Journal of Healthcare Informatics Research (2024) 8:506–522
https://doi.org/10.1007/s41666-024-00168-3

RESEARCH ARTICLE

MELEP: A Novel Predictive Measure of Transferability in Multi-label ECG Diagnosis

Cuong V. Nguyen[1] · Hieu Minh Duong[1] · Cuong D. Do[1,2]

Received: 17 November 2023 / Revised: 4 May 2024 / Accepted: 4 June 2024 /
Published online: 15 June 2024

Abstract

In practical electrocardiography (ECG) interpretation, the scarcity of well-annotated data is a common challenge. Transfer learning techniques are valuable in such situations, yet the assessment of transferability has received limited attention. To tackle this issue, we introduce MELEP, which stands for *Muti-label Expected Log of Empirical Predictions*, a measure designed to estimate the effectiveness of knowledge transfer from a pre-trained model to a downstream multi-label ECG diagnosis task. MELEP is generic, working with new target data with different label sets, and computationally efficient, requiring only a single forward pass through the pre-trained model. To the best of our knowledge, MELEP is the first transferability metric specifically designed for multi-label ECG classification problems. Our experiments show that MELEP can predict the performance of pre-trained convolutional and recurrent deep neural networks, on small and imbalanced ECG data. Specifically, we observed strong correlation coefficients (with absolute values exceeding 0.6 in most cases) between MELEP and the actual average F1 scores of the fine-tuned models. Our work highlights the potential of MELEP to expedite the selection of suitable pre-trained models for ECG diagnosis tasks, saving time and effort that would otherwise be spent on fine-tuning these models.

Keywords Electrocardiography · Computer-aided diagnosis · Transferability · Decision support systems

✉ Cuong D. Do
cuong.dd@vinuni.edu.vn

Cuong V. Nguyen
cuong.nv@vinuni.edu.vn

Hieu Minh Duong
hieu.dm@vinuni.edu.vn

[1] College of Engineering and Computer Science, VinUniversity, Hanoi, Vietnam

[2] VinUni-Illinois Smart Health Center, VinUniversity, Hanoi, Vietnam

1 Introduction

Electrocardiogram (ECG) signals are non-invasive and cost-effective tools for the early detection and accurate diagnosis of heart-related disease, one of the leading causes of death worldwide. Early diagnosis and treatment can improve patient outcomes, making ECG signals essential for improving health and well-being. Recently, automatic ECG interpretation has gained significant popularity and witnessed remarkable progress. This advancement can be attributed to the wide-scale digitization of ECG data and the evolution of deep learning techniques. Notably, deep neural networks (DNN) have achieved classification performance on par with cardiologists, as demonstrated by Hannun et al. [1] and Ribeiro et al. [2]. These outstanding achievements have partly been due to the availability of extensive human-labeled datasets, consisting of 91,232 and 2,322,513 ECG recordings, respectively. However, ECG datasets used in practice are often much smaller, due to the expensive and time-consuming data collection and annotation process. Consequently, it becomes challenging to achieve desirable results when training DNNs from scratch. Transfer learning is often useful in such scenarios, resulting in improved performance [3, 4] and faster convergence [5]. Fortunately, there exists some large, publicly available ECG datasets, which enable DNNs to learn important latent features, then transfer the learned knowledge to our main task, typically with much less annotated data. There are two most commonly used transfer learning techniques: head retraining [3, 6] and fine-tuning [7, 8]. Both replace the top classification layer to match the number of target task's outputs; however, whereas the former freezes all feature extractor layers and only updates the top layer's parameters during training on the target dataset, the latter does not have such a constraint and makes all layers trainable. Research suggested that fine-tuning leads to better performance [4, 9, 10], thus it has been accepted as a de facto standard.

Given the effectiveness of fine-tuning, a new problem arises: how do we select the best pre-trained checkpoints among a large candidate pool? A checkpoint is a model pre-trained on a source dataset, with a specific set of hyperparameter settings. It is straightforward to actually do the fine-tuning and then select the top ones; however, this method is obviously expensive and difficult to scale. Transferability estimation [11, 12] aims to address the above bottleneck by developing a metric that indicates how effectively transfer learning can apply to the target task, ideally with minimal interaction with it. Good estimation is likely to facilitate the checkpoint selection process. In the domain of computer vision, several transferability measures were developed. Tran et al. [39] introduced negative conditional entropy between the source and target label sets. Bao et al. [14] proposed a transferability measure called H-score, which was based on solving a Maximal HGR Correlation problem [15–17]. Nguyen et al. [9] and Huang et al. [18] developed LEEP and TransRate, respectively, two efficient estimates with no expensive training on target tasks. However, those measures only apply to multi-class classification problems and thus cannot be directly applicable to multi-label tasks such as ECG diagnosis, in which a patient may suffer from more than one cardiovascular disease.

Key Contributions

- We propose MELEP, a transferability measure that can directly apply to multi-label classification problems in automatic ECG interpretation. To the best of our knowledge, we are the first to develop such a measure to estimate the effectiveness of transfer learning for multi-label ECG.
- We conducted the first extensive experiment of transfer learning for 12-lead ECG data. We focused on small downstream datasets and covered a wide range of source checkpoints, which were produced from multiple source datasets and representatives of the two most popular DNN architectures for time-series analysis: convolutional and recurrent neural networks.

Our article is structured as follows: first, we provide the mathematical foundation behind MELEP and describe the intuition and its properties. Then, four 12-lead ECG datasets and two DNN architectures are introduced, which build the backbone of our experiments. We evaluate the ability of MELEP to predict the fine-tuning performance of a convolutional neural network by conducting extensive experiments with multiple checkpoints produced from pre-training the model on different source datasets. To show the versatility of MELEP, we replicate the experiment with a recurrent neural network, affirming that its capability is not tied to a specific model architecture. Next, we demonstrate the effectiveness of MELEP in a real-world scenario, which is selecting the best checkpoints among a group of pre-trained candidates. Finally, we discuss some notable properties, extensions, and applications of MELEP and suggest promising directions for future study.

2 Materials and Methods

2.1 Multi-label Expected Log of Empirical Predictions (MELEP)

Consider transfer learning from one multi-label classification task to another.

Let:

- Θ be the pre-trained model on the source task.
- $\mathcal{L}_s = \{0, 1, ..., \mathcal{Z}-1\}$ be the source label set of size $|\mathcal{L}_s| = \mathcal{Z}$.
- $\mathcal{L}_t = \{0, 1, ..., \mathcal{Y}- 1\}$ be the target label set of size $|\mathcal{L}_t| = \mathcal{Y}$.
- $\mathcal{D} = \{(x_1, \mathbf{y_1}), ..., (x_n, \mathbf{y_n})\}$ be the target dataset of size n. $\mathbf{y_i}$ is a label vector of size $\mathcal{Y}$.
- $(y, z) \in \mathcal{L}_t \times \mathcal{L}_s$ be a pair of target-source labels taken from the two sets.
- (t, s) be the values of (y, z). In the ECG classification context, the label values are binary, so $(t, s) \in \{0, 1\} \times \{0, 1\}$.

Then, MELEP is computed as follows:

1. Step 1: Compute the dummy label distributions of the target data over the source label set, denoted by a vector $\hat{\mathbf{y}}_i = \Theta(x_i)$ of size $\mathcal{Z}$, by forward passing each data point to the pre-trained model.
2. Step 2: Consider each pair of target-source labels (y, z). Let θ_{iz} denote the value of $\hat{\mathbf{y}}_i$ at the z^{th} column, i.e., the predicted probability that the sample x_i belongs to label z.

(a) Compute its 2 × 2 empirical joint distribution matrix $\hat{\mathbf{P}}_{yz}(t, s)$, with value at row t column s is as follows:

$$\hat{P}_{yz}(t, s) = \frac{1}{n} \sum_{i:y_{iz}=t} (\theta_{iz})_s \tag{1}$$

where $\sum_{i:y_{iz}=t}$ means we select all samples x_i with the z^{th} ground-truth label y_{iz} equal to t. With corresponding values of s, $(\theta_{iz})_1$ and $(\theta_{iz})_0$ are the probabilities that the label z can and cannot be assigned to the sample x_i, respectively.

(b) Compute the empirical marginal distribution vector (of size 2) with respect to the source label z:

$$\begin{aligned}\hat{P}_z(s) &= \frac{1}{n} \sum_{i=1}^{n} (\theta_{iz})_s \\ &= \hat{P}_{yz}(0, s) + \hat{P}_{yz}(1, s) \end{aligned} \tag{2}$$

(c) Compute the 2 × 2 empirical conditional distribution matrix $\hat{\mathbf{P}}_{y|z}(t, s)$ of the target label y given the source label z, with value at row t column s is as follows:

$$\hat{P}_{y|z}(t|s) = \frac{\hat{P}_{yz}(t, s)}{\hat{P}_z(s)} \tag{3}$$

For any input x_i, consider a binary classifier that predicts whether x_i belongs to label y by first randomly drawing $\mathcal{Z}$ dummy labels from $\Theta(x_i)$, then averaging the likelihood of y based on $\mathcal{Z}$ empirical conditional distributions $\hat{\mathbf{P}}_{y|z}$. This process is repeated for all $\mathcal{Y}$ target labels. The set of binary classifiers is called the ***E**mpirical **P**redictor* (EP). MELEP is defined as the average negative log-likelihood of the EP across all target labels, as follows:

3. Step 3: Compute the Expected Logarithm of Empirical Prediction with respect to the label pair (y, z):

$$\phi(\Theta, \mathcal{D}, y, z) = -\frac{1}{n} \sum_{i=1}^{n} \log \left(\sum_{s=0}^{1} \hat{P}_{y|z}(y_{iz}|s)(\theta_{iz})_s \right) \tag{4}$$

4. Step 4: Compute MELEP by taking the weighted average of $\phi(\theta, \mathcal{D}, y, z)$ over all target-source label pairs:

$$\Phi(\Theta, \mathcal{D}) = \frac{1}{\mathcal{Y}} \sum_{y} w_y \times \frac{1}{\mathcal{Z}} \sum_{z} \phi(\Theta, \mathcal{D}, y, z) \tag{5}$$

where w_y are the weights of the target label y in the target dataset, i.e., the ratio of the number of positive samples to the number of negative samples of y. Note that we do not take the source weights into consideration, because in practice, it makes sense to assume that we do not know the source label distribution prior to fine-tuning.

From its definition, MELEP is always positive, and smaller values indicate superior transferability. Intuitively, MELEP can be regarded as a distance metric, indicating how "close" the pre-trained model Θ and the target dataset $\mathcal{D}$ are. The closer the distance, the better the transfer.

The measure is *generic*, meaning that it can be applied to all types of checkpoints, and works without any prior knowledge of the pre-training process, such as data distribution, hyperparameter settings, optimizer, and loss functions. Furthermore, the computation of MELEP is *efficient*, which renders it practically useful. This lightweight property is inherited from the original LEEP [9], with the calculation involving only a single forward pass through the target dataset $\mathcal{D}$, requiring no training on the downstream task.

2.2 Datasets

We used publicly available 12-lead ECG datasets in this work. The first source was the public training dataset from the China Physiological Signal Challenge 2018 (CPSC2018) [19]. This dataset comprises 6877 ECG records, each associated with at most nine diagnostic categories: NORM (representing normal ECG patterns), AF (Atrial Fibrillation), I-AVB (First-degree atrioventricular block), LBBB (Left Bundle Branch Block), RBBB (Right Bundle Branch Block), PAC (Premature Atrial Contraction), PVC (Premature ventricular contraction), STD (ST-segment Depression), and STE (ST-segment Elevated).

The second dataset was PTB-XL [20], containing 21,837 records from 18,885 patients and a total of 44 diagnostics statements. The dataset's authors organized these diagnostic labels into a hierarchical structure [21], categorizing the 44 labels into five broader superclasses: NORM (normal ECG), MI (Myocardial Infarction), STTC (ST/T-Changes), HYP (Hypertrophy), and CD (Conduction Disturbance). We followed this structure and focused on these five superclasses when conducting experiments with the PTB-XL dataset.

Our third dataset, known as the Georgia dataset [22], consists of 10,344 ECGs that reflect the demographic characteristics of the Southeastern United States. The data covers a diverse range of 67 unique diagnoses. However, for our research, we focused on a subset of ten specific labels, which had the most substantial number of samples: NORM, AF, I-AVB, PAC, SB (Sinus Bradycardia), LAD (left axis deviation), STach (Sinus Tachycardia), TAb (T-wave Abnormal), TInv (T-wave Inversion), and LQT (Prolonged QT interval).

The last source was the Chapman University, Shaoxing People's Hospital, and Ningbo First Hospital database [23, 24], which we will refer to as the CSN dataset for brevity. This dataset contains 45,152 12-lead ECG records, each lasting for 10 s and sampled at 500 Hz. There are a total of 94 unique labels, among which we focused on 20 labels with more than 1000 records for our experiments. These 20 labels are SB, NORM, STach, TAb, TInv, AF, STD, LAD, PAC, I-AVB, PVC, AFL (Atrial Flutter), LVH (Left Ventricular Hypertrophy), STC (S-T changes), SA (Sinus Arrhythmia), LQRSV (Low QRS Voltages), PR (pacing rhythm), NSTTA (Nonspecific ST-T Abnormality), CRBBB (complete Right Bundle Branch Block), and QAb (Q-wave Abnormal).

Table 1 summarizes key statistics of the four data sources. In terms of data preprocessing, we applied the following procedures:

- Downsampling: We reduced the sampling frequency of all ECG records from 500 to 100 Hz. This helps reduce computational load while retaining essential information.
- Cropping: For ECG records longer than the desired duration (ten seconds), we cropped them to meet this target by keeping only the first 10-s data points. This step ensures that all records have consistent lengths for training.

It is worth noting that only a tiny fraction of records have durations shorter than 10 s: six out of 6877 in the CPSC2018 dataset, 52 out of 10,334 in the Georgia dataset, and none in the PTB-XL and CSN datasets. Therefore, instead of padding these records to meet the desired duration, which could potentially introduce unwanted noise or artifacts into the signals, they were simply omitted from our experiments.

We used the CSN and PTB-XL datasets for fine-tuning due to their relatively large amount of records. When fine-tuning models on the former, we pre-trained our models using three source datasets: CPSC2018, PTB-XL, and Georgia. When fine-tuning on the latter, we only used two source datasets: CPSC2018 and Georgia.

For pre-training, we partitioned each of the source datasets into training and test sets as follows. For PTB-XL, we followed the recommended split in [20], pre-training our models on the first eight folds, and testing on the tenth fold. For the CPSC2018 and Georgia datasets, we kept 33% of the amount of data in the test set and allocated the remaining for pre-training.

2.3 Deep Learning Models

We investigated two widely used deep learning architectures for time-series analysis:

- Convolutional neural network (CNN): We utilized ResNet1d101, which is a 1D variant of ResNet101 [25]. The architecture of the ResNet1d101 model is illustrated in Fig. 1. The ResNet family was originally introduced to work with

Table 1 Statistics of datasets used in this work

Dataset	Number of records	Number of labels	Sampling rate (Hz)	Duration (sec)
CPSC2018 [19]	6877	9	500	6–60
PTB-XL [20]	21,837	44	500 & 1000	10
Georgia [22]	10,344	62	500	10
CSN [23, 24]	45,152	94	500	10

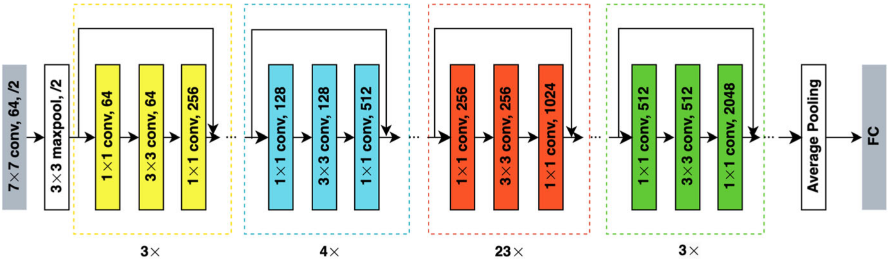


Fig. 1 ResNet1d101 architecture

image data, performing well in healthcare applications such as medical imaging [26–29]. Yet their power of capturing useful patterns in data still demonstrates strong performance when applied to time-series ECG data [30, 31].

- Recurrent neural network (RNN): The bidirectional long short-term memory (Bi-LSTM) architecture [32] was used. The structure of the Bi-LSTM model is visually presented in Fig. 2. LSTM is also a popular choice when dealing with ECG data, as it is capable of capturing long-term dependencies within the sequences [33–36].

Since the source datasets have varying numbers of labels, the last fully connected layer of the models was adjusted to align with the respective number of outputs. During pre-training, each model was trained on a source training set for 50 epochs, using Adam optimizer [37] with a learning rate of 0.01. At the end of each epoch, we recorded the average F1 score on the test set, which served as an early stopping criterion. We observed that Bi-LSTM experienced overfitting when training beyond the early stopping point, whereas ResNet1d101 mostly converged.

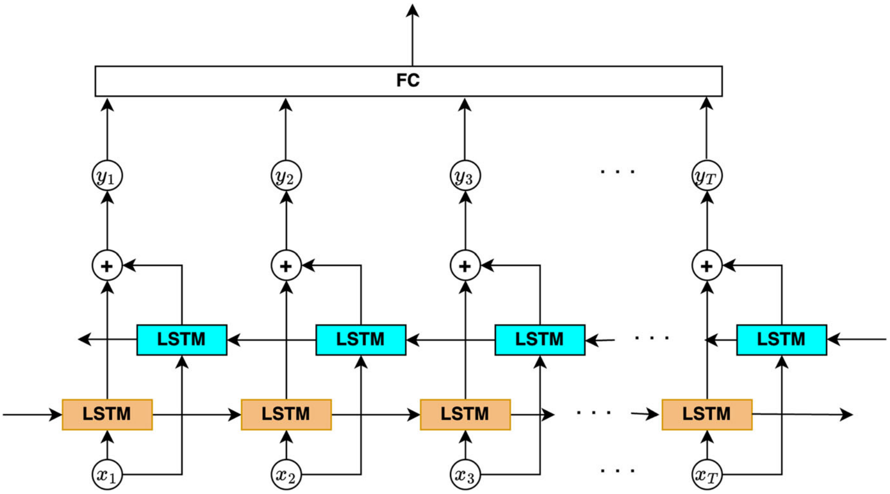


Fig. 2 Bi-LSTM architecture

2.4 Performance Metric

To evaluate the fine-tuning performance of a pre-trained model, we relied on the weighted average F1 score across all labels in the target dataset. F1 score was chosen as the evaluation metric due to its robustness in handling class imbalances [38], a common feature of ECG data, compared to accuracy.

Suppose that each ECG record in the target dataset has N diagnostic labels. Let denote:

- TP_i: true positives for the i^{th} label
- FP_i: false positives for the i^{th} label
- FN_i: false negatives for the i^{th} label

Then, the precision (P_i), recall (R_i), and F1 score (F_{1_i}) for the i^{th} label are computed as follows:

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

$$R_i = \frac{TP_i}{TP_i + FN_i}$$

$$F_{1i} = \frac{2 \times P_i \times R_i}{P_i + R_i}$$

The performance metric is computed as the weighted average F1 score across all labels within the target dataset, given by the following:

$$\text{Average F1} = \sum_{i=1}^{N} w_i \times F_{1i}$$

where the weight w_i represents the ratio of true instances of the i^{th} label.

3 Experiments and Results

In this section, we show the potential of MELEP in predicting the performance of fine-tuning a pre-trained model on a target dataset. In practice, transfer learning is often used when dealing with limited human-annotated data. Therefore, we focused on investigating MELEP in the context of small target datasets. The code and resources used for experiments can be found at github.com/cuongvng/melep-ecg.

3.1 Relation Between MELEP and Model Performance of CNN Fine-Tuned on CSN

We first experimented with the convolutional model ResNet1d101. This model was pre-trained on three different source datasets: PTB-XL, CPSC2018, and Georgia, as described in Section 2.2, resulting in three respective source checkpoints.

Each source checkpoint was then undergone an experiment with a wide range of target tasks sampled from the CSN dataset. To construct these tasks, we started with the set of 20 labels in the CSN dataset with at least 1000 positive samples, as in Section 2.2. N labels were then randomly sampled without replacement from the set, where N varied from 2 to 10. This step ensured that the target tasks would cover a diverse set of target labels. Records with no positive values for the N selected labels were filtered out to avoid creating a sparse dataset and to guarantee that every sample left contained at least one positive label. We then randomly select 1000 records among the remaining to form a data fold. The process was repeated 100 times to generate a total of 100 data folds for our experiment.

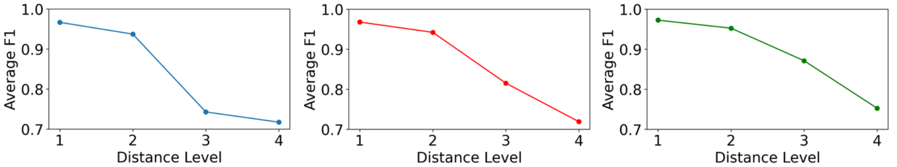
For each fold, we further split it into training and test subsets with a 7:3 ratio, i.e., 700 training records and 300 test records. Subsequently, we compute MELEP using the pre-trained checkpoint and the training subset only, following the algorithm described in Section 2.1. Prior to fine-tuning the model, we replaced the top fully connected layer of the checkpoint, adjusting the number of output neurons to match the target number of labels N. The entire modified model was then fine-tuned on the training subset for 50 epochs with early stopping, using Adam optimizer [37], and then evaluated on the test subset using weighted average F1 score across the N labels of the given fold. Ultimately, we gathered 100 points of (MELEP, average F1) representing the correlation between MELEP and the fine-tuning performance of the source checkpoint across a wide range of target tasks.

We then performed the Pearson correlation analysis between MELEP and the performance, following a similar approach used in assessing transferability on multi-class computer vision tasks [9, 13]. The first three rows in Table 2 show the results of the three ResNet1d101 checkpoints in this experiment, revealing strong negative correlations between MELEP and average F1 scores, all of which are below −0.6. To visualize this relationship, Fig. 3 classifies the MELEP values into four distinct distance levels. Within each level, we calculated the mean of average F1 scores from all the folds with

Table 2 Pearson correlation coefficients between MELEP and average of F1 scores in the experiments described in Sections 3.1, 3.2 and 3.3

Model	Source data	Target data	Details in	Pearson (r)	Correlation	p-value
ResNet1d101	PTB-XL	CSN	Sec. 3.1	−0.639		8.1×10^{-13}
	CPSC2018	CSN	Sec. 3.1	−0.631		2.0×10^{-12}
	Georgia	CSN	Sec. 3.1	−0.608		1.9×10^{-11}
	CPSC2018	PTB-XL	Sec. 3.3	−0.476		5.7×10^{-7}
	Georgia	PTB-XL	Sec. 3.3	−0.500		1.1×10^{-7}
Bi-LSTM	PTB-XL	CSN	Sec. 3.2	−0.691		1.7×10^{-15}
	CPSC2018	CSN	Sec. 3.2	−0.670		2.6×10^{-14}
	Georgia	CSN	Sec. 3.2	−0.665		4.2×10^{-14}
	CPSC2018	PTB-XL	Sec. 3.3	−0.551		2.8×10^{-9}
	Georgia	PTB-XL	Sec. 3.3	−0.517		3.5×10^{-8}

Strong negative correlations were observed for most cases, indicating MELEP's potential to predict fine-tuning performance with only a single forward pass required. All correlations are statistically significant



(a) PTB-XL → CSN (b) CPSC2018 → CSN (c) Georgia → CSN

Fig. 3 Relation of MELEP (partitioned as four distance levels) and fine-tuning performance of ResNet1d101 on target tasks sampled from the CSN dataset. The lower the MELEP (the closer the distance), the better transferability

MELEP falling into that level. The lower the MELEP, the closer the distance, implying easier transferability.

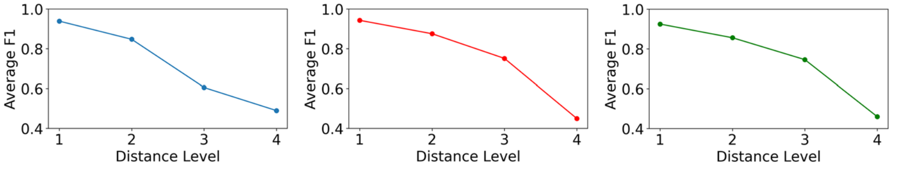
3.2 Relation Between MELEP and Model Performance of RNN Fine-Tuned on CSN

To illustrate the applicability of MELEP to RNN, we repeated the experiment in Section 3.1 with Bi-LSTM as the source model. Similar to ResNet1d101, the Bi-LSTM model was pre-trained on three source datasets: PTB-XL, CPSC2018, and Georgia. We leveraged the same set of 100 CSN data folds which were previously constructed for the CNN experiment, and applied the identical fine-tuning procedure.

In Table 2 (specifically, the first three rows of the Bi-LSTM section), we observe a robust correlation, even stronger than that observed with ResNet1d101, between MELEP and average F1 scores. This correlation is visually depicted in Fig. 4, where MELEP is categorized into the same distance levels as described in Section 3.1. The trend remains consistent: the closer the distance, the better the transfer.

3.3 Relation Between MELEP and Performance of Models Fine-Tuned on PTB-XL

In this experiment, we explored the use of MELEP on a different target dataset, specifically PTB-XL, chosen for its relatively large amount of records. We followed the same procedure outlined in Section 3.1 to construct 100 target data folds, with the only difference being the number of labels N. These label sets ranged from two to five and were



(a) PTB-XL → CSN (b) CPSC2018 → CSN (c) Georgia → CSN

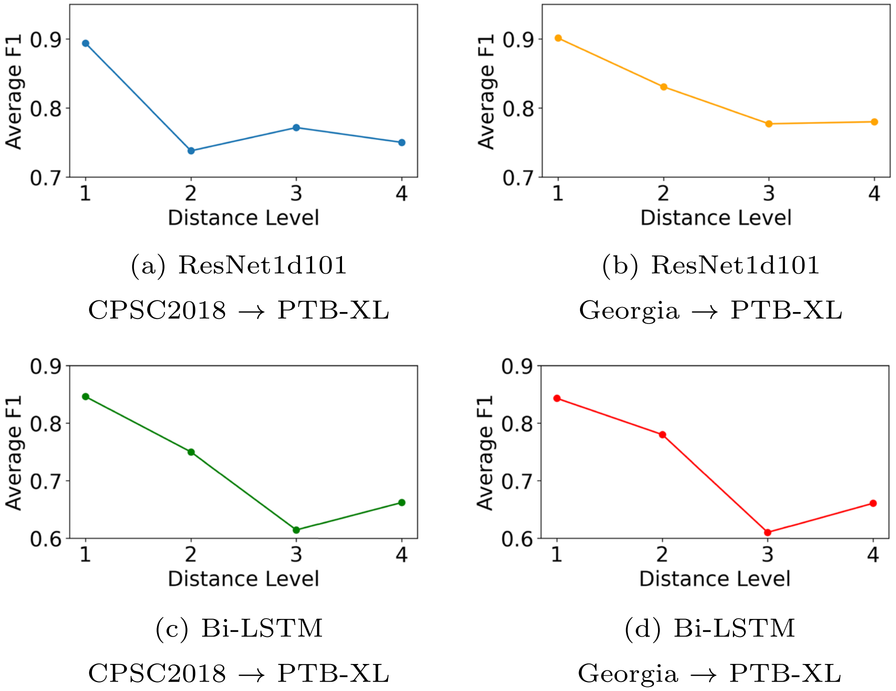
Fig. 4 Relation of MELEP (partitioned as four distance levels) and fine-tuning performance of Bi-LSTM on target tasks sampled from the CSN dataset. The lower the MELEP (the closer the distance), the better transferability

derived from the five superclasses covering the whole PTB-XL dataset, as described in Section 2.2.

We considered four different checkpoints: ResNet1d101 and Bi-LSTM models pre-trained on the CPSC2018 and Georgia datasets. The results in Table 2 indicate a moderate correlation between MELEP and transfer performance, with most correlation coefficients below −0.5. These correlations, while still significant, are slightly weaker than what was observed in the experiment with the CSN dataset (Sections 3.1 and 3.2), as shown in Fig. 5, where the predictive trend of MELEP is disrupted, with an increase instead of a decrease at one distance level (the 2^{nd} level for ResNet1d101 pre-trained on CPSC2018 and the 3^{rd} level for other checkpoints).

3.4 MELEP for Checkpoint Selection

This experiment demonstrates the use of MELEP in practice to effectively estimate fine-tuning performance in a multi-label classification task before the actual fine-tuning process takes place. Consider a checkpoint selection problem, where the goal is to choose the best candidate from a set of given source checkpoints for a target task.



(a) ResNet1d101 CPSC2018 → PTB-XL

(b) ResNet1d101 Georgia → PTB-XL

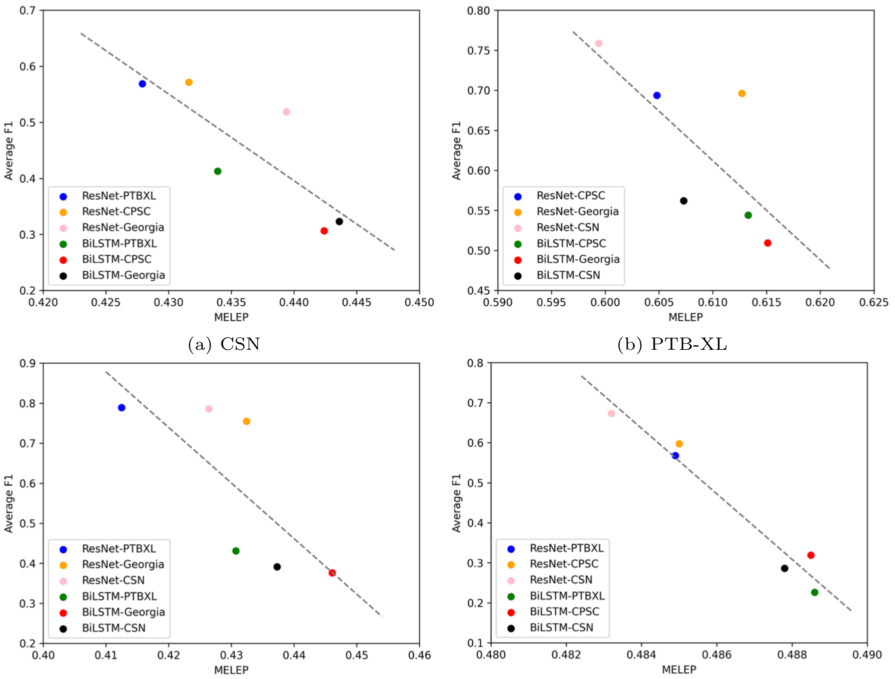
(c) Bi-LSTM CPSC2018 → PTB-XL

(d) Bi-LSTM Georgia → PTB-XL

Fig. 5 Relation of MELEP (partitioned as four distance levels) and fine-tuning performance of ResNet1d101 and Bi-LSTM on target tasks sampled from the PTB-XL dataset

In this scenario, we had eight checkpoint candidates: ResNet1d101-PTBXL, ResNet1d101-CPSC, ResNet1d101-Georgia, ResNet1d101-CSN, BiLSTM-PTBXL, BiLSTM-CPSC, BiLSTM-Georgia, BiLSTM-CSN. These checkpoints were obtained by pre-training two DNNs (ResNet1d101 and BiLSTM) on four datasets (PTBXL, CPSC2018, Georgia, and CSN). To simulate the context of fine-tuning a small target dataset, for each of the four datasets, we generated four target folds of 1000 records, following the random process outlined in Section 3.1, with a full set of 5, 9, 10, and 20 labels, respectively. For a given target fold, two checkpoints pre-trained on the same dataset were excluded to ensure fair comparison. For example, we did not consider the ResNet1d101-PTBXL and BiLSTM-PTBXL for experiments with the target PTBXL fold. Subsequently, we divided each fold into training and test subsets with a 7:3 ratio. The training subset was used for computing MELEP and fine-tuning, while the test set was reserved for performance evaluation.

In Fig. 6, we display the MELEP values and their corresponding average F1 scores for all checkpoint candidates for each target task, along with the reference best-fit lines. The four graphs illustrate the effectiveness of MELEP in predicting the performance



(a) CSN

(b) PTB-XL

(c) CPSC2018

(d) Georgia

Fig. 6 MELEP for checkpoint selection problem with corresponding target dataset. The consistent trend demonstrates MELEP's effectiveness in predicting the fine-tuning performance, supporting the pre-selection of the best pre-trained models

of the given checkpoints on the target task: lower MELEP should indicate a better average F1 score.

4 Discussions and Conclusion

We introduced MELEP, a novel transferability measure that is directly applicable to multi-label ECG diagnosis. The measure is built upon the foundation of LEEP [9], adapting from single-label multi-class problems in computer vision to multi-label binary-class ones in the ECG domain. We conducted extensive experiments to empirically illustrate the effectiveness of MELEP in predicting the performance of transfer learning in various ECG classification tasks. In this section, we discuss some notable properties, extensions, and applications of MELEP alongside promising directions for future study.

Source Model Dependence MELEP computation is based on a source checkpoint, which is a source model pre-trained on a source task. This may imply that better performance of the source model may result in better MELEP (thus better transferability). Empirically, in our experiments, pre-trained ResNet outperformed pre-trained Bi-LSTM on all source tasks, indicated in Table 3. Finetuned ResNet mostly achieved better MELEP scores than and outperformed fine-tuned Bi-LSTM on most target tasks in the checkpoint selection experiment in Section 3.4. This implication had also been discussed in [39], in which they theoretically showed that the source task hardness could affect their transferability metric NCE score. However, the extent to which better source models result in better MELEP requires further systematic investigation. In Table 2, despite underperforming ResNet, Bi-LSTM surprisingly achieved better Pearson correlation coefficients between MELEP and fine-tuning performance. This suggests that factors beyond model performance may influence MELEP. Kornblith et al. [4] might provide valuable insights into this question, as they pointed out that regularization and training settings had an impact on their transferability metric, ImageNet Top-1 Accuracy. Analyzing the impact of those dependence sources is an interesting topic to explore in the future.

Table 3 Performance of pre-trained models on the source task (evaluated on the test subset)

Model	Source data	Average F1
ResNet1d101	PTB-XL	0.744
	CPSC2018	0.740
	Georgia	0.593
	CSN	0.603
Bi-LSTM	PTB-XL	0.541
	CPSC2018	0.412
	Georgia	0.167
	CSN	0.277

Data Dependence In addition, MELEP is also dependent on the source dataset. Equations (4) and (5) show that the cardinality of the source label set contributes to the MELEP score. While MELEP can technically be applied even when the source and target datasets have different label sets, it is reasonable to expect that substantial overlap between the two sets would facilitate transferability. Conversely, in cases of minimal overlap, the source model would exhibit greater uncertainty regarding inputs from the target dataset, resulting in a more balanced dummy probability distribution (like random guesses) across non-overlapping diagnostic categories. For example, suppose that AF (atrial fibrillation) is a new label in the target dataset and does not appear in the source one, the uncertainty about AF would result in a dummy probability of positive AF being reduced close to 0.5. Therefore, any negative log-likelihood components related to AF will be larger, leading to a larger MELEP, indicating harder transferability. This effect amplifies with an increased number of non-overlapping categories. Note that here we assume that the source model is good enough, otherwise, even with overlapping labels, the dummy probability distributions may be worse than a random guess, leading to poor transferability predictions.

Considerable Extensions As mentioned in (5), we do not consider source label weights in the MELEP formula. This exclusion is based on the assumption that we lack prior knowledge of the source label distribution used in pre-training. However, in situations where this information is known, it is more sensible to take the source weights into account. Additionally, there is another variant that deserves consideration for its practical versatility. Instead of aggregating the weighted average of $\phi(\theta, \mathcal{D}, y, z)$ into a single value as in (5), we can output a vector of size $\mathcal{Y}$, indicating the transferability measures for each target label. Such an approach is well suited in scenarios where the performances on certain labels hold more significance than others.

Potential Applications Apart from the source checkpoint selection use case demonstrated in Section 3.4, MELEP can be useful for continual learning algorithms that are based on neural architecture changes or selection of data points in replay buffers [40, 41], facilitating the decision-making process. Additionally, in federated learning, where data is often allocated across multiple sources [29, 42] can utilize MELEP to facilitate local model selection and fine-tuning. Furthermore, multi-task learning [43, 44], which often depends on the selection of deep parameter-sharing networks and a combination of task labels, can also benefit from MELEP. Finally, MELEP holds the potential to assist in the selection of hyperparameters for Bayesian optimization [45]. We leave these directions for future work.

Funding This work was supported by VinUni Seed Grant 2020.

Data Availability Datasets used in this work can be found here: https://physionet.org/content/challenge-2021/1.0.3/#files.

Declarations

Ethical Approval Not applicable

Competing Interests The authors declare no competing interests.

References

1. Hannun AY, Rajpurkar P, Haghpanahi M, Tison GH, Bourn C, Turakhia MP, Ng AY (2019) Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. Nat Med 25(1):65–69
2. Ribeiro AH, Ribeiro MH, Paix ao GM, Oliveira DM, Gomes PR, Canazart JA, Ferreira MP, Andersson CR, Macfarlane PW, Meira Jr W et al (2020) Automatic diagnosis of the 12-lead ECG using a deep neural network. Nat Commun 11(1):1760
3. Sharif Razavian A, Azizpour H, Sullivan J, Carlsson S (2014) Cnn features o-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 806–813
4. Kornblith S, Shlens, J., Le, Q.V (2019) Do better ImageNet models transfer better? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2661–2671
5. He K, Girshick R, Dollár P (2019) Rethinking ImageNet pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 4918–4927
6. Donahue J, Jia Y, Vinyals O, Homan J, Zhang N, Tzeng E, Darrell T (2014) DeCaf: a deep convolutional activation feature for generic visual recognition. In: International conference on machine learning, pp 647–655. PMLR
7. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? Advances in neural information processing systems 27
8. Pan SJ, Yang Q (2009) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
9. Nguyen C, Hassner T, Seeger M, Archambeau C (2020) LEEP: a new measure to evaluate transferability of learned representations. In: International conference on machine learning, pp 7294–7305. PMLR
10. Nguyen CV, Do CD (2024) Transfer learning in ECG diagnosis: is it effective? arXiv:2402.02021
11. Ammar HB, Eaton E, Taylor ME, Mocanu DC, Driessens K, Weiss G,Tuyls K (2014) An automated measure of MDP similarity for transfer in reinforcement learning. In: Workshops at the twenty-eighth AAAI conference on articial intelligence, vol 1
12. Sinapov J, Narvekar S, Leonetti M, Stone P (2015) Learning inter-task transferability in the absence of target task samples. In: Proceedings of the 2015 international conference on autonomous agents and multiagent systems, pp 725–733
13. Tran AT, Nguyen CV, Hassner T (2019) Transferability and hardness of supervised classification tasks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1395–1405
14. Bao Y, Li Y, Huang SL, Zhang L, Zheng L, Zamir A, Guibas L (2019) An information-theoretic approach to transferability in task transfer learning. In: 2019 IEEE International Conference on Image Processing (ICIP), pp 2309–2313. IEEE
15. Hirschfeld HO (1935) A connection between correlation and contingency. In: Mathematical proceedings of the Cambridge philosophical society, vol 31, pp 520–524. Cambridge University Press
16. Gebelein H (1941) Das statistische problem der korrelation als variations-und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung. ZAMMJournal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 21(6):364–379
17. Rényi A (1959) On measures of dependence. Acta Math Hungar 10(3–4):441–451
18. Huang LK, Huang J, Rong Y, Yang Q, Wei Y (2022) Frustratingly easy transferability estimation. In: International conference on machine learning, pp 9201–9225. PMLR
19. Liu F, Liu C, Zhao L, Zhang X, Wu X, Xu X, Liu Y, Ma C, Wei S, He Z et al (2018) An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. Journal of Medical Imaging and Health Informatics 8(7):1368–1373
20. Wagner P, Strodtho N, Bousseljot RD, Kreiseler D, Lunze FI, Samek W, Schaeter T (2020) PTB-XL, a large publicly available electrocardiography dataset. Scientic data 7(1):154
21. Strodtho N, Wagner P, Schaeter T, Samek W (2020) Deep learning for ECG analysis: benchmarks and insights from PTB-XL. IEEE J Biomed Health Inform 25(5):1519–1528
22. Alday EAP, Gu A, Shah AJ, Robichaux C, Wong AKI, Liu C, Liu F, Rad AB, Elola A, Seyedi S et al (2020) Classication of 12-lead ECGs: the Physionet/Computing in Cardiology Challenge 2020. Physiol Meas 41(12)
23. Zheng J, Zhang J, Danioko S, Yao H, Guo H, Rakovski C (2020) A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. Scientic data 7(1):48

24. Zheng J, Chu H, Struppa D, Zhang J, Yacoub SM, El-Askary H, Chang A, Ehwerhemuepha L, Abudayyeh I, Barrett A et al (2020) Optimal multi-stage arrhythmia classification approach. Scientic Reports 10(1):2898
25. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
26. Xu W, Fu YL, Zhu D (2023) ResNet and its application to medical image processing: research progress and challenges. Computer Methods and Programs in Biomedicine 107660
27. Harrison P, Hasan R, Park K (2023) State-of-the-art of breast cancer diagnosis in medical images via convolutional neural networks (CNNs). Journal of Healthcare Informatics Research 1–46
28. Yu H, Yang LT, Zhang Q, Armstrong D, Deen MJ (2021) Convolutional neural networks for medical image analysis: state-of-the-art, comparisons, improvement and perspectives. Neurocomputing 444:92–110
29. Riedel P, Schwerin R, Schaudt D, Hafner A, Späte C (2023) ResNetFed: federated deep learning architecture for privacy-preserving pneumonia detection from COVID-19 chest radiographs. Journal of Healthcare Informatics Research 1–22
30. Wang C, Yang S, Tang X, Li B (2019) A 12-lead ECG arrhythmia classification method based on 1D densely connected CNN. In: Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting: First International Workshop, MLMECH 2019, and 8th Joint International Workshop, CVII-STENT 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1, pp 72–79.Springer
31. Zhu J, Xin K, Zhao Q, Zhang Y (2019) A multi-label learning method to detect arrhythmia based on 12-lead ECGs. In: Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting: First International Workshop, MLMECH 2019, and 8th Joint International Workshop, CVII-STENT 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1, pp 11–19. Springer
32. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
33. Luo C, Jiang H, Li Q, Rao N (2019) Multi-label classification of abnormalities in 12-lead ECG using 1D CNN and LSTM. In: Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting: First International Workshop, MLMECH 2019, and 8th Joint International Workshop, CVII-STENT 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 1, pp 55–63. Springer
34. Mostayed A, Luo J, Shu X, Wee W (2018) Classification of 12-lead ECG signals with bi-directional LSTM network. arXiv:1811.02090
35. Lv QJ, Chen HY, Zhong WB, Wang YY, Song JY, Guo SD, Li LX, Chen CYC (2019) A multi-task group Bi-LSTM networks application on electrocardiogram classification. IEEE Journal of Translational Engineering in Health and Medicine 8:1–11
36. Gupta P, Malhotra P, Narwariya J, Vig L, Shro G (2020) Transfer learning for clinical time series analysis using deep neural networks. Journal of Healthcare Informatics Research 4(2):112–137
37. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
38. Jeni LA, Cohn JF, De La Torre F (2013) Facing imbalanced data-recommendations for the use of performance metrics. In: 2013 Humaine association conference on affective computing and intelligent interaction, pp 245–251. IEEE
39. Tran AT, Nguyen CV, Hassner T (2019) Transferability and hardness of supervised classification tasks. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1395–1405
40. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (2019) Continual lifelong learning with neural networks: a review. Neural Netw 113:54–71
41. Kiyasseh D, Zhu T, Clifton D (2021) A clinical deep learning framework for continually learning from cardiac signals across diseases, time, modalities, and institutions. Nat Commun 12(1):4221
42. Baumgartner M, Veeranki SPK, Hayn D, Schreier G (2023) Introduction and comparison of novel decentral learning schemes with multiple data pools for privacy-preserving ECG classification. Journal of Healthcare Informatics Research 7(3):291–312
43. Ji J, Chen X, Luo C, Li P (2018) A deep multi-task learning approach for ECG data analysis. In: 2018 IEEE EMBS International conference on Biomedical & Health Informatics (BHI), pp 124–127. IEEE
44. Hsieh ME, Tseng V (2021) Boosting multi-task learning through combination of task labels-with applications in ECG phenotyping. Proceedings of the AAAI Conference on Articial Intelligence 35:7771–7779

45. Li H, Lin Z, An Z, Zuo S, Zhu W, Zhang Z, Mu Y, Cao L, Garcia JDP (2022) Automatic electrocardiogram detection and classification using bidirectional long short-term memory network improved by Bayesian optimization. Biomed Signal Process Control 73

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.