Decentralized Attention Fails Centralized Signals: Rethinking Transformers for Medical Time Series

Anonymous authors
Paper under double-blind review

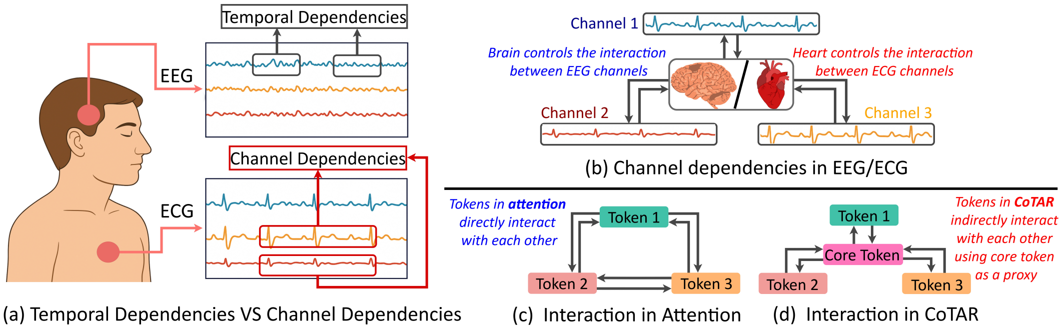
Abstract

Accurate analysis of Medical time series (MedTS) data, such as Electroencephalography (EEG) and Electrocardiography (ECG), plays a pivotal role in healthcare applications, including the diagnosis of brain and heart diseases. MedTS data typically exhibits two critical patterns: **temporal dependencies** within individual channels and **channel dependencies** across multiple channels. While recent advances in deep learning have leveraged Transformer-based models to effectively capture temporal dependencies, they often struggle with modeling channel dependencies. This limitation stems from a structural mismatch: MedTS signals are inherently centralized, whereas the Transformer's attention is decentralized, making it less effective at capturing global synchronization and unified waveform patterns. To address this mismatch, we propose **CoTAR** (Core Token Aggregation-Redistribution), a centralized MLP-based module tailored to replace the decentralized attention. Instead of allowing all tokens to interact directly, as in attention, CoTAR introduces a global core token that acts as a proxy to facilitate the inter-token interaction, thereby enforcing a centralized aggregation and redistribution strategy. This design not only better aligns with the centralized nature of MedTS signals but also reduces computational complexity from quadratic to linear. Experiments on five benchmarks validate the superiority of our method in both effectiveness and efficiency, achieving up to a **12.13%** improvement on the APAVA dataset, with merely 33% memory usage and 20% inference time compared to the previous state-of-the-art. Code and all training scripts are available in `https://anonymous.4open.science/r/TeCh-24`

1 Introduction

Medical time series (MedTS) data are temporal sequences of physiological data used to monitor a subject's health status (Badr et al., 2024), such as Electroencephalography (EEG) for neurological assessment (Arif et al., 2024; Jafari et al., 2023) and Electrocardiography (ECG) for cardiac diagnosis (Xiao et al., 2023; Wang et al., 2023). Accurate classification of MedTS facilitates early anomaly detection, timely diagnosis, and personalized treatment (Liu et al., 2021; Murat et al., 2020). This requires adequate modeling for two critical patterns: *temporal dependencies* within individual channels and *channel dependencies* across multiple channels, as illustrated in Figure 1 (a). Temporal dependencies reflect the intrinsic signal dynamics over time within each channel, such as oscillatory rhythms and event-related potentials for EEG (Niedermeyer & da Silva, 2005b), and P&T wave for ECG (Goldberger et al., 2000b). In contrast, channel dependencies capture the interactions and entanglements among multiple channels, such as functional connectivity for EEG (Stam, 2005) and the biophysical geometry of the heart for ECG (Macfarlane et al., 2005).

Previous deep-learning methods have achieved remarkable performance by focusing on modeling temporal dependencies, using architectures such as recurrent neural networks (RNNs) (Roy et al., 2019; Alhagry et al., 2017), convolutional neural networks (CNNs) (Wang et al., 2024a; Lawhern et al., 2018), or CNN–attention hybrids (Miltiadous et al., 2023a). However, each of these methods has limitations: RNNs suffer from sequential bottlenecks and difficulty capturing long-term dependencies, while CNNs are limited by local receptive fields and struggle with global temporal context. In contrast, Transformer (Vaswani et al., 2017) employs a decentralized attention mechanism, where each token can directly interact with all other tokens, which enables global receptive fields, allowing it to capture long-range and complex temporal dependencies effectively. This makes Transformer-based models deliver state-of-the-art MedTS classification performance (Wang et al., 2024b; Mobin et al., 2025). Despite their success in modeling temporal dependencies, Transformers face fundamental challenges when applied to modeling channel dependencies in MedTS. As illustrated in Figure 1 (b), **MedTS signals typically originate from a centralized biological source**. For example, EEG rhythms emerge from thalamo–cortical circuits synchronizing cortical neurons into coherent scalp oscillations (Schaul, 1998; Scherg et al., 2019a), and ECG waveforms arise when impulses from the sinoatrial node propagate uniformly across the heart's conduction network (Rieta & Alcaraz, 1999a; AlGhatrif & Lindsay, 2012b). In contrast, Transformer's attention operates as a decentralized graph (Figure 1 (c)): every token attends equally to every other token (Vaswani et al., 2017). This uniform treatment of inter-channel interactions overlooks the inherent central coordination present in MedTS data. As a result, the attention mechanism tends to dilute the principal, centrally driven patterns—such as the cardiac pacemaker rhythms—and thus fails to capture the global synchronization and unified waveform features that are essential for accurate modeling of channel dependencies in MedTS.


Temporal Dependencies
EEG
Channel Dependencies
ECG
Channel 1
Brain controls the interaction between EEG channels
Heart controls the interaction between ECG channels
Channel 2
Channel 3
Tokens in attention directly interact with each other
Token 1
Token 2
Token 3
Tokens in CoTAR indirectly interact with each other using core token as a proxy
Core Token

(b) Channel dependencies in EEG/ECG

(a) Temporal Dependencies VS Channel Dependencies (c) Interaction in Attention (d) Interaction in CoTAR

Figure 1: (a): Illustration of Temporal dependencies within each channel, and channel dependencies across channels. (b): Interaction between channels in EEG/ECG signals is centrally controlled by the brain/heart. (c): Attention module is a decentralized structure, where each token attends to all other tokens equally. (d): The proposed Core Token Aggregation-Redistribution (CoTAR) module operates in a centralized manner, with a core token as a proxy.

To address this mismatch between the centralized nature of MedTS and the decentralized structure of attention, we ask: *can we maintain the benefits of attention (flexible, dynamic cross-channel interaction) while renovating it to reflect the centralized organization of MedTS?* Inspired by star-shaped architectures in distributed systems—where a central server mediates all communication for improved efficiency and robustness (Roberts & Wessler, 1970; Guo et al., 2019)—we propose CoTAR (**Co**re **T**oken **A**ggregation-**R**edistribution): a lightweight, MLP-based module that seamlessly replaces the conventional attention. Instead of pairwise token interactions, CoTAR introduces a global core token that first aggregates information from all tokens and then redistributes it into each token, enabling centralized and flexible communication (Figure 1 (d)). This architecture not only better mirrors the central coordination inherent in signals like EEG and ECG, but also reduces the computational complexity of token interaction from **quadratic** to **linear**. This shift enables significant gains in scalability and efficiency, particularly for long or high-dimensional sequences common in medical applications (Arif et al., 2024; Jafari et al., 2023).

With CoTAR, we propose **TeCh**, a unified CoTAR-based framework that adaptively captures **Te**mporal dependencies, **Ch**annel dependencies, or both, by tuning the tokenization strategy (Temporal, Channel, or Dual). Such flexibility is particularly desirable in real-world medical time series, where not all datasets simultaneously exhibit strong temporal and inter-channel patterns.

We conduct extensive experiments across five MedTS datasets, including three EEG datasets and two ECG datasets. Results show that Tech not only achieves the best performance across all datasets but also introduces significantly lower resource consumption, highlighting its superior effectiveness, efficiency, and potential for broader real-world applications.

2 Related Work

Medical Time Series. Medical time series (MedTS) are time series data collected from the human body, used for disease diagnosis (Liu et al., 2021; Xiao et al., 2023), health monitoring (Badr et al., 2024), and brain-computer interfaces (BCIs) (Musk et al., 2019; Altaheri et al., 2023). MedTS include EEG (Tang et al., 2021), ECG (Xiao et al., 2023), EMG (Xiong et al., 2021), and EOG (Jiao et al., 2020), each offering crucial information for medical applications. For example, EEG and ECG data are critical in assessing brain and heart health (Tang et al., 2021; Xiao et al., 2023). Such MedTS are characterized by temporal dependencies within each channel and channel dependencies between channels. Temporal dependencies include oscillatory rhythms and event-related potentials for EEG (Niedermeyer & da Silva, 2005b), P wave and T wave for ECG (Goldberger et al., 2000b). While the channel dependencies consist of functional connectivity for EEG (Stam, 2005), biophysical geometry of the heart for ECG (Macfarlane et al., 2005). Accurate modeling of these two patterns presents unique challenges. Recently, deep learning methods have significantly advanced the field of MedTS classification by providing precise temporal dependencies modeling using RNNs (Roy et al., 2019; Alhagry et al., 2017), CNNs (Lawhern et al., 2016), and Transformer (Wang et al., 2024b; Mobin et al., 2025), but the channel dependencies remain underexplored (Li et al., 2024; Fan et al., 2025; Kim et al., 2025).

Transformers for Time Series. Transformer-based models have been extensively adopted for time series analysis, with growing attention to both temporal and channel dependencies. For example, Informer (Zhou et al., 2021) proposes the Temporal embedding that aggregates values across channels as a token to model temporal dependencies. Autoformer (Wu et al., 2021) utilizes seasonal and trend decomposition to capture disentangled temporal information. PatchTST (Nie et al., 2023) splits the series from one channel into multiple patches, which improves the extraction of long-term temporal variations. iTransformer (Liu et al., 2024) embeds the whole series of a channel into the Variate embedding, which maintains its complete context, thereby enhancing channel dependencies modeling. Finally, Leddam (Yu et al., 2024) introduces a dual attention module to capture both temporal and channel dependencies.

Though the effective extraction of temporal dependencies has been addressed in MedTS using Temporal embedding and Transformer (Wang et al., 2024b; Mobin et al., 2025), the mismatch between the current decentralized attention structure and the centrally organized MedTS fails the Transformer in channel dependencies modeling. To address this, we propose a centralized MLP-based Core Token Aggregation-Redistribution (CoTAR) module, which delivers higher channel dependencies modeling ability while introducing only **Linear** complexity. By replacing attention using CoTAR, we propose a framework that can adaptively model **Te**mporal dependencies or **Ch**annel dependencies or both (denoted as **TeCh**) by tuning the tokenization strategy (Temporal, Channel, or Dual), whose effectiveness and efficiency are validated on five benchmarks.

3 Preliminaries

Subject-Independent Setting. Medical time series (MedTS) data exhibit a hierarchical structure—spanning subjects (individuals), sessions (recordings per visit), trials (repeated measurements), and samples (short segments used for diagnosis model training) (Wang et al., 2024a). In clinical diagnosis tasks, the goal is to predict disease status at the subject level using tools such as deep models trained on MedTS samples. To ensure clinically meaningful evaluations, we adopt the '***Subject-Independent***' protocol (Wang et al., 2024c;b), which splits the dataset by subjects. Each subject—and all associated samples—appears exclusively in either the training, validation, or test set. This setting better reflects real-world deployment, where models must generalize to unseen patients, therefore providing a practical comparison.

Problem Formulation. *Consider an input MedTS sample $X \in \mathbb{R}^{T \times C}$, where T denotes the number of timestamps and C represents the number of channels. Our objective is to learn a function that can predict the corresponding label $\hat{Y} \in \mathbb{R}^{K}$. Here, K denotes the number of classes, such as various disease types or different stages of one disease.*

4 METHOD

4.1 ATTENTION *vs* CORE TOKEN AGGREGATION-REDISTRIBUTION

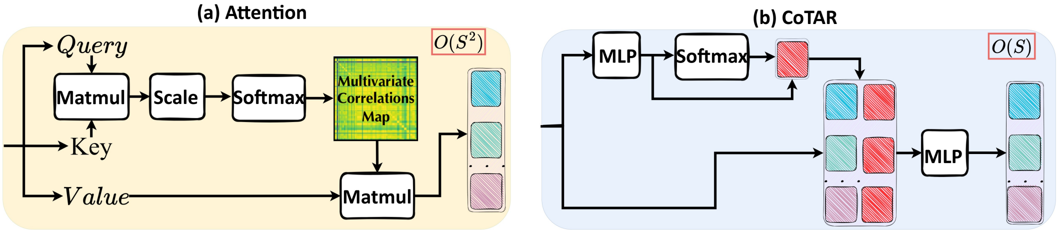


Figure 2: Illustration of attention and Core Token Aggregation-Redistribution (CoTAR). Attention is organized in a decentralized way where each token directly interacts with all tokens, introducing a ***Quadratic*** complexity. CoTAR first aggregates a core token and then redistributes it across channels to facilitate centralized channel interaction, bringing only **Linear** complexity.

The standard Attention. Transformer has demonstrated strong performance in many domains due to its ability to capture complex inter-token relationships, benefiting from the attention mechanism. Formally, for an input embedding $O \in \mathbb{R}^{S \times D}$ (where S is the number of tokens and D the embedding dimension), as in Figure 2 (a), attention operates via:

$$\begin{aligned} Q = OW_Q + b_q, \quad K = OW_K + b_k, \quad V = OW_V + b_v, \\ A = Softmax(\frac{QK^T}{\sqrt{D}})V, \quad Q, K, V, A \in \mathbb{R}^{S \times D}. \end{aligned} \tag{1}$$

As mentioned before, such a decentralized structure does not fit the centrally controlled MedTS data. Besides, its quadratic complexity stemmed from the matrix multiplications between $Query$ and Key, making it inefficient for long and high-dimensional MedTS (Albuquerque et al., 2019).

Core Token Aggregation-Redistribution (CoTAR).. To better match the MedTS and break the scalability bottleneck of attention, we borrow insight from the star-shaped centralized system in software engineering. Traditional peer-to-peer structure lets the clients communicate directly with each other, which is time- and resource-consuming. So a more reliable and efficient way is to set a server to aggregate and exchange the information between clients (Roberts & Wessler, 1970; Guo et al., 2019). Motivated by this, we propose the Core Token Aggregation-Redistribution (CoTAR), a plug-in module that can seamlessly replace attention, as shown in Figure 2 (b). CoTAR first projects the token of each channel, aggregates global context across channels into a core vector, and redistributes it back to every token. Given input $O \in \mathbb{R}^{S \times D}$, where S denotes the number of tokens and D the hidden dimension, CoTAR performs aggregation and redistribution as follows:

$$\begin{aligned} &\tilde{O} = \mathrm{GELU}(OW_1 + b_1)W_2 + b_2, && W_1 \in \mathbb{R}^{D \times D},\ b_1 \in \mathbb{R}^{D},\ W_2 \in \mathbb{R}^{D \times D_c},\ b_2 \in \mathbb{R}^{D_c}, \\ &O_w = \mathrm{Softmax}(\tilde{O}, \mathrm{dim} = 0), && \tilde{O} \in \mathbb{R}^{S \times D_c},\ O_w \in \mathbb{R}^{S \times D_c}, \\ &\tilde{C_o} = \mathrm{Sum}(\tilde{O} \odot O_w, \mathrm{dim} = 0), && \tilde{C_o} \in \mathbb{R}^{D_c}, \\ &C_o = \mathrm{Repeat}(\tilde{C_o}, \mathrm{time} = S, \mathrm{dim} = 0), && C_o \in \mathbb{R}^{S \times D_c}, \\ &O_{C_o} = \mathrm{Concat}([O, C_o], \mathrm{dim} = 1), && O_{C_o} \in \mathbb{R}^{S \times (D + D_c)}, \\ &A = \mathrm{GELU}(O_{C_o}W_3 + b_3)W_4 + b_4, && W_3 \in \mathbb{R}^{(D + D_c) \times D},\ W_4 \in \mathbb{R}^{D \times D},\ b_3, b_4 \in \mathbb{R}^{D}. \end{aligned} \tag{2}$$

D_c is the dimension of core token, $\tilde{C_o}$ **is the obtained core token** by aggregating information across all channels, and $A \in \mathbb{R}^{S \times D}$ is the final output. CoTAR employs a centralized structure that first gets the global core token by aggregating information from all channels. Then the core token is redistributed into each token. This realizes an indirect interaction between channels using the core token as a proxy (like the brain/heart in EEG/ECG). And since each token only needs to interact with a single core token, it only brings **Linear** complexity. Thus, CoTAR delivers higher effectiveness with lower resource consumption.

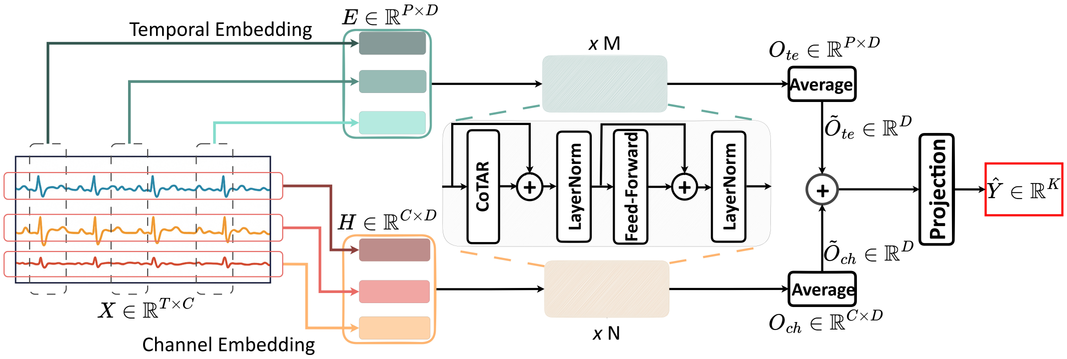

Temporal Embedding
$E \in \mathbb{R}^{P\times D}$
x M
$O_{te} \in \mathbb{R}^{P\times D}$
Average
$\tilde{O}_{te} \in \mathbb{R}^{D}$
CoTAR
LayerNorm
Feed-Forward
LayerNorm
Projection
$\hat{Y} \in \mathbb{R}^{K}$
$H \in \mathbb{R}^{C\times D}$
$\tilde{O}_{ch} \in \mathbb{R}^{D}$
Average
$O_{ch} \in \mathbb{R}^{C\times D}$
x N
$X \in \mathbb{R}^{T\times C}$
Channel Embedding

Figure 3: **Overview of TeCh.** MedTS signals $X \in \mathbb{R}^{T\times C}$ are embedded into Temporal embedding and Channel embedding. Then, each embedding is processed using Transformer encoders, with attention replaced by CoTAR. The final output representation from each branch is averaged across channels and added, then projected to the final predicted logits $\hat{Y} \in \mathbb{R}^{K}$.

4.2 Overview of TeCh

The proposed Tech framework is illustrated in Figure 3. The raw MedTS is embedded into Temporal and Channel embedding, each is processed using a set of Transformer Encoders (M for Temporal and N for Channel, M and N are tunable to match with data, and the Temporal or Channel branch will be removed if $M = 0$ or $N = 0$); the learned representations are average across channels, fused and projected to the final output $\hat{Y} \in \mathbb{R}^{K}$.

Adaptive Dual Tokenization. Existing methods mainly rely on Temporal embedding that treats single or multiple timestamps across channels as a token, favoring temporal dependencies modeling while hindering channel dependencies extraction (Liu et al., 2024; Yu et al., 2024). So we take a balanced adaptive consideration of both patterns by using Adaptive Dual Tokenization.

Specifically, we form a temporal token by aggregating one or multiple timestamps across channels:

$$\begin{aligned} E_{i,:} &= \mathrm{vec}(X_{(i-1)L:iL,:})W_t + b_t + W^{tpos}_{i,:}, \\ i &= 1, \dots, P, \quad P = \lceil T/L \rceil, \\ W_t &\in \mathbb{R}^{LC\times D}, \quad b_t \in \mathbb{R}^{D}, \quad W^{tpos} \in \mathbb{R}^{P\times D}. \end{aligned} \tag{3}$$

where vec : $\mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ flattens a 2D tensor into a 1D tensor, L is a predefined hyperparameter that decides the granularity, W^{tpos} is the classical position embedding (Vaswani et al., 2017). This will result in Temporal embedding $E \in \mathbb{R}^{P\times D}$. Then, following ***iTransformer*** (Liu et al., 2024), we form a channel token by aggregating the whole series across all timestamps of a channel:

$$\begin{aligned} H_{j,:} &= X^{\top}_{:,j} W_c + b_c + W^{cpos}_{j,:}, \quad j = 1, \dots, C, \\ W_c &\in \mathbb{R}^{T\times D}, \quad b_c \in \mathbb{R}^{D}, \quad W^{cpos} \in \mathbb{R}^{C\times D}. \end{aligned} \tag{4}$$

This will result in Channel embedding $H \in \mathbb{R}^{C\times D}$. By embedding the whole series of each channel as a token, the unique semantic information of each individual channel is well-retained. Such a channel-centric token is proven to be effective in modeling multivariate correlations (Qiu et al., 2024; Wang et al., 2024d; Han et al., 2024).

In the real world, not all signals simultaneously exhibit strong temporal and inter-channel patterns. Thereby, our Adaptive Dual Tokenization strategy can better match with them by tuning M and N.

Classification Paradigm. After Adaptive Dual Tokenization, the Temporal embedding E and Channel embedding H are processed using M and N standard Transformer Encoders with attention replaced by CoTAR, respectively. Then the learned Temporal representation $O_{te} \in \mathbb{R}^{P\times D}$ from the Temporal embedding is averaged across channels into $\tilde{O}_{te} \in \mathbb{R}^{D}$. Similarly, the learned Channel representation $O_{ch} \in \mathbb{R}^{C\times D}$ from the Channel embedding is averaged into $\tilde{O}_{ch} \in \mathbb{R}^{D}$. Notably, if we set $M = 0$ or $N = 0$, this will remove the Temporal or Channel branch, and $\tilde{O}_{te} = 0$ or $\tilde{O}_{ch} = 0$. The final predicted logits are obtained via:

$$\hat{Y} = (\tilde{O}_{te} + \tilde{O}_{ch})W_y + b_y, \quad W_y \in \mathbb{R}^{D\times K}, \quad b_y \in \mathbb{R}^{K}. \tag{5}$$

With the Adaptive Dual Tokenization strategy, our Tech can adaptively model temporal dependencies or channel dependencies or both, and CoTAR allows for more effective and efficient token correlation extraction. These innovations make Tech a powerful, stable, and scalable framework for MedTS classification.

5 Experiments

5.1 Experiment Setting

We compare our **Tech** with 10 Transformer-based baselines across five MedTS datasets, including 3 EEG datasets, 2 ECG datasets. Our method is evaluated under the ***Subject-Independent*** setting, where training, validation, and test sets are split based on subjects. Additionally, we also conduct extensive experiments on two human activity recognition (HAR) datasets to test the generalizability.

Table 1: **The information of utilized datasets,** including the number of subjects, samples, classes, sample channels, and timestamps (TS).

Dataset	#-Subject	#-Sample	#-Class	#-Channel	#-TS
ADFTD	88	69,752	3	19	256
APAVA	23	5,967	2	16	256
TDBrain	72	6,240	2	33	256
PTB	198	64,356	2	15	250
PTB-XL	17,596	191,400	5	12	250
FLAPP	8	13123	10	6	100
UCI-HAR	30	10,299	6	9	128

Datasets. (1) **APAVA** (Escudero et al., 2006) is an EEG dataset where each sample is assigned a binary label indicating whether the subject has Alzheimer's disease. (2) **TDBrain** (van Dijk et al., 2022) is an EEG dataset with a binary label assigned to each sample, indicating whether the subject has Parkinson's disease. (3) **ADFTD** (Miltiadous et al., 2023b;a) is an EEG dataset with a three-class label for each sample, categorizing the subject as Healthy, having Frontotemporal Dementia, or Alzheimer's disease. (4) **PTB** (PhysioBank, 2000) is an ECG dataset where each sample is labeled with a binary indicator of Myocardial Infarction. (5) **PTB-XL** (Wagner et al., 2020) is an ECG dataset with a five-class label for each sample, representing various heart conditions. (6) **FLAAP** (Kumar & Suresh, 2022) is a smartphone-based HAR dataset that records accelerometer and gyroscope data for activity pattern recognition. (7) **UCI-HAR** (Anguita et al., 2013) comprises accelerometer and gyroscope data collected via waist-mounted smartphones, widely used for evaluating HAR models. Table 1 provides critical information, such as subjects, channels, and timestamps. The data preprocessing and dataset split follow Medformer (Wang et al., 2024b).

Baselines. We compare with 10 cutting-edge time series Transformer-based methods: Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022), Informer (Zhou et al., 2021), iTransformer (Liu et al., 2024), MTST (Zhang et al., 2024), Nonformer (Liu et al., 2022), PatchTST (Nie et al., 2023), Reformer (Kitaev et al., 2019), vanilla Transformer (Vaswani et al., 2017), and Medformer (Wang et al., 2024b) (state-of-the-art Transformer-based MedTS classification model).

Implementation. We employ six evaluation metrics: accuracy, precision (macro-averaged), recall (macro-averaged), F1 score (macro-averaged), AUROC (macro-averaged), and AUPRC (macro-averaged). The training process is conducted with five random seeds (42-46) to compute the mean and standard deviation. All experiments are run on an NVIDIA RTX 4090 GPU. The results of all baselines on the five MedTS datasets are directly taken from Medformer (Wang et al., 2024b). And the results on the two HAR datasets are reproduced using the official code from Medformer (Wang et al., 2024b). We save the model with the best F1 score on the validation set.

Table 2: **Results on five MedTS datasets.** The training, validation, and test sets are distributed based on subject IDs. The best is **Bolded** and second is *Underlined*.

Datasets	Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC	Avg
ADFTD (3-Classes)	**Autoformer**	45.25±1.48	43.67±1.94	42.96±2.03	42.59±1.85	61.02±1.82	43.10±2.30	46.60±1.87
	FEDformer	46.30±0.59	46.05±0.76	44.22±1.38	43.91±1.37	62.62±1.75	46.11±1.44	48.70±1.04
	Informer	48.45±1.96	46.54±1.68	46.06±1.84	45.74±1.38	65.87±1.27	47.60±1.30	50.21±1.41
	iTransformer	52.60±1.59	46.79±1.27	47.28±1.29	46.79±1.13	67.26±1.16	49.53±1.21	51.38±1.27
	MTST	45.60±2.03	44.70±1.33	45.05±1.30	44.31±1.74	62.50±0.81	45.16±0.85	47.39±1.19
	Nonformer	49.95±1.05	47.71±0.97	47.46±1.50	46.96±1.35	66.23±1.37	47.33±1.78	50.61±1.17
	PatchTST	44.37±0.95	42.40±1.13	42.06±1.48	41.97±1.37	60.08±1.50	42.49±1.79	45.73±1.37
	Reformer	50.78±1.17	49.64±1.49	*49.89±1.67*	47.94±0.69	*69.17±1.58*	**51.73±1.94**	*51.69±1.59*
	Transformer	50.47±2.14	49.13±1.83	48.01±1.53	48.09±1.59	67.93±1.59	48.93±2.02	50.26±1.62
	Medformer	*53.27±1.54*	*51.02±1.57*	**50.71±1.55**	**50.65±1.51**	**70.93±1.19**	*51.21±1.32*	**51.80±1.39**
	TeCh	**54.54±0.70**	**53.02±0.87**	49.25±1.01	*48.84±1.72*	68.67±1.05	50.62±1.26	50.82±1.10
APAVA (2-Classes)	**Autoformer**	68.64±1.82	68.48±2.10	68.77±2.27	68.06±1.94	75.94±3.61	74.38±4.05	70.72±2.63
	FEDformer	74.94±2.15	74.59±1.50	73.56±3.55	73.51±3.39	83.72±1.97	82.94±2.37	77.21±2.49
	Informer	73.11±4.40	75.17±6.06	69.17±4.56	69.47±5.06	70.46±4.91	70.75±5.27	71.02±4.71
	iTransformer	74.55±1.66	74.77±2.10	71.76±1.72	72.30±1.79	*85.59±1.55*	*84.39±1.57*	76.40±1.73
	MTST	71.14±1.59	79.30±0.97	65.27±2.28	64.01±3.16	68.87±2.34	71.06±1.60	69.10±2.07
	Nonformer	71.89±3.81	71.80±4.58	69.44±3.56	69.74±3.84	70.55±2.96	70.78±4.08	70.03±3.80
	PatchTST	67.03±1.65	78.76±1.28	59.91±2.02	55.97±3.10	65.65±0.28	67.99±0.76	65.22±1.68
	Reformer	78.70±2.00	*82.50±3.95*	75.00±1.61	75.93±1.82	73.94±1.40	76.04±1.14	77.52±2.32
	Transformer	76.30±4.72	77.64±5.95	73.09±5.01	73.75±5.38	72.50±6.60	73.23±7.60	74.42±5.04
	Medformer	*78.74±0.64*	81.11±0.84	*75.40±0.66*	*76.31±0.71*	83.20±0.91	83.66±0.92	*79.06±0.78*
	TeCh	**86.86±1.09**	**86.85±1.29**	**86.10±1.00**	**86.30±1.06**	**94.02±0.52**	**93.79±0.56**	**88.65±1.10**
TDBrain (2-Classes)	**Autoformer**	87.33±3.79	88.06±3.56	87.33±3.79	87.26±3.84	93.81±2.26	93.32±2.42	89.02±3.28
	FEDformer	78.13±1.98	78.52±1.91	78.13±1.98	78.04±2.01	86.56±1.86	86.48±1.99	80.81±1.79
	Informer	89.02±2.50	89.43±2.14	89.02±2.50	88.98±2.54	96.64±0.68	96.75±0.63	91.81±1.67
	iTransformer	74.67±1.06	74.71±1.06	74.67±1.06	74.65±1.06	83.37±1.14	83.73±1.27	77.14±1.12
	MTST	76.96±3.76	77.24±3.59	76.96±3.76	76.88±3.83	85.27±4.46	82.81±5.64	79.85±3.95
	Nonformer	87.88±2.48	88.86±1.84	87.88±2.48	87.78±2.56	*97.05±0.68*	*96.99±0.68*	*91.74±1.62*
	PatchTST	79.25±3.79	79.60±4.09	79.25±3.79	79.20±3.77	87.95±4.96	86.36±6.67	81.60±4.01
	Reformer	87.92±2.01	88.64±1.40	87.92±2.01	87.85±2.08	96.30±0.54	96.40±0.45	90.02±1.58
	Transformer	87.17±1.67	87.99±1.68	87.17±1.67	87.10±1.68	96.28±0.92	96.34±0.81	89.68±1.40
	Medformer	*89.62±0.81*	*89.68±0.78*	*89.62±0.81*	*89.62±0.81*	96.41±0.35	96.51±0.33	90.81±0.60
	TeCh	**93.21±0.61**	**93.39±0.58**	**93.21±0.61**	**93.20±0.61**	**98.68±0.19**	**98.72±0.17**	**95.07±0.29**
PTB (2-Classes)	**Autoformer**	73.35±2.10	72.11±2.89	63.24±3.17	63.69±3.84	78.54±3.48	74.25±3.53	69.03±3.33
	FEDformer	76.05±2.54	77.58±3.61	66.10±3.55	67.14±4.37	85.93±4.31	82.59±5.42	76.40±3.30
	Informer	78.69±1.68	82.87±1.02	69.19±2.90	70.84±3.47	92.09±0.53	90.02±0.60	80.45±1.87
	iTransformer	*83.89±0.71*	*88.25±1.18*	76.39±1.01	79.06±1.06	91.18±1.16	*90.93±0.98*	*84.63±1.02*
	MTST	76.59±1.90	79.88±1.90	66.31±2.95	67.38±3.71	86.86±2.75	83.75±2.84	76.13±2.17
	Nonformer	78.66±0.49	82.77±0.86	69.12±0.87	70.90±1.00	89.37±2.51	86.67±2.38	79.75±1.25
	PatchTST	74.74±1.62	76.94±1.51	63.89±2.71	64.36±3.38	88.79±0.91	83.39±0.96	75.02±1.68
	Reformer	77.96±2.13	81.72±1.61	68.20±3.35	69.65±3.88	91.13±0.74	88.42±1.30	79.18±2.17
	Transformer	77.37±1.02	81.84±0.66	67.14±1.80	68.47±2.19	90.08±1.76	87.22±1.68	78.52±1.35
	Medformer	83.50±2.01	85.19±0.94	*77.11±3.39*	*79.18±3.31*	*92.81±1.48*	90.32±1.54	84.02±1.94
	TeCh	**85.96±2.52**	**89.92±0.74**	**79.43±4.13**	**81.97±4.07**	**94.57±0.70**	**94.36±0.66**	**89.94±2.37**
PTB-XL (5-Classes)	**Autoformer**	61.68±2.72	51.60±1.64	49.10±1.52	48.85±2.27	82.04±1.44	51.93±1.71	57.53±1.88
	FEDformer	57.20±9.47	52.38±6.09	49.04±7.26	47.89±8.44	82.13±4.17	52.31±7.03	56.83±7.08
	Informer	71.43±0.32	62.64±0.60	59.12±0.47	60.44±0.43	88.65±0.09	64.76±0.17	67.84±0.35
	iTransformer	69.28±0.22	59.59±0.45	54.62±0.18	56.20±0.19	86.71±0.10	60.27±0.21	64.44±0.23
	MTST	72.14±0.27	63.84±0.72	60.01±0.81	61.43±0.38	88.97±0.33	65.83±0.51	68.70±0.50
	Nonformer	70.56±0.55	61.57±0.66	57.75±0.72	59.10±0.66	88.32±0.36	63.40±0.79	66.78±0.62
	PatchTST	*73.23±0.25*	*65.70±0.64*	**60.82±0.76**	**62.61±0.34**	*89.74±0.19*	**67.32±0.22**	*69.90±0.40*
	Reformer	71.72±0.43	63.12±1.02	59.20±0.75	60.69±0.18	88.80±0.24	64.72±0.47	68.04±0.52
	Transformer	70.59±0.44	61.57±0.65	57.62±0.35	59.05±0.25	88.21±0.16	63.36±0.29	66.73±0.36
	Medformer	72.87±0.23	64.14±0.42	60.60±0.46	62.02±0.37	89.66±0.13	66.39±0.22	69.28±0.30
	TeCh	**73.53±0.07**	**65.92±0.52**	*60.61±0.59*	*62.44±0.27*	**90.03±0.12**	*67.19±0.25*	**69.95±0.30**

5.2 MAIN RESULT

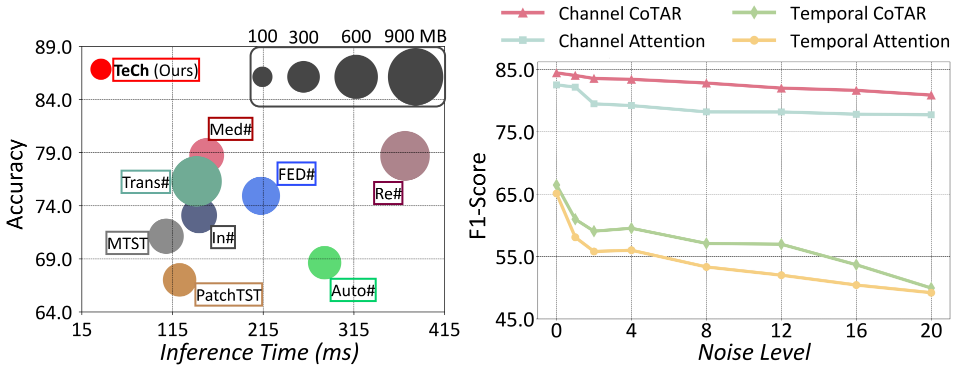
Table 2 presents the results under the Subject-Independent setup. Our Tech consistently outperforms Medformer (the previous state-of-the-art) across all six metrics on four datasets, achieving up to **12.13%** improvement in the ***average of all metrics*** on the APAVA dataset. Even in the challenging case of ADFTD, Tech remains comparable to Medformer (Avg: 51.80 vs. 50.82). Aggregated across all six metrics on these five MedTS datasets, Tech achieves an overall **4.59%** performance gain over Medformer, which is remarkable. In Table 3, Tech substantially outperforms Medformer across all metrics on both datasets, with an average improvement of **4.23%**. Since HAR tasks involve multi-sensor channels and fine-grained activity classes, these consistent and significant gains indicate that Tech generalizes better to noisy, high-variation, multi-channel time series inputs. In terms of robustness, Tech also outperforms Medformer, as reflected in the lower average ***std*** across all datasets (0.86 vs. 0.96, a **10.42%** reduction). These results demonstrate that Tech is both more effective and more robust than Medformer.

Table 3: **Results of two HAR datasets.** To evaluate the performance of our method on general time series, we test it on two human activity recognition (HAR) datasets: FLAAP and UCI-HAR, which exhibit potential channel correlations inherently. The best is **Bolded** and second is *Underlined*.

Datasets	Models	Accuracy	Precision	Recall	F1 score	AUROC	AUPRC	Avg
FLAAP (10-Classes)	**Autoformer**	38.93±1.01	38.22±1.31	37.40±1.17	33.51±1.14	74.12±0.35	35.77±0.91	42.99±0.98
	FEDformer	59.51±9.03	59.84±8.10	58.57±8.97	57.73±9.99	89.75±5.37	60.88±9.63	64.38±8.52
	Informer	72.87±0.89	73.20±0.97	72.76±0.92	72.59±0.96	95.91±0.24	77.57±1.21	77.48±0.86
	iTransformer	75.15±0.48	75.09±0.53	75.14±0.47	74.91±0.51	96.64±0.14	80.81±0.60	79.62±0.46
	MTST	70.57±0.54	71.09±0.73	70.97±0.73	70.61±0.57	94.56±0.18	73.28±0.99	75.18±0.62
	Nonformer	74.85±1.76	75.19±1.37	74.51±1.85	74.39±1.80	96.43±0.27	79.29±1.90	79.11±1.49
	PatchTST	56.34±0.31	56.36±0.63	55.29±0.32	55.58±0.45	89.24±0.11	58.92±0.36	61.95±0.36
	Reformer	71.13±1.64	71.20±1.81	70.57±1.66	70.54±1.79	95.16±0.42	73.80±2.09	75.40±1.57
	Transformer	76.36±1.21	76.53±1.25	76.23±0.98	76.05±1.16	*96.65±0.11*	80.70±0.63	*80.42±0.89*
	Medformer	*76.44±0.64*	*76.61±1.13*	*76.63±1.36*	*76.25±0.65*	95.44±0.26	*81.12±1.60*	80.41±0.94
	TeCh	**80.60**±0.30	**80.29**±0.24	**80.36**±0.32	**80.23**±0.24	**97.67**±0.10	**86.18**±0.31	**84.22**±0.25
UCI-HAR (6-Classes)	**Autoformer**	41.86±2.46	49.62±11.48	44.30±2.55	32.69±2.60	83.72±2.53	58.56±4.67	51.79±4.38
	FEDformer	76.89±9.59	75.66±9.46	77.56±9.79	75.03±9.77	95.16±4.66	83.28±8.14	80.37±8.89
	Informer	88.33±1.26	88.28±1.20	88.47±1.20	88.20±1.29	98.36±0.14	94.20±0.33	89.81±0.77
	iTransformer	*92.41±0.63*	*92.24±0.63*	*92.33±0.67*	*92.39±0.64*	*99.07±0.07*	96.01±0.39	*93.74±0.47*
	MTST	90.99±0.84	90.96±0.79	90.92±0.85	90.83±0.88	98.21±0.11	*96.14±0.59*	93.17±0.51
	Nonformer	91.04±0.58	90.98±0.60	91.14±0.56	91.01±0.60	99.02±0.09	96.07±0.37	93.37±0.47
	PatchTST	87.67±0.39	88.37±0.43	87.97±0.37	88.02±0.38	98.50±0.09	93.86±0.40	90.56±0.34
	Reformer	88.70±1.14	88.82±1.03	88.82±1.13	88.59±1.19	98.68±0.26	94.60±1.07	91.53±0.80
	Transformer	89.36±1.74	89.33±1.70	89.49±1.69	89.33±1.75	98.87±0.23	95.58±0.68	91.83±1.13
	Medformer	89.62±0.81	89.70±0.18	89.80±0.14	89.62±0.81	98.11±0.06	94.80±0.72	91.61±0.45
	TeCh	**94.15**±0.96	**94.27**±0.96	**94.30**±0.97	**94.26**±0.98	**99.32**±0.05	**96.74**±0.18	**95.02**±0.35

5.3 Ablation Study

Model Efficiency Analysis. Since CoTAR introduces only **Linear** complexity compared to the **Quadratic** complexity of attention, our Tech achieves higher performance with significantly lower resource consumption, as in Figure 4 ***(a)***. Compared to Medformer, Tech delivers 8% better accuracy while using just 33% of the memory usage and 20% of the inference time.



(a) Efficiency analysis on APAVA

(b) Robust Test on PTB

Figure 4: ***(a)***: Efficiency and Effectiveness analysis of **TeCh** and other baselines on APAVA dataset with batch size $B = 128$. '#' stands for 'former' to save space. ***(b)***: Robustness of attention and **CoTAR** to noise when using Channel or Temporal embedding. We consistently increase the intensity β (the standard deviation) of Gaussian random noise from 0.0 to 20.0 on the last channel of the PTB dataset. F1-Score is used to quantify the change.

Robustness Analysis. To test the robustness of attention and CoTAR, we introduce noise progressively during training by adding perturbations to the last channel of the PTB dataset. This formulated as $\hat{X}_{:,C} = X_{:,C} + \beta \cdot \text{noise}$, where $X_{:,C}, \hat{X}_{:,C} \in \mathbb{R}^{1\times T}$ is the last channel, $noise \in \mathbb{R}^{1\times T}$ is Gaussian noise with mean 0 and standard deviation 1, $\beta \in \mathbb{R}^1$ controls the noise intensity. Then, the processed sample $\hat{X}_{:,C}$ is embedded into Channel embedding (Liu et al., 2024) or Temporal embedding (Wang et al., 2024b). Figure 4 ***(b)*** reveals that attention is highly sensitive to noise. This is because attention is a decentralized structure, which means each channel can be directly influenced by the corrupted, noisy channel. In contrast, our CoTAR employed a centralized strategy, which prevents the noisy channel from directly interfering with others, therefore enhancing the robustness to noise. Meanwhile, compared to Temporal embedding, which is a more common practice in previous

Table 4: Ablation result of the proposed ***Dual Tokenization*** strategy. We include a general Human Activity dataset, UCI-HAR, to test its generalizability. *(i)* w/o: No tokenization is performed and directly uses the raw series as input-without representation learning, a single linear projection as classifier. *(ii)* Temporal: Only Temporal embedding is used. *(iii)* Channel: Only Channel embedding is used. *(iv)* Dual: Both Temporal and Channel embedding are used. The best is **Bolded**

	ADFTD		**APAVA**		**TDBrain**		**PTB**		**UCI-HAR**	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
w/o	33.79±0.64	32.67±0.53	50.68±0.86	50.13±0.88	53.79±1.21	53.77±1.20	72.62±1.30	64.84±2.05	54.22±0.47	51.72±0.47
Temporal	53.78±0.72	**49.10±1.60**	55.93±5.06	53.71±5.56	**93.21±0.61**	**93.20±0.61**	74.74±0.55	62.90±1.15	91.56±0.63	91.52±0.62
Channel	47.06±1.35	32.92±0.90	75.68±1.80	73.54±2.49	67.58±1.04	67.54±1.06	**85.96±2.52**	**81.97±4.07**	92.98±0.44	93.00±0.48
Both	**54.54±0.70**	48.84±1.72	**86.86±1.09**	**86.30±1.06**	89.79±0.96	89.77±0.97	84.15±2.06	79.11±3.43	**94.15±0.96**	**94.26±0.98**

Table 5: Ablation result of the proposed ‘Core Token Aggregate-Redistribut’ (CoTAR) module. *(i)* w/o: No Token interaction is performed, which means directly removing the CoTAR module. *(ii)* Attention: Replacing CoTAR with the Attention module. *(iii)* CoTAR: baseline with the CoTAR module. The best is **Bolded**.

	ADFTD		**APAVA**		**TDBrain**		**PTB**		**UCI-HAR**	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
w/o	53.32±0.67	47.26±0.53	83.31±0.95	81.99±1.18	92.69±0.75	92.67±0.76	85.28±2.32	80.82±3.69	92.40±0.19	92.55±0.21
Attention	52.77±1.00	48.65±1.22	83.42±1.60	82.09±0.28	90.40±2.18	90.35±2.23	85.74±1.45	81.93±2.22	93.13±0.59	93.21±0.60
CoTAR	**54.54±0.70**	**48.84±1.72**	**86.86±1.09**	**86.30±1.06**	**93.21±0.61**	**93.20±0.61**	**85.96±2.52**	**81.97±4.07**	**94.15±0.96**	**94.26±0.98**

work (Mobin et al., 2025; Wang et al., 2024b), Channel embedding delivers higher robustness and classification performance. This aligns with general time series analysis, where Channel embedding is more suitable for modeling channel dependencies, as it can better preserve the unique context of each channel, even when noise is entangled (Liu et al., 2024; Wang et al., 2024d).

Ablation Study on ‘Adaptive Dual Tokenization’. The results in Table 4 demonstrate the effectiveness of the proposed Adaptive Dual Tokenization design. When skipping the representation learning phase (the ***w/o*** setting), the performance significantly deteriorates across all datasets, highlighting the necessity of structured token embedding. Temporal tokenization excels on TDBrain, while Channel tokenization excels on PTB. And combining both yields an 11% improvement of Accuracy and a 13% improvement of F1-Score on APAVA. Moreover, Dual tokenization also excels on the UCI-HAR dataset, a well-known benchmark for Human Activity (HAR) tasks. Since HAR tasks involve multi-sensor channels and fine-grained activity classes, the significant gains of Dual Tokenization indicate that by simultaneously capturing both patterns, Tech can generalize to noisy, high-variation, multi-channel time series. These findings confirm that the Adaptive Dual Tokenization strategy enables Tech to better align with the unique characteristics of each dataset, providing more versatile modeling of Temporal dependencies or Channel dependencies, or both.

Ablation Study on ‘Core Token Aggregate-Redistribute’. Table 5 provides a comprehensive ablation study validating the effectiveness of the proposed Core Token Aggregate-Redistribute (CoTAR) module, which yields consistent performance gains across all five datasets and both metrics. Moreover, CoTAR also demonstrates competitive or lower standard deviations, indicating higher robustness. These results suggest that CoTAR not only captures richer inter-token dependencies through core-token centric redistribution but also leads to more stable and generalizable representations, thereby justifying its architectural necessity.

6 Conclusion

Existing Transformer models suffer from the mismatch between the centralized nature of medical time series (MedTS) and the decentralized structure of the attention module. This work proposes the Core Token Aggregation-Redistribution (**CoTAR**) module, which models inter-token relationships in a centralized way using a core token as a proxy, to replace attention seamlessly. Beyond being more effective in channel dependencies modeling, it also reduces complexity from quadratic to linear. Based on CoTAR, our **Tech** framework can adaptively capture temporal dependencies or channel dependencies, or both, and achieves superior performance and efficiency on three EEG and two ECG datasets. This work demonstrates the effectiveness of introducing domain-specific inductive biases into deep learning architectures for MedTS analysis and paves the way for more effective and scalable solutions.

REFERENCES

Isabela Albuquerque, João Monteiro, Olivier Rosanne, Abhishek Tiwari, Jean-François Gagnon, and Tiago H Falk. Cross-subject statistical shift estimation for generalized electroencephalography-based mental workload assessment. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 3647–3653. IEEE, 2019.

Majd AlGhatrif and Joseph Lindsay. A brief review: history to understand fundamentals of electrocardiography. *Journal of Community Hospital Internal Medicine Perspectives*, 2(1):14383, 2012a.

Majd AlGhatrif and Joseph Lindsay. A brief review: history to understand fundamentals of electrocardiography. *Journal of community hospital internal medicine perspectives*, 2(1):14383, 2012b.

Salma Alhagry, Aly Aly Fahmy, and Reda A El-Khoribi. Emotion recognition based on eeg using lstm recurrent neural network. *International Journal of Advanced Computer Science and Applications*, 8(10), 2017.

Hamdi Altaheri, Ghulam Muhammad, Mansour Alsulaiman, Syed Umar Amin, Ghadir Ali Altuwaijri, Wadood Abdul, Mohamed A Bencherif, and Mohammed Faisal. Deep learning techniques for classification of electroencephalogram (eeg) motor imagery (mi) signals: A review. *Neural Computing and Applications*, 35(20):14681–14722, 2023.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3, 2013.

Aniqa Arif, Yihe Wang, Rui Yin, Xiang Zhang, and Ahmed Helmy. Ef-net: Mental state recognition by analyzing multimodal eeg-fnirs via cnn. *Sensors*, 24(6):1889, 2024.

Yara Badr, Usman Tariq, Fares Al-Shargie, Fabio Babiloni, Fadwa Al Mughairbi, and Hasan Al-Nashash. A review on evaluating mental stress by deep learning using eeg signals. *Neural Computing and Applications*, pp. 1–26, 2024.

György Buzsáki. *Rhythms of the Brain*. Oxford University Press, 2006.

Stanislas Dehaene and Jean-Pierre Changeux. The global neuronal workspace model of conscious access: from neuronal architectures to clinical applications. *Neuron*, 69(2):200–213, 2011.

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.

Wei Fan, Jingru Fei, Dingyu Guo, Kun Yi, Xiaozhuang Song, Haolong Xiang, Hangting Ye, and Min Li. Towards multi-resolution spatiotemporal graph learning for medical time series classification. In *Proceedings of the ACM on Web Conference 2025*, pp. 5054–5064, 2025.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000a.

Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000b.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. *arXiv preprint arXiv:1902.09113*, 2019.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *arXiv preprint arXiv:2404.14197*, 2024. URL `https://arxiv.org/abs/2404.14197`.

Mahboobeh Jafari, Afshin Shoeibi, Marjane Khodatars, Sara Bagherzadeh, Ahmad Shalbaf, David López García, Juan M Gorriz, and U Rajendra Acharya. Emotion recognition in eeg signals using deep learning methods: A review. *Computers in Biology and Medicine*, pp. 107450, 2023.

Yingying Jiao, Yini Deng, Yun Luo, and Bao-Liang Lu. Driver sleepiness detection from eeg and eog signals using gan and lstm networks. *Neurocomputing*, 408:100–111, 2020.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. *Artificial Intelligence Review*, 58(7):1–95, 2025.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2019.

Prabhat Kumar and S Suresh. Flaap: An open human activity recognition (har) dataset for learning and finding the associated activity patterns. *Procedia Computer Science*, 212:64–73, 2022.

Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance. Eegnet: A compact convolutional network for eeg-based brain-computer interfaces. *CoRR*, abs/1611.08024, 2016.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, 15(5):056013, 2018.

Huayu Li, Ana S Carreon-Rascon, Xiwen Chen, Geng Yuan, and Ao Li. Mts-lof: medical time-series representation learning via occlusion-invariant features. *IEEE Journal of Biomedical and Health Informatics*, 2024.

Xinwen Liu, Huan Wang, Zongjin Li, and Lang Qin. Deep learning in ecg diagnosis: A review. *Knowledge-Based Systems*, 227:107187, 2021.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *NeurIPS*, 35:9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *ICLR*, 2024.

Jiecheng Lu, Xu Han, Yan Sun, and Shihao Yang. CATS: Enhancing multivariate time series forecasting by constructing auxiliary time series as exogenous variables. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=11DAGDe0UR`.

PW Macfarlane, B Devine, and E Clark. The university of glasgow (uni-g) ecg analysis program. In *Computers in Cardiology, 2005*, pp. 451–454. IEEE, 2005.

Andreas Miltiadous, Emmanouil Gionanidis, Katerina D Tzimourta, Nikolaos Giannakeas, and Alexandros T Tzallas. Dice-net: a novel convolution-transformer architecture for alzheimer detection in eeg signals. *IEEE Access*, 2023a.

Andreas Miltiadous, Katerina D Tzimourta, Theodora Afrantou, Panagiotis Ioannidis, Nikolaos Grigoriadis, Dimitrios G Tsalikakis, Pantelis Angelidis, Markos G Tsipouras, Euripidis Glavas, Nikolaos Giannakeas, et al. A dataset of scalp eeg recordings of alzheimer's disease, frontotemporal dementia and healthy subjects from routine eeg. *Data*, 8(6):95, 2023b.

Md Kamrujjaman Mobin, Md Saiful Islam, Sadik Al Barid, and Md Masum. Cardioformer: Advancing ai in ecg analysis with multi-granularity patching and resnet. *arXiv preprint arXiv:2505.05538*, 2025.

Fatma Murat, Ozal Yildirim, Muhammed Talo, Ulas Baran Baloglu, Yakup Demir, and U Rajendra Acharya. Application of deep learning techniques for heartbeats detection using ecg signals-analysis and review. *Computers in biology and medicine*, 120:103726, 2020.

Elon Musk et al. An integrated brain-machine interface platform with thousands of channels. *Journal of medical Internet research*, 21(10):e16194, 2019.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *ICLR*, 2023.

Ernst Niedermeyer and F Lopes da Silva. *Electroencephalography: Basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005a.

Ernst Niedermeyer and FH Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005b.

PhysioToolkit PhysioBank. Physionet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. *arXiv preprint arXiv:2412.10859*, 2024.

José J Rieta and Raúl Alcaraz. The genesis of the electrocardiogram (ecg). *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–15, 1999a.

José J Rieta and Raúl Alcaraz. The genesis of the electrocardiogram (ecg). In *Wiley Encyclopedia of Electrical and Electronics Engineering*. Wiley, 1999b.

Lawrence G Roberts and Barry D Wessler. Computer network development to achieve resource sharing. In *Proceedings of the May 5-7, 1970, spring joint computer conference*, pp. 543–549, 1970.

Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of Neural Engineering*, 16(5):051001, aug 2019. doi: 10.1088/1741-2552/ab260c. URL `https://dx.doi.org/10.1088/1741-2552/ab260c`.

Neil Schaul. The fundamental neural mechanisms of electroencephalography. *Electroencephalography and clinical neurophysiology*, 106(2):101–107, 1998.

Michael Scherg, Patrick Berg, Nobukazu Nakasato, and Sándor Beniczky. Taking the eeg back into the brain: the power of multiple discrete sources. *Frontiers in neurology*, 10:855, 2019a.

Michael Scherg, Patrick Berg, Nobukazu Nakasato, and Sándor Beniczky. Taking the eeg back into the brain: the power of multiple discrete sources. *Frontiers in Neurology*, 10:855, 2019b.

Anil K. Seth, Adam B. Barrett, and Lionel Barnett. Granger causality analysis in neuroscience and neuroimaging. *Journal of Neuroscience*, 35(8):3293–3297, 2015.

Olaf Sporns. Networks of the brain. *MIT Press*, 2010.

Cornelis J Stam. Nonlinear dynamical analysis of eeg and meg: review of an emerging field. *Clinical neurophysiology*, 116(10):2266–2301, 2005.

Siyi Tang, Jared Dunnmon, Khaled Kamal Saab, Xuan Zhang, Qianying Huang, Florian Dubost, Daniel Rubin, and Christopher Lee-Messer. Self-supervised graph neural networks for improved electroencephalographic seizure analysis. In *ICLR*, 2021.

Thomas W. Valente, Katie Coronges, Cynthia Lakon, and Elizabeth Costenbader. How correlated are network centrality measures? *Connections*, 28(1):16–26, 2008.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

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.

Yihe Wang, Yu Han, Haishuai Wang, and Xiang Zhang. Contrast everything: A hierarchical contrastive framework for medical time-series. *NeurIPS*, 36, 2024a.

Yihe Wang, Nan Huang, Taida Li, Yujun Yan, and Xiang Zhang. Medformer: A multi-granularity patching transformer for medical time-series classification. In *NeurIPS*, 2024b. URL https://openreview.net/forum?id=jfkid2HwNr.

Yihe Wang, Taida Li, Yujun Yan, Wenzhan Song, and Xiang Zhang. How to evaluate your medical time series classification? *arXiv preprint arXiv:2410.03057*, 2024c.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024d. URL https://arxiv.org/abs/2407.13278.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024e. URL https://openreview.net/forum?id=INAeUQ04lT.

Zekai Wang, Stavros Stavrakis, and Bing Yao. Hierarchical deep learning with generative adversarial network for automatic cardiac diagnosis from ecg signals. *Computers in Biology and Medicine*, 155:106641, 2023.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *NeurIPS*, 34:22419–22430, 2021.

Jingyun Xiao, Ran Liu, and Eva L Dyer. GAFormer: Enhancing timeseries transformers through group-aware embeddings. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=c56TWtYp0W.

Qiao Xiao, Khuan Lee, Siti Aisah Mokhtar, Iskasymar Ismail, Ahmad Luqman bin Md Pauzi, Qiuxia Zhang, and Poh Ying Lim. Deep learning-based ecg arrhythmia classification: A systematic review. *Applied Sciences*, 13(8):4964, 2023.

Dezhen Xiong, Daohui Zhang, Xingang Zhao, and Yiwen Zhao. Deep learning for emg-based human-machine interaction: A review. *IEEE/CAA Journal of Automatica Sinica*, 8(3):512–533, 2021.

Guoqi Yu, Jing Zou, Xiaowei Hu, Angelica I Aviles-Rivero, Jing Qin, and Shujun Wang. Revitalizing multivariate time series forecasting: Learnable decomposition with inter-series dependencies and intra-series variations modeling. In *ICML*, 2024. URL https://openreview.net/forum?id=87CYNyCGOo.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *NeurIPS*, 35:3988–4003, 2022.

Yitian Zhang, Liheng Ma, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Multi-resolution time-series transformer for long-term forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 4222–4230. PMLR, 2024.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, pp. 27268–27286. PMLR, 2022.

A Data Augmentation Banks

In the embedding stage, we apply data augmentation to the input time series. We utilize a bank of data augmentation techniques to enhance the model's robustness and generalization. During the forward pass in training, each time series will pick one augmentation from available augmentation options with equal probability. The data augmentation methods include temporal flipping, channel shuffling, temporal masking, frequency masking, jittering, and dropout, and can be further expanded to include more choices. We provide a detailed description of each technique below.

Temporal Flippling We reverse the MedTS data along the temporal dimension. The probability of applying this augmentation is controlled by a parameter *prob*, with a default value of 0.5.

Channel Shuffling We randomly shuffle the order of MedTS channels. The probability of applying channel shuffling is controlled by the parameter *prob*, also set by default to 0.5.

temporal masking We randomly mask some timestamps across all channels. The proportion of timestamps masked is controlled by the parameter *ratio*, with a default value of 0.1.

Frequency Masking First introduced in (Zhang et al., 2022) for contrastive learning, this method involves converting the MedTS data into the frequency domain, randomly masking some frequency bands, and then converting it back. The proportion of frequency bands masked is controlled by the parameter *ratio*, with a default value of 0.1.

Jittering Random noise, ranging from 0 to 1, is added to the raw data. The intensity of the noise is adjusted by the parameter *scale*, which is set by default to 0.1.

Dropout Similar to the dropout layer in neural networks, this method randomly drops some values. The proportion of values dropped is controlled by the parameter *ratio*, with a default setting of 0.1.

B Data Preprocessing

We obtain all the well-preprocessed datasets from **Medformer** (Wang et al., 2024b)(`https://github.com/DL4mHealth/Medformer`).

B.1 APAVA Preprocessing

The **A**lzheimer's **P**atients' Relatives **A**ssociation of **Va**lladolid (APAVA) dataset[1], referenced in the paper (Escudero et al., 2006), is a public EEG time series dataset with 2 classes and 23 subjects, including 12 Alzheimer's disease patients and 11 healthy control subjects. On average, each subject has 30.0 ± 12.5 trials, with each trial being a 5-second time sequence consisting of 1280 timestamps across 16 channels. Before further preprocessing, each trial is scaled using the standard scaler. Subsequently, we segment each trial into 9 half-overlapping samples, where each sample is a 1-second time sequence comprising 256 timestamps. This process results in 5,967 samples. Each sample has a subject ID to indicate its originating subject. For the training, validation, and test set splits, we employ the subject-independent setup. Samples with subject IDs $\{15,16,19,20\}$ and $\{1,2,17,18\}$ are assigned to the validation and test sets, respectively. The remaining samples are allocated to the training set.

B.2 TDBrain Preprocessing

The TDBrain dataset[2], referenced in the paper (van Dijk et al., 2022), is a large permission-accessible EEG time series dataset recording brain activities of 1274 subjects with 33 channels. Each subject has two trials: one under eye open and one under eye closed setup. The dataset includes a total of 60 labels, with each subject potentially having multiple labels indicating multiple diseases simultaneously. In this paper, we utilize a subset of this dataset containing 25 subjects with Parkinson's disease and 25 healthy controls, all under the eye-closed task condition. Each eye-closed trial is segmented into non-overlapping 1-second samples with 256 timestamps, and any samples shorter

[1] `https://osf.io/jbysn/`

[2] `https://brainclinics.com/resources/`

than 1 second are discarded. This process results in 6,240 samples. Each sample is assigned a subject ID to indicate its originating subject. For the training, validation, and test set splits, we employ the subject-independent setup. Samples with subject IDs $\{18,19,20,21,46,47,48,49\}$ are assigned to the validation set, while samples with subject IDs $\{22,23,24,25,50,51,52,53\}$ are assigned to the test set. The remaining samples are allocated to the training set.

B.3 ADFTD Preprocessing

The **A**lzheimer's **D**isease and **F**ron**T**otemporal **D**ementia (ADFTD) dataset[3], referenced in the papers (Miltiadous et al., 2023b;a), is a public EEG time series dataset with 3 classes, including 36 Alzheimer's disease (AD) patients, 23 Frontotemporal Dementia (FTD) patients, and 29 healthy control (HC) subjects. The dataset has 19 channels, and the raw sampling rate is 500Hz. Each subject has a trial, with trial durations of approximately 13.5 minutes for AD subjects (min=5.1, max=21.3), 12 minutes for FD subjects (min=7.9, max=16.9), and 13.8 minutes for HC subjects (min=12.5, max=16.5). A bandpass filter between 0.5-45Hz is applied to each trial. We downsample each trial to 256Hz and segment them into non-overlapping 1-second samples with 256 timestamps, discarding any samples shorter than 1 second. This process results in 69,752 samples. For the training, validation, and test set splits, we employ the subject-independent setup by allocating 60%, 20%, and 20% of total subjects with their corresponding samples into the training, validation, and test sets, respectively.

B.4 PTB Preprocessing

The PTB dataset[4], referenced in the paper (PhysioBank, 2000), is a public ECG time series recording from 290 subjects, with 15 channels and a total of 8 labels representing 7 heart diseases and 1 health control. The raw sampling rate is 1000Hz. For this paper, we utilize a subset of 198 subjects, including patients with Myocardial infarction and healthy control subjects. We first downsample the sampling frequency to 250Hz and normalize the ECG signals using standard scalers. Subsequently, we process the data into single heartbeats through several steps. We identify the R-Peak intervals across all channels and remove any outliers. Each heartbeat is then sampled from its R-Peak position, and we ensure all samples have the same length by applying zero padding to shorter samples, with the maximum duration across all channels serving as the reference. This process results in 64,356 samples. For the training, validation, and test set splits, we employ the subject-independent setup. Specifically, we allocate 60%, 20%, and 20% of the total subjects, along with their corresponding samples, into the training, validation, and test sets, respectively.

B.5 PTB-XL Preprocessing

The PTB-XL dataset[5], referenced in the paper (Wagner et al., 2020), is a large public ECG time series dataset recorded from 18,869 subjects, with 12 channels and 5 labels representing 4 heart diseases and 1 healthy control category. Each subject may have one or more trials. To ensure consistency, we discard subjects with varying diagnosis results across different trials, resulting in 17,596 subjects remaining. The raw trials consist of 10-second time intervals, with sampling frequencies of 100Hz and 500Hz versions. For our paper, we utilize the 500Hz version, then we downsample to 250Hz and normalize using standard scalers. Subsequently, each trial is segmented into non-overlapping 1-second samples with 250 timestamps, discarding any samples shorter than 1 second. This process results in 191,400 samples. For the training, validation, and test set splits, we employ the subject-independent setup. Specifically, we allocate 60%, 20%, and 20% of the total subjects, along with their corresponding samples, into the training, validation, and test sets, respectively.

[3] `https://openneuro.org/datasets/ds004504/versions/1.0.6`

[4] `https://physionet.org/content/ptbdb/1.0.0/`

[5] `https://physionet.org/content/ptb-xl/1.0.3/`

C Implementation Details

C.1 Implementation Details of All Baselines

We implement all the baselines based on the Medformer (Wang et al., 2024b), which integrates all methods under the same framework and training techniques to ensure a strict fair comparison. The compared 10 baseline time series transformer methods are Autoformer (Wu et al., 2021), FEDformer (Zhou et al., 2022), Informer (Zhou et al., 2021), iTransformer (Liu et al., 2024), MTST (Zhang et al., 2024), Nonformer (Liu et al., 2022), PatchTST (Nie et al., 2023), Reformer (Kitaev et al., 2019), Medformer (Wang et al., 2024b), and vanilla Transformer (Vaswani et al., 2017).

For Medformer, we directly reproduced its result using their official implementations. For all other methods, we employ 6 layers for the encoder, with the self-attention dimension D set to **128** and the hidden dimension of the feed-forward networks set to **256**. The optimizer used is Adam, with a learning rate of 1e-4. The batch size is set to $\{32,32,128,128,128\}$ for the datasets APAVA, TDBrain, ADFD, PTB, and PTB-XL, respectively. Training is conducted for 100 epochs, with early stopping triggered after 10 epochs without improvement in the F1-Score on the validation set. We save the model with the best F1 score on the validation set and evaluate it on the test set. We employ six evaluation metrics: Accuracy, Precision (macro-averaged), Recall (macro-averaged), F1-Score (macro-averaged), AUROC (macro-averaged), and AUPRC (macro-averaged). Each experiment is run with 5 random seeds and fixed training, validation, and test sets to compute the average results and standard deviations.

Autoformer Autoformer (Wu et al., 2021) employs an auto-correlation mechanism to replace self-attention for time series forecasting. Additionally, they use a time series decomposition block to separate the time series into trend-cyclical and seasonal components for improved learning. The raw source code is available at `https://github.com/thuml/Autoformer`.

FEDformer FEDformer (Zhou et al., 2022) leverages frequency domain information using the Fourier transform. They introduce frequency-enhanced blocks and frequency-enhanced attention, which are computed in the frequency domain. A novel time series decomposition method replaces the layer norm module in the transformer architecture to improve learning. The raw code is available at `https://github.com/MAZiqing/FEDformer`.

Informer Informer (Zhou et al., 2021) is the first paper to employ a one-forward procedure instead of an autoregressive method in time series forecasting tasks. They introduce ProbSparse self-attention to reduce complexity and memory usage. The raw code is available at `https://github.com/zhouhaoyi/Informer2020`.

iTransformer iTransformer (Liu et al., 2024) questions the conventional approach of embedding attention tokens in time series forecasting tasks and proposes an inverted approach by embedding the whole series of channels into a token. They also invert the dimension of other transformer modules, such as the layer norm and feed-forward networks. The raw code is available at `https://github.com/thuml/iTransformer`.

MTST MTST (Zhang et al., 2024) uses the same token embedding method as Crossformer and PatchTST. It highlights the importance of different patching lengths in forecasting tasks and designs a method that can take different sizes of patch tokens as input simultaneously. The raw code is available at `https://github.com/networkslab/MTST`.

Nonformer Nonformer (Liu et al., 2022) analyzes the impact of non-stationarity in time series forecasting tasks and its significant effect on results. They design a de-stationary attention module and incorporate normalization and denormalization steps before and after training to alleviate the over-stationarization problem. The raw code is available at `https://github.com/thuml/Nonstationary_Transformers`.

PatchTST PatchTST (Nie et al., 2023) embeds a sequence of single-channel timestamps as a patch token to replace the attention token used in the vanilla transformer. This approach enlarges the receptive field and enhances forecasting ability. The raw code is available at `https://github.com/yuqinie98/PatchTST`.

Reformer Reformer (Kitaev et al., 2019) replaces dot-product attention with locality-sensitive hashing. They also use a reversible residual layer instead of standard residuals. The raw code is available at `https://github.com/lucidrains/reformer-pytorch`.

Transformer Transformer (Vaswani et al., 2017), commonly known as the vanilla transformer, is introduced in the well-known paper "Attention is All You Need." It can also be applied to time series by embedding each timestamp of all channels as an attention token. The PyTorch version of the code is available at `https://github.com/jadore801120/attention-is-all-you-need-pytorch`.

Medformer Medformer (Wang et al., 2024b) is a multi-granularity patching transformer specifically designed for medical time-series classification. It constructs patch tokens at multiple temporal resolutions to capture both fine-grained local dependencies and long-range contextual patterns. This design improves the model's ability to handle heterogeneous temporal dynamics in physiological signals. The raw code is available at `https://github.com/DL4mHealth/Medformer`.

C.2 Implementation Details of our TeCh

Our Tech is trained with a unified batch size ($B = 128$) and dimension of core token $D_c = \frac{1}{4}D$ across all datasets. The selection of other critical hyperparameters is listed in Table 6. We present the pseudo-code of the proposed CoTAR module in Algorithm 1.

Table 6: Critical hyperparameters for **TeCh** by dataset. We listed the model dimension ($\boldsymbol{D}$), patch length of Temporal embedding ($\boldsymbol{L}$), number of temporal encoders ($\boldsymbol{M}$), number of channel encoders ($\boldsymbol{N}$), and learning rate (**lr**).

Dataset	$\boldsymbol{D}$	$\boldsymbol{L}$	$\boldsymbol{M}$	$\boldsymbol{N}$	**lr**
ADFTD	128	1	6	6	3e−5
APAVA	256	1	6	6	1e−4
TDBRAIN	128	6	6	0	1e−4
PTB	256	1	0	3	1e−4
PTB-XL	128	8	5	0	1e−4
UCI-HAR	256	12	5	6	1e−4
FLAAP	512	1	6	0	1e−4

Algorithm 1 Pseudo-Code of Core Token Aggregation-Redistribution (CoTAR).

```
Require: Input tensor: $O \in \mathbb{R}^{S \times D}$.
Require: Parameters: Linear mapping layers Lin1, Lin2, Lin3, Lin4, dimension of core token $D_c$.
Require: Definition: $\text{Lin1} : \mathbb{R}^D \to \mathbb{R}^D,\ \text{Lin2} : \mathbb{R}^D \to \mathbb{R}^{D_c}$,
Require: Definition: $\text{Lin3} : \mathbb{R}^{D+D_c} \to \mathbb{R}^D,\ \text{Lin4} : \mathbb{R}^D \to \mathbb{R}^D$.
  $\tilde{O} \leftarrow \text{Lin2}(\text{GELU}(\text{Lin1}(O))),\ \tilde{O} \in \mathbb{R}^{S \times D_c}$,   ▷ First MLP to obtain core representation
  $O_w \leftarrow \text{Softmax}(\tilde{O}, \text{dim} = 0),\ O_w \in \mathbb{R}^{S \times D_c}$,   ▷ Attention-like weights across channels
  $\tilde{C_o}^d = \sum_{i=1}^{S} \tilde{O}^{i,d} \odot O_w^{i,d},\ \tilde{C}_o \in \mathbb{R}^{D_c}$,   ▷ Weighted sum across channels to get core token
  $C_o \leftarrow \text{Repeat}(\tilde{C}_o, N \text{ times}),\ C_o \in \mathbb{R}^{S \times D_c}$,   ▷ Repeat to align the channel dimension of input
  $O_{Co} \leftarrow [O; C_o],\ O_{Co} \in \mathbb{R}^{S \times (D+D_c)}$,   ▷ Concatenate along last dimension
  $A \leftarrow \text{Lin4}(\text{GELU}(\text{Lin3}(O_{Co}))),\ A \in \mathbb{R}^{S \times D}$.   ▷ Fuse information through second MLP
  Return $A \in \mathbb{R}^{S \times D}$
```

C.3 Full Ablation Results

To save space in the main text, we only present the ablation result of five representative datasets. We provide the full results on all datasets in Table 7 and Table 8.

Table 7: Full ablation result of the proposed ***Dual Tokenization*** strategy. *(i)* w/o: No tokenization is performed and directly uses the raw series as input-without representation learning, a single linear projection as classifier. *(ii)* Temporal: Only Temporal embedding. *(iii)* Channel: Only Channel embedding. *(iv)* Dual: Both Temporal and Channel. The best is **Bolded**

	ADFTD		APAVA		TDBrain		PTB		PTB-XL		FLAAP		UCI-HAR	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
w/o	33.79±0.64	32.67±0.53	50.68±0.86	50.13±0.88	53.79±1.21	53.77±1.20	72.62±1.30	64.84±2.05	30.95±0.13	20.61±0.51	28.54±2.34	25.08±1.33	54.22±0.47	51.72±0.47
Temporal	53.78±0.72	**49.10±1.60**	55.93±5.06	53.71±5.56	**93.21±0.61**	**93.20±0.61**	74.74±0.55	62.90±1.15	**73.53±0.07**	**62.44±0.27**	**80.60±0.30**	**80.23±0.24**	91.56±0.63	91.52±0.62
Channel	47.06±1.35	32.92±0.90	75.68±1.80	73.54±2.49	67.58±1.04	67.54±1.06	**85.96±2.52**	**81.97±4.07**	69.18±0.21	54.76±0.47	77.48±0.13	77.06±0.17	92.98±0.44	93.00±0.48
Both	**54.54±0.70**	48.84±1.72	**86.86±1.09**	**86.30±1.06**	89.79±0.96	89.77±0.97	84.15±2.06	79.11±3.43	73.15±0.09	62.13±0.16	78.03±0.31	77.86±0.30	**94.15±0.96**	**94.26±0.98**

Table 8: Full ablation result of the proposed 'Core Token Aggregate-Redistribut' (CoTAR) module. *(i)* w/o: No Token interaction is performed, which means directly removing the CoTAR module. *(ii)* Attention: Replacing CoTAR with the Attention module. *(iii)* CoTAR: baseline with the CoTAR module. The best is **Bolded**.

	ADFTD		APAVA		TDBrain		PTB		PTB-XL		FLAAP		UCI-HAR	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
w/o	53.32±0.67	47.26±0.53	83.31±0.95	81.99±1.18	92.69±0.75	92.67±0.76	85.28±2.32	80.82±3.69	72.25±0.38	59.48±0.59	74.48±0.46	74.00±0.53	92.40±0.19	92.55±0.21
Attention	52.77±1.00	48.65±1.22	83.42±1.60	82.09±0.28	90.40±2.18	90.35±2.23	85.74±1.45	81.93±2.22	72.01±0.22	60.96±0.21	77.16±0.76	76.87±0.77	93.13±0.59	93.21±0.60
CoTAR	**54.54±0.70**	**48.84±1.72**	**86.86±1.09**	**86.30±1.06**	**93.21±0.61**	**93.20±0.61**	**85.96±2.52**	**81.97±4.07**	**73.53±0.07**	**62.44±0.27**	**80.60±0.30**	**80.23±0.24**	**94.15±0.96**	**94.26±0.98**

C.4 Comparison with Cutting-edge Temporal Models

To position TeCh within the broader landscape beyond current MedTS classifiers and relative to general time-series backbones exhibiting partial similarity, we present a comparative analysis that maps overlaps and distinctions between recent backbones and TeCh.

(i) *Methods employed a dual-dependencies modeling.* We select two representative works: GAFormer (ICLR24) (Xiao et al., 2024) and Leddam (ICML24) Yu et al. (2024). GAFormer enhances token representations with group-aware embeddings for series clustering; Leddam introduces learnable decomposition into inter-series dependencies and intra-series variations; TeCh utilizes Adaptive Dual Tokenization (Temporal/Channel/Dual). Though all capture dual dependencies (temporal and inter-channel), GAFormer and Leddam target forecasting and are Transformer-based, thus decentralizing inter-channel interactions via attention, whereas TeCh uses a centralized CoTAR to better align with MedTS' biologically centralized sources (brain/heart). TeCh focuses on MedTS classification with physiological interpretability and linear complexity, while GAFormer/Leddam primarily focus on time series forecasting with quadratic attention costs. Consequently, GAFormer and Leddam are well-suited for broad forecasting scenarios; TeCh's centralized communication is more appropriate for MedTS channel dependencies. This is validated in our comparative result in Table 9. (Since there is no official implementation of GAFormer, and the information in the paper is not enough to reproduce, we take Leddam as baseline for its high reproducibility.)

(ii) *Methods employed global or auxiliary tokens.* We select two representative works: CATS (ICML24) (Lu et al., 2024) and TimeXer (NIPS24) (Wang et al., 2024e). They both employ global/auxiliary tokens that are parameter-initialized and learned jointly with the model, remaining largely input-agnostic while aggregating/redistributing information (often tied to exogenous-variable modeling). In contrast, TeCh's core token is generated adaptively from each input (subject) via CoTAR, making it data-conditional and thus better suited to MedTS heterogeneity where the "central source" differs across individuals. Moreover, TimeXer and CATS still operate within decentralized quadratic attention, while TeCh enforces centralized communication and achieves linear complexity. Additionally, TimeXer focuses on forecasting with exogenous variables and CATS constructs auxiliary time series to aid prediction, whereas TeCh targets MedTS classification with physiologically aligned central coordination. This dynamic, per-input core token mitigates the risk of poorer generalization from pre-defined global/aux tokens in clinical settings, as validated in Table 9. (Since there is no official implementation of CATS, and the information in the paper is not enough to reproduce, we take TimeXer as baseline for its high reproducibility.)

Table 9: We compare our **Tech** with two representative models in general time series analysis that are similar to ours in certain respects. *(i)* ***Leddam (Yu et al., 2024)***: like GAFormer (Xiao et al., 2024) and our Tech, all employ a dual-dependency modeling structure. *(ii)* ***TimeXer (Wang et al., 2024e)***: like CATS (Lu et al., 2024) and our Tech, all employ global or auxiliary tokens to aggregate and redistribute information. The best is **Bolded**.

	ADFTD		APAVA		TDBrain		PTB		PTB-XL	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
Leddam	53.14±0.67	46.64±0.80	75.92±1.78	74.08±2.38	71.27±0.88	71.22±0.97	83.84±1.61	78.76±2.77	67.41±0.38	51.84±0.58
TimeXer	52.96±0.50	43.41±0.85	72.44±0.43	70.09±0.86	72.48±1.57	72.56±1.45	83.32±0.72	78.43±0.99	66.14±0.18	50.00±0.30
Tech (Ours)	**54.54±0.70**	**48.84±1.72**	**86.86±1.09**	**86.30±1.06**	**93.21±0.61**	**93.20±0.61**	**85.96±2.52**	**81.97±4.07**	**73.53±0.07**	**62.44±0.27**

Table 10: To further validate the generalizability of Tech, we further conduct a five-fold cross-validation based on the subject ID. We ensure that the classes within each dataset are balanced. We select the second-best Medformer as the baseline. The best is **Bolded**.

	ADFTD		APAVA		TDBrain		PTB		PTB-XL	
	Accuracy	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**	**Accuracy**	**F1-Score**
Medformer	53.41±3.05	49.03±3.97	68.01±9.13	66.63±9.71	82.92±9.03	81.13±9.16	83.30±5.46	72.46±5.17	71.76±0.66	61.10±0.70
Tech (Ours)	**55.05**±**2.43**	**49.82**±**2.82**	**80.66**±**6.53**	**79.62**±**6.79**	**87.06**±**6.71**	**86.00**±**6.62**	**89.48**±**3.18**	**84.59**±**2.84**	**73.65**±**0.41**	**62.79**±**0.52**

C.5 Five-fold Cross validation Result

To mitigate the bias of a fixed subject-independent split, we further performed a five-fold cross-validation based on subject IDs, ensuring balanced class distributions. As shown in Table 10, TeCh consistently surpasses Medformer across all datasets. For example, on APAVA, TeCh improves Accuracy and F1-Score by +12.6% and +13.0%, while on PTB, the gains reach +6.2% and +12.1%, respectively. TeCh also yields lower ***standard deviation*** (e.g., 9.71 vs. 6.79 on APAVA F1-Score), indicating greater robustness. These results confirm that TeCh generalizes more effectively across subjects and remains robust to inter-subject noise, benefiting from CoTAR's centralized aggregating-redistributing mechanism.

C.6 Centralization Analysis

To formally quantify the degree of centralization in a multivariate time series $\mathbf{J} \in \mathbb{R}^{S\times T}$, where S is the number of channels, and T is the length, we introduce two complementary metrics:

(1) Spectral Centralization Index (SCI):

$$\mathrm{SCI}(\mathbf{X}) = \frac{\lambda_{\max}\left(\frac{1}{T-1}(\mathbf{X}-\bar{\mathbf{X}})(\mathbf{X}-\bar{\mathbf{X}})^\top\right)}{\mathrm{Tr}\left(\frac{1}{T-1}(\mathbf{X}-\bar{\mathbf{X}})(\mathbf{X}-\bar{\mathbf{X}})^\top\right)}, \quad \bar{\mathbf{X}} = \frac{1}{T}\mathbf{X}\mathbf{1}_T.$$

(2) Dynamic Influence Centralization (DIC):

Let $Z = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}], \quad Y = [\mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_T]$,

estimate $\mathbf{A} = YZ^\dagger$, where $\mathbf{x}_t$ is the t-th column of $\mathbf{X}$.

$$\mathrm{DIC}(\mathbf{X}) = \frac{\max_i s_i - \bar{s}}{\bar{s}}, \quad \bar{s} = \frac{1}{S}\sum_i s_i, \quad s_i = \sum_j |A_{ji}|. \tag{6}$$

SCI measures spatial dominance as the energy concentration in the principal component of the covariance matrix (Jolliffe & Cadima, 2016), while DIC captures temporal dominance as the normalized imbalance of out-strengths in a first-order vector autoregressive model (Seth et al., 2015; Valente et al., 2008). As shown in Table 11, EEG and ECG datasets exhibit significantly higher centralization values than general-purpose datasets (Energy: ETTh2, ETTm2 (Zhou et al., 2021), Climate: Weather (Wu et al., 2021)). This confirms that MedTS possesses inherently centralized structures, where a few dominant channels or physiological processes govern the global dynamics. In contrast, energy and climate datasets are more decentralized. These results quantitatively validate our hypothesis and explain why TeCh's centralized aggregating–redistributing design is particularly effective for MedTS.

Table 11: Quantitative comparison of the **centralized property** across datasets. Beyond MedTS, we also include three general multivariate time series datasets for comparison. We measure centralization using: (1) **Spectral Centralization Index (SCI)**, the ratio of the largest eigenvalue to total variance, and (2) **Dynamic Influence Centralization (DIC)**, the normalized out-strength imbalance of a first-order VAR model. Higher values indicate stronger centralized behavior.

	EEG			ECG		Energy		Climate
Metric/Dataset	**ADFTD**	**APAVA**	**TDBrain**	**PTB**	**PTB-XL**	**ETTh2**	**ETTm2**	**Weather**
SCI	0.918	0.520	0.616	0.622	0.652	0.397	0.296	0.381
DIC	0.668	0.731	0.747	0.825	0.777	0.241	0.119	0.342

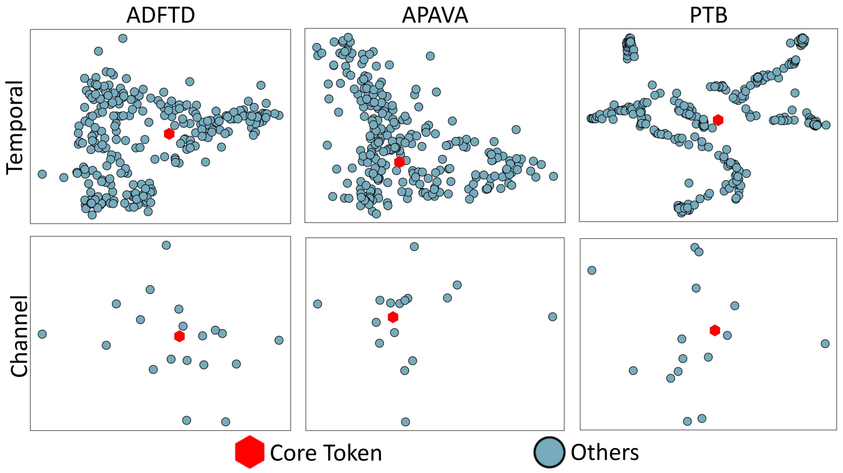


Figure 5: T-SNE visualization of the ***core token*** generated by CoTAR and other tokens. We visualize the embedding space of both temporal and channel.

C.7 Visualization of Core Token

In Figure 5, we visualized the ***core token*** generated by CoTAR and other embeddings across both temporal and channel spaces. Interestingly, in both embedding spaces, the core token consistently occupies a central position, suggesting that it captures a latent global physiological state integrating information across sensors (channel dimension) and across time (temporal dimension).

In the temporal space, this behavior reflects cross-temporal integration, which aggregates patterns over time into a stable representation of the system's evolving state. For EEG, such temporal aggregation resembles slow cortical dynamics, in which distributed neuronal populations maintain low-frequency coherence (e.g., alpha or beta bands) to stabilize perception and working-memory states (Niedermeyer & da Silva, 2005a; Buzsáki, 2006; Scherg et al., 2019b). For ECG, it parallels the beat-to-beat coordination within the cardiac cycle: the sinus node's rhythmic discharge orchestrates each P–QRS–T sequence, and the consistent temporal integration of these cycles ensures stable and regular cardiac pacing (AlGhatrif & Lindsay, 2012a; Goldberger et al., 2000a). Thus, the core token can be interpreted as a latent summary of temporal coherence in both neural and cardiac dynamics.

In the channel space, such centralization mirrors spatial integration across sensors. For EEG, this aligns with the global workspace and hub-based integration observed in frontoparietal networks that unify activity from distributed cortical regions (Dehaene & Changeux, 2011; Sporns, 2010). For ECG, it reflects pacemaker synchronization across myocardial conduction pathways, where a central excitation orchestrates coherent activation throughout the heart (Rieta & Alcaraz, 1999b; AlGhatrif & Lindsay, 2012a).

Together, these observations indicate that CoTAR's centralized proxy learns physiologically interpretable representations of both temporal and spatial coordination, effectively mirroring the centralized integration mechanisms that underlie real biological systems.