asym 40%	
	Avg_F1	*#Best*	*Avg_F1*	*#Best*
Vanilla	0.608	2/128	0.499	3/128
SIGUA	0.652	2/128	0.544	3/128
Co-teaching	0.696	10/128	0.590	5/128
Mixup-BMM	0.630	2/128	0.530	1/128
DivideMix	0.449	1/128	0.399	5/128
Sel-CL	0.601	5/128	0.542	12/128
SREA	0.700	17/128	0.589	18/128
CTW	0.733	30/128	0.621	23/128
DREAM	**0.739**	**63/128**	**0.669**	**58/128**

In addition, to further strengthen the persuasiveness of our experimental results, we conducted experiments on 128 UCR datasets under both 30% symmetric noise and 40% asymmetric noise. The results are reported in Table 2. As observed, DREAM consistently surpasses other baseline methods across a larger and more diverse set of datasets, further validating its robustness and effectiveness in handling noisy labels in time-series classification.

4.3 Ablation Studies

To comprehensively evaluate the contribution of each module in DREAM, we conduct a series of ablation studies from three perspectives: the effect of Dual Data-centric Sample Separation, the impact of Circular Embedding MixUp, and the significance of Consistency Learning. The ablation results are summarized in Table 3. Additionally, we evaluated the effects of different augmentation strategies, with detailed settings provided in the Appendix C.2.

Effect of Dual Data-centric Sample Separation. We evaluate our strategy by comparing two variants: *w/o APM Threshold*, which removes the adaptive pseudo-margin threshold, and *w/o Data Simulation*, which eliminates the simulated mislabeled samples used to calibrate the threshold. Both variants show notable performance degradation. Notably, the removal of the APM threshold leads to



(a) λ_{nois} (b) λ_{dc} (c) Number of Neighbors k (d) MixUp Strength β

Figure 3: Performance of DREAM with different hyperparameters.

Table 3: Ablation studies of DREAM. The red arrows (↓) indicate performance degradation.

Methods	Sym 60%	Asym 40%
DREAM	**0.644**	**0.734**
DREAM w/o APM Threshold	0.581$_{\downarrow 0.063}$	0.689$_{\downarrow 0.045}$
DREAM w/o Data Simulation	0.626$_{\downarrow 0.018}$	0.713$_{\downarrow 0.021}$
DREAM w Stack Mixup	0.608$_{\downarrow 0.036}$	0.702$_{\downarrow 0.032}$
DREAM w/o Mixup	0.610$_{\downarrow 0.034}$	0.714$_{\downarrow 0.020}$
DREAM w/o $\mathcal{L}_{nois}$	0.579$_{\downarrow 0.065}$	0.661$_{\downarrow 0.073}$
DREAM w/o $\mathcal{L}_{avg}^{dc}$	0.591$_{\downarrow 0.053}$	0.690$_{\downarrow 0.044}$

Table 4: Performance of different augmentation strategies.

Augmentation	Sym 30%	Asym 30%
Time-Warping	**0.801**	**0.793**
GaussNoise	0.769$_{\downarrow 0.032}$	0.751$_{\downarrow 0.042}$
Convolve	0.755$_{\downarrow 0.046}$	0.756$_{\downarrow 0.037}$
Drift	0.770$_{\downarrow 0.031}$	0.751$_{\downarrow 0.042}$
Crop	0.776$_{\downarrow 0.025}$	0.782$_{\downarrow 0.011}$

the most substantial drop, emphasizing the critical role of dynamically adjusting the clean-noisy separation boundary. In contrast, fixed thresholds either exclude too many clean samples or include excessive noisy ones. Additionally, the data simulation component enhances threshold estimation by introducing noise-aware pseudo-label guidance, which leads to more stable and accurate sample partitioning. Precision visualizations are provided in Appendix C.3.

Effect of Circular Embedding MixUp. To investigate the role of embedding fusion, we consider two variants: *w Stack MixUp*, which replaces the circular embedding MixUp with a simple stacked MixUp strategy, and *w/o MixUp*, which removes embedding MixUp entirely. Both variants lead to performance degradation, where the stacked MixUp approach outperforms the version without MixUp. These findings suggest that merely combining embeddings without structural consideration leads to suboptimal interaction between clean and noisy samples. In contrast, circular embedding MixUp introduces staggered, controlled perturbations that help the model learn more generalizable features by preserving semantic relationships and smoothing decision boundaries.

Effect of Consistency Learning. We further analyze the impact of consistency learning by removing its two main components: *w/o* $\mathcal{L}_{nois}$, which removes the consistency loss designed to align noisy samples with their pseudo-labels, and *w/o* $\mathcal{L}_{avg}^{dc}$, which excludes the discrepancy-aware loss that enforces consistency in feature-level representations. The results show that both losses contribute to performance, but their impact differs in magnitude. Specifically, removing $\mathcal{L}_{nois}$ causes a more severe drop (0.065 under symmetric noise and 0.073 under asymmetric noise) than removing $\mathcal{L}_{avg}^{dc}$ (0.053 and 0.044, respectively). This observation is reasonable, since $\mathcal{L}_{nois}$ directly affects classification by correcting noisy samples, whereas $\mathcal{L}_{avg}^{dc}$ primarily enhances the robustness of feature representations, thereby contributing to generalization in a more indirect manner.

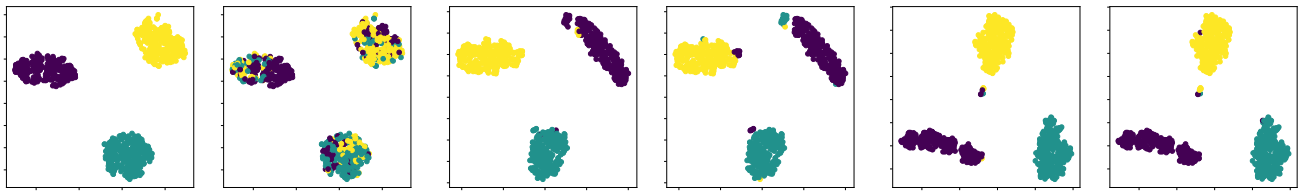
Impact of Augmentation Strategies. To validate our choice of data augmentation, we compare Time-Warping against four other common strategies: GaussNoise, Convolve, Drift, and Crop. As shown in Table 4, augmentations like Time-Warping, Drift, and Crop, which modify the temporal dimension of the data, lead to the highest performance improvements. Notably, Time-Warping, which is utilized in our DREAM, preserves the intrinsic temporal dependencies within the data while introducing variability, thereby enhancing the model's ability to generalize across diverse time-series. In contrast, augmentations such as GaussNoise and Convolve, which modify data values through noise or smoothing, have a less favorable impact. While these value perturbations promote robustness to minor fluctuations, they fail to preserve the underlying temporal structure as effectively as temporal modifications, ultimately resulting in reduced model performance.

4.4 Parameter Analysis

We further study the sensitivity of DREAM to several key hyperparameters, including the loss weight λ, the number of neighbors k used for margin threshold, and the MixUp strength β. The results are shown in Fig. 3.

Effect of Loss Weight λ. We evaluate the impact of λ by varying λ_{nois} within the range of $[0, 1.5]$ and λ_{dc} within the range of $[0, 1]$, using 30% symmetric noise on the Epilepsy, FaceFour, and OSULeaf datasets. Performance improves as λ increases, peaking at $\lambda_{nois} = 1$ and $\lambda_{dc} = 0.5$, then declines. The initial improvement is attributed to the model's ability to effectively leverage noisy data to extract both discriminative and invariant features. However, as λ_{nois} continues to increase, the model tends to overfit to noisy pseudo-labels. Similarly, a higher λ_{dc} results in the learned representations that are less pertinent to the classification task, hindering performance.

Number of Neighbors k of Margin Threshold. We examine how model performance varies with the number of neighbors k



(a) Vanilla (b) CTW (c) DREAM

Figure 4: t-SNE visualization on the CBF dataset under the 30% symmetric noise setting. For each method, the left plot shows the embedding colored with the predicted label, while the right plot shows the embedding colored with true labels y_i.

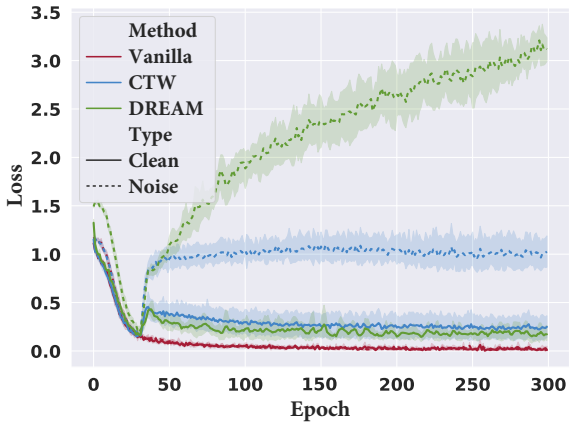


Figure 5: Loss curves of clean and noisy samples of the CBF dataset at 60% symmetric noise level. The shadowed area represents the 95% confidence interval.

on model performance by varying it within the range of $[5, 25]$ on three datasets: Epilepsy, FaceFour, and OSULeaf, under 30% symmetric noise. From the results, it is evident that the model performance remains relatively stable across different k values, with a notable improvement when k is set to 10. This indicates that an appropriate choice of k effectively captures the local structure of the data, thereby facilitating more accurate sample selection and enhancing overall model performance.

Effect of MixUp Strength β. We analyze the effect of MixUp strength β by varying it within the range of $[0, 0.2]$ on four datasets: ArrowHead, EthanolConcentration, FingerMovements, and OSULeaf, under 30% symmetric noise. Increasing β up to 0.1 slightly improves performance, indicating that an appropriate MixUp strength can effectively combine noisy and clean samples, thereby enhancing feature representation and improving model robustness. However, higher β values degrade performance, as excessive MixUp strength distorts the intrinsic characteristics of the samples.

4.5 Further Analysis

To further explore the effectiveness of the model from a qualitative perspective, we conduct t-SNE visualization on the learned embeddings and perform loss analysis.

Visualization. We conduct t-SNE to visualize the learned embeddings on the CBF dataset. As shown in Fig. 4, the Vanilla exhibits a significant overlap between clean and noisy samples, indicating that the model struggles to distinguish between the two sets. In contrast, DREAM demonstrates several key advantages. First, the number of misclassified samples of DREAM is notably smaller compared to Vanilla and CTW, showing the efficiency of our method in handling noisy labels. Second, we can observe a distinct cluster at the center of the DREAM embedding. This cluster represents samples that are difficult to classify, which are adaptively assigned to a virtual class $C+1$ for better isolation and handling. In this way, DREAM can effectively handle these ambiguous points, resulting in a robust classification performance under noisy data.

Loss Analysis. Fig. 5 presents the loss curves of the different models. While the Vanilla approach converges more quickly on clean data, the simultaneous reduction in loss on noisy instances implies a growing tendency of the model to overfit erroneous labels over time. In contrast, the CTW method maintains a certain gap between clean and noisy data, keeping the loss of noisy data relatively stable. Our DREAM further amplifies this distinction, with the loss of clean data continuing to decrease and the loss of noisy samples continuing to increase. This indicates that our margin threshold-based sample selection could effectively separate clean samples, preventing DREAM from overfitting to noisy labels.

5 Conclusion

In this work, we propose a novel framework, DREAM, to tackle the challenge of LNL in time series data. DREAM is designed to not only separate clean and noisy samples, but also promote their interaction to fully exploit the noisy data. Specifically, we perform data augmentation and representation extraction to construct an affinity graph, from which a flexible margin threshold is learned to select confident clean samples based on both neighborhood structure and model predictions. To bridge the gap between clean and noisy data, we introduce a circular embedding mixup mechanism and a delta-based consistency regularization, facilitating direct and structured interactions. Extensive experiments on diverse time series benchmarks demonstrate the robustness and adaptability of DREAM across varying noise levels.

Acknowledgments

Tao Ren is supported by the National Natural Science Foundation of China (62276058, 41774063), the Fundamental Research Funds for the Central Universities (N25GFZ011). Yifan Wang is supported by the Fundamental Research Funds for the Central Universities in UIBE (Grant No. 23QN02).

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *Proceedings of the International Conference on Machine Learning*. PMLR, 312–321.

[2] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. 2014. Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*. IEEE, 106–112.

[3] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. 2018. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075* (2018).

[4] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2020. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proceedings of the International Conference on Learning Representations*.

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Proceedings of the Conference on Neural Information Processing Systems*. 5049–5059.

[6] Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. 2022. Cross modal retrieval with querybank normalisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5194–5205.

[7] WANG Botao, Jia Li, Yang Liu, Jiashun Cheng, Yu Rong, Wenjia Wang, and Fugee Tsung. 2023. Deep insights into noisy pseudo labeling on graph data. In *Proceedings of the Conference on Neural Information Processing Systems*. 76214–76228.

[8] Andrea Castellani, Sebastian Schmitt, and Barbara Hammer. 2021. Estimating the electrical power output of industrial devices with end-to-end time-series classification in the presence of label noise. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 469–484.

[9] Yuzhou Chen, Ignacio Segovia-Dominguez, Baris Coskunuzer, and Yulia Gel. 2022. TAMP-S2GCNets: coupling time-aware multipersistence knowledge representation with spatio-supra graph convolutional networks for time-series forecasting. In *Proceedings of the International Conference on Learning Representations*.

[10] Mark Collier, Basil Mustafa, Efi Kokiopoulou, Rodolphe Jenatton, and Jesse Berent. 2021. Correlated input-dependent label noise in large-scale image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1551–1560.

[11] Filipe R Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. 2021. Propmix: Hard sample filtering and proportional mixup for learning with noisy labels. In *British Machine Vision Conference*. 187.

[12] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. 2020. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition* 108 (2020), 107561.

[13] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. 2019. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica* 6, 6 (2019), 1293–1305.

[14] Yijun Dong, Yuege Xie, and Rachel Ward. 2023. Adaptively Weighted Data Augmentation Consistency Regularization for Robust Optimization under Concept Shift. In *Proceedings of the International Conference on Machine Learning*. PMLR, 8296–8316.

[15] Erik Englesson and Hossein Azizpour. 2021. Consistency regularization can improve robustness to label noise. *arXiv preprint arXiv:2110.01242* (2021).

[16] Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. 2021. Can cross entropy loss be robust to label noise?. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2206–2212.

[17] Fahimeh Fooladgar, Minh Nguyen Nhat To, Parvin Mousavi, and Purang Abolmaesumi. 2024. Manifold DivideMix: A semi-supervised contrastive learning framework for severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4012–4021.

[18] Theodore Frankel. 2011. *The geometry of physics: an introduction*. Cambridge university press.

[19] Peikun Guo, Huiyuan Yang, and Akane Sano. 2023. Empirical study of mix-based data augmentation methods in physiological time series data. In *2023 IEEE 11th International Conference on Healthcare Informatics (ICHI)*. IEEE, 206–213.

[20] Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. 2020. Sigua: Forgetting may make learning with noisy labels more robust. In *Proceedings of the International Conference on Machine Learning*. PMLR, 4006–4016.

[21] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proceedings of the Conference on Neural Information Processing Systems*. 8536–8546.

[22] Hui He, Qi Zhang, Simeng Bai, Kun Yi, and Zhendong Niu. 2022. CATN: Cross attentive tree-aware network for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4030–4038.

[23] Peng Hu, Xi Peng, Hongyuan Zhu, Liangli Zhen, and Jie Lin. 2021. Learning cross-modal retrieval with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5403–5413.

[24] Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia. 2021. Simple: Similar pseudo label exploitation for semi-supervised classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15099–15108.

[25] Zhe Huang, Xiaowei Yu, Dajiang Zhu, and Michael C Hughes. 2024. InterLUDE: Interactions between Labeled and Unlabeled Data to Enhance Semi-Supervised Learning. In *Proceedings of the International Conference on Machine Learning*.

[26] Shenwang Jiang, Jianan Li, Jizhou Zhang, Ying Wang, and Tingfa Xu. 2023. Dynamic Loss for Robust Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 12 (2023), 14420–14434. doi:10.1109/TPAMI.2023.3311636

[27] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. 2022. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9676–9686.

[28] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*.

[29] Hansang Lee, Haeil Lee, Helen Hong, and Junmo Kim. 2022. Noisy Label Classification Using Label Noise Selection with Test-Time Augmentation Cross-Entropy and NoiseMix Learning. In *MICCAI Workshop on Data Augmentation, Labelling, and Imperfections*. Springer, 74–82.

[30] Jisoo Lee and Sae-Young Chung. 2020. Robust training with ensemble consensus. In *Proceedings of the International Conference on Learning Representations*.

[31] Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *Proceedings of the International Conference on Learning Representations*.

[32] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. 2022. Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 316–325.

[33] Weijie Liu, Chong Wang, Shenghao Yu, Chenchen Tao, Jun Wang, and Jiafei Wu. 2022. Novel instance mining with pseudo-margin evaluation for few-shot object detection. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2250–2254.

[34] Zhen Liu, Dongliang Chen, Wenbin Pei, Qianli Ma, et al. 2023. Scale-teaching: robust multi-scale training for time series classification with noisy labels. In *Proceedings of the Conference on Neural Information Processing Systems*. 33726–33757.

[35] Minrong Lu and Xuerong Xu. 2024. TRNN: An efficient time-series recurrent neural network for stock price prediction. *Information Sciences* 657 (2024), 119951.

[36] Peitian Ma, Zhen Liu, Junhao Zheng, Linghao Wang, and Qianli Ma. 2023. CTW: Confident Time-Warping for Time-Series Label-Noise Learning.. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 4046–4054.

[37] Qing Miao, Xiaohe Wu, Chao Xu, Yanli Ji, Wangmeng Zuo, Yiwen Guo, and Zhaopeng Meng. 2024. Learning with noisy labels using collaborative sample selection and contrastive semi-supervised learning. *Knowledge-Based Systems* 296 (2024), 111860.

[38] Sujay Nagaraj, Walter Gerych, Sana Tonekaboni, Anna Goldenberg, Berk Ustun, and Thomas Hartvigsen. 2024. Learning from Time Series under Temporal Label Noise. *arXiv preprint arXiv:2402.04398* (2024).

[39] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. 2023. ClimaX: A foundation model for weather and climate. In *Proceedings of the International Conference on Machine Learning*.

[40] Youngmin Oh, Beomjun Kim, and Bumsub Ham. 2021. Background-aware pooling and noise-aware loss for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6913–6922.

[41] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1944–1952.

[42] Iustin Sirbu, Robert-Adrian Popovici, Cornelia Caragea, Ștefan Trăușan-Matu, and Traian Rebedea. 2025. MultiMatch: Multihead Consistency Regularization Matching for Semi-Supervised Text Classification. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 2792–2808.

[43] Iustin Sîrbu, Robert-Adrian Popovici, Traian Rebedea, and Ștefan Trăușan-Matu. 2025. Multihead Average Pseudo-Margin Learning for Disaster Tweet Classification. *Information* 16, 6 (2025), 434.

[44] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems* 33 (2020), 596–608.

[45] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. 2020. Metnet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140* (2020).

[46] Tiberiu Sosea and Cornelia Caragea. 2023. MarginMatch: Improving semi-supervised learning with pseudo-margins. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15773–15782.
[47] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*. 216–220.
[48] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the International Conference on Machine Learning*. PMLR, 6438–6447.
[49] Hongxin Wei, Huiping Zhuang, Renchunzi Xie, Lei Feng, Gang Niu, Bo An, and Yixuan Li. 2023. Mitigating memorization of noisy labels by clipping the model prediction. In *Proceedings of the International Conference on Machine Learning*. PMLR, 36868–36886.
[50] Qi Wei, Lei Feng, Haobo Wang, and Bo An. 2024. Debiased sample selection for combating noisy labels. *arXiv preprint arXiv:2401.13360* (2024).
[51] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478* (2020).
[52] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. 2020. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems* 33 (2020), 7597–7610.
[53] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep convolutional neural networks on multichannel time series for human activity recognition.. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 3995–4001.
[54] Jiachen Yao, Yikai Zhang, Songzhu Zheng, Mayank Goswami, Prateek Prasanna, and Chao Chen. 2023. Learning to Segment from Noisy Annotations: A Spatial Correction Approach. In *Proceedings of the International Conference on Learning Representations*.
[55] Xichen Ye, Yifan Wu, Yiwen Xu, Xiaoqiang Li, Weizhong Zhang, and Yifan Chen. 2024. Active Negative Loss: A Robust Framework for Learning with Noisy Labels. *arXiv preprint arXiv:2412.02373* (2024).
[56] Nan Yin, Li Shen, Chong Chen, Xian-Sheng Hua, and Xiao Luo. 2024. SPORT: A Subgraph Perspective on Graph Classification with Label Noise. *ACM Transactions on Knowledge Discovery from Data* (2024).
[57] Nan Yin, Li Shen, Mengzhu Wang, Xiao Luo, Zhigang Luo, and Dacheng Tao. 2023. Omg: Towards effective graph classification against label noise. *IEEE Transactions on Knowledge and Data Engineering* 35, 12 (2023), 12873–12886.
[58] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption?. In *Proceedings of the International Conference on Machine Learning*. PMLR, 7164–7173.
[59] Xiaowei Yu, Yao Xue, Lu Zhang, Li Wang, Tianming Liu, and Dajiang Zhu. 2024. Exploring the Impact of Information Entropy Change in Learning Systems. (2024).
[60] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Proceedings of the Conference on Neural Information Processing Systems*. 18408–18419.
[61] Hongyi Zhang. 2018. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations*.
[62] Weiqi Zhang, Jianfeng Zhang, Jia Li, and Fugee Tsung. 2023. A co-training approach for noisy time series learning. In *Proceedings of the International Conference on Information and Knowledge Management*. 3308–3318.
[63] Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the Conference on Neural Information Processing Systems*. 8792–8802.
[64] Zhi Zhou, Yi-Xuan Jin, and Yu-Feng Li. 2024. Rts: learning robustly from time series data with noisy label. *Frontiers of Computer Science* 18, 6 (2024), 186332.

A Dataset and Data Preparation

A.1 Dataset Specifications

We conduct experiments on 13 benchmark datasets from the UCR and UEA repositories. Among them, those belonging to UCR include ArrowHead, CBF, FaceFour, MelbournePedestrian, OSULeaf, Plane, and Symbols. The datasets from UEA include Trace, Epilepsy, NATOPS, EthanolConcentration, FaceDetection, and FingerMovements. The statistics of the datasets are shown in Table 5.

Table 5: Statistics of 13 benchmark datasets from UCR and UEA repositories.

Dataset	Classes	Samples	Dimensions	Length	Type
ArrowHead	3	211	1	251	Image
CBF	3	930	1	128	Simulated
FaceFour	4	112	1	350	Image
MelbournePedestrian	10	3650	24	24	Traffic
OSULeaf	6	442	1	427	Image
Plane	7	210	1	144	Image
Symbols	6	1020	1	398	Image
Trace	4	200	1	275	Sensor
Epilepsy	4	275	3	207	HAR
NATOPS	6	360	24	51	HAR
EthanolConcentration	4	524	3	1751	Other
FaceDetection	2	9414	144	62	-
FingerMovements	2	416	28	50	EEG

A.2 Noise Type Definitions

The following three types of noise are introduced to assess the model's performance under varying noise conditions:

- **Symmetric Noise (Sym):** Uniform label corruption across all classes, where each sample has a probability p of being mislabeled. And the probability of mislabeling the sample as another class is $\frac{p}{c-1}$, where c is the number of classes.
- **Asymmetric Noise (Asym):** This noise denotes a type of class-dependent label noise, where errors occur between specific classes. For instance, class A can be mislabeled as class B, class B as class C, and so forth, with a probability p, reflecting structured, non-random mislabeling.
- **Instance-Dependent Noise (IDN):** The probability of label corruption varies based on the characteristics of each sample, following a procedure in previous work [52].

B Implementation Details

The encoder $f(\cdot)$ consists of four convolutional blocks with *stride* = 2, *padding* = 2, and *filters* = [128, 128, 256, 256], where each block consists of a 1D convolution layer to encode the input data. The encoder is concluded by a linear projection layer that reduces the output dimension to 32, and the final linear classifier consists of 128 hidden units.

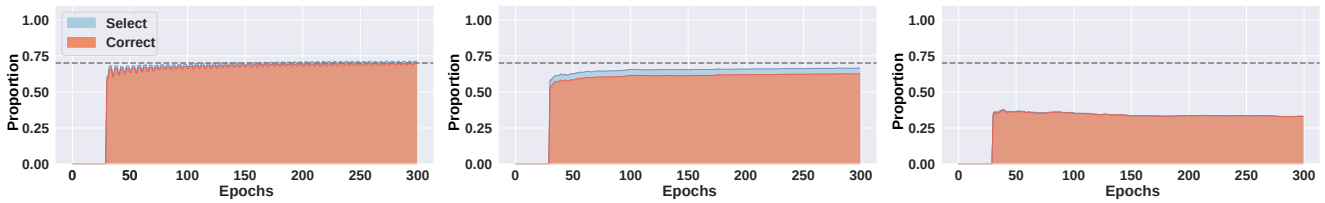
C More Experimental Details

C.1 Detailed Experimental Results

We provide the detailed experimental results for DREAM across all 13 benchmark datasets under various noise types and ratios in Table 6. These results correspond to the performance of DREAM presented in Table 1, providing the average F1-score and standard deviation over five independent runs for each setting.

C.2 Different Augmentation Strategies Setting

We evaluate the impact of various augmentation strategies through experiments on 13 datasets with 30% symmetric and 30% asymmetric noise. The detailed experimental settings and parameter configurations for these strategies, implemented via the *tsaug* library, are listed in Table 7.



(a) DREAM (b) DREAM w/o Data Simulation (c) DREAM w/o APM Threshold

Figure 6: Proportion of the selected samples on the ArrowHead dataset. The grey line represents the real clean sample rate, while the blue and red lines indicate the proportion of selected samples and clean samples relative to the total set, respectively.

Table 6: The average F1-score and standard deviation of DREAM on 13 datasets with different noise settings.

Datasets	Sym				Asym				IDN	
	15%	30%	45%	60%	10%	20%	30%	40%	30%	40%
ArrowHead	0.853(0.027)	0.860(0.052)	0.856(0.037)	0.635(0.157)	0.872(0.054)	0.886(0.024)	0.831(0.055)	0.776(0.070)	0.842(0.038)	0.715(0.093)
CBF	1.000(0.000)	1.000(0.000)	0.978(0.006)	0.769(0.096)	1.000(0.000)	1.000(0.000)	0.993(0.007)	0.943(0.019)	0.995(0.005)	0.991(0.006)
FaceFour	0.930(0.035)	0.929(0.033)	0.738(0.132)	0.562(0.108)	0.964(0.034)	0.936(0.068)	0.904(0.060)	0.755(0.128)	0.800(0.066)	0.729(0.186)
MelbournePedestrian	0.870(0.007)	0.872(0.007)	0.884(0.007)	0.858(0.012)	0.869(0.010)	0.881(0.005)	0.873(0.004)	0.850(0.007)	0.883(0.007)	0.876(0.004)
OSULeaf	0.890(0.034)	0.855(0.049)	0.807(0.064)	0.704(0.069)	0.898(0.057)	0.878(0.048)	0.855(0.051)	0.819(0.049)	0.881(0.056)	0.827(0.081)
Plane	1.000(0.000)	1.000(0.000)	0.995(0.01)	0.906(0.091)	0.995(0.010)	1.000(0.000)	0.957(0.074)	0.910(0.093)	0.985(0.020)	0.79(0.154)
Symbols	0.986(0.005)	0.976(0.009)	0.985(0.008)	0.967(0.010)	0.991(0.002)	0.988(0.007)	0.976(0.012)	0.937(0.037)	0.983(0.005)	0.971(0.009)
Trace	1.000(0.000)	1.000(0.000)	1.000(0.000)	0.788(0.147)	1.000(0.000)	1.000(0.000)	1.000(0.000)	0.912(0.115)	1.000(0.000)	0.936(0.061)
Epilepsy	0.951(0.033)	0.860(0.065)	0.672(0.085)	0.449(0.034)	0.949(0.048)	0.874(0.045)	0.865(0.085)	0.718(0.038)	0.869(0.049)	0.722(0.101)
NATOPS	0.814(0.032)	0.683(0.033)	0.534(0.050)	0.446(0.043)	0.853(0.013)	0.788(0.035)	0.711(0.067)	0.615(0.070)	0.712(0.031)	0.617(0.046)
EthanolConcentration	0.237(0.029)	0.227(0.024)	0.261(0.036)	0.262(0.031)	0.232(0.020)	0.217(0.038)	0.214(0.032)	0.203(0.039)	0.231(0.028)	0.213(0.027)
FingerMovements	0.574(0.041)	0.540(0.050)	0.568(0.044)	0.516(0.031)	0.599(0.035)	0.567(0.054)	0.515(0.028)	0.519(0.032)	0.526(0.026)	0.561(0.047)
FaceDetection	0.648(0.017)	0.612(0.015)	0.532(0.013)	0.512(0.013)	0.659(0.033)	0.645(0.024)	0.619(0.019)	0.588(0.027)	0.610(0.018)	0.574(0.010)
Avg	0.827(0.020)	0.801(0.026)	0.755(0.038)	0.644(0.065)	0.837(0.024)	0.820(0.027)	0.793(0.038)	0.734(0.056)	0.794(0.027)	0.732(0.063)

Table 7: Augmentation Strategies and Parameters.

Strategies	Settings
TimeWarp	Number of Speed Changes = 5, Maximum Speed Ratio = 3
GaussNoise	Scale = 0.015
Convolve	Window = *'flattop'*, Size = 10
Drift	Maximum Drift = 0.2, Number of Drift Points = 5
Crop	Size = Original Length × 2/3, Adjusted Size = Original Length

C.3 Comparison of Sample Selection Strategies

To further assess the effectiveness of the proposed dual data-centric sample separation strategy, we conduct a detailed comparison of different sample selection approaches during training. Fig. 6 presents the proportion of total selected samples and correctly identified clean samples on the ArrowHead dataset. As shown in Fig. 6(a), the complete DREAM framework, which integrates the adaptive pseudo-margin threshold with noise-aware data simulation, achieves near-optimal performance. The selected set and the clean subset coincide almost perfectly and converge to the actual clean sample rate after a short warm-up period. This highlights that DREAM effectively distinguishes clean samples from noisy ones with high precision, even under severe label corruption. In contrast, while the variant without data simulation maintains high precision, its recall is reduced, as the selection criteria become less adaptive to the actual noise distribution. Removing the APM threshold leads to a severe failure in sample separation, with the model identifying only a small fraction of clean samples.