H-Tuning: Toward Low-Cost and Efficient ECG-based Cardiovascular Disease Detection with Pre-Trained Models

Rushuang Zhou [1 2] **Yuan-Ting Zhang** [3 4 5] **Yining Dong** [6 2]

Abstract

Fine-tuning large-scale pre-trained models provides an effective solution to alleviate the label scarcity problem in cardiovascular diseases (CVDs) detection using electrocardiogram (ECG). However, as the pre-trained models scale up, the computational costs for fine-tuning and inference become unaffordable on low-level devices deployed for clinical applications. Additionally, maintaining the model performance under low budgets in computational resources remains a significant challenge. However, a comprehensive study that can address them in a joint framework is still lacking. Here, we propose a holistic method (H-Tuning) for low-cost and efficient fine-tuning of pre-trained models on downstream datasets. Then, the inference costs of the models fine-tuned by H-Tuning are further reduced significantly using a knowledge distillation technique. Experiments on four ECG datasets demonstrate that H-Tuning reduces the GPU memory consumption during fine-tuning by 6.34 times while achieving comparable CVDs detection performance to standard fine-tuning. With the knowledge distillation technique, the model inference latency and the memory consumption are reduced by 4.52 times and 19.83 times. As such, the proposed joint framework allows for the utilization of pre-trained models with high computation efficiency and robust performance, exploring a path toward low-cost and efficient CVDs detection. Code is available at *https://github.com/KAZABANA/H-Tuning*

[1]Department of Biomedical Engineering, City University of Hong Kong, Hong Kong, China [2]Hong Kong Center for Cerebro-Cardiovascular Health Engineering, Hong Kong, China [3]Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong, China [4]Hong Kong Institutes of Medical Engineering, Hong Kong, China [5]The AICARE Bay Lab, Guangdong Medical University, Dong Guan, China [6]Department of Data Science, City University of Hong Kong, Hong Kong, China. Correspondence to: Yining Dong <yinidong@cityu.edu.hk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. Introduction

Acting as the number one cause of death, cardiovascular diseases threaten the lives of millions of people worldwide (Kelly et al., 2010; Mc Namara et al., 2019). In recent years, deep-learning models have been successful in diagnosing cardiovascular diseases (CVDs) using electrocardiography (ECG)(Hannun et al., 2019; Ribeiro et al., 2020; Strodthoff et al., 2020). However, gathering sufficient labeled data and computational resources required to train and implement deep learning models from scratch is still very expensive and time-consuming, especially for developers from resource-limited communities. Fortunately, fine-tuning large-scale pre-trained models provides an effective solution to reduce the requirement of labeled data in downstream datasets. For example, Vaid et al. (2023) pre-trained a large-scale vision transformer on a huge dataset with eight million ECG recordings, which demonstrated better transferability and performance than traditional network architectures. Subsequently, many studies have tried to transfer the success of large pre-trained foundation models from the computer vision and natural language processing domains to medical intelligence domains. For example, foundation models in retinal imaging (Zhou et al., 2023), cancer imaging (Pai et al., 2024), and digital pathology (Wang et al., 2024) were developed for enhancing disease diagnosis performance on downstream datasets. In the field of ECG-based CVDs detection, researchers also validated the effectiveness of fine-tuning foundation models in reducing the requirement on labeled data (Han & Ding, 2024; Mathew et al., 2024; McKeen et al., 2024; Pham et al., 2024).

However, fine-tuning pre-trained models becomes computationally expensive as the pre-trained models scale up. Specifically, standard fine-tuning optimizes all the parameters of pre-trained models using gradient backpropagation (first-order optimization), requiring a large amount of GPU memory. Unlike industrial servers, such computational costs are unaffordable for low-level devices. Hu et al. (2022) proposed the low-rank adaptation (LoRA) to alleviate this problem and maintain the model performance on downstream datasets. Specifically, it freezes the pre-trained weights and injects trainable low-rank matrices to fine-tune the pre-trained models, reducing the number of trainable parame-

ters and the GPU memory consumption in saving parameter gradients. However, LoRA cannot avoid caching network activations required during the backpropagation process, greatly limiting its efficiency in reducing GPU memory consumption. Malladi et al. (2023) addressed this problem by developing a backpropagation-free method (MeZO), which can fine-tune pre-trained models with just forward pass. MeZO calculates parameter gradients using a zeroth-order optimization method and waives the requirement for activations caching. It can achieve comparable performance with fine-tuning on natural language processing tasks. However, MeZO's performance would collapse without the assistance of prompt engineering (Malladi et al., 2023). Consequently, its effectiveness in medical image and physiological signal processing is questionable as prompt engineering is not feasible in these domains. In summary, the high and often prohibitive computational costs associated with fine-tuning pre-trained models have greatly impeded their widespread application in cardiac healthcare within resource-limited communities. This phenomenon highlights the need for computationally efficient fine-tuning methods with robust performance in the field of ECG-based CVDs detection. Apart from fine-tuning costs, pre-trained models also introduce high inference costs, which is also a pressing issue when deploying them on low-level devices. Specifically, deploying large-scale models is feasible on personal computers and laptops but is very difficult on mobile devices, such as phones and smartwatches. It limits the development of mobile cardiac health in the era of pre-trained models. Fortunately, the knowledge distillation technique (Hinton et al., 2015) can transfer the knowledge of large-scale teacher models to small-scale student models, providing a solution to alleviate this problem. In recent years, many studies have validated its effectiveness in ECG-based CVDs detection. For example, Sepahvand & Abdali-Mohammadi (2022) validated that student models can achieve comparable CVDs detection performance with large-scale teacher models while demonstrating low computational costs. Despite their effectiveness in reducing inference costs, the computational costs of training or fine-tuning teacher models for knowledge distillation are not considered.

In summary, the high fine-tuning and inference costs of pre-trained models hinder their application in ECG-based CVD detection, especially in resource-limited environments. However, a comprehensive study that addresses them in a joint workflow is still lacking. Additionally, how to maintain the fine-tuning performance under low computational budgets still needs to be explored. In this study, a holistic framework (H-Tuning) is proposed to fine-tune pre-trained models with low computational costs while maintaining good model performance. By integrating a knowledge distillation technique, a joint workflow for low-cost and efficient ECG-based CVDs detection is developed (Fig.1). Specifically, a mix-order optimization method is first designed to accurately estimate gradient information with low GPU memory footprints during the fine-tuning process. A low-rank adaptation technique is then integrated to reduce the number of trainable parameters. Subsequently, the shallow and deep layers of pre-trained models are updated with different schemes to ensure low fine-tuning costs while avoiding significant drops in detection performance. Experiment results on four downstream datasets demonstrate that H-Tuning achieves comparable performance with standard fine-tuning while reducing the GPU memory consumption by 6.34 times. To reduce the inference costs of the fine-tuned models for mobile cardiac healthcare, a knowledge distillation technique is utilized to transfer the knowledge from large-scale fine-tuned models to tiny student models. In comparison to the teacher models, the inference latency, the GPU memory consumption, and the number of parameters of the student models are reduced by 4.52, 19.83, and 194.23 times, respectively. Notably, student models also achieve similar CVDs detection performance when compared to the teacher models. Additionally, the knowledge transfer process improves model performance in recognizing CVDs from 1-lead and 3-lead ECG, facilitating efficient mobile healthcare using wearable ECG devices. With robust CVDs detection performance and low computational costs, the proposed workflow can be deployed for accurate and mobile cardiac healthcare, providing an effective solution for low-cost and efficient ECG-based CVDs detection. The major contributions of the proposed paradigm are listed below:

- First, a mix-order optimization method is proposed to accurately estimate the gradient information using a coarse-to-fine estimation mechanism. Compared with traditional optimization methods, it greatly reduces the computation costs for fine-tuning pre-trained models while maintaining the CVDs detection performance of the fine-tuned models.

- Second, a holistic framework (H-Tuning) is developed to integrate the mix-order optimization with low-rank adaptation and a novel layer-dependent model update scheme, enhancing both computational efficiency and robustness. Then, a knowledge distillation technique is introduced to reduce the computational costs for model inference.

- Third, a comprehensive workflow is proposed for reducing the fine-tuning and inference costs of pre-trained models while maintaining diagnostic performance, exploring a new path toward low-cost and efficient CVDs detection using ECG.

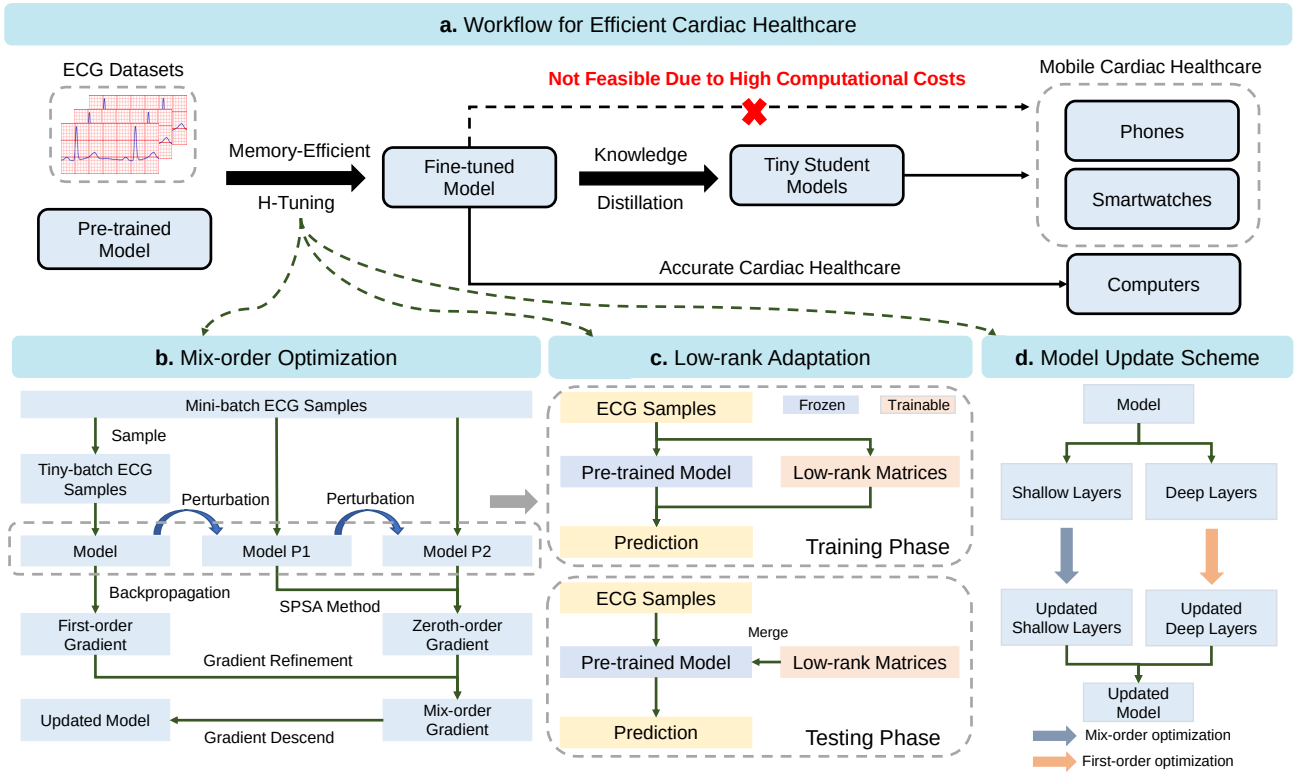


Figure 1. **The workflow of efficient CVDs detection and cardiac healthcare with ECG data.** Using the proposed H-Tuning framework, a large-scale pre-trained model is first fine-tuned on the downstream ECG datasets in a memory-efficient way. Then, to further reduce the computational cost, the knowledge of the large-scale fine-tuned model is transferred into tiny student models using a knowledge distillation method.

2. Method

2.1. Preliminaries about Zeroth-order Optimization

Fine-tuning pre-trained models using gradient backpropagation provides a robust method for solving downstream CVDs detection tasks. However, the gradient backpropagation process requires saving the network activations during the forward process, which consumes a prohibitive amount of GPU memory. On the contrary, zeroth-order optimization estimates the gradients using the loss differences, waiving the need to store the network activations (Spall, 1992; Flaxman et al., 2005; Duchi et al., 2015). Specifically, the gradients of a model with parameters θ can be estimated by the simultaneous perturbation stochastic approximation (SPSA) method (Flaxman et al., 2005),

$$\widehat{\nabla}\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathcal{L}(\theta+\mu z_i)-\mathcal{L}(\theta-\mu z_i)}{2\mu}z_i, \quad (1)$$

where $z_i \in \mathbb{R}^d$ is a random vector sampled from the standard Gaussian distribution $N(0, I_d)$, μ is the perturbation scale and n is the number of function queries. Note that a large n value reduces the variance of the estimated gradients but increases the complexity of forward computation. Based on the estimated gradients at the t-th iteration, the parameters θ can be updated by the gradient descent process as

$$\theta_{t+1} = \theta_t - \eta\widehat{\nabla}\mathcal{L}(\theta_t). \quad (2)$$

Assuming that the loss function $\mathcal{L}(\theta)$ is L-Lipschitz smooth, the differences between $\mathbb{E}_z\left[\widehat{\nabla}\mathcal{L}(\theta)\right]$ and the true gradients $\nabla\mathcal{L}(\theta)$ can be bounded as (Nesterov & Spokoiny, 2017; Gao et al., 2018),

$$\left\|\mathbb{E}_z\left[\widehat{\nabla}\mathcal{L}(\theta)\right]-\nabla\mathcal{L}(\theta)\right\|^2 \leq \frac{\mu^2L^2d^2}{4} \quad (3)$$

When $\mu \to 0$, Eq. (1) provides us an unbiased estimation of the true gradients of θ : $\mathbb{E}_z\left[\widehat{\nabla}\mathcal{L}(\theta)\right] = \nabla\mathcal{L}(\theta)$, which indicates the feasibility of the zeroth-order methods in optimization. In the implementation, μ is usually set to a small constant (1e-5, 1e-3), and n is set to 1 for efficient model training, which makes $\widehat{\nabla}\mathcal{L}(\theta)$ become a biased estimation of the true gradients (Malladi et al., 2023).

2.2. Mix-order Optimization

Although the zeroth-order methods demonstrate low GPU memory footprints, the models fine-tuned with them cannot achieve comparable performance to the models fine-tuned with the first-order methods, which calculate the gradient information using backpropagation (Li et al., 2024). In natural language processing tasks, prompt engineering is adopted to make the optimization trajectory well-behaved, which enables the zeroth-order methods to optimize the whole network (Malladi et al., 2023; Zhang et al., 2024). However, prompt engineering is only feasible when text data is available for model training, which greatly limits the effectiveness of the zeroth-order methods in ECG-based CVDs detection. In response to the aforementioned limitations, we propose a mix-order optimization method, which enables memory-efficient and robust model fine-tuning without prompt engineering.

Given a random batch of ECG signals $\mathcal{B} = \{x_i, y_i\}_{i=1}^{N}$ and a model with parameters θ, the zeroth-order method (SPSA, Eq. (1)) is first utilized to provide an initial estimation of the parameter gradients, defined as $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$. Here, $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$ can be regarded as an approximation of $\widehat{\nabla}\mathcal{L}(\theta)$. Considering that multiple CVDs can be diagnosed from each ECG segment, the CVDs detection process should be summarized as a multi-label classification task. Consequently, the loss function utilized in our study can be defined as a multi-label binary cross-entropy loss,

$$\begin{aligned}&\mathcal{L}(\mathcal{B};\theta)\\&= -\frac{1}{NC}\sum_{i=1}^{N}\sum_{c=1}^{C}(1-y_{i,c})\log(1-p_{i,c}) + y_{i,c}\log p_{i,c},\end{aligned} \tag{4}$$

where $p_{i,c} = P(c = 1|\theta; x_i)$ is the model prediction on class c and C is the number of categories. According to Eq.(3), the variance of $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$ increases rapidly as the model size scales up, which results in slow convergence and poor model performance (Gautam et al., 2024). To address this problem, we propose to utilize the first-order method (gradient backpropagation) to refine the norm and the direction of the estimated gradients $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$. In order to avoid heavy computational burdens, backpropagation is only conducted on a tiny subset $\mathcal{B}_1 = \{x_i, y_i\}_{i=1}^{N_1}$ sampled from $\mathcal{B}$, with $N_1 \ll N$. Specifically, the gradient refinement process can be formulated as,

$$\begin{aligned}&\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)_\lambda\\&= \lambda\nabla\mathcal{L}(\mathcal{B}_1;\theta) + (1-\lambda)\frac{\|\nabla\mathcal{L}(\mathcal{B}_1;\theta)\|}{\|\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)\|}\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta),\end{aligned} \tag{5}$$

where $\nabla\mathcal{L}(\mathcal{B}_1;\theta)$ is the parameter gradients estimated by the gradient backpropagation on the tiny subset $\mathcal{B}_1$, λ is a hyperparameter balancing the first-order and the zeroth-order methods. The direction of $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$ depends on the random vector z, which might have a large discrepancy with the true gradients. Consequently, we introduce $\nabla\mathcal{L}(\mathcal{B}_1;\theta)$ to refine the direction of the estimated gradients with an important factor of λ. Compared with the gradients $\nabla\mathcal{L}(\mathcal{B}_1;\theta)$ estimated by the first-order methods, $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$ has a much larger expected Frobenius norm when $n = 1$, which can be formulated as (Malladi et al., 2023),

$$\begin{aligned}\mathbb{E}\left[\|\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)\|^2\right] &= d\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B};\theta)\|^2\right]\\&\geq \frac{dN_1}{N}\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B}_1;\theta)\|^2\right].\end{aligned} \tag{6}$$

The proof is provided in Appendix A.1. When $1 \leq N_1 \ll N \ll d$, we have,

$$\mathbb{E}\left[\|\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)\|^2\right] \gg \mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B}_1;\theta)\|^2\right]. \tag{7}$$

$\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)_\lambda$ is a weighted sum of the gradients estimated by the first-order and the zeroth-order methods. Without norm refinement, its direction and Frobenius norm would be dominated by $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$ unless $\lambda \approx 1$. A naive solution is simply dividing the $\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)$ by $\sqrt{\frac{dN_1}{N}}$. However, as shown in Eq.(1), the random vector z is sampled from the standard Gaussian distribution. This suggests that the discrepancy between $\|\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)/\sqrt{\frac{dN_1}{N}}\|$ and $\|\nabla\mathcal{L}(\mathcal{B}_1;\theta)\|$ is not stable and will affect the refinement process. To address this problem, we normalize the magnitude of $\|\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)\|$ using $\|\nabla\mathcal{L}(\mathcal{B}_1;\theta)\|$ and Eq.(5).

Compared with the standard zeroth-order optimization, the proposed mix-order optimization provides a more stable and accurate estimation of the parameter gradients in both direction and Frobenius norm. Without prompt engineering, this advantage enables the mix-order optimization to demonstrate a better convergence behavior than the zero-order methods. Additionally, the extra computational burdens introduced by the gradient refinement process are ignorable because the backpropagation process is only conducted on the tiny subset $\mathcal{B}_1$.

2.3. Hybrid Tuning for Lightweight and Robust Model Training

While the proposed mix-order optimization method demonstrates robust performance, its computational efficiency and robustness can be further improved by two techniques. First, we integrated our mix-order optimization with low-rank adaptation, which approximates the incremental update of the pre-trained weights by two low-rank matrices (Hu et al., 2022). With inputs X, the output of a linear layer with parameters $\theta = \{W_0, b\}$ can be formulated as,

$$h = (W_0 + BA)X + b, \tag{8}$$

where $B \in \mathbb{R}^{d_{\text{in}} \times r}$ and $A \in \mathbb{R}^{r \times d_{\text{out}}}$, and the rank $r \ll \min(d_{\text{in}}, d_{\text{out}})$. During model training, pre-trained weight W_0 is frozen while the bias term b and the low-rank matrices A, B are trainable. Compared with tuning the whole linear layer, the low-rank adaptation greatly reduces the number of trainable parameters, which decreases the variance in the SPSA process according to Eq.(3). Besides, it can significantly reduce the storage consumption of the fine-tuned models, as we only need to save the optimized low-rank matrices.

The second technique is applying different optimization schemes to different layers distributed at different locations of the model. Specifically, we define the last $M = 2$ linear layers as the deep layers and the remaining layers as the shallow layers. It is well known that the deep layers contain the task-variant knowledge while the shallow layers contain the domain-invariant knowledge (Sharif Razavian et al., 2014; Tajbakhsh et al., 2016). More importantly, fine-tuning the deep layers using first-order optimization is computationally efficient, whereas fine-tuning the shallow layers using the same method introduces heavy computational burdens. Consequently, we propose tuning the deep layers using first-order optimization to provide good adaptability to the model on downstream tasks. Additionally, we tune the shallow layers using the proposed mix-order optimization method to improve computational efficiency during the model training. In summary, the proposed pipeline can be formulated as,

$$\theta_{t+1}^m = \begin{cases} \theta_t^m - \eta \nabla \mathcal{L}(\mathcal{B}; \theta_t^m), & m \geq K - M \\ \theta_t^m - \eta \widehat{\nabla} \mathcal{L}(\mathcal{B}; \theta_t^m)_\lambda, & m < K - M \end{cases}, \tag{9}$$

where K is the total number of layers within the model, $\nabla \mathcal{L}(\mathcal{B}; \theta_t^m)$ is computed using the gradient backpropagation and $\widehat{\nabla} \mathcal{L}(\mathcal{B}; \theta_t^m)_\lambda$ is estimated using Eq.(5). The complete algorithm of H-Tuning is presented in Appendix A.2.

2.4. Knowledge Distillation for Fast Model Inference

In application scenarios with very limited computation resources, the model inference efficiency should be high enough to satisfy the hardware requirements. Although the model fine-tuning process can be carried out on the cloud servers, the inference process should be performed on the local devices. When mobile and continuous monitoring is required, the inference speed should be high, and the corresponding memory footprints should be low. Consequently, we first conduct the proposed H-Tuning on a large pre-trained model on downstream tasks and then transfer its knowledge to a tiny model using a classic knowledge distillation method (Hinton et al., 2015). The optimized parameters of the large fine-tuned model are defined as θ_T, and the parameters of the tiny student model are defined as θ_S. During the knowledge distillation process, the large model is frozen, and the training loss of the student model is formulated as,

$$\begin{aligned} &\mathcal{L}_{\mathcal{K}}(\mathcal{B}; \theta_S) \\ &= -\frac{1}{NC} \sum_{i=1}^{N} \sum_{c=1}^{C} (1 - p_{i,c}^T) \log(1 - p_{i,c}^S) + p_{i,c}^T \log p_{i,c}^S, \end{aligned} \tag{10}$$

$$\mathcal{L}_{\mathcal{S}}(\mathcal{B}; \theta_S) = \mathcal{L}(\mathcal{B}; \theta_S) + \mathcal{L}_{\mathcal{K}}(\mathcal{B}; \theta_S), \tag{11}$$

where $p_{i,c}^S = P(c = 1 | \theta_S; x_i^s)$, $p_{i,c}^T = P(c = 1 | \theta_T; x_i)$. $\mathcal{B} = \{x_i, y_i\}_{i=1}^N$ is a random batch of ECG signals and the corresponding ground truth for knowledge distillation. To allow the student models to handle ECG recordings with various numbers of leads and meet the requirements for mobile cardiac healthcare, the input x_i^s can be ECG signals with different numbers of leads. If only 1-lead ECG is accessible for the student model, we use lead I of x_i to generate x_i^s. If only 3-lead ECG is accessible for the student model, we use leads II, V1, and V5 of x_i to generate x_i^s. Considering the number of trainable parameters in the student model is very small, we use the gradient backpropagation to optimize its parameters θ_S.

3. Results

3.1. Datasets and Signal Pre-processing

In this study, the Chapman-Shaoxing database (Zheng et al., 2020b), the Georgia 12-lead ECG Challenge (G12EC) database (Alday et al., 2020), the Physikalisch-Technische Bundesanstalt (PTB-XL) database (Wagner et al., 2020), and the Ningbo database (Zheng et al., 2020a) are used for the performance evaluation of our H-Tuning framework. The four datasets were also included in the Physionet 2020/2021 challenge (Alday et al., 2020; Reyna et al., 2022). To be specific, there are 10,646 12-lead ECG recordings in the Chapman-Shaoxing database. In addition, the G12EC database, the PTB-XL database, and the Ningbo database contain 10344, 21837, and 34,905 12-lead ECG recordings, respectively. The recordings from the above databases last around 10 seconds with a sampling rate of 500 Hz and are annotated with multi-label CVDs ground truths. Note that multiple CVDs can be recognized from each recording simultaneously. Consequently, automatic CVDs detection can be formulated as a multi-label classification task. To ensure sufficient representation across the training, validation, and test sets, only CVD categories containing more than 200 ECG recordings are included during model training and evaluation. For signal pre-processing, a band-pass (1-47Hz) is applied to remove the potential noises from the raw ECG recordings, such as the power-line interference and the motion artifacts. Then, z-score normalization is utilized to normalize the filtered signals.

Table 1. Performance of H-Tuning and the compared models on four public datasets. The average performance across four seeds is presented. The dashed lines separate the memory-efficient fine-tuning methods and the fine-tuning methods with first-order optimization. The unit of memory is Gigabyte (GB).

	G12EC Dataset			PTB-XL Dataset			Ningbo Dataset			Chapman Dataset		
Methods	Memory	Macro AUC	Macro $F_{\beta=2}$	Memory	Macro AUC	Macro $F_{\beta=2}$	Memory	Macro AUC	Macro $F_{\beta=2}$	Memory	Macro AUC	Macro $F_{\beta=2}$
Full FT	9.214	0.869	0.588	9.212	0.919	0.618	9.211	0.933	**0.591**	9.211	0.930	0.623
LoRA	8.754	0.870	**0.592**	8.754	0.919	0.618	8.754	**0.934**	0.580	8.754	**0.932**	**0.636**
LP	1.416	0.852	0.545	1.416	0.897	0.582	1.416	0.906	0.530	1.416	0.898	0.579
MeZO	1.815	0.532	0.327	1.814	0.515	0.241	1.814	0.504	0.231	1.814	0.499	0.277
MeZO + LoRA	1.437	0.507	0.316	1.444	0.483	0.227	1.444	0.530	0.234	1.444	0.481	0.278
Addax	2.000	0.863	0.573	2.003	0.898	0.578	2.003	0.909	0.509	2.004	0.929	0.596
Addax + LoRA	1.453	0.857	0.568	1.451	0.899	0.582	1.451	0.901	0.495	1.451	0.907	0.588
LoHO	2.002	0.851	0.554	2.000	0.902	0.588	2.001	0.907	0.535	1.998	0.905	0.574
LoHO + LoRA	1.453	0.851	0.557	1.453	0.900	0.581	1.453	0.909	0.535	1.453	0.904	0.580
H-Tuning	1.453	**0.870**	0.586	1.453	**0.923**	**0.628**	1.453	0.931	0.550	1.453	0.929	0.634

Table 2. Ablation study of H-Tuning. The average performance across four seeds is presented. The unit of memory is Gigabyte (GB).

	G12EC Dataset			PTB-XL Dataset			Ningbo Dataset			Chapman Dataset		
Methods	Memory	Macro AUC	Macro $F_{\beta=2}$	Memory	Macro AUC	Macro $F_{\beta=2}$	Memory	Macro AUC	Macro $F_{\beta=2}$	Memory	Macro AUC	Macro $F_{\beta=2}$
Without SPSA gradient estimation	1.451	0.866	0.575	1.451	0.919	0.616	1.451	0.921	0.525	1.451	0.927	0.612
Without gradient refinement	1.453	0.851	0.557	1.453	0.900	0.581	1.453	0.909	0.535	1.453	0.904	0.580
Without gradient normalization	1.453	0.870	**0.588**	1.453	0.917	0.609	1.453	0.924	0.536	1.453	0.928	0.596
Without low-rank adaptation	2.002	0.856	0.564	2.000	0.907	0.588	2.000	0.924	0.527	2.001	0.922	0.603
H-Tuning	1.453	**0.870**	0.586	1.453	**0.923**	**0.628**	1.453	**0.931**	**0.550**	1.453	**0.929**	**0.634**

3.2. Experiment Protocols and Evaluation Metrics

In our experiments, a pre-trained model provided by Zhou et al. (2024) acts as the backbone for all the compared fine-tuning methods. Specifically, the backbone has 50.494 million parameters and is pre-trained on the Clinical Outcomes in Digital Electrocardiology (CODE) dataset (Ribeiro et al., 2019; 2020). Subsequently, the backbone is fine-tuned and evaluated on the four public datasets with a limited number of labeled samples. For each dataset, a training set and a held-out test set are randomly sampled in a ratio of 1: 9. Then, a validation set is collected from the training set and accounts for 20% of it. In our implementation, the validation set is used to select the best checkpoint during the fine-tuning process. During the model evaluation process, we utilize two metrics to quantify the performance of different methods in multi-label CVDs detection: macro $F_{\beta=2}$ and macro AUC. Following the settings in Strodthoff et al. (2020), we set the β value to be 2. Additionally, we record the peak GPU memory footprints of different methods during fine-tuning. In our experiments, we compare H-Tuning with several baseline models: Full Fine-Tuning (Full FT), Low-Rank Adaptation (LoRA) (Hu et al., 2022), Linear Probing (LP), Memory-Efficient Zeroth-Order Optimization (MeZO) (Malladi et al., 2023), Low-order Hybrid Optimizer (LoHO) (Chen et al., 2025), and Addition of Gradient Estimates through Memory-Efficient Execution (Addax) (Li et al., 2024). Full FT, LoRA, and LP are three popular adaptation approaches that use first-order optimization to update the trainable parameters. MeZO is the baseline model in zeroth-order optimization. Addax and LoHO combine the first-order and zeroth-order optimization methods, serving as state-of-the-art models in memory-efficient fine-tuning. At the same time, we integrate them with LoRA to formulate new comparison methods (MeZO + LoRA, Addax + LoRA, LoHO + LoRA), which have less trainable parameters than them. Note that the low-rank matrices are implemented in every layer of the pre-trained backbone. Details of the compared methods are provided in Appendix A.3.

3.3. Comparison of H-Tuning with Existing Methods

As presented in Table 1, we report the CVDs detection performance and GPU memory footprints of the proposed H-Tuning and the compared methods. It can be observed that H-Tuning is the only memory-efficient method that achieves comparable performance with Full FT and LoRA. Specifically, H-Tuning achieves an average macro $F_{\beta=2}$

of 0.600 across four datasets and only has performance losses of 0.5% and 0.7% compared with Full FT and LoRA, respectively. More importantly, H-Tuning achieves a remarkable reduction in GPU memory footprints by 6.02 to 6.34 times compared with LoRA and Full FT, indicating its flexibility and compatibility in clinical devices with limited GPU memory. Specifically, LoRA and Full FT use first-order optimization methods to calculate the parameter gradients, which require large GPU memory to save the activation outputs for backpropagation on the mini-batch samples. In contrast, H-Tuning estimates the gradients using the proposed mix-order optimization, significantly reducing the need to store the activation outputs and the overall GPU memory footprints. Compared with other memory-efficient methods, H-Tuning demonstrates the best CVDs detection performance without significantly increasing the GPU memory footprints. Compared with the method with the lowest GPU memory footprints (LP), the extra GPU memory consumption introduced by H-Tuning is ignorable in model training (0.037 GB). However, the improvements in CVDs detection performance are remarkable. For example, H-Tuning achieves a marco $F_{\beta=2}$ score of 0.634 on the Chapman dataset, outperforming LP by 5.5%. In Appendix A.4, we provide detailed comparisons of different methods using more evaluation metrics.

3.4. Ablation Studies

In this section, ablation studies are performed to demonstrate the contribution of the modules implemented in the proposed H-Tuning. As shown in Table 2, we successively remove the components from the H-Tuning and report the corresponding CVDs detection performance across four datasets. (1) Utilizing SPSA methods to estimate the parameter gradients improves model performance. When the SPSA gradient estimation module is removed from the H-Tuing ($\lambda = 1$), backpropagation cannot calculate the gradient information with enough batch sizes due to limited GPU memory spaces. Therefore, the CVDs detection performance decreases in all datasets. (2) A remarkable degeneration in model performance is observed when the gradient refinement module is removed ($\lambda = 0$). It indicates that refining the initial gradients estimated by the SPSA method benefits the optimization process. (3) Applying gradient normalization (Eq.(5)) during the refinement process has a positive effect on the model performance. Specifically, an obvious performance enhancement is attained when the module is implemented. Ablation studies of H-Tuning on more evaluation metrics are provided in Appendix A.4

3.5. Sensitivity Analyses

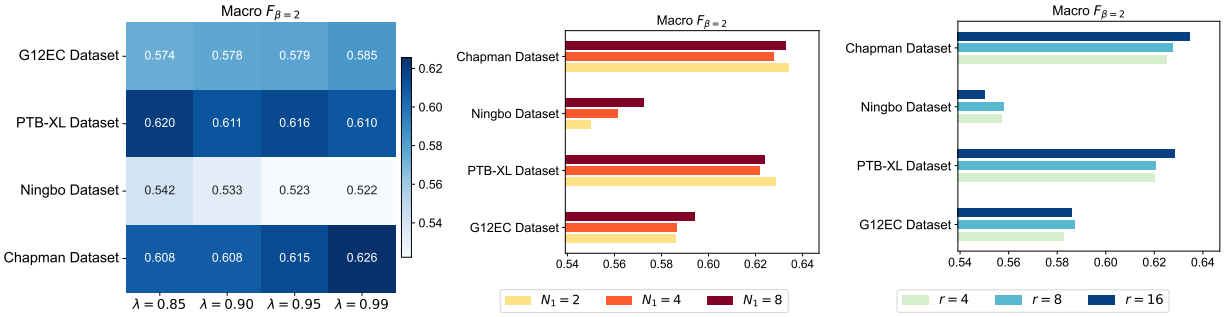
(1) The effect of the control parameter for mix-order optimization. In the proposed mix-order optimization module, we control the importance of the zeroth-order SPSA method and the first-order gradient backpropagation by a hyperparameter λ (Eq.(5)). Here, we adjust its value from 0.85 to 0.99 and present the corresponding model performance across four datasets. As shown in Fig.2a, the experiment results reveal that the model performance is not sensitive to the varying λ, suggesting the stability of the proposed module.

(2) The effect of the backpropagation batch size. In the gradient refinement process, we conduct gradient backpropagation on a tiny subset $\mathcal{B}_1 = \{x_i, y_i\}_{i=1}^{N_1}$ to refine the estimated gradients in Eq.(5). As shown in Fig.2b, we adjust N_1 from 2 to 8 and report the corresponding model performance across four datasets. It can be observed that the model performance generally improves as N_1 increases on the Ningbo and the G12EC datasets. However, increasing N_1 has a limited impact on the PTB-XL and the Chapman datasets. This phenomenon indicates that the difficulty of recovering the true gradient information varies across datasets.

(3) The effect of the rank parameter for low-rank adaptation. The proposed H-Tuning framework uses low-rank adaptation (Hu et al., 2022) to reduce the number of trainable parameters during the mix-order optimization process. The rank of all the low-rank matrices is controlled by the parameter r, and the number of trainable parameters decreases as r decreases. As presented in Fig.2c, we adjust r from 4 to 16 and provide the corresponding model performance across four datasets. The experiment results demonstrate that the model performance improves as r increases. This phenomenon can be explained by the various representation capacities of the fine-tuned models with different r. Specifically, increasing the number of trainable parameters enables the low-rank matrices to store more task-specific information in CVDs detection, thus improving the model performance. On the other hand, it is important to note that it will also increase the storage costs of the matrices. Consequently, the value rank parameter r should be carefully selected according to the maximum storage spaces and the expected CVDs detection performance.

3.6. Inference Costs Reduction through Knowledge Distillation

Apart from the training process, the computational costs of the fine-tuned models during the inference process are important for their implementations in clinical practices. Therefore, through a knowledge distillation technique (Hinton et al., 2015), we compress the fine-tuned large-scale model into a tiny student model, which is able to be deployed at low-level devices for inference. Considering that most mobile ECG devices only have 1-3 leads, we also investigate the performance of the student model under various lead configurations. As shown in Table 3, we report the CVDs detection performance of the student models



(a) Effect of the parameter λ. (b) Effect of the parameter N_1. (c) Effect of the parameter r.

Figure 2. The sensitivity analysis on the H-Tuning.

Table 3. Overall CVDs detection performance of the student models with various numbers of ECG leads. The average performance across four seeds is presented. 'None' represents that the student models are optimized using the ground truths without teachers' assistance.

		G12EC Dataset		PTB-XL Dataset		Ningbo Dataset		Chapman Dataset	
Teacher	Student	Macro AUC	Macro $F_{\beta=2}$	Macro AUC	Macro $F_{\beta=2}$	Macro AUC	Macro $F_{\beta=2}$	Macro AUC	Macro $F_{\beta=2}$
None	12-Lead	0.847	0.558	0.903	0.592	0.921	0.530	0.927	0.617
12-Lead	12-Lead	0.868	0.582	0.921	0.628	**0.937**	**0.568**	**0.933**	0.632
None	12-Lead	0.834	0.537	0.886	0.559	0.903	0.500	0.911	0.577
3-Lead	3-Lead	0.860	0.574	0.905	0.596	0.928	0.549	0.919	0.614
None	1-Lead	0.772	0.457	0.830	0.462	0.844	0.420	0.847	0.498
12-Lead	1-Lead	0.795	0.495	0.843	0.509	0.866	0.469	0.862	0.538
Teacher's performance		**0.870**	**0.586**	**0.923**	**0.628**	0.931	0.550	0.929	**0.634**

across four datasets. Note that we use leads II, V1, and V5 to simulate 3-lead ECG signals and use lead I to simulate 1-lead ECG signals. The performance of the teacher models fine-tuned by H-Tuning is presented for comparison. The results demonstrate that the student models with 12-lead ECG achieve similar and even better diagnostic performance than the teacher models. Additionally, it can be observed that the participation of teacher models introduces remarkable improvements in student models' performance, highlighting the effectiveness of the knowledge transfer process. Specifically, the average improvements on macro $F_{\beta=2}$ score across four datasets are 2.81%, 3.98%, and 4.36% for 12-lead, 3-lead, and 1-lead ECG signals. We also present the computational efficiency of the teacher and student models with different numbers of ECG leads in Fig.3. Specifically, the GPU memory footprints, the inference time, and the number of parameters of different models are utilized to quantify their computational efficiency. It can be observed that the student models have only 0.26 million parameters. They achieve an inference time of 1.28 ms and only consume 18.05 MB of GPU memory during the ECG screening process. Compared with the large-scale teacher model, the student models speed up the inference speed by 4.52 times and decrease the GPU memory footprints and the number of parameters by 19.83 times and 194.23 times, respectively. Consequently, the results demonstrate the high inference efficiency of the student models, highlighting their great potential in cardiac healthcare using low-level mobile devices. The architecture of the teacher and student models is provided in Appendix A.3.

3.7. External Validation on A Wearable 12-lead ECG Dataset

External validation is an important approach to evaluate the generalization performance of the proposed method on unseen datasets. Specifically, We combine the G12EC, PTB-XL, Ningbo, and Chapman datasets to fine-tune the pre-trained model and generate three teacher models using three methods (Full FT, LoRA (Hu et al., 2022) and the proposed H-Tuning). Following the dataset splitting method defined in Section.3.1, only 10% of the labeled data within the training set are used for fine-tuning. Subsequently, the knowledge distillation technique defined in Eq.(10) is utilized to create three corresponding student models. Then, an external validation set consisting of 7000 wearable 12-lead ECG signals is used to evaluate the CVDs detection performance of our classifiers in mobile cardiac healthcare. The external dataset is provided by (Lai et al., 2023), which contains 60 types of CVDs. Here, only the CVDs that

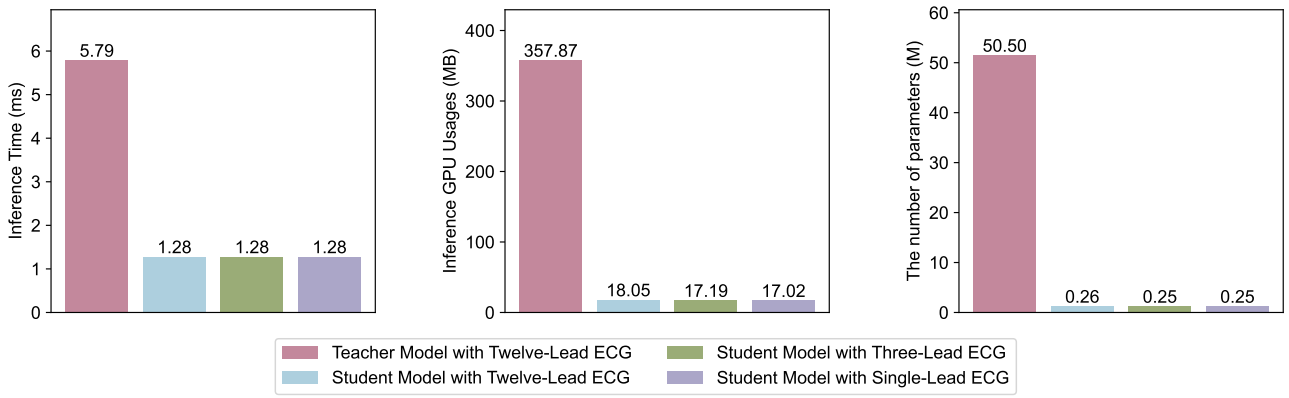


Figure 3. Inference efficiency of the teacher model and the student models with different numbers of ECG leads. The batch size for the four models is set to 4.

Table 4. External validation on the wearable 12-lead ECG dataset. The average performance across four seeds is presented.

Methods	Ranking Loss ↓	Coverage ↓	Macro AUC ↑	MAP ↑	Macro $G_{\beta=2}$ ↑	Macro $F_{\beta=2}$ ↑
Teacher Models						
Full FT	0.137	5.595	0.870	**0.600**	0.314	0.570
LoRA	0.134	5.440	0.879	0.598	**0.319**	**0.579**
H-Tuning	0.141	5.484	0.866	0.575	0.312	0.567
Student Models						
Full FT	0.135	5.462	0.867	0.566	0.287	0.534
LoRA	0.129	5.384	0.874	0.582	0.302	0.543
H-Tuning	**0.127**	**5.297**	**0.880**	0.598	0.311	0.551

co-exist in the external dataset and the training set are selected for model evaluation, including NSR, QAb, TAb, SB, SA, PAC, AF, AFL, PVC, IAVB, BBB, CRBBB, IRBBB, CLBBB, and PR. The CVD detection performance of the six classifiers on the external dataset is shown in Table 4. The results demonstrate that the teacher model generated by H-Tuning achieves performance similar to Full FT and LoRA. As shown in Table 1, the GPU memory consumptions of H-Tuning are 6.34 times smaller than Full FT. The above two observations reveal that H-Tuning can provide a low-cost and effective solution to fine-tuning pre-trained models. Additionally, our student model performs better than the compared methods. This phenomenon could be explained by the fact that strong teachers might not have a better teaching ability than weaker teachers (Huang et al., 2022).

4. Conclusion

In this paper, we propose the H-Tuning framework for fine-tuning pre-trained models in low GPU memory consumption. Experiment results on four downstream datasets demonstrate that the proposed H-Tuning outperforms other memory-efficient methods in ECG-based CVDs detection with remarkable superiority. Additionally, H-Tuning significantly reduces GPU memory usage and achieves comparable performance with first-order fine-tuning methods. Subsequently, the fine-tuned models are compressed into small-scale student models through a knowledge distillation technique. Compared with large-scale models, small-scale student models have significantly lower inference costs and are more suitable for mobile cardiac healthcare. In conclusion, this paper designs a joint workflow to reduce the fine-tuning and inference costs of large-scale pre-trained models in ECG-based CVDs detection. We hope the proposed workflow could pave the way toward low-cost and efficient CVDs detention with pre-trained models. In the future, we plan to explore the applications of H-Tuning on other physiological signals, such as electroencephalograms.

Impact Statement

Our H-Tuning provides a pathway toward low-cost and efficient ECG-based cardiovascular disease detection by designing a joint workflow to reduce the fine-tuning and inference costs of large-scale pre-trained models. It can accelerate and popularize the clinical applications of pre-trained ECG analysis models in resource-limited communities. Additionally, it has the potential to be generalized to many other fields in machine learning domains.

Acknowledgements

This work was supported in part by InnoHK Project at Hong Kong Centre for Cerebro-cardiovascular Health Engineering (COCHE), in part by the the National Natural Science Foundation of China (22322816) and in part by the City University of Hong Kong Project (9610640).

References

Alday, E. A. P., Gu, A., Shah, A. J., Robichaux, C., Wong, A.-K. I., Liu, C., Liu, F., Rad, A. B., Elola, A., Seyedi, S., et al. Classification of 12-lead ECGs: the physionet/-computing in cardiology challenge 2020. *Physiological measurement*, 41(12):124003, 2020.

Chen, M., Huang, Y.-L., and Wen, Z. Towards efficient low-order hybrid optimizer for language model fine-tuning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23605–23613, Apr. 2025. doi: 10.1609/aaai.v39i22.34530.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Wibisono, A. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pp. 385–394, USA, 2005. Society for Industrial and Applied Mathematics. ISBN 0898715857.

Fuster, V., Rydén, L. E., Cannom, D. S., Crijns, H. J., Curtis, A. B., Ellenbogen, K. A., Halperin, J. L., Heuzey, J.-Y. L., Kay, G. N., Lowe, J. E., Olsson, S. B., Prystowsky, E. N., Tamargo, J. L., Wann, S., MEMBERS, A. T. F., Smith, S. C., Jacobs, A. K., Adams, C. D., Anderson, J. L., Antman, E. M., Halperin, J. L., Hunt, S. A., Nishimura, R., Ornato, J. P., Page, R. L., Riegel, B., GUIDELINES, E. C. F. P., Priori, S. G., Blanc, J.-J., Budaj, A., Camm, A. J., Dean, V., Deckers, J. W., Despres, C., Dickstein, K., Lekakis, J., McGregor, K., Metra, M., Morais, J., Osterspey, A., Tamargo, J. L., and Zamorano, J. L. Acc/aha/esc 2006 guidelines for the management of patients with atrial fibrillation. *Circulation*, 114(7):e257–e354, 2006. doi: 10.1161/CIRCULATIONAHA.106.177292.

Gao, X., Jiang, B., and Zhang, S. On the information-adaptive variants of the admm: an iteration complexity perspective. *Journal of Scientific Computing*, 76:327–363, 2018.

Gautam, T., Park, Y., Zhou, H., Raman, P., and Ha, W. Variance-reduced zeroth-order methods for fine-tuning language models. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15180–15208. PMLR, 21–27 Jul 2024.

Han, Y. and Ding, C. Foundation models in electrocardiogram: A review. *arXiv preprint arXiv:2410.19877*, 2024.

Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., and Ng, A. Y. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65–69, 2019.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network, 2015.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Huang, T., You, S., Wang, F., Qian, C., and Xu, C. Knowledge distillation from a stronger teacher. *Advances in Neural Information Processing Systems*, 35:33716–33727, 2022.

Kashou, A. H., Basit, H., and Chhabra, L. *Electrical Right and Left Axis Deviation*. StatPearls Publishing, Treasure Island (FL), 2023.

Kelly, B. B., Fuster, V., et al. *Promoting cardiovascular health in the developing world: a critical challenge to achieve global health*. National Academies Press, 2010.

Lai, J., Tan, H., Wang, J., Ji, L., Guo, J., Han, B., Shi, Y., Feng, Q., and Yang, W. Practical intelligent diagnostic algorithm for wearable 12-lead ECG via self-supervised learning on large-scale dataset. *Nature Communications*, 14(1):3741, 2023.

Li, Z., Zhang, X., and Razaviyayn, M. Addax: Memory-efficient fine-tuning of language models with a combination of forward-backward and forward-only passes. In *5th Workshop on practical ML for limited/low resource settings*, 2024.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

Mathew, G., Barbosa, D., Prince, J., and Venkatraman, S. Foundation models for cardiovascular disease detection via biosignals from digital stethoscopes. *npj Cardiovascular Health*, 1(1):25, 2024.

Mc Namara, K., Alzubaidi, H., and Jackson, J. K. Cardiovascular disease as a leading cause of death: how are pharmacists getting involved? *Integrated pharmacy research and practice*, pp. 1–11, 2019.

McKeen, K., Oliva, L., Masood, S., Toma, A., Rubin, B., and Wang, B. ECG-FM: An open electrocardiogram foundation model. *arXiv preprint arXiv:2408.05178*, 2024.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Pai, S., Bontempi, D., Hadzic, I., Prudente, V., Sokač, M., Chaunzwa, T. L., Bernatz, S., Hosny, A., Mak, R. H., Birkbak, N. J., et al. Foundation model for cancer imaging biomarkers. *Nature machine intelligence*, 6(3):354–367, 2024.

Pham, M., Saeed, A., and Ma, D. C-MELT: Contrastive enhanced masked auto-encoders for ECG-language pre-training. *arXiv preprint arXiv:2410.02131*, 2024.

Reyna, M., Sadr, N., Gu, A., Perez Alday, E. A., Liu, C., Seyedi, S., Shah, A., and Clifford, G. Will two do? varying dimensions in electrocardiography: The physionet/computing in cardiology challenge 2021. *PhysioNet*, 2022. doi: 10.13026/34va-7q14.

Ribeiro, A. H., Ribeiro, M. H., Paixão, G. M., Oliveira, D. M., Gomes, P. R., Canazart, J. A., Ferreira, M. P., Andersson, C. R., Macfarlane, P. W., Meira Jr, W., et al. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nature communications*, 11(1):1760, 2020.

Ribeiro, A. L. P., Paixao, G. M., Gomes, P. R., Ribeiro, M. H., Ribeiro, A. H., Canazart, J. A., Oliveira, D. M., Ferreira, M. P., Lima, E. M., de Moraes, J. L., et al. Tele-electrocardiography and bigdata: the code (clinical outcomes in digital electrocardiography) study. *Journal of electrocardiology*, 57:S75–S78, 2019.

Sepahvand, M. and Abdali-Mohammadi, F. A novel method for reducing arrhythmia classification from 12-lead ECG signals to single-lead ECG with minimal loss of accuracy through teacher-student knowledge distillation. *Information Sciences*, 593:64–77, 2022.

Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

Spall, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 1992.

Strodthoff, N., Wagner, P., Schaeffter, T., and Samek, W. Deep learning for ECG analysis: Benchmarks and insights from PTB-XL. *IEEE Journal of Biomedical and Health Informatics*, 25(5):1519–1528, 2020.

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., and Liang, J. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

Vaid, A., Jiang, J., Sawant, A., Lerakis, S., Argulian, E., Ahuja, Y., Lampert, J., Charney, A., Greenspan, H., Narula, J., et al. A foundational vision transformer improves diagnostic performance for electrocardiograms. *NPJ Digital Medicine*, 6(1):108, 2023.

Wagner, P., Strodthoff, N., Bousseljot, R.-D., Kreiseler, D., Lunze, F. I., Samek, W., and Schaeffter, T. PTB-XL, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

Wang, X., Zhao, J., Marostica, E., Yuan, W., Jin, J., Zhang, J., Li, R., Tang, H., Wang, K., Li, Y., et al. A pathology foundation model for cancer diagnosis and prognosis prediction. *Nature*, pp. 1–9, 2024.

Zhang, Y., Li, P., Hong, J., Li, J., Zhang, Y., Zheng, W., Chen, P.-Y., Lee, J. D., Yin, W., Hong, M., Wang, Z., Liu, S., and Chen, T. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *Forty-first International Conference on Machine Learning*, 2024.

Zheng, J., Chu, H., Struppa, D., Zhang, J., Yacoub, S. M., El-Askary, H., Chang, A., Ehwerhemuepha, L., Abudayyeh, I., Barrett, A., et al. Optimal multi-stage arrhythmia classification approach. *Scientific reports*, 10(1):2898, 2020a.

Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., and Rakovski, C. A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific data*, 7(1):48, 2020b.

Zhou, R., Clifton, L., Liu, Z., Chan, K. W. Y., Clifton, D. A., Zhang, Y.-T., and Dong, Y. CE-SSL: Computation-efficient semi-supervised learning for ECG-based cardiovascular diseases detection. 2024.

Zhou, Y., Chia, M. A., Wagner, S. K., Ayhan, M. S., Williamson, D. J., Struyven, R. R., Liu, T., Xu, M., Lozano, M. G., Woodward-Court, P., et al. A foundation model for generalizable disease detection from retinal images. *Nature*, 622(7981):156–163, 2023.

A. Appendix.

A.1. Proof of Eq.(6)

Consider a model with trainable parameter θ and a batch size of N, we can calculate its gradients using gradient backpropagation and a loss function $\mathcal{L}$ as,

$$\nabla\mathcal{L}(\mathcal{B};\theta) = \frac{1}{N}\sum_{i=1}^{N}\nabla\mathcal{L}(x_i, y_i;\theta), \tag{12}$$

where x_i, y_i are ECG sample and its label within the mini-batch $\mathcal{B}$. The computational costs of gradient backpropagation are expensive as the model scales up. To address this, we can use the zeroth-order method to estimate the gradient as

$$\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n}\frac{\mathcal{L}(x_i, y_i;\theta+\mu z_j) - \mathcal{L}(x_i, y_i;\theta-\mu z_j)}{2\mu}z_j, \tag{13}$$

where $z \in \mathbb{R}^d$ is a random vector sampled from the standard Gaussian distribution $N(0, I_d)$, μ is the perturbation scale and n is the number of function queries. As shown in Malladi et al. (2023),

$$\mathbb{E}\left[\|\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)\|^2\right] = \frac{d+n-1}{n}\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B};\theta)\|^2\right] \tag{14}$$

where d is the number of trainable parameters in the model. In our study, we set $n = 1$ to ensure low latency in calculating the zeroth-order gradients, which yields that,

$$\begin{aligned}
\mathbb{E}\left[\|\widehat{\nabla}\mathcal{L}(\mathcal{B};\theta)\|^2\right] &= d\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B};\theta)\|^2\right] = d(\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B};\theta) - \mathbb{E}\left[\nabla\mathcal{L}(\mathcal{B};\theta)\right]\|^2\right] + \|\mathbb{E}\left[\nabla\mathcal{L}(\mathcal{B};\theta)\right]\|^2) \\
&= \frac{d}{N}(\mathbb{E}\left[\|\nabla\mathcal{L}(x_1, y_1;\theta)\|^2\right] - \|\nabla\mathcal{L}(\theta)\|^2) + d\|\nabla\mathcal{L}(\theta)\|^2 \\
&= \frac{N_1}{N_1}\frac{d}{N}(\mathbb{E}\left[\|\nabla\mathcal{L}(x_1, y_1;\theta)\|^2\right] - \|\nabla\mathcal{L}(\theta)\|^2) + d\|\nabla\mathcal{L}(\theta)\|^2 \\
&= \frac{dN_1}{N}\frac{1}{N_1}(\mathbb{E}\left[\|\nabla\mathcal{L}(x_1, y_1;\theta)\|^2\right] - \|\nabla\mathcal{L}(\theta)\|^2) + d\|\nabla\mathcal{L}(\theta)\|^2 \\
&= \frac{dN_1}{N}\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B}_1;\theta) - \mathbb{E}\left[\nabla\mathcal{L}(\mathcal{B}_1;\theta)\right]\|^2\right] + d\|\nabla\mathcal{L}(\theta)\|^2 \\
&\geq \frac{dN_1}{N}\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B}_1;\theta) - \mathbb{E}\left[\nabla\mathcal{L}(\mathcal{B}_1;\theta)\right]\|^2\right] + d\frac{N_1}{N}\|\nabla\mathcal{L}(\theta)\|^2 \\
&= \frac{dN_1}{N}\mathbb{E}\left[\|\nabla\mathcal{L}(\mathcal{B}_1;\theta)\|^2\right],
\end{aligned} \tag{15}$$

where $\mathcal{L}(\theta)$ is the true gradient of θ and $\mathcal{B}_1 = \{x_i, y_i\}_{i=1}^{N_1}$ is the tiny mini-batch for the proposed mix-order optimization ($N_1 \ll N$).

A.2. Algorithm of H-Tuning

The algorithm of the proposed H-Tuning is presented in Algorithm 1.

A.3. Implementation Details

This section introduces the details of the H-Tuning algorithm and the knowledge distillation process. We use the pre-trained model provided by Zhou et al. (2024) for ECG-based CVDs detection on the downstream dataset. It has 50.494 million parameters and is pre-trained on the Clinical Outcomes in Digital Electrocardiology (CODE) dataset (Ribeiro et al., 2019; 2020). The backbone consists of three convolution blocks, twelve self-attention blocks, and one classification block. The number of convolution channels within the convolution blocks and the hidden layer dimension control the number of trainable parameters. We fine-tuned the pre-trained model on downstream datasets using the proposed H-Tuning method. The fine-tuned model acts as the teacher model during the knowledge distillation process. In this study, the student model shares a similar architecture with the teacher model but only has 0.26 million trainable parameters. Specifically, the

Algorithm 1 The H-Tuning algorithm

Require:
- Labeled dataset $\mathcal{D} = \{X, Y\}$; Learning rate η; Batch sizes N; Batch sizes of the tiny subset N_1.
- The trainable parameters of the large pre-trained model θ; λ for the gradient refinement process.

Ensure: Fine-tuned large model with the updated parameters θ_T;

$\theta_1 = \theta$
for 1 to T **do**
  sample a random batch $\mathcal{B} = \{x_i, y_i\}_{i=1}^{N}$ from $\mathcal{D}$;
  apply data augmentation to $\mathcal{B}$;
  sample a tiny subset $\mathcal{B}_1 = \{x_i, y_i\}_{i=1}^{N_1}$ from $\mathcal{B}$;
  Simultaneous Perturbation Stochastic Approximation (SPSA)
  Based on Eq.(13), generate an initial estimation of the parameter gradients $\widehat{\nabla}\mathcal{L}(\mathcal{B}; \theta_t)$.
  Gradient Refinement
  Conduct the gradient backpropagation on the tiny subset $\mathcal{B}_1$ to calculate $\nabla\mathcal{L}(\mathcal{B}_1; \theta_t)$.
  Calculate $\widehat{\nabla}\mathcal{L}(\mathcal{B}; \theta)_\lambda$ using $\widehat{\nabla}\mathcal{L}(\mathcal{B}; \theta_t)$, $\nabla\mathcal{L}(\mathcal{B}_1; \theta_t)$ and the Eq.(5).
  Parameter Optimization
  Conduct the gradient descent process to update the parameters θ_t using Eq.(9).
  apply an early-stop strategy to avoid overfitting;
end for

convolution channels and the hidden layer dimension of the teacher model are 512, while the number is 64 for the student model. Specifically, Adam optimizer is utilized to conduct the gradient descent process defined in Eq.(9), with a learning rate of $\eta = 0.002$. The batch size N and N_1 defined in the proposed mix-order optimization are set to 128 and 2, respectively. Additionally, the controlling weight λ is searched within a set of $\{0.85, 0.90, 0.95, 0.99\}$. The perturbation scale μ for the SPSA process is searched within a set of $\{0.001, 0.0001\}$, and the number of queries n is set to 1. The rank r of the low-rank adaptation process is set to 16, and the number of deep layers M is set to 2.

For the comparison methods, the same pre-trained model is used for fine-tuning. Adam optimizer is adopted to update the model with a learning rate $\eta = 0.002$. To implement LP, we freeze all the layers within the model except the last two linear layers. The perturbation scales μ for MeZO, LoHO, and Addax are searched within a set of $\{0.001, 0.0001\}$, and the number of queries n is set to 1. The rank r of the compared methods with low-rank adaptation process (LoRA, MeZO + LoRA, Addax + LoRA, LoHO + LoRA) is set to 16. The training batch size of the proposed H-Tuning and the compared methods is set to 128. All the experiments are conducted in a single NVIDIA A6000 graphics processing unit using the Pytorch library.

A.4. Extended Experimental Results on More Evaluation Metrics.

- (1) **Detailed comparisons of H-Tuning and the baseline models on more evaluation metrics (Table 5).** Apart from macro AUC and macro $F_{\beta=2}$, four metrics on multi-label classification are used to evaluate the performance of different fine-tuning methods: ranking loss, coverage, mean average precision (MAP), macro $G_{\beta=2}$. For the ranking loss and coverage, a lower value indicates a better detection performance. However, for the MAP, macro AUC, macro $G_{\beta=2}$, and macro $F_{\beta=2}$, the greater the better. We also report the number of trainable parameters (Params) of different methods to measure their storage consumption during fine-tuning. The higher the number of trainable parameters, the higher the parameter storage consumption. The average time for each iteration during fine-tuning is also included for comparison (Time/iter). Experimental results on six evaluation metrics demonstrate the superiority of H-Tuning against other memory-efficient fine-tuning methods in CVDs detection. For instance, H-Tuning demonstrates an average MAP of 0.535 across four datasets, outperforming the best competitor (Addax) by 2.8%. Additionally, it can be observed that H-Tuning achieves comparable performance to fine-tuning methods with the first-order optimization (Full FT and LoRA) on three datasets but fails to achieve this in the Ningbo dataset. This phenomenon indicates that its performance might not be stable in certain datasets, which deserves further investigation in future works. As for the computational costs, H-Tuning consumes significantly less GPU memory than Full FT and LoRA. Similar to LoRA(Hu et al., 2022), it can fine-tune pre-trained models in a parameter-efficient manner, greatly decreasing the parameter storage costs by 23.7 times compared with the Full FT. One limitation is that H-Tuning cannot accelerate the fine-tuning process.

- (2) **Detailed ablation studies of H-Tuning on more evaluation metrics for multi-label CVDs detection (Table 6).** It can be observed that removing gradient estimation, refinement, or normalization module from the H-Tuning results in performance degradations on all datasets.
- (3) **Detailed CVDs detection performance under different ECG lead configurations on more evaluation metrics for multi-label CVDs detection (Table 7).** Under all ECG lead configurations, the knowledge distillation process increases the CVDs detection performance of the student models on all evaluation metrics. In Fig.4, we visualize the AUC of different student models on each CVDs from the four datasets. The CVDs analyzed in our study and their abbreviations can be found in Table 8. For example, the models with 1-lead ECG perform well in detecting atrial fibrillation (AF) but poorly in recognizing left axis deviation (LAD). This is because 1-lead ECG (lead I) has the ability to capture the patterns of determining AF, such as irregular RR intervals and the absence of P waves(Fuster et al., 2006). However, as the critical pattern of LAD is abnormal QRS complexes in lead II and lead aVF(Kashou et al., 2023), it is not visible in 1-lead ECG (lead I).

A.5. Performance Comparisons of Different Models under Limited Supervision

In the domain of medical intelligence, collecting labeled labels is expensive and time-consuming, limiting the sample size of the datasets for model fine-tuning. Therefore, evaluating the performance of different fine-tuning methods under minimal supervision is important, as it provides a stress test for their robustness in clinical practices. For the four datasets, we decrease the ratio between the training set and the held-out test set from 1:9 to 0.5:9.5, and present the average CVDs detection performance of different fine-tuning methods across the four datasets in Fig.5. For simplicity, we visualize the methods with the top eight performances. It can be observed that the superiority of H-Tuning compared with other memory-efficient methods persists under limited supervision. For instance, H-Tuning achieves an average MAP of 0.503 across the four datasets, outperforming Addax by 3.03%. More importantly, it outperforms the Full FT in ranking loss, coverage, and macro AUC. Besides, its performance losses compared with Full FT are within 1.5% for the remaining evaluation metrics. As shown in Fig.6, we utilize paired t-test to evaluate whether the performance differences between H-Tuning and all compared methods are significant. The results demonstrate that H-Tuning significantly outperforms other memory-efficient methods on six evaluation metrics, such as Addax. Additionally the performance differences between H-Tuning and memory-inefficient methods (Full FT, LoRA) are not significant on five metrics. These phenomena indicate the robustness of the proposed H-Tuning framework under different supervision levels, highlighting its effectiveness in clinical practices.

Table 5. Performance of H-Tuning and the compared models on four public datasets. The average performance and the standard deviation across four seeds are presented. The dashed lines separate the memory-efficient fine-tuning methods and the traditional fine-tuning methods.

Methods	Params	Memory	Time/iter	Ranking Loss ↓	Coverage ↓	Macro AUC ↑	MAP ↑	Macro $G_{\beta=2}$ ↑	Macro $F_{\beta=2}$ ↑
G12EC Dataset									
Full FT	50.493 M	9.214 GB	0.401 s	0.086±0.004	3.657±0.113	0.869±0.006	0.517±0.012	0.336±0.009	0.588±0.014
LoRA	2.135 M	8.754 GB	0.428 s	**0.084±0.005**	3.621±0.134	0.870±0.009	**0.522±0.008**	**0.336±0.007**	**0.592±0.011**
LP	0.272 M	1.416 GB	0.152 s	0.095±0.004	3.837±0.086	0.852±0.003	0.483±0.002	0.306±0.004	0.545±0.005
MeZO	50.493 M	1.815 GB	0.339 s	0.513±0.050	11.325±0.811	0.532±0.047	0.133±0.026	0.111±0.010	0.327±0.012
MeZO + LoRA	2.135 M	1.437 GB	0.386 s	0.479±0.010	10.761±0.187	0.507±0.025	0.115±0.009	0.103±0.004	0.316±0.009
Addax	50.493 M	2.000 GB	0.349 s	0.086±0.006	3.664±0.130	0.863±0.006	0.497±0.009	0.319±0.008	0.573±0.008
Addax + LoRA	2.135 M	1.453 GB	0.403 s	0.093±0.008	3.786±0.166	0.857±0.005	0.478±0.008	0.312±0.009	0.568±0.006
LoHO	50.493 M	2.002 GB	0.303 s	0.098±0.004	3.877±0.098	0.851±0.003	0.484±0.003	0.309±0.004	0.554±0.009
LoHO + LoRA	2.135 M	1.453 GB	0.341 s	0.095±0.002	3.825±0.029	0.851±0.002	0.480±0.002	0.308±0.003	0.557±0.009
H-Tuning	2.135 M	1.453 GB	0.409 s	0.085±0.004	**3.612±0.069**	**0.870±0.002**	0.506±0.006	0.330±0.005	0.586±0.010
PTB-XL Dataset									
Full FT	50.494 M	9.212 GB	0.413 s	0.026±0.002	2.550±0.069	0.919±0.002	**0.558±0.001**	0.387±0.006	0.618±0.011
LoRA	2.135 M	8.754 GB	0.442 s	0.026±0.001	2.537±0.057	0.919±0.006	0.558±0.002	0.385±0.006	0.618±0.006
LP	0.272 M	1.416 GB	0.155 s	0.032±0.001	2.721±0.025	0.897±0.002	0.509±0.001	0.354±0.003	0.582±0.006
MeZO	50.494 M	1.814 GB	0.347 s	0.368±0.135	9.890±2.169	0.515±0.021	0.116±0.023	0.104±0.014	0.241±0.024
MeZO + LoRA	2.135 M	1.444 GB	0.394 s	0.398±0.088	10.155±1.362	0.483±0.024	0.104±0.013	0.096±0.006	0.227±0.008
Addax	50.494 M	2.003 GB	0.358 s	0.031±0.005	2.677±0.163	0.898±0.009	0.510±0.014	0.354±0.010	0.578±0.015
Addax + LoRA	2.135 M	1.451 GB	0.408 s	0.029±0.001	2.628±0.044	0.899±0.004	0.505±0.007	0.361±0.006	0.582±0.007
LoHO	50.494 M	2.000 GB	0.311 s	0.031±0.001	2.679±0.027	0.902±0.002	0.517±0.002	0.360±0.003	0.588±0.004
LoHO + LoRA	2.135 M	1.453 GB	0.355 s	0.032±0.001	2.700±0.038	0.900±0.001	0.509±0.004	0.352±0.007	0.581±0.009
H-Tuning	2.135 M	1.453 GB	0.412 s	**0.025±0.001**	**2.470±0.034**	**0.923±0.003**	0.552±0.004	**0.393±0.003**	**0.628±0.002**
Ningbo Dataset									
Full FT	50.496 M	9.211 GB	0.428 s	**0.027±0.001**	**2.639±0.034**	0.933±0.002	**0.541±0.005**	**0.366±0.005**	**0.591±0.004**
LoRA	2.137 M	8.754 GB	0.457 s	0.027±0.001	2.661±0.035	**0.934±0.002**	0.536±0.006	0.355±0.006	0.580±0.008
LP	0.274 M	1.416 GB	0.172 s	0.034±0.001	2.893±0.040	0.906±0.001	0.462±0.002	0.314±0.006	0.530±0.008
MeZO	50.496 M	1.814 GB	0.359 s	0.509±0.056	13.938±1.235	0.504±0.031	0.091±0.018	0.086±0.012	0.231±0.017
MeZO + LoRA	2.137 M	1.444 GB	0.417 s	0.489±0.094	13.560±2.056	0.530±0.018	0.085±0.006	0.082±0.006	0.234±0.014
Addax	50.496 M	2.003 GB	0.370 s	0.041±0.010	3.155±0.377	0.909±0.018	0.443±0.045	0.303±0.019	0.509±0.029
Addax + LoRA	2.137 M	1.451 GB	0.425 s	0.040±0.005	3.128±0.188	0.901±0.016	0.426±0.029	0.293±0.014	0.495±0.022
LoHO	50.496 M	2.001 GB	0.328 s	0.034±0.001	2.916±0.030	0.907±0.001	0.466±0.003	0.316±0.006	0.535±0.012
LoHO + LoRA	2.137 M	1.453 GB	0.369 s	0.034±0.001	2.875±0.016	0.909±0.001	0.463±0.002	0.321±0.003	0.535±0.004
H-Tuning	2.137 M	1.453 GB	0.429 s	0.028±0.001	2.689±0.039	0.931±0.003	0.497±0.010	0.333±0.006	0.550±0.006
Chapman Dataset									
Full FT	50.492 M	9.211 GB	0.363 s	0.032±0.002	2.245±0.057	0.930±0.005	0.587±0.006	0.417±0.008	0.623±0.008
LoRA	2.134 M	8.754 GB	0.394 s	**0.029±0.001**	**2.189±0.036**	**0.932±0.003**	**0.596±0.003**	**0.428±0.002**	**0.636±0.005**
LP	0.271 M	1.416 GB	0.140 s	0.038±0.002	2.352±0.023	0.898±0.008	0.530±0.003	0.381±0.007	0.579±0.018
MeZO	50.492 M	1.814 GB	0.314 s	0.388±0.097	7.951±1.338	0.499±0.064	0.114±0.019	0.107±0.011	0.277±0.014
MeZO + LoRA	2.134 M	1.444 GB	0.361 s	0.518±0.062	9.926±1.011	0.481±0.009	0.129±0.020	0.115±0.009	0.278±0.004
Addax	50.492 M	2.004 GB	0.324 s	0.031±0.004	2.216±0.112	0.929±0.005	0.579±0.008	0.398±0.006	0.596±0.007
Addax + LoRA	2.134 M	1.451 GB	0.368 s	0.034±0.007	2.302±0.202	0.907±0.025	0.529±0.042	0.391±0.029	0.588±0.038
LoHO	50.492 M	1.998 GB	0.280 s	0.036±0.001	2.309±0.029	0.905±0.002	0.535±0.005	0.374±0.006	0.574±0.013
LoHO + LoRA	2.134 M	1.453 GB	0.313 s	0.035±0.001	2.303±0.021	0.904±0.001	0.541±0.003	0.382±0.002	0.580±0.005
H-Tuning	2.134 M	1.453 GB	0.381 s	0.030±0.002	2.195±0.034	0.929±0.002	0.586±0.011	0.420±0.005	0.634±0.009

Table 6. Detailed ablation study of H-Tuning.

Methods	Memory	Time/iter	Ranking Loss ↓	Coverage ↓	Macro AUC ↑	MAP ↑	Macro $G_{\beta=2}$ ↑	Macro $F_{\beta=2}$ ↑
G12EC Dataset								
Without SPSA gradient estimation	1.451 GB	0.388 s	0.090±0.002	3.760±0.067	0.866±0.002	0.500±0.004	0.324±0.006	0.575±0.009
Without gradient refinement	1.453 GB	0.341 s	0.095±0.002	3.825±0.029	0.851±0.002	0.480±0.002	0.308±0.003	0.557±0.009
Without gradient normalization	1.453 GB	0.395 s	0.091±0.005	3.715±0.061	0.870±0.002	0.504±0.005	0.328±0.005	**0.588±0.004**
Without low-rank adaptation	2.002 GB	0.357 s	0.096±0.003	3.900±0.038	0.856±0.005	0.492±0.006	0.318±0.006	0.564±0.012
H-Tuning	1.453 GB	0.409 s	**0.085±0.004**	**3.612±0.069**	**0.870±0.002**	**0.506±0.006**	**0.330±0.005**	0.586±0.010
PTB-XL Dataset								
Without SPSA gradient estimation	1.451 GB	0.394 s	0.027±0.002	2.531±0.078	0.919±0.005	0.542±0.009	0.379±0.009	0.616±0.010
Without gradient refinement	1.453 GB	0.355 s	0.032±0.001	2.700±0.038	0.900±0.001	0.509±0.004	0.352±0.007	0.581±0.009
Without gradient normalization	1.453 GB	0.403 s	0.027±0.002	2.553±0.064	0.917±0.003	0.537±0.006	0.376±0.005	0.609±0.007
Without low-rank adaptation	2.000 GB	0.368 s	0.032±0.004	2.727±0.140	0.907±0.006	0.526±0.013	0.364±0.010	0.588±0.015
H-Tuning	1.453 GB	0.412 s	**0.025±0.001**	**2.470±0.034**	**0.923±0.003**	**0.552±0.004**	**0.393±0.003**	**0.628±0.002**
Ningbo Dataset								
Without SPSA gradient estimation	1.451 GB	0.406 s	0.034±0.003	2.910±0.115	0.921±0.013	0.469±0.023	0.318±0.006	0.525±0.010
Without gradient refinement	1.453 GB	0.369 s	0.034±0.001	2.875±0.016	0.909±0.001	0.463±0.002	0.321±0.003	0.535±0.004
Without gradient normalization	1.453 GB	0.420 s	0.033±0.002	2.859±0.062	0.924±0.004	0.478±0.013	0.322±0.013	0.536±0.014
Without low-rank adaptation	2.000 GB	0.381 s	0.035±0.003	2.971±0.114	0.924±0.004	0.474±0.014	0.316±0.010	0.527±0.017
H-Tuning	1.453 GB	0.429 s	**0.028±0.001**	**2.689±0.039**	**0.931±0.003**	**0.497±0.010**	**0.333±0.006**	**0.550±0.006**
Chapman Dataset								
Without SPSA gradient estimation	1.451 GB	0.357 s	0.034±0.007	2.275±0.145	0.927±0.004	0.578±0.013	0.402±0.010	0.612±0.015
Without gradient refinement	1.453 GB	0.313 s	0.035±0.001	2.303±0.021	0.904±0.001	0.541±0.003	0.382±0.002	0.580±0.005
Without gradient normalization	1.453 GB	0.369 s	0.034±0.002	2.300±0.044	0.928±0.005	0.572±0.010	0.398±0.013	0.596±0.021
Without low-rank adaptation	2.001 GB	0.327 s	0.035±0.004	2.289±0.082	0.922±0.003	0.565±0.016	0.401±0.011	0.603±0.014
H-Tuning	1.453 GB	0.381 s	**0.030±0.002**	**2.195±0.034**	**0.929±0.002**	**0.586±0.011**	**0.420±0.005**	**0.634±0.009**

Table 7. Detailed CVDs detection performance of the student models with various numbers of ECG leads.

Teacher	Student	Ranking Loss ↓	Coverage ↓	Macro AUC ↑	MAP ↑	Macro $G_{\beta=2}$ ↑	Macro $F_{\beta=2}$ ↑
				G12EC Dataset			
None	12-Lead	0.091±0.002	3.779±0.077	0.847±0.004	0.481±0.010	0.311±0.008	0.558±0.003
12-Lead	12-Lead	**0.085±0.003**	3.627±0.079	0.868±0.003	**0.510±0.007**	**0.335±0.002**	0.582±0.005
None	3-Lead	0.102±0.003	4.050±0.087	0.834±0.003	0.465±0.007	0.295±0.005	0.537±0.005
12-Lead	3-Lead	0.088±0.002	3.712±0.046	0.860±0.002	0.497±0.004	0.325±0.003	0.574±0.008
None	1-Lead	0.159±0.015	5.120±0.312	0.772±0.011	0.351±0.014	0.225±0.013	0.457±0.018
12-Lead	1-Lead	0.133±0.009	4.685±0.194	0.795±0.011	0.399±0.012	0.260±0.009	0.495±0.012
Teacher's performance		0.085±0.004	**3.612±0.069**	**0.870±0.002**	0.506±0.006	0.330±0.005	**0.586±0.010**
				PTB-XL Dataset			
None	12-Lead	0.029±0.001	2.620±0.032	0.903±0.004	0.527±0.004	0.360±0.008	0.592±0.014
12-Lead	12-Lead	**0.024±0.000**	**2.448±0.016**	0.921±0.003	**0.559±0.005**	**0.394±0.003**	0.628±0.009
None	3-Lead	0.032±0.001	2.721±0.037	0.886±0.007	0.496±0.006	0.336±0.004	0.559±0.015
12-Lead	3-Lead	0.027±0.001	2.570±0.020	0.905±0.004	0.528±0.004	0.372±0.003	0.596±0.004
None	1-Lead	0.055±0.003	3.328±0.079	0.830±0.004	0.381±0.005	0.240±0.007	0.462±0.009
12-Lead	1-Lead	0.048±0.002	3.175±0.066	0.843±0.007	0.431±0.003	0.285±0.002	0.509±0.004
Teacher's performance		0.025±0.001	2.470±0.034	**0.923±0.003**	0.552±0.004	0.393±0.003	**0.628±0.002**
				Ningbo Dataset			
None	12-Lead	0.030±0.001	2.781±0.057	0.921±0.002	0.485±0.003	0.323±0.002	0.530±0.004
12-Lead	12-Lead	**0.026±0.001**	**2.623±0.047**	**0.937±0.002**	**0.520±0.007**	**0.350±0.007**	**0.568±0.010**
None	3-Lead	0.036±0.001	2.987±0.037	0.903±0.001	0.450±0.008	0.300±0.008	0.500±0.009
12-Lead	3-Lead	0.029±0.000	2.756±0.022	0.928±0.001	0.495±0.005	0.337±0.005	0.549±0.006
None	1-Lead	0.057±0.003	3.630±0.124	0.844±0.008	0.340±0.004	0.234±0.003	0.420±0.007
12-Lead	1-Lead	0.050±0.001	3.541±0.037	0.866±0.003	0.395±0.005	0.268±0.004	0.469±0.007
Teacher's performance		0.028±0.001	2.689±0.039	0.931±0.003	0.497±0.010	0.333±0.006	0.550±0.006
				Chapman Dataset			
None	12-Lead	**0.029±0.001**	**2.176±0.021**	0.927±0.002	0.574±0.004	0.410±0.011	0.617±0.008
12-Lead	12-Lead	0.030±0.001	2.197±0.031	**0.933±0.003**	**0.587±0.003**	**0.423±0.009**	0.632±0.009
None	3-Lead	0.037±0.001	2.356±0.041	0.911±0.007	0.532±0.016	0.381±0.010	0.577±0.020
12-Lead	3-Lead	0.033±0.001	2.283±0.022	0.919±0.003	0.559±0.006	0.408±0.004	0.614±0.011
None	1-Lead	0.053±0.004	2.677±0.124	0.847±0.007	0.441±0.010	0.318±0.006	0.498±0.011
12-Lead	1-Lead	0.054±0.002	2.779±0.045	0.862±0.004	0.471±0.008	0.346±0.004	0.538±0.001
Teacher's performance		0.030±0.002	2.195±0.034	0.929±0.002	0.586±0.011	0.420±0.005	**0.634±0.009**

Table 8. Description of the cardiovascular diseases analyzed in our study.

Original annotation	Abbreviations	Original annotation	Abbreviations
G12EC Dataset			
atrial fibrillation	AF	1st degree av block	IAVB
incomplete right bundle branch block	IRBBB	left axis deviation	LAD
left bundle branch block	LBBB	low qrs voltages	LQRSV
nonspecific intraventricular conduction disorder	NSIVCB	sinus rhythm	NSR
premature atrial contraction	PAC	prolonged qt interval	LQT
qwave abnormal	QAb	right bundle branch block	RBBB
sinus arrhythmia	SA	sinus bradycardia	SB
sinus tachycardia	STach	t wave abnormal	TAb
t wave inversion	TInv	ventricular premature beats	VPB
PTB-XL Dataset			
atrial fibrillation	AF	complete right bundle branch block	CRBBB
1st degree av block	IAVB	incomplete right bundle branch block	IRBBB
left axis deviation	LAD	left anterior fascicular block	LAnFB
left bundle branch block	LBBB	nonspecific intraventricular conduction disorder	NSIVCB
sinus rhythm	NSR	premature atrial contraction	PAC
pacing rhythm	PR	prolonged pr interval	LPR
qwave abnormal	QAb	right axis deviation	RAD
sinus arrhythmia	SA	sinus bradycardia	SB
sinus tachycardia	STach	t wave abnormal	TAb
t wave inversion	TInv		
Ningbo Dataset			
atrial flutter	AFL	bundle branch block	BBB
complete left bundle branch block	CLBBB	complete right bundle branch block	CRBBB
1st degree av block	IAVB	incomplete right bundle branch block	IRBBB
left axis deviation	LAD	left anterior fascicular block	LAnFB
low qrs voltages	LQRSV	nonspecific intraventricular conduction disorder	NSIVCB
sinus rhythm	NSR	premature atrial contraction	PAC
pacing rhythm	PR	poor R wave Progression	PRWP
premature ventricular contractions	PVC	prolonged qt interval	LQT
qwave abnormal	QAb	right axis deviation	RAD
sinus arrhythmia	SA	sinus bradycardia	SB
sinus tachycardia	STach	t wave abnormal	TAb
t wave inversion	TInv		
Chapman Dataset			
atrial fibrillation	AF	atrial flutter	AFL
1st degree av block	IAVB	left axis deviation	LAD
left bundle branch block	LBBB	low qrs voltages	LQRSV
nonspecific intraventricular conduction disorder	NSIVCB	sinus rhythm	NSR
premature atrial contraction	PAC	qwave abnormal	QAb
right axis deviation	RAD	right bundle branch block	RBBB
sinus bradycardia	SB	sinus tachycardia	STach
t wave abnormal	TAb	ventricular premature beats	VPB

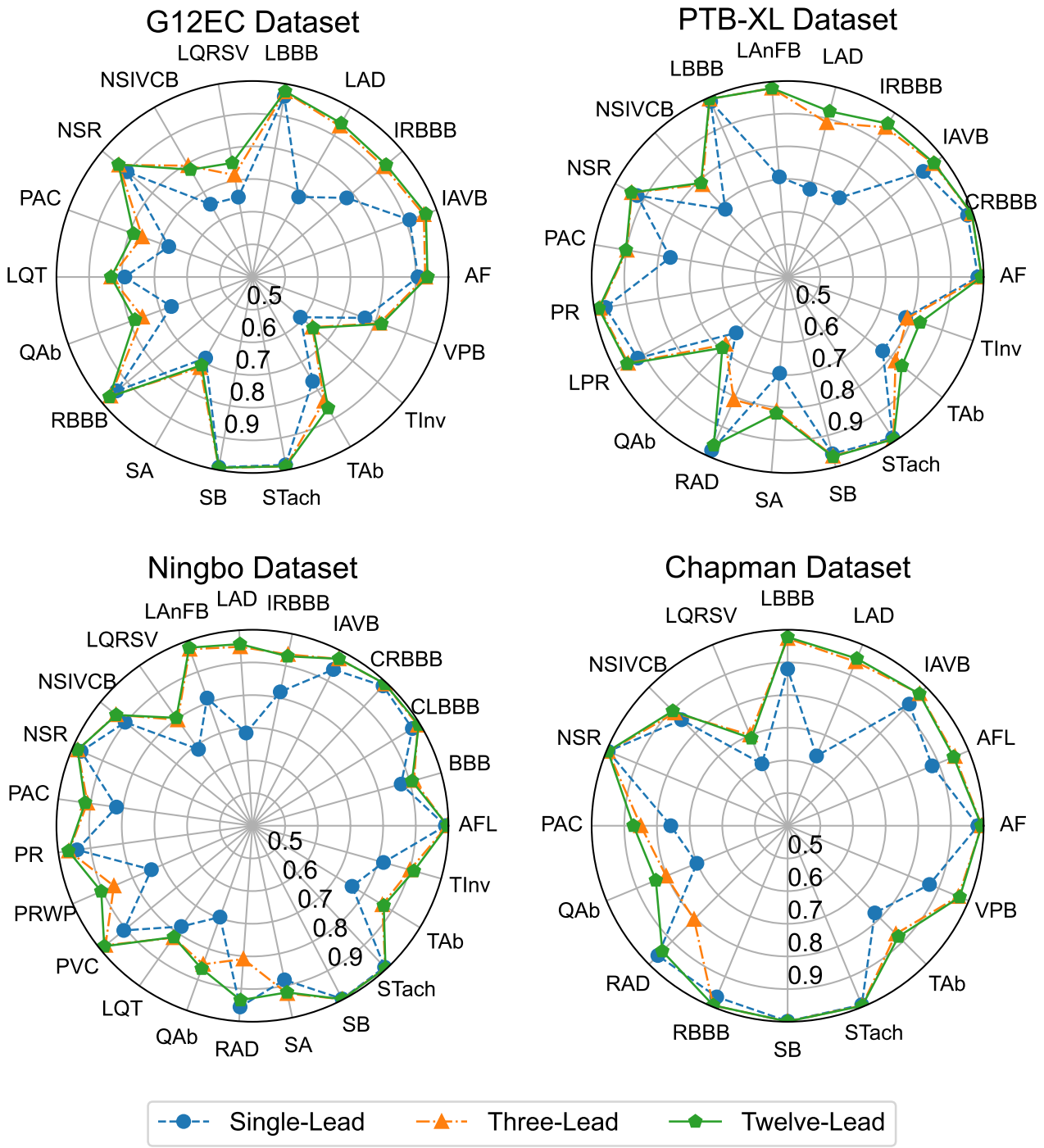


Figure 4. AUC of different student models on various CVDs. The green lines denote the student model trained with 12-lead ECG signals, the orange dashed-dotted lines denote the student model trained with 3-lead ECG signals, and the blue dashed lines denote the student model trained with 1-lead ECG signals.

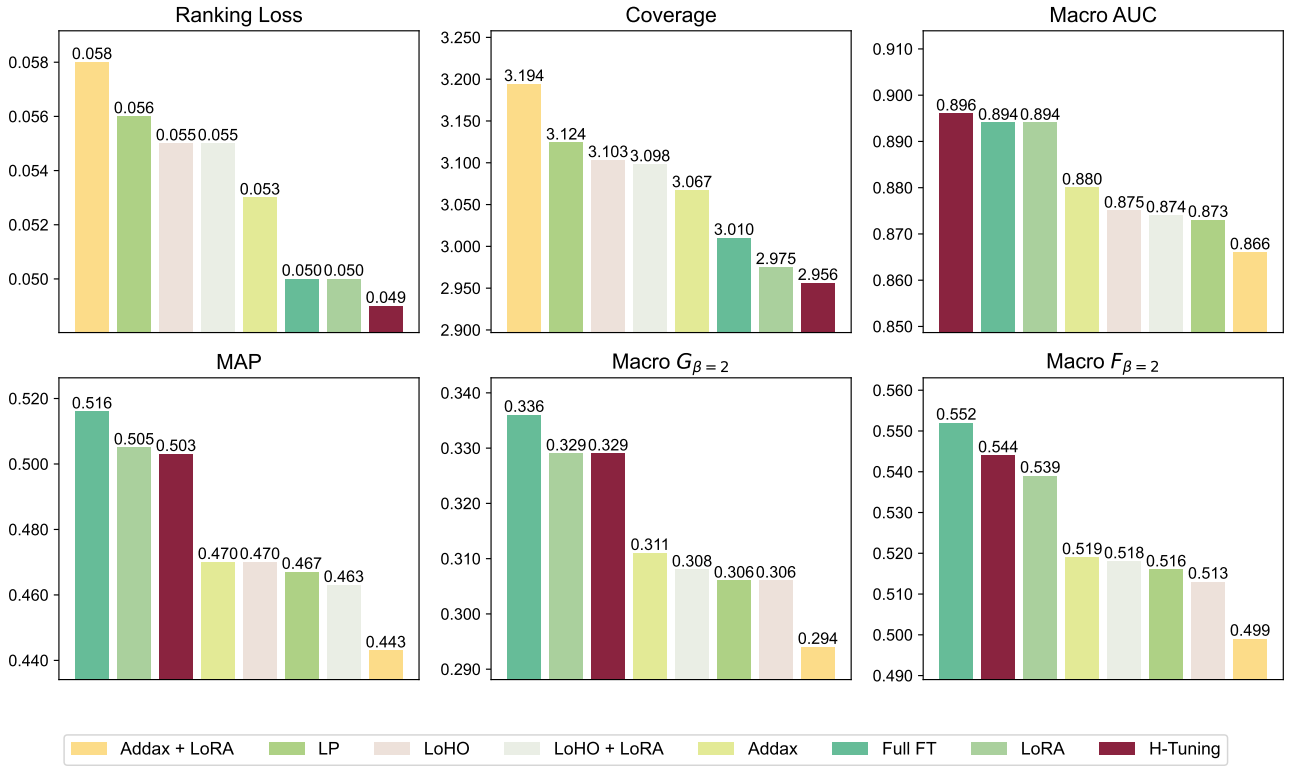


Figure 5. CVDs detection performance of different models under very limited supervision (training data: testing data = 0.5 : 9.5).

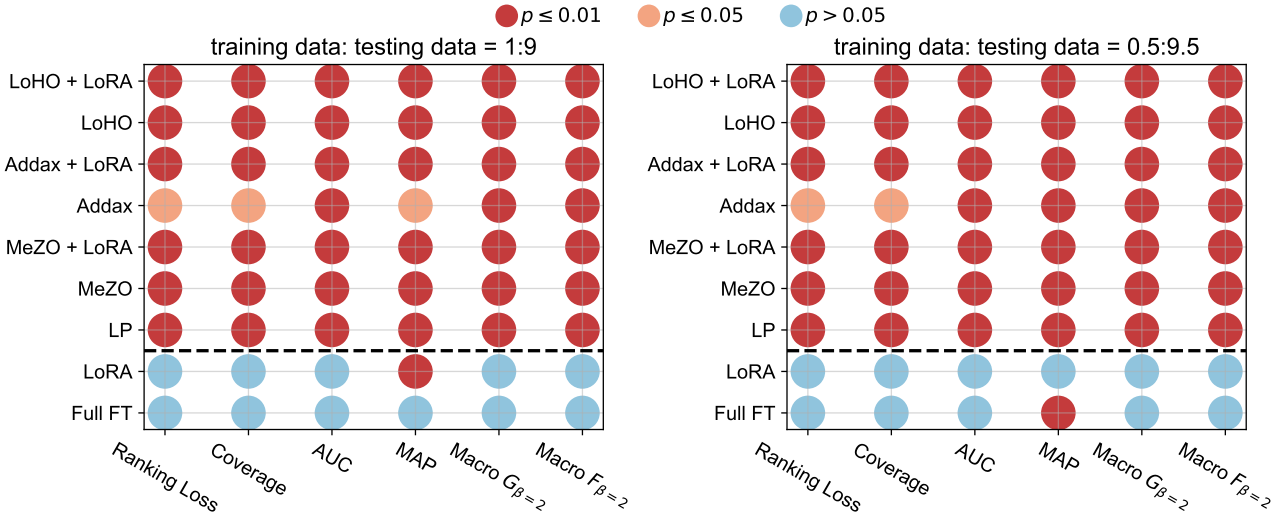


Figure 6. Paired t-test results for the model performance under different supervision levels. We check if the averaged performance differences on four datasets between H-Tuning and the compared methods are significant. Each circle indicates a t-test result between H-Tuning and a compared method after false discovery rate correction. The circle colors denote different significant levels. The black dashed lines separate the memory-efficient and memory-inefficient fine-tuning methods.

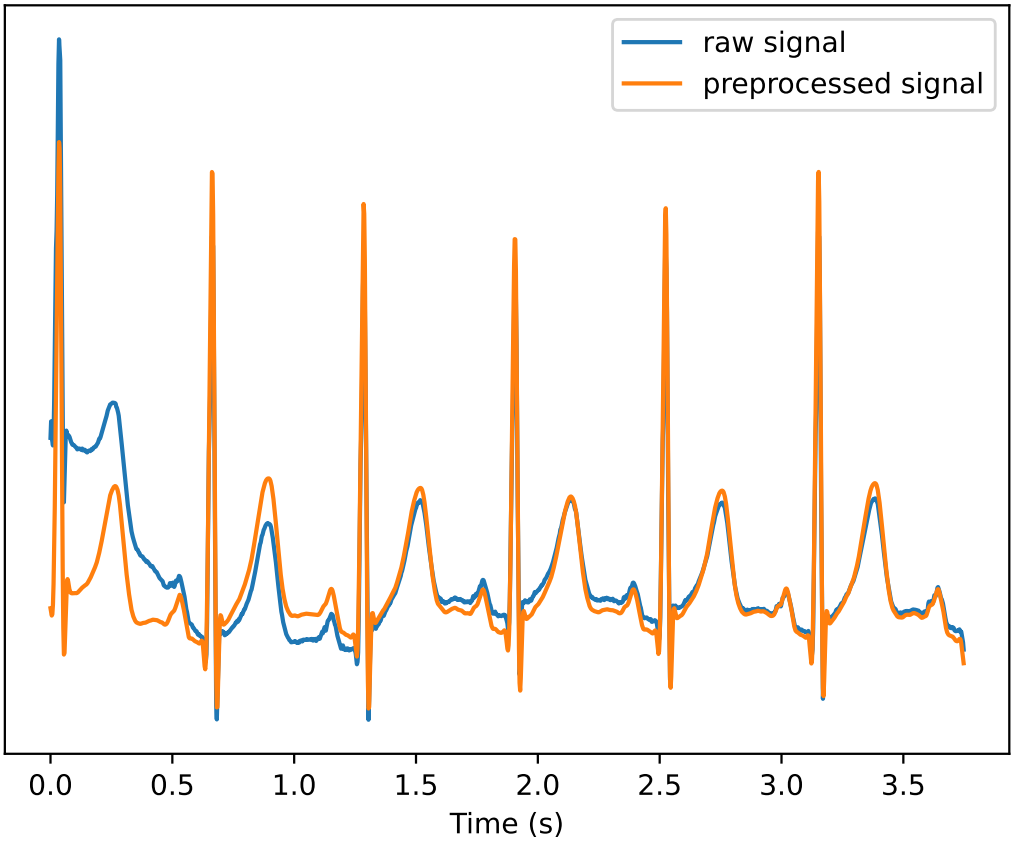


Figure 7. Visualization of the raw and the preprocessed lead I ECG .