DISCREPANCY-AWARE CONTRASTIVE ADAPTATION IN MEDICAL TIME SERIES ANALYSIS

Yifan Wang[*]
School of Medicine, The Chinese University of Hong Kong, Shenzhen
yifanwang13@link.cuhk.edu.cn

Hongfeng Ai[*]
School of Medicine, The Chinese University of Hong Kong, Shenzhen
aihongfeng@cuhk.edu.cn

Ruiqi Li[*]
University of the Chinese Academy of Sciences
liruiqi1@sia.cn

Maowei Jiang[*]
Shenzhen International Graduate School, Tsinghua University
jiangmaowei@sia.cn

Ruiyuan Kang
Wave and Machine Intelligence Department,
Technology Innovation Institute
ruiyuan.kang@tii.ae

Jiahua Dong
Mohamed bin Zayed University of
Artificial Intelligence
dongjiahua1995@gmail.com

Cheng Jiang[†]
School of Medicine, The Chinese University of Hong Kong, Shenzhen
jiangcheng@cuhk.edu.cn

Chenzhong Li[†]
School of Medicine, The Chinese University of Hong Kong, Shenzhen
lichenzhong@cuhk.edu.cn

ABSTRACT

In medical time series disease diagnosis, two key challenges are identified. First, the high annotation cost of medical data leads to overfitting in models trained on label-limited, single-center datasets. To address this, we propose incorporating external data from related tasks and leveraging AE-GAN to extract prior knowledge, providing valuable references for downstream tasks. Second, many existing studies employ contrastive learning to derive more generalized medical sequence representations for diagnostic tasks, usually relying on manually designed diverse positive and negative sample pairs. However, these approaches are complex, lack generalizability, and fail to adaptively capture disease-specific features across different conditions. To overcome this, we introduce LMCF (Learnable Multi-views Contrastive Framework), a framework that integrates a multi-head attention mechanism and adaptively learns representations from different views through inter-view and intra-view contrastive learning strategies. Additionally, the pre-trained AE-GAN is used to reconstruct discrepancies in the target data as disease probabilities, which are then integrated into the contrastive learning process. Experiments on three target datasets demonstrate that our method consistently outperforms other seven baselines, highlighting its significant impact on healthcare applications such as the diagnosis of

[*] These authors contributed equally.

[†] Corresponding author.

myocardial infarction, Alzheimer's disease, and Parkinson's disease. We release the source code at xxxxx.

Keywords Medical Time Series · Contrastive Learning · Multi-View Representation · Discrepancy Estimation · Disease Diagnosis

1 Introduction

Medical diagnosis tasks, such as Alzheimer's disease Parkinsonism require high precision and exhibit low tolerance for errors due to their critical impact on patient outcomes. In data-driven methods, time-series data is accordingly preferred to provide comprehensive information to support the accurate modeling [1, 2]. However, the time-series data used are naturally complex with hierarchical architectures [3]. In contrast, the reliable labeled data is limited in its amount and diversity [4, 5], as they are often from the same center [6, 7]. These characteristics result in two key challenges: (1) difficulty in learning effective patterns and (2) increased risk of overfitting, which undermines the generalizability of data-driven medical diagnosis models.

To facilitate better pattern extraction, domain knowledge has been used to structure time-series data across multiple levels.[8, 3, 9]. In addition, representation learning methods, particularly contrastive learning, are employed to derive more informative representations. Although these endeavors improve the performance of ML model, the human bias introduced in data leveling and positive-negative pairing [10, 11, 12] in contrastive learning also constrain the model performance. On the other hand, to mitigate overfitting, external normal samples have been introduced as auxiliary resources to compensate for the limitations of training data and to alleviate the risk of "single-center bias"[13, 14]. We admit the contribution of previous attempts, and think about that (1) whether we can design a more robust contrastive learning algorithm to avoid the labeling expense and bias, to better capture the effective patterns. (2) whether we can better utilize the external resource to avoid overfitting due to the data limitation.

Accordingly, we propose **Discrepancy Adaptive Contrastive learning** for medical diagnosis tasks. It includes two modules:

(1) Discrepancy Estimator. Leveraging large-scale public normal data, we train a reconstructor based on an encoder-decoder architecture with GAN-style enhancement (AE-GAN). This reconstructor effectively models the distribution of normal samples. When encountering abnormal inputs, the resulting reconstruction error quantifies the discrepancy between the input and the learned normal distribution. This discrepancy signal supplements the limited abnormal data, provides additional supervision, and helps mitigate overfitting during training.

(2) Adaptive Contrastive Learner. Instead of relying on handcrafted positive-negative sample pairing, the learner employs Multi-Head Attention (MHA) to adaptively identify contrastive relationships. Specifically, it contrasts embeddings across attention heads (Inter-View) and across different patients (Intra-View). This mechanism enables the model to extract informative representations while reducing representational redundancy. The resulting features are further fine-tuned in a supervised manner for downstream medical diagnosis tasks.

The experimental results demonstrate that our method outperforms the SOTA methods across three representative clinical datasets, covering myocardial infarction, Alzheimer's disease, and Parkinson's disease. Notably, it still has decent performance when merely 10% of the data is labeled, which highlights the effectiveness of the DAAC design.

2 Related Work

Medical time series Medical diagnosis often relies on physiological time-series signals such as EEG [15], ECG [16], EMG [17], and EOG [18], which reflect complex temporal dynamics. Unlike images or texts, these data exhibit subtle and hierarchical variations [10], increasing modeling difficulty. To address this, various representation learning methods have been proposed. TS2Vec [19] organizes data across temporal scales, CLOCS [9] models spatial-temporal invariance, and COMET [8] leverages four-level hierarchy to improve representation granularity. However, medical datasets are often limited to single-center collections with costly manual annotations [20, 21], leading to data scarcity and overfitting [22, 23].

Supervised learning for time series Supervised methods have shown strong performance in medical diagnostics, such as U-Sleep for EEG staging [24], L-SeqSleepNet for cross-night generalization [25], and Inception-based networks for EOG classification [26]. To address annotation bottlenecks, techniques like SimGAN [27] generate realistic ECG signals, with GAN-based augmentation now mainstream for ECG [28] and EEG [29]. Yet, detecting rare events and adapting across domains remains challenging [30, 31, 32, 33], with supervised learning constrained by label imbalance and limited diversity.

Contrastive learning for time series Self-supervised learning—especially contrastive methods—has become a promising alternative for time-series modeling [34]. By constructing positive and negative pairs through data augmentation [35, 19, 36, 37], these approaches learn discriminative features without labels. For example, SoftCLT [38] introduces soft assignment to improve flexibility but is sensitive to noise; CPC [39] predicts future representations but suffers from negative sampling bias; TS-TCC [40] contrasts temporal and contextual views yet struggles with dynamic or short sequences. Improving representation robustness in low-data regimes remains an open challenge.

3 Proposed Method:DAAC

4 Proposed Method:DAAC

As discussed above, in order to better extract more effective feature extraction and also avoid model overfitting on limited data, DAAC improved the paradigm of contrastive learning + downstream fine-tuning with two sophisticatedly designed modules. (1) **Discrepancy Estimator (E_D)** To avoid overfitting on limited data, we embed the information of the external normal data distribution into an Encoder-Decoder architecture reconstructor via GAN-style training (Sec.4.1), which is named as Discrepancy Estimator (E_D). Because the well-trained reconstructor captures the normal data distribution, and thus the reconstruction error reflects discrepancies of abnormal samples from normal ones. Appendix X demonstrates this experiment in detail. Such a design, utilizing the complementary information from normal data to alleviate the limitation of the given target data. The calculated discrepancy is used as a feature to augment the original data. (2) **Adaptive Contrastive Learner (E_L)**. To better capture effective features without further labeling (Sec. 4.2), we use the Multi-Head Attention (MHA) to embed features, which are sufficiently optimized through adaptively differentiating the embedding between attention heads and patients (inter-view and intra-view contrastive mechanisms, respectively). Meanwhile, features from diverse levels are used to provide more comprehensive information.

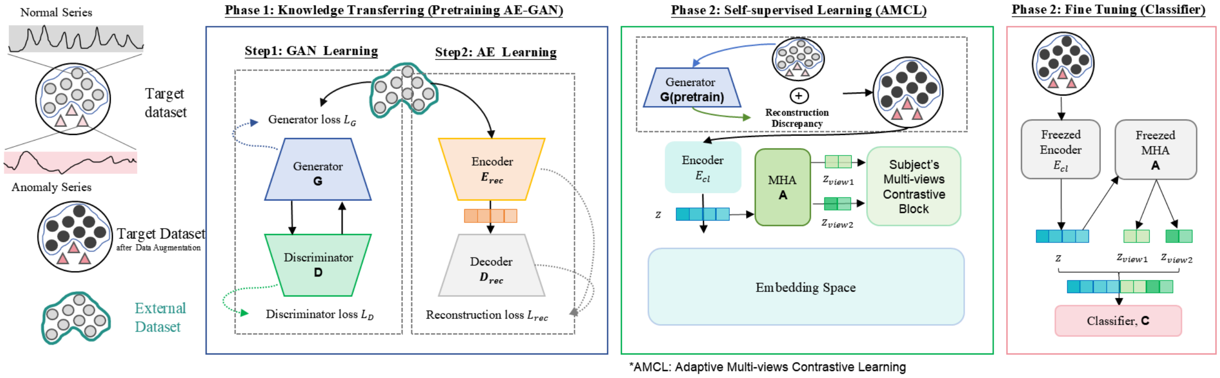

Figure 1: Overview of our three-stage training framework. We first pretrain a Discrepancy Estimator (E_D) on normal data to extract reconstruction-based features. These features are then fused with raw inputs for contrastive representation learning with an MHA encoder. Finally, the learned encoder is fine-tuned with supervision for downstream tasks.

As shown in Figure 1, the training process consists of three stages. **Stage 1: Training and inference of Discrepancy Estimator**. A Discrepancy Estimator (E_D) is trained using data from normal patients in $\mathcal{D}'$, whose samples are expressed as $\mathbf{x}'_i$. Then, it is used to perform inference on all samples in $\mathcal{D}$, whose samples are expressed as $\mathbf{x}_i$, the calculated overall reconstruction error is used as an additional feature. fused with the raw data, resulting in a feature-augmented dataset with the number of features increased from F to $F + 1$. **Stage 2: Training of Adaptive Contrastive Learner (E_L)** An encoder E_L with a multi-head attention block embed the augmented dataset into feature space, with the consideration of diverse data level and diverse views. Respecively, The well-trained encoder generates series-level features $\mathbf{h}_i \in \mathbb{R}^{T \times C}$ with C dimensions, and view-level features $\mathbf{g}_i \in \mathbb{R}^{T \times V \times d}$ with V views and d dimensions for each view. **Stage 3: Supervised fine-tuning** These representations are then used to train the classifier for specific downstream medical diagnostic tasks.

4.1 Discrepancy Estimator

Although the target dataset $\mathcal{D}$ contains limited and non-diverse data, the massive normal data $\mathcal{D}'$ from other institutions are available. Although it varies from $\mathcal{D}$, but both of them share similar characteristics, which accordingly provides complementary information for extracting better feature and alleviating overfitting on targeted dataset $\mathcal{D}$. As mentioned above, we utilize the discrepancy to reflect the information of the normal dataset, an Encoder-Decoder Discrepancy Estimator (E_D) enhanced by GAN-style training is used to provide the function.

4.1.1 Encoder-Decoder Generator

The generator G_{gen} is composed of an encoder E_{gen} and a decoder D_{gen}, which work together to learn the distribution of healthy samples $\mathbf{x}'_i$ from the external dataset D'. In details, The encoder maps the healthy samples $\mathbf{x}'_i$ to a latent space representation and the decoder reconstructs the data from the representation to the generated data $\hat{\mathbf{x}}'_i$:

$$\hat{\mathbf{x}}'_i = G_{gen}(\mathbf{x}'_i) = D_{gen}(E_{gen}(\mathbf{x}'_i)) \tag{1}$$

The generator is trained to minimize the reconstruction loss, ensuring that the generated data $\hat{\mathbf{x}}'_i$ closely approximates the original healthy samples $\mathbf{x}'_i$. This enables the generator to capture the underlying distribution of healthy data, which is critical for subsequent feature augmentation and knowledge transfer.

4.1.2 Discriminator

In order to maximally enhance the performance of generator, The discriminator D_{dis} is used to distinguish between real healthy samples $\mathbf{x}'_i$ from the external dataset D' and generated data $\hat{\mathbf{x}}'_i$. The output of D_{dis} represents the probability that the sample is from healthy subject. The details about the loss function can be seen in Appendix D.

4.1.3 Inference on Internal Data

After pretraining, we use the Generator model to perform inference on all samples $\bar{\mathbf{x}}_i$ of D, and the reconstruction error $\mathcal{E}$ is computed by:

$$\mathcal{E} = \text{MSE}(G_{gen}(\bar{\mathbf{x}}_i), \bar{\mathbf{x}}_i) \tag{2}$$

Since the generator is pretrained on healthy samples $\mathbf{x}'_i$ from D', the reconstruction error $\mathcal{E}$ is smaller for healthy samples $\mathbf{x}_i$ in the target dataset and larger for abnormal samples $\mathbf{x}_i$. Thus, the magnitude of $\mathcal{E}$ indirectly reflects the abnormality of the sample $\mathbf{x}_i$, i.e., the probability of disease. In our design, we compute sequence-level reconstruction error rather than point-wise error. This decision is motivated by the observation that point-wise errors are often highly sensitive to local fluctuations and noise, which are common in medical time series due to physiological variability or sensor artifacts. In contrast, sequence-level feedback captures the global consistency and overall structural deviation from expected normal patterns, leading to more robust and stable anomaly signals. Additional details can be found in Appendix F

The reconstruction error $\mathcal{E}$ is concatenated with the original features x_i of the target dataset to form an augmented feature set:

$$\mathbf{x}_i = \text{concat}(\bar{\mathbf{x}}_i, \mathcal{E}) \tag{3}$$

The augmented dataset $\mathbf{x}_i$ is used for subsequent model development.

4.2 Adaptive Contrastive Learner

Conventional contrastive learning relies on manually designed heuristics for constructing positive and negative pairs, which may introduce noise. To address this limitation, we propose an adaptive contrastive loss that leverages multi-head attention to guide the model in learning which views should be pulled together or pushed apart. This approach eliminates the reliance on external labels and handcrafted pairing strategies, while producing robust and semantically rich representations. The Adaptive Contrastive Learner E_L of our approach is designed with two components.

The first component is a dilated convolutional network $h(\mathbf{x})$, which broadens the receptive field of the sequence. This network is crucial for capturing a broader range of information and enhancing the generalizability of the temporal representations. On the other hand, inspired from the local transformations to craft diverse data views in the latent space for contrastive learning [41], we recognized that MHA can significantly reduce the manual cost of designing positive and negative sample pairs from different perspectives of data. Consequently, we incorporated a second component into our encoder, which is our innovative learnable multi-views network $g(\mathbf{x})$.

It introduces MHA to contrastive learning for extracting varying view representations adaptively. By leveraging both inter-wise and intra-wise contrastive losses, we ensure that the representations not only encompass diverse views but also exhibit distinct separability among different subjects within each view as Figure 2 shown.

Furthermore, to capture the intrinsic hierarchical information embedded in medical data, we adopt the concept from COMET [8]. Our contrastive learning considers subject, trial, epoch, and temporal, and optimizes the model using multiple InfoNCE losses [39]. Ultimately, we obtain representations that are beneficial for downstream medical time-series tasks.

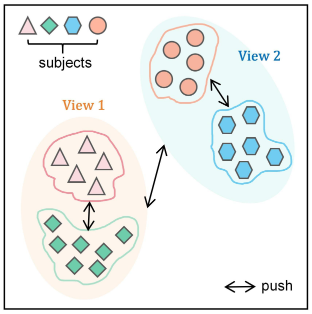

Figure 2: Mechanism of Multi-View Contrastive Learning

4.2.1 Subject-wise Contrastive Loss L_S

Subject may exhibit unique physiological characteristics and pathological variations. Consequently, the subject-wise contrastive learning is designed to enhance the discriminative representation of subject-specific features by increasing the similarity among samples from the same subject and the separability among samples from different subjects.

Specifically, let $\mathbf{x}_{i,s}$ denote the i-th anchor sample in the given batch, which is from subject s. The positive sample set S_i^+ for $\mathbf{x}_{i,s}$ consists of all other samples $\mathbf{x}_{j,s^+}$ with the same subject s^+. Correspondingly, the negative sample set S_i^- includes all samples $\mathbf{x}_{j,s^-}$ from different subjects s^-. Based on this delineation of positive and negative samples, we define the subject-level contrastive loss function as follows:

$$L_S = \frac{-1}{M} \sum_{i=1}^{M} \sum_{\mathbf{x}_{j,s^+} \in S_i^+} \log \frac{\exp\left(s(\mathbf{h}_{i,s}, \mathbf{h}_{j,s^+})/\tau\right)}{\sum_{\mathbf{x}_{j,s^-} \in S_i^-} \exp\left(s(\mathbf{h}_{i,s}, \mathbf{h}_{j,s^-})/\tau\right)} \tag{4}$$

where $j \neq i$ and $s(\cdot, \cdot)$ denotes the cosine similarity between the respective representations. $\exp(\cdot)$ is the exponential function and M represents the batch size. $\mathbf{h}_{i,s}$ corresponds to the output of the network $h(x)$ applied to the sample $\mathbf{x}_{i,s}$, i.e., $\mathbf{h}_{i,s} = h(\mathbf{x}_{i,s})$. Similarly, $\mathbf{h}_{j,s^+}$ and $\mathbf{h}_{j,s^-}$ are derived from $h(\mathbf{x}_{j,s^+})$ and $h(\mathbf{x}_{j,s^-})$. The parameter τ serves as a temperature parameter.

4.2.2 Trial-wise Contrastive Loss L_R

Trials may demonstrate distinctive experimental conditions and variations in outcomes. To address this, trial-wise contrastive learning enhances trial-specific feature representation by promoting similarity within trials and separability between trials.

Concretely, assume $\mathbf{x}_{i,r}$ represent the i-th anchor sample in the batch, sourced from trial r. The set of positive samples R_i^+ associated with $\mathbf{x}_{i,r}$ encompasses all other samples $\mathbf{x}_{j,r^+}$ that are from the same trial r^+. Conversely, the set of negative samples R_i^- consists of all samples $\mathbf{x}_{j,r^-}$ originating from different trials r^-. Therefore, we establish the trial-wise contrastive loss function as below:

$$L_R = \frac{-1}{M} \sum_{i=1}^{M} \sum_{\mathbf{x}_{j,r^+} \in R_i^+} \log \frac{\exp\left(s(\mathbf{h}_{i,r}, \mathbf{h}_{j,r^+})/\tau\right)}{\sum_{\mathbf{x}_{j,r^-} \in R_i^-} \exp\left(s(\mathbf{h}_{i,r}, \mathbf{h}_{j,r^-})/\tau\right)} \tag{5}$$

where, $\mathbf{h}_{i,r}$ denotes the representation obtained by applying the network $h(\mathbf{x})$ to the sample $\mathbf{x}_{i,r}$, formally expressed as $\mathbf{h}_{i,r} = h(\mathbf{x}_{i,r})$. Consistent with this notation, $\mathbf{h}_{j,r^+}$ and $\mathbf{h}_{j,r^-}$ represent the feature representations resulting from the application of $h(\mathbf{x})$ to the corresponding positive and negative samples $\mathbf{x}_{j,r^+}$ and $\mathbf{x}_{j,r^-}$.

4.2.3 Epoch-wise Contrastive Loss L_E

For epoch-wise contrastive learning, our hypothesis is that a sample which undergoes minor modifications, like temporal masking, should maintain a similar pattern compared to other sample. Hence, for each sample $\mathbf{x}_i$ in batch, we employ a data augmentation strategy that is random continuous masking. This strategy involves applying two independent augmentations to $\mathbf{x}_i$, resulting in augmented samples $\tilde{\mathbf{x}}_i^1$ and $\tilde{\mathbf{x}}_i^2$. Since these augmented samples originate from the same anchor sample $\mathbf{x}_i$, they are encouraged to have high similarity between their feature representations and are considered as positive pair $(\tilde{\mathbf{x}}_i^1, \tilde{\mathbf{x}}_i^2)$. In contrast, negative sample pairs are formed from augmented samples derived from different anchor samples, such as $\left(\tilde{\mathbf{x}}_i^1, \tilde{\mathbf{x}}_j^1\right)$ and $\left(\tilde{\mathbf{x}}_i^1, \tilde{\mathbf{x}}_j^2\right)$. Finally, the epoch-level loss function is formulated as:

$$L_E = \frac{-1}{M} \sum_{i=1}^{M} \log \frac{\exp\left(\tilde{\mathbf{h}}_i^1 \cdot \tilde{\mathbf{h}}_i^2\right)}{\sum\limits_{j=1, j \neq i}^{M} \left(\exp\left(\tilde{\mathbf{h}}_i^1 \cdot \tilde{\mathbf{h}}_j^1\right) + \exp\left(\tilde{\mathbf{h}}_i^1 \cdot \tilde{\mathbf{h}}_j^2\right)\right)} \tag{6}$$

where $\tilde{\mathbf{h}}_i^1 = h(\tilde{\mathbf{x}}_i^1)$ and $\tilde{\mathbf{h}}_i^2 = h(\tilde{\mathbf{x}}_i^2)$.

4.2.4 Temporal-wise Contrastive Loss L_T

The objective of this loss function is to ensure that two augmented samples originating from the same sample exhibit similar representations when observed at the same timestamp, while their representations differ when observed at other timestamps. Temporal-wise contrastive learning leverages the temporal consistency within the data. In this context, let us consider the anchor sample $\mathbf{x}_{i,t}$ of the i-th data point at time t within a batch. By applying temporal masking, we generate two augmented samples as $\tilde{\mathbf{x}}_{i,t}^1$ and $\tilde{\mathbf{x}}_{i,t}^2$. The pair $(\tilde{\mathbf{x}}_{i,t}^1, \tilde{\mathbf{x}}_{i,t}^2)$ is classified as a positive sample pair because they are derived from the same anchor sample at timestamp t. Conversely, negative sample pairs are defined as $(\tilde{\mathbf{x}}_{i,t}^1, \tilde{\mathbf{x}}_{i,t^-}^1)$ and $(\tilde{\mathbf{x}}_{i,t}^1, \tilde{\mathbf{x}}_{i,t^-}^2)$, where $t \neq t^-$. These negative pairs are constructed from augmented samples originating from different temporal points, thereby encouraging the model to learn distinct representations for different time observations. The Temporal-wise contrastive loss is shown as:

$$L_T = \frac{-1}{T \cdot M} \sum_{t=1}^{T} \sum_{i=1}^{M} \log Q_{i,t} \tag{7}$$

$$Q_{i,t} = \frac{\exp\left(\tilde{\mathbf{h}}_{i,t}^1 \cdot \tilde{\mathbf{h}}_{i,t}^2\right)}{\sum\limits_{t=1, t\neq t^-}^{T} \left(\exp\left(\tilde{\mathbf{h}}_{i,t}^1 \cdot \tilde{\mathbf{h}}_{i,t^-}^1\right) + \exp\left(\tilde{\mathbf{h}}_{i,t}^1 \cdot \tilde{\mathbf{h}}_{i,t^-}^2\right)\right)} \tag{8}$$

where $\tilde{\mathbf{h}}_{i,t}^1 = h(\tilde{\mathbf{x}}_{i,t}^1)$. Similarly, analogous notations with h are derived from the network $h(\mathbf{x})$.

4.2.5 Inter-view Contrastive Loss L_{IRV}

The inter-view contrastive loss we propose aims to encourage the model to capture informative features across different augmented views adaptively. Essentially, inter-view loss drives the representations from different views to be distinguishable from each other, thereby promoting the diversity of views.

Let us consider a batch of samples as $\mathbf{x}_i$, where $i = 1, \ldots, M$. For each sample $\mathbf{x}_i$, we apply data augmentation to generate two augmented samples, $\tilde{\mathbf{x}}_i^1$ and $\tilde{\mathbf{x}}_i^2$. These augmented samples are subsequently fed into the MHA module, denoted as $g(\mathbf{x})$. This module generates representations $\tilde{\mathbf{g}}_{i,v}^k$ for each augmented branch $k \in \{1, 2\}$ and different views v, where $g(\mathbf{x}_i^k) = \tilde{\mathbf{g}}_{(i,v)}^k$. The view representations from the given batch are then aggregated into $\tilde{\mathbf{g}}_v^k = \left\{\tilde{\mathbf{g}}_{1,v}^k, \tilde{\mathbf{g}}_{2,v}^k, \ldots, \tilde{\mathbf{g}}_{M,v}^k\right\}$.

Assume we have two views for each branch that is $V = 2$, To facilitate inter-view contrastive learning, we define positive sample pairs as those from the same view, such as $(\tilde{\mathbf{g}}_1^1, \tilde{\mathbf{g}}_1^2)$ and $(\tilde{\mathbf{g}}_2^1, \tilde{\mathbf{g}}_2^2)$. Conversely, negative sample pairs are those from different views, such as $(\tilde{\mathbf{g}}_1^1, \tilde{\mathbf{g}}_2^1)$ and $(\tilde{\mathbf{g}}_2^1, \mathbf{g}_1^2)$. Hence, the inter-view contrastive loss is formulated as:

$$L_{IRV} = \frac{-1}{V} \sum_{i=1}^{V} \log \frac{\exp(\mathrm{s}(\tilde{\mathbf{g}}_i^1, \tilde{\mathbf{g}}_i^2))}{\sum_{j=1, j\neq i}^{V} \exp(\mathrm{s}(\tilde{\mathbf{g}}_i^1, \tilde{\mathbf{g}}_j^2))} \tag{9}$$

4.2.6 Intra-View Contrastive Loss L_{IAV}

In addition to inter-view loss, we also introduce the intra-view loss to further refine the representations within the same view. This loss is designed to enhance the discriminative power of representations by contrasting samples from the same view but different subjects, thus encouraging the model to capture subject-specific view features more effectively.

Similarly, assume we have subject-specific samples $\mathbf{x}_{i,s}$, where i ranges from 1 to M. For each of them, we apply data augmentation, resulting in augmented samples $\tilde{\mathbf{x}}_i^1$ and $\tilde{\mathbf{x}}_i^2$. These augmented samples are then processed through a MHA network $g(\mathbf{x})$, which produces subject-specific view representations $\tilde{\mathbf{g}}_{i,v}^k$ for each augmented branch $k \in \{1, 2\}$ and various views v, i.e., $\tilde{\mathbf{g}}_{i,v,s}^k = g(\mathbf{x}_{i,s}^k)$.

Suppose we have V views and S subjects. To enable intra-view contrastive learning, we identify positive sample pairs as those from the same view and subject. Therefore, we have positive pairs that are $\left(\tilde{\mathbf{g}}_{i,v,s}^1, \tilde{\mathbf{g}}_{i,v^+,s^+}^2\right)$. On the other hand, The negative sample pairs are defined as the samples that differ in subject at the given view, which are $\left(\tilde{\mathbf{g}}_{i,v,s}^1, \tilde{\mathbf{g}}_{i,v^+,s^-}^2\right)$

$$\begin{aligned} L_{IAV} &= \frac{-1}{V \cdot M} \sum_{v=1}^{V} \sum_{i=1}^{M} \sum_{\mathbf{x}_{j,v^+,s^+} \in S_i^+} \log K_{i,v,s} \\ K_{i,v,s} &= \frac{\exp(s(\tilde{\mathbf{g}}_{i,v,s}^1 \cdot \tilde{\mathbf{g}}_{j,v^+,s^+}^2)/\tau)}{\sum\limits_{\mathbf{x}_{j,v^{a+},s^-} \in S_i^-} \exp(s(\tilde{\mathbf{g}}_{i,v,s}^1 \cdot \tilde{\mathbf{g}}_{j,v^+,s^-}^2)/\tau)} \end{aligned} \tag{10}$$

where S_i^+ includes the augmented sample set of $\tilde{\mathbf{x}}_{j,v^+,s^+}^2$ that belong to the same subject as the given augmented sample $\tilde{\mathbf{x}}_{i,v,s}^2$ under the same view v. Conversely, S_i^- consists of the augmented samples that come from the same view but different subjects.

Thus, our view loss comprises both inter-view and intra-view contrastive losses:

$$\mathcal{L}_V = L_{IRV} + L_{IAV} \tag{11}$$

Finally, the overall contrastive loss function L is defined as:

$$\mathcal{L} = \lambda_S \cdot L_S + \lambda_R \cdot L_R + \lambda_E \cdot L_E + \lambda_T \cdot L_T + \lambda_V \cdot L_V \tag{12}$$

where $\lambda_S : \lambda_R : \lambda_E : \lambda_T : \lambda_V = 1 : 1 : 1 : 1 : 2$ in our practice. The details about loss weight sensitivity could be seen in Appendix D.2

4.3 Fine Tuninig

In order to use the Encoder E_L of DAAC for downstream tasks, we utilize fine-tuning as follows:

$$\hat{y}_i = C(E_L(\mathbf{x}_i)) = C(h(\mathbf{x}_i), g(\mathbf{x}_i)) \tag{13}$$

where $\hat{y}$ is the classification result for downstream medical disease diagnosis, and the training optimization objective is binary cross-entropy.

5 Experiments

This study utilized three target datasets corresponding to Alzheimer's disease, myocardial infarction, and Parkinson's disease. Additionally, three external datasets from independent institutions were incorporated to facilitate cross-center knowledge transfer. A detailed description of all datasets and the preprocessing protocols is provided in Appendix C. All datasets are publicly available and had received institutional review board (IRB) approval prior to their release.

All experiments were conducted on a workstation equipped with four NVIDIA RTX 4090 GPUs (24GB each). The models were implemented in PyTorch 1.13.1 with CUDA 11.6. All results were obtained under the same hardware configuration to ensure fairness and consistency in comparison.

5.1 Results on Partial Fine-tuning

To evaluate the representation quality of our proposed DAAC quickly, we adopt the Partial Fine-Tuning (PFT). PFT refers to combining the Encoder with frozen parameters and a trainable logistic regression linear classifier C. In this setup, we utilizes 100% of the labeled data for training the supervised classification. The PFT results on the AD dataset are shown in Table 1. Our proposed DAAC model excels in all metrics, including Accuracy, Precision, Recall, F1 score, AUROC, and AUPRC, demonstrating the power of our representation learning.

Table 1: Partial Fine-tuning Results. **ACL** = Adaptive Contrastive Learner; **DE** = Discrepancy Estimator; **COMET+ACL**, **COMET+DE** represent ablations with only ACL or DE added to COMET; **DAAC** is the final model combining both ACL and DE. Bold indicates the best performance, and underline denotes the second best.

Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
TS2vec	66.48 ± 5.33	67.72 ± 5.09	67.40 ± 5.20	66.32 ± 5.46	74.12 ± 6.88	72.96 ± 7.21
TF-C	77.03 ± 2.80	75.79 ± 5.07	64.27 ± 5.03	64.85 ± 5.56	80.71 ± 4.03	79.27 ± 4.15
Mixing-up	46.16 ± 1.38	52.62 ± 4.90	50.81 ± 1.32	37.37 ± 1.98	64.42 ± 6.49	62.85 ± 6.07
TS-TCC	59.71 ± 8.63	61.66 ± 8.63	60.33 ± 3.26	58.66 ± 8.39	67.53 ± 10.04	68.33 ± 9.37
SimCLR	57.16 ± 2.05	56.67 ± 3.91	53.57 ± 2.21	49.11 ± 4.26	56.67 ± 3.91	52.10 ± 1.41
CLOCS	66.99 ± 2.84	67.17 ± 2.96	67.33 ± 2.99	66.91 ± 2.88	73.71 ± 3.62	72.58 ± 3.54
COMET	76.09 ± 4.21	77.36 ± 3.97	74.68 ± 4.62	74.80 ± 4.83	81.30 ± 4.97	80.50 ± 5.31
COMET+ACL	77.42 ± 4.40	78.64 ± 3.73	76.11 ± 4.98	76.26 ± 5.26	82.44 ± 5.19	81.30 ± 5.56
COMET+DE	79.60 ± 6.01	**80.41 ± 5.86**	78.50 ± 6.51	78.75 ± 6.46	**86.32 ± 6.54**	**85.85 ± 6.99**
DAAC	**80.48 ± 5.97**	80.34 ± 6.00	**80.04 ± 6.27**	**80.12 ± 6.22**	85.96 ± 7.17	85.40 ± 7.52

5.2 Results on Full Fine-tuning

Unlike PFT, Full Fine-tuning (FFT) adapts all model parameters to the given target medical task, which means the Encoder and the classifier are trained together with supervision. FFT further enhances the representation in the downstream task. The results of FFT on the target dataset are exemplified using the AD, PD, PTB dataset in Table 2.

5.3 Visualization of Views

To verify the learnable multi-view learning capability of ACL, we employed the UMAP [42] to reduce the dimensionality of view representations to two dimensions and visualize them as shown in Fig. 3. The visualizations provide strong validation from the following two aspects:

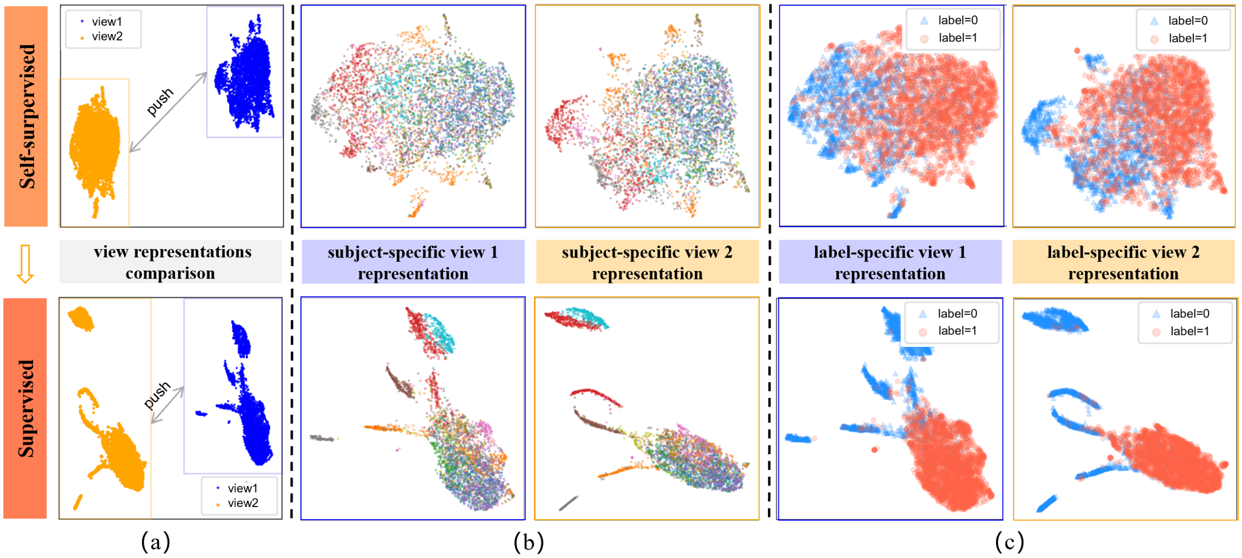

Figure 3: Visualization of Multi-View Representations on the AD Dataset. The figure compares contrastive view representations obtained through self-supervised contrastive learning and 100% supervised fine-tuning.

The Disparities between Diverse Views: By comparing the view representations as Fig. 3 (a), it can be seen that the designed inter-view contrastive loss enables the representations of view 1 and view 2 under different samples to show obvious disparities, effectively distinguishing the representations of each view.

The Intra-view Subject Discrepancies: The intra-view contrastive loss we propose fully considers the subject identity, setting the positive samples as those samples with the same view and subject as the anchor sample, and the negative samples as those with the same view but different subjects with the anchor sample. From the visualization results of patient-specific view representations in Fig. 3 (b), it can be found that there are differences in the representations of some different subjects under each view. In the self-supervised part, although label is not introduced, when observing the representations of different views with label coloring in Fig. 3 (c), the representation differences between diseased subjects and normal subjects in the AD dataset can still be found. This indicates that the view representations have good generalization ability, which is beneficial for downstream disease diagnosis tasks.

In addition, Fig. 3 also shows that after adding 100% labeled data for supervised fine-tuning, the clustering of view representations is further enhanced. Overall, through this visualization, it is sufficient to prove the effectiveness of the adaptive contrastive learning.

6 Conclusion

In this work, we tackle two fundamental challenges in medical time-series diagnosis: the scarcity of labeled data and the limitations of conventional contrastive learning that rely on handcrafted sample pairs. We propose DAAC, a novel framework that enhances representation learning through a combination of external knowledge and adaptive contrastive strategies. By introducing a Discrepancy Estimator trained on large-scale normal data, we effectively quantify and incorporate abnormality as an auxiliary feature, reducing overfitting caused by limited target data. Additionally, our Adaptive Contrastive Learner, equipped with multi-head attention and hierarchical contrastive losses, enables the model to automatically discover meaningful relationships across diverse views and patient conditions without manual design. Together, these components allow DAAC to extract robust, generalizable features from complex medical time-series data. Experimental results on three diagnostic tasks confirm that DAAC achieves state-of-the-art performance, even with minimal labeled data, highlighting its potential for real-world deployment in low-resource clinical settings.

While DAAC demonstrates its effectiveness, the current discrepancy estimation may still be suboptimal in fully leveraging external information. Although care is taken to prevent external data from overwhelming the target dataset, the representation of discrepancy remains somewhat coarse. Future work could explore more precise and adaptive strategies for modeling discrepancy, aiming to better balance external knowledge integration with the inherent distribution of the target data. Additionally, we plan to extend DAAC to other modalities that share similar hierarchical or temporal structures, broadening its applicability. We also acknowledge that jointly optimizing all five loss weights may further improve performance, and we have included this direction as a key part of our future work.

References

[1] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. Advances in neural information processing systems, 31, 2018.

[2] Jiaxin Gao, Qinglong Cao, and Yuntian Chen. Auto-regressive moving diffusion models for time series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 16727–16735, 2025.

[3] Joren Brunekreef, Eric Marcus, Ray Sheombarsing, Jan-Jakob Sonke, and Jonas Teuwen. Kandinsky conformal prediction: Efficient calibration of image segmentation algorithms. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4135–4143, 2024.

[4] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478, 2020.

[5] José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. Big data, 9(1):3–21, 2021.

[6] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. IEEE transactions on knowledge and data engineering, 35(1):857–876, 2021.

[7] Enjun Zhu, Haiyu Feng, Long Chen, Yongqiang Lai, and Senchun Chai. Mp-net: a multi-center privacy-preserving network for medical image segmentation. IEEE Transactions on Medical Imaging, 2024.

[8] Yihe Wang, Yu Han, Haishuai Wang, and Xiang Zhang. Contrast everything: A hierarchical contrastive framework for medical time-series. Advances in Neural Information Processing Systems, 36, 2024.

[9] Dani Kiyasseh, Tingting Zhu, and David A Clifton. Clocs: Contrastive learning of cardiac signals across space, time, and patients. In International Conference on Machine Learning, pages 5606–5615. PMLR, 2021.

[10] Ziyu Liu, Azadeh Alavi, Minyi Li, and Xiang Zhang. Self-supervised contrastive learning for medical time series: A systematic review. Sensors, 23(9):4221, 2023.

[11] Youngkyu Hong and Eunho Yang. Unbiased classification through bias-contrastive and bias-balanced learning. Advances in Neural Information Processing Systems, 34:26449–26461, 2021.

[12] Inwoo Hwang, Sangjun Lee, Yunhyeok Kwak, Seong Joon Oh, Damien Teney, Jin-Hwa Kim, and Byoung-Tak Zhang. Selecmix: Debiased learning by contradicting-pair sampling. Advances in Neural Information Processing Systems, 35:14345–14357, 2022.

[13] Xing Wu, Jie Pei, Cheng Chen, Yimin Zhu, Jianjia Wang, Quan Qian, Jian Zhang, Qun Sun, and Yike Guo. Federated active learning for multicenter collaborative disease diagnosis. IEEE transactions on medical imaging, 42(7):2068–2080, 2022.

[14] Emma M Harding-Esch, Emma C Cousins, S-LC Chow, Laura T Phillips, Catherine L Hall, Nicholas Cooper, Sebastian S Fuller, Achyuta V Nori, Rajul Patel, Sathish Thomas-William, et al. A 30-min nucleic acid amplification point-of-care test for genital chlamydia trachomatis infection in women: a prospective, multi-center study of diagnostic accuracy. EBioMedicine, 28:120–127, 2018.

[15] B Mca Savers, HA Beagley, and WR Henshall. The mechanism of auditory evoked eeg responses. Nature, 247(5441):481–483, 1974.

[16] Chin-Sheng Lin, Wei-Ting Liu, Dung-Jang Tsai, Yu-Sheng Lou, Chiao-Hsiang Chang, Chiao-Chin Lee, Wen-Hui Fang, Chih-Chia Wang, Yen-Yuan Chen, Wei-Shiang Lin, et al. Ai-enabled electrocardiography alert intervention and all-cause mortality: a pragmatic randomized clinical trial. Nature Medicine, 30(5):1461–1470, 2024.

[17] Sike Ni, Mohammed AA Al-qaness, Ammar Hawbani, Dalal Al-Alimi, Mohamed Abd Elaziz, and Ahmed A Ewees. A survey on hand gesture recognition based on surface electromyography: Fundamentals, methods, applications, challenges and future trends. Applied Soft Computing, page 112235, 2024.

[18] Geer Teng, Yue He, Hengjun Zhao, Dunhu Liu, Jin Xiao, and S Ramkumar. Design and development of human computer interface using electrooculogram with deep learning. Artificial intelligence in medicine, 102:101765, 2020.

[19] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 8980–8987, 2022.

[20] Guoqiang Xu, Weize Zhang, Yujing Zhang, Ziqing Xiong, Zhengyu Deng, and Lejian Sun. Anomaly transformer: Time series anomaly detection with association discrepancy. In International Conference on Learning Representations (ICLR), 2022.

[21] Yu Zheng, Haokun Lin, Zicheng Li, Zhaowei Wang, and Yiming Zhao. Dabaclt: A data augmentation bias-aware contrastive learning framework for time series representation. Applied Sciences, 13(13):7908, 2023.

[22] Chenghao Luo, Ziyue Wang, Hao Zhang, and Liang Zhao. Time series contrastive learning with information-aware augmentations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 9744–9752, 2023.

[23] Aneta Neumann, Wanru Gao, Carola Doerr, Frank Neumann, and Markus Wagner. Discrepancy-based evolutionary diversity optimization. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 991–998, 2018.

[24] Mathias Perslev, Sune Darkner, Lykke Kempfner, Miki Nikolic, Poul Jørgen Jennum, and Christian Igel. U-sleep: resilient high-frequency sleep staging. NPJ digital medicine, 4(1):72, 2021.

[25] Huy Phan, Kristian P Lorenzen, Elisabeth Heremans, Oliver Y Chén, Minh C Tran, Philipp Koch, Alfred Mertins, Mathias Baumert, Kaare B Mikkelsen, and Maarten De Vos. L-seqsleepnet: Whole-cycle long sequence modeling for automatic sleep staging. IEEE Journal of Biomedical and Health Informatics, 27(10):4748–4757, 2023.

[26] Zheng Zeng, Linkai Tao, Jun Hu, Ruizhi Su, Long Meng, Chen Chen, and Wei Chen. Multi-scale inception-based deep fusion network for electrooculogram-based eye movements classification. Biomedical Signal Processing and Control, 103:107377, 2025.

[27] Tomer Golany, Kira Radinsky, and Daniel Freedman. Simgans: Simulator-based generative adversarial networks for ecg synthesis to improve deep ecg classification. In International Conference on Machine Learning, pages 3597–3606. PMLR, 2020.

[28] Laurenz Berger, Max Haberbusch, and Francesco Moscato. Generative adversarial networks in electrocardiogram synthesis: Recent developments and challenges. Artificial Intelligence in Medicine, 143:102632, 2023.

[29] Ran An, Zuogang Shang, Yuhong Liu, Xin Wu, Jing Ren, and Zhibin Zhao. Imbalanced domain adaptation net for multi-class electrocardiography signals classification. Biomedical Signal Processing and Control, 108:107912, 2025.

[30] Yongle Chen, Sida Su, Dan Yu, Hao He, Xiaojian Wang, Yao Ma, and Hao Guo. Cross-domain industrial intrusion detection deep model trained with imbalanced data. IEEE Internet of Things Journal, 10(1):584–596, 2022.

[31] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. Advances in Neural Information Processing Systems, 35:3988–4003, 2022.

[32] Eun Som Jeon, Suhas Lohit, Rushil Anirudh, and Pavan Turaga. Robust time series recovery and classification using test-time noise simulator networks. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[33] Jiawei Chen. Addressing spatial-temporal heterogeneity: General mixed time series analysis via latent continuity recovery and alignment. Advances in Neural Information Processing Systems, 37:17910–17946, 2024.

[34] Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[35] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR, 2020.

[36] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li, and Cuntai Guan. Self-supervised contrastive representation learning for semi-supervised time-series classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[37] Jiexi Liu and Songcan Chen. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 13918–13926, 2024.

[38] Seunghan Lee, Taeyoung Park, and Kibok Lee. Soft contrastive learning for time series. arXiv preprint arXiv:2312.16424, 2023.

[39] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

[40] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. arXiv preprint arXiv:2106.14112, 2021.

[41] Tim Schneider, Chen Qiu, Marius Kloft, Decky Aspandi Latif, Steffen Staab, Stephan Mandt, and Maja Rudolph. Detecting anomalies within time series using local neural transformations. arXiv preprint arXiv:2202.03944, 2022.

[42] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.

[43] Andreas Miltiadous, Katerina D Tzimourta, Theodora Afrantou, Panagiotis Ioannidis, Nikolaos Grigoriadis, Dimitrios G Tsalikakis, Pantelis Angelidis, Markos G Tsipouras, Evripidis Glavas, Nikolaos Giannakeas, et al. A dataset of eeg recordings from: Alzheimer's disease, frontotemporal dementia and healthy subjects. OpenNeuro, 1:88, 2023.

[44] Md Fahim Anjum, Soura Dasgupta, Raghuraman Mudumbai, Arun Singh, James F Cavanagh, and Nandakumar S Narayanan. Linear predictive coding distinguishes spectral eeg features of parkinson's disease. Parkinsonism & related disorders, 79:79–85, 2020.

[45] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. Scientific data, 7(1):1–15, 2020.

[46] J Escudero, Daniel Abásolo, Roberto Hornero, Pedro Espino, and Miguel López. Analysis of electroencephalograms in alzheimer's disease patients with multiscale entropy. Physiological measurement, 27(11):1091, 2006.

[47] Hanneke Van Dijk, Guido Van Wingen, Damiaan Denys, Sebastian Olbrich, Rosalinde Van Ruth, and Martijn Arns. The two decades brainclinics research archive for insights in neurophysiology (tdbrain) database. Scientific data, 9(1):333, 2022.

[48] Ralf-Dieter Bousseljot, Dieter Kreiseler, and Andreas Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. Biomedizinische Technik/Biomedical Engineering, 40(Supplement 1):317–318, 1995.

A Preliminary Knowledge on Medical Time Series

A.1 Hierarchical Structuring of Medical Time Series

To better understand the structure of medical time series data, this section provides a comprehensive explanation of its hierarchical organization, using EEG signals from Alzheimer's disease as an example. The dataset is categorized into four hierarchical levels: Subject, Trial, Epoch, and Temporal, with each level representing a unique granularity of the data (Appendix figure B1). Unlike conventional time series, which typically consist of sample and observation levels, medical time series incorporate two additional layers: patient and trial. These extra layers enhance the dataset's structure, enabling the development of methods tailored to the specific characteristics and analytical requirements of medical time series. As illustrated in Appendix Table B1, many existing approaches utilize only a subset of these levels, underscoring the importance of leveraging the complete hierarchy for more effective analysis.

Definition 1: Subject

A **subject** $\mathbf{p}_j \in \mathbb{R}^G$ in medical time series data represents the collection of multiple trials corresponding to a single individual. The subscript j denotes the subject ID. Trials associated with the same subject may differ due to variations in data collection timeframes, sensor placement, or patient conditions. A subject's data is generally represented as an aggregate of trials. Each trial within a subject $\mathbf{p}_j$ is typically segmented into smaller samples to facilitate representation learning. To denote all the samples belonging to a specific subject $\mathbf{p}_j$, we use the notation $\mathbf{P}_j$.

Definition 2: Trial

A **trial** $\mathbf{r}_k$ refers to a continuous set of observations collected over an extended time period (e.g., 30 minutes). This can also be referred to as a **record**. The subscript k denotes the trial ID. Trials are generally too lengthy (e.g., consisting of hundreds of thousands of observations) to be directly processed by deep learning models. Therefore, they are usually divided into smaller subsequences, referred to as samples or segments. The collection of all samples derived from a trial $\mathbf{r}_k$ is denoted by $\mathbf{R}_k$.

Definition 3: Epoch

An **epoch** e_m is a sequence of consecutive observations derived from a trial, typically spanning a shorter time period (e.g., several seconds). In EEG (electroencephalography), an epoch refers to a specific time window extracted from the continuous EEG signal. These time windows are used to isolate signal segments for further analysis, often aligned with specific events or experimental conditions. Each epoch is composed of multiple observations measured at regular intervals over a time range defined by M timestamps. To denote a specific epoch within a trial, we use e_m, where m represents the epoch index. The set of observations forming an epoch is

$$e_m = \{e_{m,t} \mid t = 1, \cdots, M\}.$$

Definition 4: Temporal

A **temporal observation** $\mathbf{x}_{n,t} \in \mathbb{R}^H$ refers to a single data point or vector recorded at a specific timestamp t. The subscript n denotes the sample index, while t represents the timestamp. For univariate time series, the temporal observation is a single real value, whereas, for multivariate time series, it is a vector with H dimensions. Temporal observations may represent physiological signals, laboratory measurements, or other health-related indicators.

A.2 Data Scarcity and Heterogeneity for Medical Time Series

Data Scarcity Medical data, especially high-quality neurodegenerative disease data, is often expensive, with high costs associated with data collection and annotation. Additionally, individual datasets typically have limited sample sizes and suffer from class imbalance issues (e.g., an unequal ratio of healthy individuals to patients). These problems lead to overfitting during model training, poor generalization, and suboptimal classification performance.

Data Heterogeneity Medical data often exhibits a multi-center characteristic, meaning datasets from different institutions show significant distributional differences. This heterogeneity makes it difficult to directly transfer models trained on one dataset to others, severely hindering the development and application of robust deep learning models.

B Supplementary Figures and Tables

C Datasets and Data Preprocessing

In the medical field, datasets from different hospitals often exhibit a certain degree of distributional shift. Models trained on data from one hospital may demonstrate unstable performance when applied to others. To enhance model generalization, we propose leveraging DE to incorporate data from other hospitals. Although distributional differences exist, we argue that the sequential patterns of normal data (i.e., data from healthy patients) can still provide valuable guidance for diagnosis in the target hospital.

In this study:

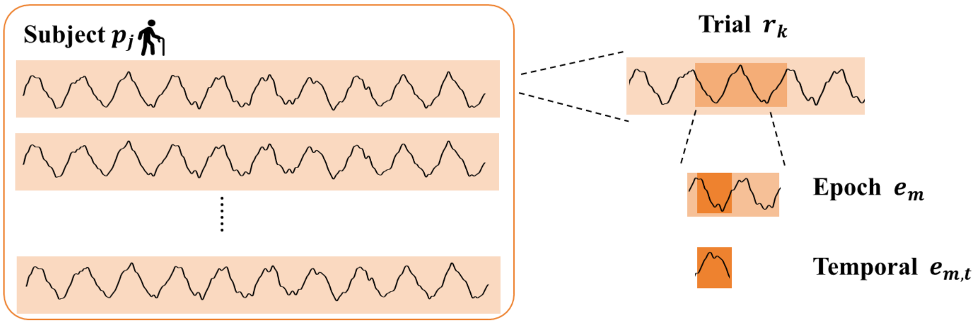

Figure B1: **Structure of Medical Time Series.**The data is organized into four levels: Subject (p_j), Trial (r_k), Epoch (e_m), and Temporal ($e_{m,t}$), capturing different granularities of the data, from subject-level recordings to temporal observations within individual epochs.

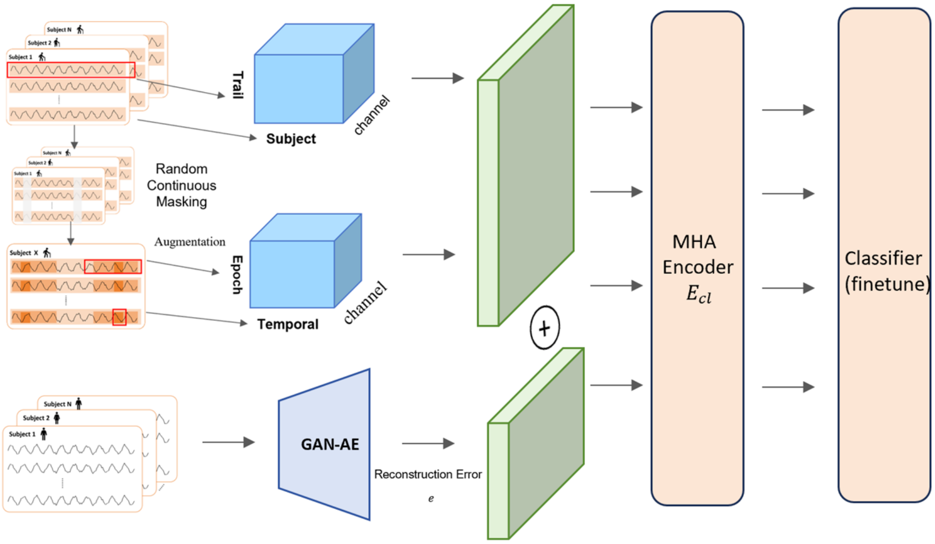

Figure B2: Overview of DAAC

The dataset from the current hospital is referred to as the **internal dataset (or target dataset)**, which participates in the second and third stages of model training. Datasets from other hospitals, used for guidance, are termed **external datasets which are from healthy patients** and contribute to the first stage of training.

C1 External Datasets

C1.1 Alzheimer's Disease Dataset

The AD dataset [43] is a publicly available EEG time-series dataset comprising three classes: 36 Alzheimer's disease (AD) patients, 23 frontotemporal dementia (FTD) patients, and 29 healthy control (HC) subjects. Recordings were obtained using 19 channels at a raw sampling rate of 500 Hz, with each subject contributing a single trial. The average trial durations were approximately 13.5 minutes for AD subjects (range: 5.1–21.3 minutes), 12 minutes for FTD subjects (range: 7.9–16.9 minutes), and 13.8 minutes for HC subjects (range: 12.5–16.5 minutes). A bandpass filter between 0.5–45 Hz was applied to all recordings. Subsequently, the data were downsampled to 256 Hz and segmented into non-overlapping 1-second windows (256 timestamps per segment), discarding any segments shorter than 1 second. Sixteen overlapping channels were selected from the original 19 channels for analysis. After preprocessing, the dataset contained a total of 69,752 samples. We considered both subject-dependent and subject-independent evaluation setups: in the subject-dependent setup, 60%, 20%, and 20% of the samples were allocated to the training, validation, and test sets, respectively; in the subject-independent setup, 60%, 20%, and 20% of the subjects (and their corresponding samples) were used for training, validation, and testing, respectively.

C1.2 Parkinson's Disease Dataset

The PD dataset used in this study consists of EEG recordings from 46 participants collected at the University of New Mexico (UNM, Albuquerque, New Mexico), as previously described in [44]. The cohort includes 22 individuals diagnosed with Parkinson's disease (PD) in an OFF-medication state and 24 healthy control (HC) subjects. EEG recordings were obtained using a 64-channel Brain Vision system at a sampling rate of 500 Hz. For consistency across datasets, 33 channels with matched naming conventions were selected for analysis. The EEG signals were high-pass filtered at 1 Hz to remove low-frequency noise, and no further preprocessing was applied. Only the first one minute of eyes-closed resting-state EEG data was used for each participant. Each 1-minute recording was segmented into non-overlapping 5-second windows, resulting in data represented as $\mathbb{R}^{n\times 33\times 1250}$, where n denotes the number of 5-second segments and 1250 is the number of timestamps per segment.

C1.3 Myocardial Infarction Dataset

The PTB-XL dataset [45] is a large-scale public ECG time-series dataset comprising recordings from 18,869 subjects across 12 channels, with five labels corresponding to four heart disease categories and one healthy control (HC) category. Each subject may contribute one or more trials; however, to ensure consistency, subjects with varying diagnostic results across trials were excluded, resulting in a final dataset of 17,596 subjects. Recordings were obtained as 10-second intervals and are available in two versions with sampling frequencies of 100 Hz and 500 Hz. In this study, we use the 500 Hz version, which is downsampled to 250 Hz and normalized using standard scalers. Each trial was then segmented into non-overlapping 1-second windows (250 timestamps per segment), discarding any segments shorter than 1 second. After preprocessing, the dataset contained 191,400 samples. A subject-independent setup was adopted for dataset splitting, allocating 60%, 20%, and 20% of the subjects (and their corresponding samples) to the training, validation, and test sets, respectively.

C2 Target Datasets

C2.1 Alzheimer's Disease Dataset

The AD dataset [46] consists of EEG recordings from 22 subjects, including 11 patients diagnosed with probable Alzheimer's disease (5 men, 6 women; mean age 72.5 ± 8.3 years) and 11 healthy controls. Patients were recruited from the Alzheimer's Patients' Relatives Association of Valladolid (AFAVA) and underwent comprehensive clinical evaluations, including neurological and physical examinations, brain imaging, and Mini-Mental State Examination (MMSE) assessments. EEG recordings were conducted at the University Hospital of Valladolid (Spain) using Profile Study Room 2.3.411 EEG equipment (Oxford Instruments) with 19 electrode positions according to the international 10–20 system. Each subject contributed over 5 minutes of EEG data recorded in a relaxed, eyes-closed, awake state to minimize artifacts. Signals were sampled at 256 Hz with 12-bit analog-to-digital precision.Each patient contributed an average of 30.0 ± 12.5 trials, with each trial spanning a 5-second interval across 16 channels. Prior to analysis, trials were standardized using a standard scaler and segmented into nine half-overlapping 1-second samples, each containing 256 timestamps.

C2.2 Parkinson's Disease Dataset

The TDBrain dataset [47] contains EEG recordings from 1,274 patients acquired across 33 channels at 500 Hz during eyes-closed (EC) and eyes-open (EO) tasks, covering 60 distinct neurological conditions, with some patients diagnosed with multiple disorders. In this study, we focus on a subset comprising 25 Parkinson's disease (PD) patients and 25 healthy controls, utilizing only the EC task for representation learning. Raw EC trials were preprocessed using a sliding window approach, where each trial was resampled to 256 Hz and divided into pseudo-trials of 2,560 timestamps (10 seconds). Each pseudo-trial was standardized and subsequently segmented into 19 half-overlapping 1-second windows, each containing 256 timestamps. Each sample was assigned a binary label (PD or healthy), along with patient and trial identifiers, where trial identifiers correspond to the generated pseudo-trials.A patient-independent split was employed, with samples from eight patients (four PD and four healthy) allocated to the validation set, another eight patients to the test set, and the remaining subjects to the training set.

C2.3 Myocardial Infarction Dataset

The PTB dataset [48] contains ECG recordings from 290 patients across 15 channels, sampled at 1,000 Hz, and covers eight types of heart diseases. In this study, we focus on a subset comprising 198 patients for a binary classification task, specifically distinguishing between myocardial infarction (MI) and healthy controls. The ECG signals were resampled to 250 Hz, normalized using a standard scaler, and segmented into individual heartbeats to preserve critical peak information, as a standard sliding window approach may lead to information loss.The segmentation process involved several steps: (1) determining the maximum heartbeat duration by calculating the median R-peak interval across all channels in each trial, excluding outliers; (2) identifying the position of the first R-peak using the median value across channels; (3) splitting raw trials into individual heartbeats based on the median R-peak interval, with segments extended symmetrically around the R-peak; and (4) applying zero-padding to ensure uniform sample lengths.A patient-independent split was adopted, with samples from 28 patients (7 healthy and 21 MI) allocated to the validation and test sets, and the remaining subjects used for training.

C3 Domain Shift Study

Although we hope that the external normal data assumes a certain level of domain alignment. In practice, however, our external and target datasets exhibit notable differences, including variations in population demographics and acquisition protocols.

For example, in the AD task, the external dataset ([43]) has an average subject age of 63.6 years, while the target dataset ([46]) has an average age of 72.5 years. When training the Discrepancy Estimator (DE) using this external dataset, we indeed observe signs of negative transfer—specifically, in Table 2, the AUROC of COMET+DE drops from 94.44 to 93.97.

Despite this, DAAC still outperforms both COMET+ACL and COMET+DE in the same setting, as shown in Table 2. This suggests that the Adaptive Contrastive Learner (ACL) effectively mitigates the potential negative impact introduced by domain-shifted external data, and in some cases, even corrects for suboptimal guidance from the DE. This robustness highlights the strength of DAAC's multi-stage design in handling real-world domain variability.

D Loss Setups

D.1 Loss Funtion of DE

The model is trained by optimizing two loss functions: the discriminator loss and the generator loss. During the training process, the discriminator is first updated using the discriminator loss (L_D), followed by the generator. The two components are updated alternately.

(1) Discriminator Loss:

$$L_D = \frac{1}{2}(L_{D_{\text{real}}} + L_{D_{\text{fake}}})$$

where $L_{D_{\text{real}}} = \text{BCE}(C(xx_i), 1)$ indicates the loss for real data, encouraging the discriminator to classify real data as 1. $L_{D_{\text{fake}}} = \text{BCE}(C(\hat{xx_i}), 0)$ represents the loss for generated data, encouraging the discriminator to classify generated data as 0. BCE is the binary cross-entropy loss function.

(2) Generator Loss:

$$L_G = L_{G_{\text{adv}}} + L_{G_{\text{recon}}}$$

where $L_{G_{\text{adv}}} = \text{BCE}(C(\hat{xx_i}), 1)$ indicates The adversarial loss, encouraging the discriminator to classify generated data as 1. $L_{G_{\text{recon}}} = \text{MSE}(\hat{xx_i}, xx_i)$ represents the reconstruction loss, ensuring that the generated data $\hat{xx_i}$ is close to the original data xx_i. MSE is the mean squared error loss function.

D.2 Loss Weight Sensitivity

Appendix G.2 Loss Weight Sensitivity Analysis

To evaluate the robustness and effectiveness of our loss function design, we performed a comprehensive sensitivity analysis on the weighting strategy of its five components. The total loss is a weighted sum of four hierarchical contrastive losses and one view contrastive loss (L_V). In our main experiments, we used a fixed configuration of 1:1:1:1:2, which we found to yield consistent robust performance across datasets.

The first four hierarchical components were assigned equal weights (1:1:1:1), following prior practices ([8]), which we hypothesized to provide a strong default setting. To test this assumption, we individually perturbed each of these four weights while keeping the others fixed, and evaluated the model on three datasets (AD, PTB, TDBrain) under both 100% and 10% data regimes. Additionally, to investigate the contribution of the proposed ACL, we varied the weight of L_V from 0 to 3 and monitored the impact on overall performance.

The full results are provided in Table **??**. We summarize the main findings as follows:

(1) **Effectiveness of L_V:** Removing L_V (i.e., setting its weight to 0) consistently led to substantial performance degradation across all evaluation metrics and datasets, highlighting its critical role in enhancing cross-view representation alignment.

(2) **Performance saturation with increasing L_V:** Increasing the weight of L_V from 1 to 2 led to noticeable gains in accuracy, F1 score, AUROC, and AUPRC. However, further increasing the weight to 3 resulted in only marginal or dataset-specific improvements, suggesting a saturation effect.

(3) **Default weight robustness:** Perturbing the weights of the first four components (e.g., 1:1:2:1:1, 1:2:1:1:1) produced only minor variations in performance, indicating that the default configuration (1:1:1:1) is a generally robust and effective setting.

These findings provide empirical justification for our selected weight configuration and demonstrate that our method performs reliably without the need for extensive tuning. Further improvements may be achievable through joint or dynamic weight optimization, which we leave as future work.

E Ablation Study

E.1 Mutual Information Analysis of Discrepancy Estimator Reconstruction Error

We evaluate the statistical relevance of the Discrepancy Estimator (DE) derived reconstruction error by computing mutual information (MI) scores between each feature and the disease label on the AD dataset. The MI score quantifies the amount of shared information between a feature and the classification target, with higher values indicating greater discriminative power. As summarized in Table E1, the reconstruction error achieves the highest MI score (0.0295), substantially outperforming all original features (maximum MI among them: 0.0159). These results demonstrate that the reconstruction error serves as a highly informative feature, supporting its design as a proxy for disease probability within the DAAC framework.

E.2 Ablation study of contrastive blocks

We conduct ablation experiments to evaluate the contribution of each contrastive loss branch in ACL. As shown in Table E2, the full model (P+R+S+O+V) achieves the best performance, and each added block improves results incrementally. Notably, adding the view loss (L_V) leads to a 3.1% F1 improvement on average. We adopt a fixed 1:1:1:1:2 weighting scheme for the five loss components, where L_V includes both inter- and intra-view terms. We did not tune these weights, as our goal was to verify the method's generality rather than overfit to a specific dataset. Despite no tuning, the model achieves SOTA results (see Table 1 and Table 2).

E.3 Ablation Study on Ratio of External Datasets

To investigate the impact of the normal population size used during pre-training, we conduct systematic ablation experiments on three datasets: AD, PTB, and TDBrain. Specifically, we vary the amount of normal samples used in the pre-training stage across four proportions: 5%, 25%, 50%, and 100%, as shown in Table E3. For each setting, we evaluate the model under four levels of labeled training data (5%, 25%, 50%, 100%) in the downstream task.

We summarize three key findings from the ablation study on ratio of external datasets:

(1) Across all datasets and downstream data scales, increasing the normal population size consistently improves model performance;

(2) The performance gain is existed even under low-resource (e.g., 5%) labeled settings, highlighting the effectiveness of leveraging normal population pre-training to reduce annotation dependency;

(3) We observe a stable and monotonic improvement trend without performance degradation, indicating the robustness and scalability of our method.

These findings support the practicality of our approach in real-world scenarios where the collection of a full-scale healthy cohort might be challenging. Even a subset of normal population data can yield substantial performance improvements, making our pre-training strategy feasible and effective for deployment.

F Discrepancy Estimator Study

F.1 DE Further Study

In our three-stage DAAC framework, the Discrepancy Estimator (DE) serves as the initial-stage module for modeling the abnormality of unlabeled target data, using external normal data as a reference. A key design choice is how to represent the discrepancy feature that serves as input to downstream stages.

We adopt a sequence-level scalar reconstruction error, computed as the mean squared error (MSE) between an input sequence and its reconstruction via an autoencoder trained only on external normal data. This design is intended to provide a robust, task-agnostic anomaly signal, especially under distribution shift.

To evaluate this design decision, we conduct an ablation study comparing our default sequence-level scalar with several finer-grained alternatives:

As shown in the table, while channel-wise MSE vector and point-wise error sequence alternatives preserve more local detail, and the cluster-wise distance reflects latent deviation from normal patterns, their performance is consistently inferior to our scalar design. This degradation can be attributed to: **(1) Sensitivity to noise** in external data; **(2) Overfitting risks** due to fine-grained alignment across domains with potential distribution shifts (e.g., different acquisition hardware or subject demographics).

In contrast, the scalar discrepancy offers a stable and compact abnormality signal, less vulnerable to non-transferable patterns. Importantly, we emphasize that this discrepancy is only used during the first stage of DAAC, and is subsequently complemented and refined by the multi-level Adaptive Contrastive Learner in Stage 2, which captures high-resolution temporal and spatial semantics.

This study demonstrates that our scalar discrepancy signal provides an effective balance between robustness and informativeness, supporting more generalizable downstream representation learning in diverse medical time-series applications.

F.2 DE Assumption Validation

We executed a KDE analysis to empirically validate the assumption. As shown in Figure F1 in the Appendix, normal samples show tightly clustered reconstruction errors around low values, while abnormal samples follow a positively skewed distribution, with a longer tail toward higher errors. This confirms that our model tends to assign higher reconstruction errors to abnormal samples. Additionally, using reconstruction error as an anomaly score yields an AUROC of 0.974 on the AD dataset, further supporting its discriminative effectiveness.

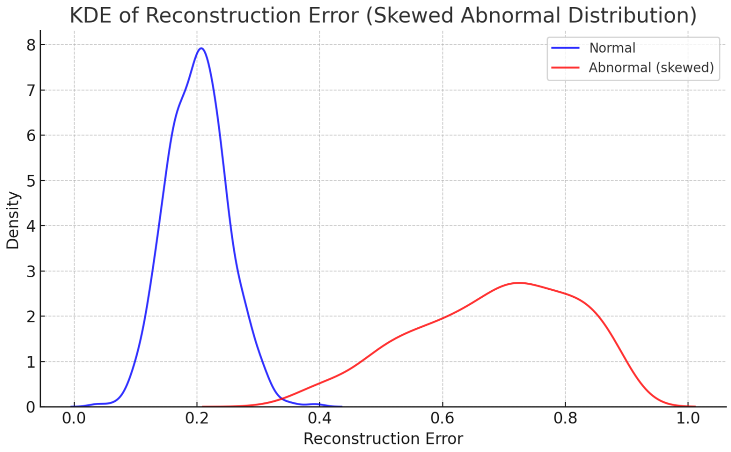

Figure F1: DE Assumption Validation

G Training Process Details

G.1 Training and Resource Requirements

We conducted all training experiments using an NVIDIA RTX 4090 GPU. The training and deployment efficiency of our model on different datasets is summarized as follows:

(1) **AD dataset**: Stage 1 training took 1.2 hours, Stage 2 took 3.2 hours, and Stage 3 took 0.2 hours. The deployed model for inference requires only 4 GB of GPU memory.

(2) **PDB dataset**: Stage 1 training took 2 hours, Stage 2 took 6.5 hours, and Stage 3 took 1.8 hours. The deployed model for inference requires only 6.2 GB of GPU memory.

(3) **TDBrain dataset**: Stage 1 training took 1.8 hours, Stage 2 took 5.5 hours, and Stage 3 took 0.5 hours. The deployed model for inference requires only 5.3 GB of GPU memory.

These results indicate that our training pipeline is efficient, and the final model is lightweight enough for practical deployment.

G.2 Model Architecture Details

To improve reproducibility, we provide detailed descriptions of the model architecture used in our experiments. The encoder is a dilated convolutional network optionally equipped with multi-head attention modules. The projection head is used for final classification. Table G1 summarizes the full configuration.

Table 2: Full fine-tuning results of AD, TDBrain, PTB datasets. We use 100% and 10% of labeled data for fine-tuning. Model names are consistent with Table 1. Bold indicates the best performance, and underline denotes the second best.

Datasets	Fraction	Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
AD	**100%**	**TS2vec**	81.26 ± 2.08	81.21 ± 2.14	81.34 ± 2.04	81.12 ± 2.06	89.20 ± 1.76	88.94 ± 1.85
		TF-C	75.31 ± 8.27	75.87 ± 8.73	74.83 ± 8.98	74.54 ± 8.85	79.45 ± 10.23	79.33 ± 10.57
		Mixing-up	65.68 ± 7.89	72.61 ± 4.21	68.25 ± 6.97	63.98 ± 9.92	84.63 ± 5.04	83.46 ± 5.48
		TS-TCC	73.55 ± 10.00	77.22 ± 6.13	73.83 ± 9.65	71.86 ± 11.59	86.17 ± 5.11	85.73 ± 5.11
		SimCLR	54.77 ± 1.97	50.15 ± 7.02	50.58 ± 1.92	43.18 ± 4.27	50.15 ± 7.02	50.42 ± 1.06
		CLOCS	78.37 ± 6.00	83.99 ± 2.11	76.14 ± 7.03	75.78 ± 7.93	91.17 ± 2.51	90.72 ± 3.05
		COMET	84.50 ± 4.46	88.31 ± 2.42	82.95 ± 5.39	83.33 ± 5.15	94.44 ± 2.37	94.43 ± 2.48
		COMET+DE	84.82 ± 10.43	88.49 ± 5.71	83.42 ± 11.99	83.08 ± 12.99	93.97 ± 4.97	93.71 ± 5.25
		COMET+ACL	91.16 ± 4.14	92.10 ± 3.66	90.50 ± 4.51	90.88 ± 4.33	96.22 ± 2.76	96.03 ± 2.82
		DAAC	**93.23 ± 5.25**	**94.01 ± 3.98**	**92.71 ± 5.86**	**92.97 ± 5.65**	**98.03 ± 1.71**	**98.03 ± 1.75**
	10%	**TS2vec**	73.28 ± 4.34	74.14 ± 4.33	73.52 ± 3.77	73.00 ± 4.18	81.66 ± 5.20	81.58 ± 5.11
		TF-C	75.66 ± 11.21	75.48 ± 11.48	75.58 ± 11.59	75.38 ± 11.47	81.38 ± 14.19	81.56 ± 13.68
		Mixing-up	59.38 ± 3.33	64.85 ± 4.38	61.94 ± 3.42	58.17 ± 3.41	75.02 ± 6.14	73.44 ± 5.82
		TS-TCC	77.83 ± 6.90	79.73 ± 7.49	76.18 ± 7.21	76.43 ± 7.56	84.12 ± 7.32	84.12 ± 7.61
		SimCLR	56.09 ± 2.25	53.81 ± 5.74	51.73 ± 2.39	44.10 ± 4.84	53.81 ± 5.74	51.08 ± 1.33
		CLOCS	76.97 ± 3.01	81.70 ± 3.21	74.69 ± 3.26	74.75 ± 3.61	86.91 ± 3.61	86.70 ± 3.64
		COMET	91.43 ± 3.12	92.52 ± 2.36	90.71 ± 3.56	91.14 ± 3.31	96.44 ± 2.84	96.48 ± 2.82
		COMET+DE	93.47 ± 3.52	93.54 ± 3.41	93.68 ± 3.16	93.43 ± 3.50	**98.27 ± 1.34**	**98.30 ± 1.34**
		COMET+ACL	92.77 ± 2.35	92.92 ± 2.43	92.55 ± 2.39	92.66 ± 2.37	97.30 ± 1.40	97.39 ± 1.40
		DAAC	**94.67 ± 1.84**	**94.93 ± 1.76**	**94.41 ± 1.99**	**94.58 ± 1.89**	98.10 ± 1.20	98.19 ± 1.15
TDBrain	**100%**	**TS2vec**	80.21 ± 1.69	81.38 ± 1.97	80.21 ± 1.69	80.07 ± 1.69	89.57 ± 2.31	89.60 ± 2.37
		TF-C	66.62 ± 1.76	67.15 ± 1.64	66.62 ± 1.76	66.35 ± 1.91	65.43 ± 6.13	66.18 ± 4.90
		Mixing-up	81.47 ± 1.07	82.11 ± 1.12	81.47 ± 1.07	81.38 ± 1.08	90.48 ± 0.89	90.51 ± 0.04
		TS-TCC	77.42 ± 2.86	77.68 ± 2.93	77.42 ± 2.86	77.37 ± 2.86	87.37 ± 3.06	87.61 ± 3.22
		SimCLR	58.43 ± 1.77	59.48 ± 1.95	58.43 ± 1.77	57.30 ± 2.07	59.48 ± 1.95	55.05 ± 1.18
		CLOCS	78.16 ± 1.13	79.49 ± 1.61	78.16 ± 1.13	77.91 ± 1.12	88.49 ± 1.95	88.83 ± 1.94
		COMET	85.47 ± 1.16	85.68 ± 1.20	85.47 ± 1.16	85.45 ± 1.16	93.73 ± 1.02	93.96 ± 0.99
		COMET+DE	90.61 ± 1.7	90.56 ± 1.49	90.16 ± 1.94	91.55 ± 2.38	95.69 ± 2.07	96.07 ± 1.85
		COMET+ACL	93.26 ± 2.33	92.84 ± 1.76	93.44 ± 1.62	93.07 ± 1.92	96.74 ± 1.98	97.60 ± 1.82
		DAAC	**95.60 ± 1.20**	**94.95 ± 1.69**	**95.73 ± 1.74**	**94.61 ± 1.26**	**97.63 ± 1.57**	**98.51 ± 1.29**
	10%	**TS2vec**	72.39 ± 1.13	74.49 ± 1.73	72.39 ± 1.13	71.80 ± 1.05	80.71 ± 1.90	80.06 ± 1.87
		TF-C	59.14 ± 3.04	59.34 ± 3.19	59.14 ± 3.04	58.98 ± 2.94	59.56 ± 4.10	59.65 ± 2.99
		Mixing-up	77.50 ± 2.07	80.09 ± 1.92	77.50 ± 2.07	76.99 ± 2.28	87.29 ± 1.34	87.13 ± 1.37
		TS-TCC	71.23 ± 1.57	78.78 ± 0.66	71.23 ± 1.57	69.18 ± 2.25	80.56 ± 1.98	80.21 ± 2.21
		SimCLR	59.79 ± 2.09	60.50 ± 1.90	59.79 ± 2.09	59.06 ± 2.82	60.50 ± 1.90	55.96 ± 1.41
		CLOCS	75.04 ± 2.65	75.97 ± 2.86	75.04 ± 2.65	74.83 ± 2.66	84.25 ± 3.27	84.37 ± 3.52
		COMET	79.28 ± 4.64	79.83 ± 4.83	79.28 ± 4.64	79.19 ± 4.62	88.39 ± 4.13	88.38 ± 3.96
		COMET+DE	81.27 ± 4.32	81.92 ± 4.29	82.30 ± 4.33	82.07 ± 4.52	90.22 ± 4.02	90.50 ± 4.09
		COMET+ACL	82.20 ± 3.08	81.53 ± 3.91	81.90 ± 3.51	82.09 ± 3.60	93.73 ± 3.27	**94.81 ± 3.49**
		DAAC	**83.66 ± 3.29**	**84.06 ± 3.06**	**83.29 ± 2.96**	**84.60 ± 3.29**	**95.27 ± 4.01**	94.71 ± 4.09
PTB	**100%**	**TS2vec**	85.14 ± 1.66	87.82 ± 2.21	76.84 ± 3.99	79.66 ± 3.63	90.50 ± 1.59	90.07 ± 1.73
		TF-C	87.50 ± 2.43	85.50 ± 3.04	82.68 ± 4.04	83.77 ± 3.50	77.59 ± 19.22	80.62 ± 15.10
		Mixing-up	87.61 ± 1.48	89.56 ± 2.80	79.30 ± 1.67	82.56 ± 2.00	89.29 ± 1.26	88.94 ± 1.04
		TS-TCC	82.24 ± 3.55	85.28 ± 5.08	69.46 ± 5.85	72.08 ± 6.85	87.60 ± 2.51	86.26 ± 3.00
		SimCLR	84.19 ± 1.32	85.85 ± 1.89	73.89 ± 2.95	76.84 ± 2.80	85.85 ± 1.89	70.70 ± 2.36
		CLOCS	86.39 ± 2.76	87.06 ± 2.81	77.95 ± 4.79	80.71 ± 4.78	90.41 ± 2.07	87.35 ± 3.36
		COMET	87.84 ± 1.98	87.67 ± 1.72	81.14 ± 3.68	83.45 ± 3.22	92.95 ± 1.56	87.47 ± 2.82
		COMET+DE	88.49 ± 2.06	88.91 ± 2.37	83.49 ± 4.59	85.67 ± 4.07	93.88 ± 1.94	88.61 ± 3.20
		COMET+ACL	91.33 ± 2.03	90.89 ± 2.83	84.20 ± 4.09	85.73 ± 3.82	94.16 ± 2.27	90.02 ± 3.05
		DAAC	**93.65 ± 2.35**	**93.28 ± 1.86**	**85.39 ± 3.84**	**87.64 ± 3.27**	**95.56 ± 1.57**	**90.28 ± 3.49**
	10%	**TS2vec**	82.49 ± 4.71	80.39 ± 5.04	83.35 ± 0.91	80.18 ± 4.04	93.03 ± 1.03	92.81 ± 0.97
		TF-C	85.37 ± 1.23	82.80 ± 2.35	79.94 ± 0.71	81.09 ± 1.14	81.57 ± 15.60	84.57 ± 8.12
		Mixing-up	87.05 ± 1.41	86.56 ± 3.24	80.61 ± 2.68	82.62 ± 1.99	89.28 ± 1.43	87.22 ± 2.76
		TS-TCC	83.38 ± 1.53	85.11 ± 2.11	72.24 ± 2.45	75.27 ± 2.64	86.06 ± 1.76	84.34 ± 2.08
		SimCLR	83.84 ± 2.15	87.19 ± 1.34	72.51 ± 4.63	75.44 ± 4.77	87.19 ± 1.34	69.99 ± 3.84
		CLOCS	88.25 ± 2.48	88.64 ± 2.12	81.40 ± 4.64	83.84 ± 4.03	91.91 ± 2.40	89.76 ± 3.94
		COMET	88.49 ± 3.28	88.98 ± 2.60	81.65 ± 6.00	84.01 ± 5.61	94.83 ± 1.08	92.48 ± 2.22
		COMET+DE	90.84 ± 3.01	89.67 ± 2.88	83.05 ± 2.89	84.68 ± 2.95	95.20 ± 2.62	94.32 ± 3.89
		COMET+ACL	89.67 ± 2.93	89.66 ± 2.47	83.36 ± 3.62	85.35 ± 3.81	95.35 ± 3.67	94.94 ± 3.71
		DAAC	**91.35 ± 2.77**	**91.56 ± 3.12**	**85.23 ± 2.65**	**85.61 ± 3.56**	**96.30 ± 3.22**	**97.87 ± 4.02**

Table B1: **Different Models Utilize Different Levels**

Models	**Subject**	**Trial**	**Epoch**	**Temporal**	**View**
SimCLR			✓		
TF-C			✓		
Mixing-up			✓		
TNC		✓			
TS2vec			✓	✓	
TS-TCC			✓	✓	
CLOCS	✓		✓		
COMET	✓	✓	✓	✓	
DAAC	✓	✓	✓	✓	✓

Table D1: Comparison of performance metrics under different hyperparameter configurations on AD, PTB, and TDBrain datasets.

Dataset	Fraction	Config	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
AD	100	1,1,1,1,0	84.50±4.46	88.31±2.42	82.95±5.39	83.33±5.15	93.60±2.37	94.10±2.48
		1,1,1,1,1	88.50±4.20	90.20±3.60	87.30±4.90	88.10±4.60	95.50±2.10	95.60±2.00
		1,1,1,1,2	**91.16±4.14**	**92.10±3.66**	90.50±4.51	90.88±4.33	**96.40±2.76**	**96.30±2.82**
		1,1,1,1,3	91.02±3.15	92.06±3.54	**90.58±5.11**	**90.98±3.98**	96.24±2.88	95.63±2.43
		1,1,1,2,1	87.20±4.30	89.30±4.00	86.80±4.60	87.30±4.30	94.65±2.90	94.75±2.80
		1,1,2,1,1	87.40±4.30	89.50±4.10	86.90±4.50	87.40±4.20	94.80±2.85	94.90±2.95
		1,2,1,1,1	86.90±4.60	89.20±4.00	86.50±4.90	87.10±4.50	94.40±2.90	94.50±2.80
		2,1,1,1,1	86.60±4.70	88.70±4.20	86.20±5.10	86.80±4.60	93.90±3.10	94.00±3.00
	10	1,1,1,1,0	91.43±3.12	92.52±2.36	90.71±3.56	91.14±3.31	92.20±2.84	92.30±2.82
		1,1,1,1,1	92.30±2.20	92.80±2.10	92.10±2.50	92.20±2.45	96.70±1.90	96.60±1.95
		1,1,1,1,2	92.77±2.35	**92.92±2.43**	92.55±2.39	**92.66±2.37**	**97.40±1.40**	97.50±1.45
		1,1,1,1,3	**92.86±2.44**	92.33±3.61	**92.63±2.01**	91.54±3.03	96.88±1.89	**97.64±1.33**
		1,1,1,2,1	92.00±2.80	92.50±2.50	91.90±2.70	92.10±2.60	95.20±2.00	95.10±2.05
		1,1,2,1,1	91.90±2.90	92.40±2.55	91.80±2.65	92.00±2.60	94.50±2.10	94.40±2.05
		1,2,1,1,1	91.70±3.00	92.10±2.80	91.60±2.75	91.90±2.70	94.00±2.20	94.10±2.15
		2,1,1,1,1	91.60±3.05	92.00±2.90	91.50±2.80	91.80±2.75	93.40±2.30	93.50±2.25
PTB	100	1,1,1,1,0	86.36±1.44	87.18±1.28	77.87±2.55	80.79±2.44	92.60±2.35	89.09±1.64
		1,1,1,1,1	89.70±1.80	89.60±2.10	82.80±3.00	85.80±2.90	94.00±2.10	90.10±2.40
		1,1,1,1,2	91.33±2.03	**90.89±2.83**	**84.20±4.09**	**87.64±3.82**	**94.16±2.27**	90.02±3.05
		1,1,1,1,3	**91.46±1.97**	89.86±2.43	83.56±3.68	87.38±4.10	94.06±2.43	**90.11±2.71**
		1,1,1,2,1	89.00±2.00	89.30±2.20	82.10±2.90	85.30±3.00	93.70±2.20	89.90±2.50
		1,1,2,1,1	88.60±2.10	88.90±2.30	81.70±3.10	84.80±3.10	93.45±2.15	89.70±2.40
		1,2,1,1,1	88.10±2.20	88.30±2.40	81.20±3.30	84.10±3.20	93.20±2.25	89.60±2.35
		2,1,1,1,1	87.80±2.30	87.90±2.50	80.70±3.40	83.80±3.30	92.90±2.30	89.40±2.30
	10	1,1,1,1,0	90.32±1.61	89.67±1.94	**85.85±2.99**	**87.35±2.36**	91.30±1.11	90.46±3.09
		1,1,1,1,1	**90.90±2.00**	**89.90±2.20**	84.60±3.00	86.30±2.80	94.10±2.30	92.20±3.10
		1,1,1,1,2	89.67±2.93	89.66±2.47	83.36±3.62	85.35±3.81	95.35±3.67	94.94±3.71
		1,1,1,1,3	89.77±2.62	88.97±3.01	84.04±3.42	86.15±4.22	**95.77±4.11**	**95.01±3.50**
		1,1,1,2,1	90.60±1.90	89.80±2.10	84.20±3.10	86.00±2.90	93.40±2.50	91.90±3.00
		1,1,2,1,1	90.50±2.10	89.60±2.00	84.00±3.20	85.80±3.00	92.90±2.60	91.70±2.80
		1,2,1,1,1	90.10±2.20	89.50±2.20	83.80±3.40	85.60±3.20	92.50±2.80	91.60±2.70
		2,1,1,1,1	89.70±2.30	89.30±2.30	83.50±3.50	85.10±3.40	91.80±3.00	91.10±2.80
TDBrain	100	1,1,1,1,0	85.47±1.16	85.68±1.20	85.47±1.16	85.45±1.16	91.80±1.02	93.96±0.99
		1,1,1,1,1	91.20±1.90	90.60±1.80	90.90±1.75	91.10±1.85	95.60±1.60	96.20±1.65
		1,1,1,1,2	93.26±2.33	**92.84±1.76**	**93.44±1.62**	93.07±1.92	**96.74±1.98**	**97.60±1.82**
		1,1,1,1,3	**93.46±2.83**	92.49±2.07	93.04±1.34	**93.43±2.13**	96.39±1.56	97.44±1.38
		1,1,1,2,1	90.70±1.85	90.30±1.95	90.60±1.80	90.80±1.90	94.80±1.75	95.60±1.70
		1,1,2,1,1	90.40±1.95	90.00±2.05	90.20±1.85	90.50±1.95	94.10±1.80	95.20±1.75
		1,2,1,1,1	89.90±2.05	89.70±2.15	89.80±2.00	90.00±2.10	93.30±1.85	94.50±1.80
		2,1,1,1,1	89.40±2.10	89.20±2.25	89.50±2.05	89.80±2.20	92.40±1.90	94.00±1.85
	10	1,1,1,1,0	79.28±4.64	79.83±4.83	79.28±4.64	79.19±4.62	88.39±4.13	88.38±3.96
		1,1,1,1,1	81.00±3.60	80.80±3.90	80.90±3.70	81.00±3.65	92.60±3.50	93.80±3.60
		1,1,1,1,2	**82.20±3.08**	**81.53±3.91**	**81.90±3.51**	**82.09±3.60**	**93.73±3.27**	**94.81±3.49**
		1,1,1,1,3	81.49±2.97	81.07±3.27	81.45±2.90	81.27±3.73	93.40±2.95	94.01±2.94
		1,1,1,2,1	80.70±3.80	80.50±3.60	80.80±3.70	80.90±3.80	91.60±3.60	92.90±3.55
		1,1,2,1,1	80.40±3.90	80.20±3.80	80.50±3.80	80.60±3.90	91.00±3.70	92.40±3.60
		1,2,1,1,1	80.10±4.00	80.00±3.90	80.20±3.90	80.30±3.90	90.40±3.80	91.80±3.70
		2,1,1,1,1	79.80±4.20	79.90±4.00	79.70±4.00	79.80±4.10	89.60±3.90	91.20±3.80

Table E1: Mutual information scores between features and disease labels on the AD dataset. The Discrepancy Estimator reconstruction loss achieves the highest mutual information.Bold indicates the best performance.

Feature	Mutual Information Score
Reconstruction Loss	**0.0295**
Feature 9	0.0159
Feature 5	0.0140
Feature 0	0.0064
Feature 15	0.0054
Feature 3	0.0053
Feature 7	0.0045
Feature 4	0.0036
Feature 11	0.0028
Feature 1	0.0023
Feature 12	0.0013
Feature 10	0.0007
Feature 8	0.0001
Feature 2	0.0001
Feature 6	0.0000
Feature 13	0.0000
Feature 14	0.0000

Table E2: Ablation study of contrastive blocks on the AD,PTB,TDBrain dataset under 100% and 10% label settings. O: baseline, S: subject, R: temporal, P: epoch, V: view. Bold indicates the best performance, and underline denotes the second best.

Dataset	Fraction	Blocks	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
AD	**100%**	O	81.69 ± 10.71	87.28 ± 13.13	79.64 ± 12.07	78.90 ± 13.78	92.54 ± 4.18	92.36 ± 4.34
		S	83.53 ± 2.89	84.35 ± 1.69	82.81 ± 3.56	82.97 ± 3.47	91.01 ± 1.96	90.81 ± 1.95
		R	78.58 ± 14.99	83.14 ± 11.66	76.31 ± 16.66	74.53 ± 18.69	85.05 ± 12.74	84.44 ± 13.22
		P	72.69 ± 13.13	78.67 ± 11.81	69.74 ± 14.50	67.04 ± 17.27	83.30 ± 13.91	82.57 ± 14.29
		S+O	85.70 ± 1.82	86.14 ± 1.63	85.09 ± 2.04	85.35 ± 1.96	92.21 ± 1.40	92.09 ± 1.39
		R+S+O	85.57 ± 4.04	88.12 ± 2.58	84.31 ± 4.79	84.73 ± 4.59	93.28 ± 2.52	92.98 ± 2.83
		P+R+S+O	84.50 ± 4.46	88.31 ± 2.42	82.95 ± 5.39	83.33 ± 5.15	94.44 ± 2.37	94.43 ± 2.48
		P+R+S+O+V	**91.16 ± 4.14**	**92.10 ± 3.66**	**90.50 ± 4.51**	**90.88 ± 4.33**	**96.22 ± 2.76**	**96.03 ± 2.82**
	10%	O	86.35 ± 9.25	87.09 ± 9.04	87.09 ± 9.04	85.55 ± 9.92	92.62 ± 8.99	92.77 ± 8.72
		S	77.30 ± 3.63	78.45 ± 3.75	78.45 ± 3.75	76.26 ± 3.91	85.43 ± 3.61	84.63 ± 3.76
		R	77.83 ± 17.59	83.45 ± 14.35	75.41 ± 14.35	75.41 ± 19.62	83.27 ± 17.00	83.27 ± 17.00
		P	71.41 ± 15.31	76.91 ± 15.56	68.70 ± 16.46	92.52 ± 2.36	**97.61 ± 16.98**	76.06 ± 16.19
		S+O	82.73 ± 2.05	84.33 ± 2.05	84.33 ± 2.71	81.51 ± 2.07	90.02 ± 2.47	90.02 ± 2.40
		R+S+O	91.19 ± 3.14	91.74 ± 3.01	90.70 ± 3.34	90.70 ± 3.34	95.86 ± 2.63	95.86 ± 2.67
		P+R+S+O	91.43 ± 3.12	92.52 ± 2.36	90.71 ± 3.56	91.14 ± 3.56	96.44 ± 2.84	96.48 ± 2.82
		P+R+S+O+V	**92.77 ± 2.35**	**92.92 ± 2.43**	**92.92 ± 2.43**	**92.55 ± 2.39**	92.66 ± 2.37	**97.39 ± 1.40**
PTB	**100%**	O	85.27 ± 1.73	84.21 ± 1.25	77.74 ± 4.01	79.78 ± 3.37	88.13 ± 2.23	84.76 ± 2.14
		S	84.30 ± 2.56	84.00 ± 2.16	75.68 ± 5.69	77.74 ± 5.13	88.66 ± 2.05	84.80 ± 2.15
		R	88.63 ± 1.43	88.42 ± 1.45	82.46 ± 2.35	84.72 ± 1.82	89.29 ± 3.96	86.21 ± 3.96
		P	88.85 ± 3.22	88.74 ± 2.30	82.78 ± 6.11	84.75 ± 5.38	94.32 ± 1.81	90.61 ± 2.81
		S+O	84.38 ± 2.34	84.33 ± 1.18	75.49 ± 4.65	77.69 ± 4.76	90.37 ± 2.59	86.57 ± 4.32
		R+S+O	85.32 ± 1.93	84.82 ± 1.78	77.54 ± 4.58	79.64 ± 3.94	92.34 ± 1.85	88.75 ± 1.98
		P+R+S+O	86.36 ± 1.44	87.18 ± 1.28	77.87 ± 2.55	80.79 ± 2.44	93.41 ± 2.35	89.09 ± 1.64
		P+R+S+O+V	**89.57 ± 1.73**	**90.05 ± 2.04**	**85.52 ± 2.18**	**85.80 ± 2.38**	**95.70 ± 2.42**	**91.13 ± 1.57**
	10%	O	86.75 ± 1.44	85.42 ± 2.10	80.88 ± 3.27	82.45 ± 2.29	90.71 ± 3.16	88.84 ± 3.59
		S	86.34 ± 0.73	84.75 ± 1.49	80.21 ± 1.78	81.90 ± 1.25	88.36 ± 1.31	85.20 ± 1.17
		R	88.12 ± 1.75	86.36 ± 2.28	84.16 ± 3.55	84.16 ± 3.55	91.80 ± 2.50	90.45 ± 1.90
		P	90.38 ± 2.23	90.33 ± 2.71	85.69 ± 4.36	87.27 ± 3.45	93.06 ± 3.86	91.83 ± 2.93
		S+O	85.84 ± 1.74	84.14 ± 0.76	80.07 ± 5.18	81.14 ± 3.82	90.28 ± 2.75	87.57 ± 3.42
		R+S+O	85.88 ± 3.08	83.60 ± 3.61	82.45 ± 1.94	82.45 ± 1.94	92.79 ± 1.79	88.68 ± 1.91
		P+R+S+O	90.32 ± 1.61	89.67 ± 1.94	85.85 ± 2.99	87.35 ± 2.36	92.78 ± 1.11	90.46 ± 3.09
		P+R+S+O+V	**91.53 ± 1.39**	**90.64 ± 1.82**	**87.63 ± 2.68**	**89.21 ± 2.11**	**93.63 ± 1.94**	**92.32 ± 2.11**
TDBrain	**100%**	O	89.92 ± 2.15	90.50 ± 2.07	89.92 ± 2.15	89.89 ± 2.17	96.79 ± 1.29	96.84 ± 1.29
		S	87.86 ± 2.85	78.98 ± 2.51	77.86 ± 2.85	77.63 ± 3.01	88.44 ± 2.23	88.96 ± 2.14
		R	94.44 ± 1.79	94.60 ± 1.75	94.44 ± 1.79	94.44 ± 1.80	98.39 ± 1.29	98.31 ± 1.46
		P	93.18 ± 3.70	93.27 ± 3.67	93.18 ± 3.70	93.18 ± 3.70	97.59 ± 2.01	97.58 ± 2.00
		S+O	80.38 ± 4.22	81.59 ± 3.70	80.38 ± 4.22	80.16 ± 4.38	91.35 ± 2.24	91.64 ± 2.05
		R+S+O	86.01 ± 4.29	86.35 ± 3.81	86.01 ± 4.29	85.95 ± 4.36	94.14 ± 2.72	94.37 ± 2.58
		P+R+S+O	85.47 ± 1.16	85.68 ± 1.20	85.47 ± 1.16	85.45 ± 1.16	93.73 ± 1.02	93.96 ± 0.90
		P+R+S+O+V	**95.11 ± 1.53**	**95.23 ± 1.43**	**95.19 ± 1.69**	**95.67 ± 1.39**	**98.54 ± 1.22**	**98.49 ± 1.10**
	10%	O	85.34 ± 4.38	86.03 ± 3.97	85.34 ± 4.38	85.25 ± 4.48	93.18 ± 3.52	93.24 ± 3.53
		S	74.02 ± 2.09	74.62 ± 2.01	74.02 ± 2.09	73.86 ± 2.17	81.92 ± 3.25	81.76 ± 3.42
		R	92.96 ± 8.22	93.02 ± 8.20	92.96 ± 8.22	92.96 ± 8.23	96.14 ± 5.80	95.93 ± 6.08
		P	89.38 ± 14.69	89.58 ± 14.72	89.38 ± 14.69	89.36 ± 14.93	93.23 ± 11.81	93.52 ± 11.10
		S+O	74.92 ± 2.57	76.60 ± 3.07	74.92 ± 2.57	74.54 ± 2.55	84.51 ± 3.07	84.40 ± 3.07
		R+S+O	81.90 ± 4.74	83.55 ± 4.02	81.90 ± 4.74	81.61 ± 4.99	91.21 ± 3.52	90.73 ± 3.57
		P+R+S+O	79.28 ± 4.64	79.83 ± 4.83	79.28 ± 4.64	79.19 ± 4.62	88.39 ± 4.13	88.38 ± 4.09
		P+R+S+O+V	**93.57 ± 4.13**	**93.02 ± 4.21**	**93.53 ± 4.31**	**94.53 ± 4.13**	**97.21 ± 4.13**	**96.68 ± 4.13**

Table E3: Comparison of model performance on AD, PTB, and TDBrain datasets under different ratios of external data.

Dataset	Label Fraction (%)	External Ratio	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC
AD	100	100%	**93.23 ± 5.25**	**94.01 ± 3.98**	**92.71 ± 5.86**	**92.97 ± 5.65**	**98.03 ± 1.71**	**98.03± 1.75**
		50%	92.21 ± 3.89	93.48 ± 3.94	92.22 ± 4.61	92.70 ± 4.13	97.10 ± 3.21	97.96 ± 3.28
		25%	91.45 ± 4.21	93.10 ± 4.55	91.73 ± 4.33	92.04 ± 4.14	96.84 ± 3.95	96.67 ± 4.22
		5%	91.23 ± 4.01	92.15± 3.06	90.70 ± 4.01	91.88 ± 4.33	96.34± 2.22	96.13 ± 2.23
	10	100%	**94.67± 1.84**	**94.93 ± 1.76**	**94.41 ± 1.99**	**94.58 ± 1.89**	**98.10 ± 1.20**	**98.19 ±1.15**
		50%	93.24 ± 2.86	94.01 ± 2.91	93.76 ± 2.63	93.22 ± 2.67	98.01 ± 1.72	97.94 ± 1.88
		25%	92.93 ± 3.19	93.56 ± 3.25	93.24 ± 3.60	93.07 ± 3.14	97.87 ± 2.10	97.66 ± 2.41
		5%	92.87 ± 2.35	93.02± 2.43	92.89 ± 2.37	93.42 ± 2.87	97.50 ± 1.45	97.49 ± 1.57
PTB	100	100%	**93.65 ± 2.35**	**93.28 ± 1.86**	**85.39 ± 3.84**	**87.64 ± 3.27**	**95.56 ± 1.57**	**90.28 ± 3.49**
		50%	92.60 ± 2.12	92.89 ± 2.63	85.30 ± 4.36	85.71 ± 3.44	95.30 ± 2.34	90.20± 3.47
		25%	92.12 ± 3.15	92.55 ± 2.90	84.54 ± 5.19	85.69± 4.31	94.27 ± 2.83	90.16± 3.98
		5%	91.50± 2.03	91.89 ± 2.83	84.40± 4.09	85.67 ± 3.71	94.23 ± 1.94	90.05± 3.20
	10	100%	**91.35± 2.77**	**91.56± 3.12**	**85.23± 2.65**	**85.61 ± 3.56**	**96.30 ± 3.22**	**97.87± 4.02**
		50%	91.05 ± 3.10	91.19 ± 2.96	84.91 ± 3.92	85.55 ± 3.62	96.26 ± 3.42	96.88 ± 3.55
		25%	90.93± 3.48	90.61 ± 3.27	84.52 ± 4.34	85.48 ± 3.89	96.07 ± 3.88	96.03 ± 4.12
		5%	90.80± 2.93	90.06 ±2.47	83.80± 4.86	85.43 ±3.27	95.91 ± 3.51	95.80± 3.39
TDBrain	100	100%	**95.60±1.20**	**94.95 ± 1.69**	**95.73 ± 1.74**	**94.61 ± 1.26**	**97.63 ± 1.57**	**98.51 ± 1.29**
		50%	95.45 ± 2.49	94.03 ± 2.12	94.88 ± 2.33	94.30 ± 2.28	97.21 ± 2.35	98.21 ± 2.22
		25%	94.22 ± 2.71	93.64 ± 2.44	94.40 ± 2.96	93.31 ± 2.73	97.13 ± 2.78	97.80 ± 2.89
		5%	93.51± 3.05	93.05± 3.11	93.55 ± 3.28	93.20± 3.16	96.88 ± 3.33	97.77 ± 3.47
	10	100%	**83.66 ± 3.29**	**84.06 ±3.06**	**83.29 ± 2.96**	**84.60 ±3.29**	**95.27 ± 4.01**	**94.71±4.09**
		50%	82.64 ± 3.52	83.78 ± 3.77	82.90 ± 3.93	83.64 ± 3.64	94.48 ± 3.84	94.00 ± 3.91
		25%	82.83 ± 4.01	83.35 ± 4.19	82.55 ± 4.37	83.50 ± 4.22	94.12 ± 4.20	93.43 ± 4.45
		5%	82.35± 4.65	82.22 ± 4.51	82.03 ± 4.60	82.66 ± 4.68	94.07± 4.82	93.04 ± 4.77

Table F1: Comparison of different discrepancy feature designs used in the Discrepancy Estimator under full fine-tuning on the AD dataset with 100% labeled data as Table 2. Discrepancy features are defined as follows: (a) **Sequence-level MSE scalar** — average MSE over all time steps and channels; (b) **Channel-wise MSE vector** — 16-dimensional vector of channel-specific MSEs; (c) **Point-wise error sequence** — 256-dimensional vector of per-timestep reconstruction errors; (d) **Cluster-wise distance** — latent-space distance to nearest cluster center from external normal data.

Discrepancy Feature	**AD AUROC**	**TDBrain AUROC**	**PTB AUROC**
(a) Sequence-level MSE scalar (ours)	**98.03**	**97.63**	**95.56**
(b) Channel-wise MSE vector	97.36	97.58	95.44
(c) Point-wise error sequence	96.71	96.86	95.37
(d) Cluster-wise distance	96.28	96.44	94.95

Table G1: Architecture details of DAAC used in our experiments.

Component	**Setting**
Encoder Type	DaulMultiHeadTSEncoder
Input Dimensions	Task-specific (e.g., 1 or 2 for AD)
Output Dimensions	320
Hidden Dimensions	64
Encoder Depth	10
Number of Heads	2
Head Dimension	160
Channel Dimension	320
Activation Function	ReLU (implicit via submodules)
Parameter Count (AD)	946,368