CiTrus: Squeezing Extra Performance out of Low-data Bio-signal Transfer Learning

Eloy Geenjaar[1]*, **Lie Lu**[2]
[1]Georgia Institute of Technology
[2]Dolby Laboratories
egeenjaar@gatech.edu, llu@dolby.com

Abstract

Transfer learning for bio-signals has recently become an important technique to improve prediction performance on downstream tasks with small bio-signal datasets. Recent works have shown that pre-training a neural network model on a large dataset (e.g. EEG) with a self-supervised task, replacing the self-supervised head with a linear classification head, and fine-tuning the model on different downstream bio-signal datasets (e.g., EMG or ECG) can dramatically improve the performance on those datasets. In this paper, we propose a new convolution-transformer hybrid model architecture with masked auto-encoding for low-data bio-signal transfer learning, introduce a frequency-based masked auto-encoding task, employ a more comprehensive evaluation framework, and evaluate how much and when (multimodal) pre-training improves fine-tuning performance. We also introduce a dramatically more performant method of aligning a downstream dataset with a different temporal length and sampling rate to the original pre-training dataset. Our findings indicate that the convolution-only part of our hybrid model can achieve state-of-the-art performance on some low-data downstream tasks. The performance often improves even further with our full model. In the case of transformer-based models, we find that pre-training especially improves performance on downstream datasets, multimodal pre-training often increases those gains further, and our frequency-based pre-training performs the best on average for the lowest and highest data regimes.

Extended version — https://arxiv.org/abs/2412.11695

1 Introduction

The wearable market is growing quickly around the world (Casselman, Onopa, and Khansa 2017). This increase in wearable usage means there is an increasing amount of bio-signal data available that can be used in the field of preventative medicine. By combining subjective assessments of experiences with objective signals extracted from bio-signals related to someone's well-being, stress level, cognitive state, etc., doctors can make more informed treatment decisions. There are many types of bio-signals, each recording a different type of information, for example, electroencephalography (EEG) non-invasively records the brain's electrical activity from outside the skull and can be used to predict attention levels (Li et al. 2011). On the other hand, photoplethysmography (PPG) records volumetric blood changes and can be used to predict stress levels (Charlton et al. 2018) from non-invasive recordings on the skin. Many more bio-signals exist, each with unique information about a potential patient.

However, bio-signals also come with three important drawbacks. First, bio-signals are noisy; with EEG, for example, the electrical activity from the brain is recorded through the skull and must thus travel through a dense bone before it arrives at the electrodes, which leads to noise. Moreover, it is harder to record activity deeper in the brain because the further away electrical current is generated from the electrode, the harder it is to reliably capture it. Additionally, normal activities while wearing wearables, such as walking, can induce movement-related noise in non-invasive bio-signals. Secondly, bio-signals suffer from a lot of subject-variability, i.e., it can be hard to generalize predictions from one person's bio-signal to the same bio-signal recorded from another person. Lastly, making predictions about a person's internal state from bio-signals is often non-trivial. These predictions require complex and non-linear transformations of the original bio-signal. This has led to the wide-scale adoption of neural networks in bio-signal research. However, neural networks require substantial amounts of labeled data to train on, and labeling bio-signals is expensive because it requires (medical) experts to go through the data and label the individual time windows of each bio-signal. It is thus imperative to develop neural network models that do not require many labeled examples to train on.

Transfer learning mitigates the need for many examples by first pre-training a neural network on a large unlabeled dataset. Transfer learning has shown to be effective for bio-signals with models pre-trained using a self-supervised task (Zhang et al. 2022; Liu et al. 2023; Dong et al. 2024). In this work, we develop a comprehensive evaluation strategy for bio-signal transfer learning and find that convolution-based architectures often outperform transformer-only models. We thus propose a new convolution transformer hybrid model that we call CiTrus, and new pre-training and fine-tuning strategies. Our model systematically improves performance over previous methods and strong baselines we

*Work completed during an internship at Dolby Laboratories

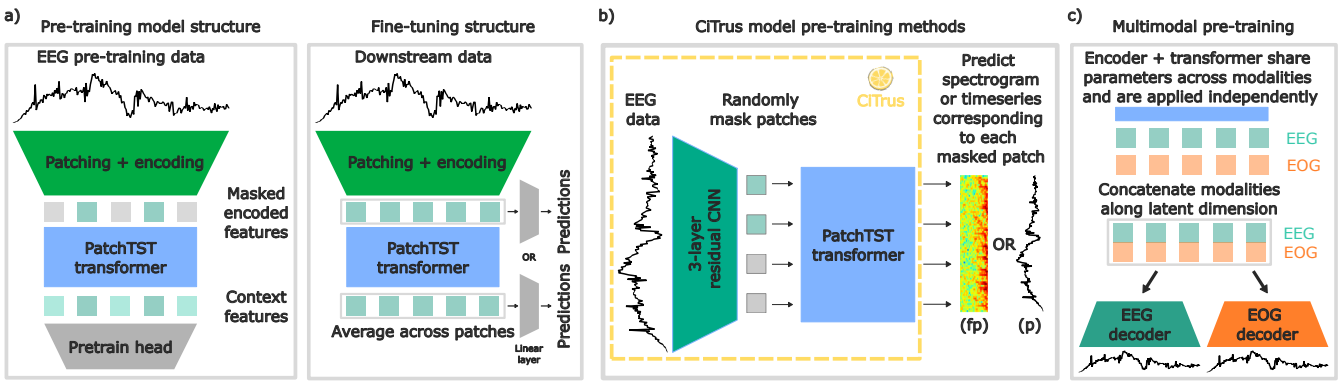

Figure 1: Subfigure a) shows the general transfer learning framework; the pre-training and fine-tuning structures. Subfigure b) shows the pre-training structure of our proposed model, and sub-figure c) shows how the structure of the model is adapted to accommodate multi-modal pre-training data.

introduce for comparison in this paper.

In summary, our main contributions are:

- The introduction of a convolution and transformer hybrid for bio-signal transfer learning (CiTrus) that outperforms previous models by a significant margin.
- A frequency-based pre-training approach that significantly improves fine-tuning performance for our proposed model.
- A new transferring approach that keeps the sampling frequency the same between the pre-training and fine-tuning dataset, and predominantly improves performance.
- A more comprehensive evaluation of transfer learning models for bio-signals. We add new datasets, evaluate models over multiple data availability regimes, and use multiple test splits. We specifically find that variance across test splits is much higher than across random model initializations.

2 Background

Transfer learning The goal of transfer learning is to pre-train a model on one dataset so that it performs well with little extra training on a new dataset with few labeled examples. Specifically with bio-signals, these downstream datasets are time series data, often with a unique temporal length, sampling frequency, and a variety of different prediction targets. Labels in the datasets are provided for windows over time, e.g. when is a cardiac rhythm normal and when is it indicative of atrial fibrillation. A common self-supervised pre-training method is masked autoencoding (He et al. 2022), which has also been used for bio-signal transfer learning previously (Liu et al. 2023; Dong et al. 2024). Specifically, masked autoencoding models can be loosely split into two parts; a patching and encoding part (encoder), and a transformer, see Figure 1a. In the simplest case, the patching and encoding part of the model linearly embeds non-overlapping windows (patches) of the input time-series into encoded features, see Figure 1a. A random subset of the encoded features is then masked by replacing them with a mask token and used as input to the transformer, which produces context features. These context features are passed through a pretrain head to reconstruct the original signal corresponding to the patches that are masked out. The pretrain head can be any type of neural network architecture, for example, a linear layer or a multi-layer perceptron (MLP). Masked auto-encoding thus forces the transformer to learn temporal relationships between patches so it can predict the masked patches.

After pre-training the model, the pretrain head is removed from the model, and in the fine-tuning stage, a linear layer is attached either to the encoded features (after the encoder) or the context features (after the transformer), see Figure 1a. The linear layer, together with the rest of the model, is then trained on the downstream bio-signal dataset. The assumption behind this method is that the relevant features and patch relationships the transformer and/or the encoder have learned are transferable to new bio-signal datasets and are a better starting point for supervised training than a randomly initialized network.

Related work In the supervised and decoding literature for EEG data, convolution-transformer hybrids have shown improvements over convolution-only architectures on a variety of tasks (Song et al. 2022; Peh, Yao, and Dauwels 2022; Gong et al. 2023; Miltiadous et al. 2023). However, all of these methods are specifically developed for EEG, are fully supervised, and have spatial convolution blocks that make it hard for them to transfer to new data with a different number of channels. Moreover, the field of masked auto-encoding for EEG and bio-signal data has recently seen many advances in same-data pre-training and fine-tuning (Chien et al. 2022; Yang, Westover, and Sun 2024), representation learning (Foumani et al. 2024), semi-supervised learning (Eldele et al. 2023), and cross-modal learning (Deldari et al. 2023). These models do not perform pre-training on one bio-signal with fine-tuning on a new bio-signal, however.

One reason why transfer learning from one bio-signal dataset to another dataset likely works is because important frequency ranges and thus low-level statistics used for prediction are similar across bio-signals (Neyshabur,

Dataset	EMG@20%			ECG@0.5%			PPG@10%			HAR@10%		
Metric	ACC	ROC	PRC	ACC	ROC	PRC	ACC	ROC	PRC	ACC	ROC	PRC
PatchTST (s)	51.46	61.87	52.07	54.15	52.34	60.85	59.98	73.89	55.32	67.67	88.62	69.83
PatchTST (p)	57.93	71.35	62.46	56.63	52.89	61.34	59.74	74.57	56.31	60.57	85.85	62.7
bioFAME (s)	50.43	61.89	53.46	58.36	51.66	60.52	60.31	73.98	55.2	54.03	84.31	56.72
bioFAME (mp)	70.93	86.94	78.78	64.26	52.89	61.36	60.21	76.67	58.51	60.5	86.85	65.54
NLPatchTST (s)	64.64	81.15	71.84	52.28	51.92	60.53	56.81	72.27	53.42	68.54	88.94	70.01
NLPatchTST (p)	70.45	87.9	80.11	54.43	52.96	61.27	60.39	75.15	56.62	65.66	87.78	67.61
NLPatchTST (mp)	74.61	90.16	82.77	56.86	53.02	61.41	59.44	74.78	56.14	60.41	86.2	63.7
SimMTM (s)	52.73	84.79	75.52	58.47	53.56	61.86	54.53	76.83	59.49	75.41	91.14	78.35
SimMTM (p)	42.52	61.3	54.13	56.77	53.08	61.5	50.58	73.7	56.07	77.37	91.76	80.14
Ci (s)	96.55	99.1	97.99	66.89	55.33	63.64	58.91	72.12	54.47	**80.71**	**93.03**	**82.8**
Ci (p)	94.74	98.72	96.73	65.62	54.76	63.07	58.57	73.85	55.77	70.7	90.18	74.46
CiTrus (s)	95.84	**99.48**	**99.4**	69.35	55.51	63.78	59.18	72.35	56.03	71.87	89.97	74.89
CiTrus (p)	90.45	96.96	94.18	64.55	54.6	62.84	62.41	75.37	59.14	73.89	90.9	75.9
CiTrus (fp)	**97.92**	99.02	98.76	**82.44**	**55.9**	**64.3**	**65.25**	**79.45**	**63.65**	65.3	87.55	68.75
CiTrus (mp)	92.9	99.01	96.86	67.72	55.2	63.6	62.98	76.15	58.64	72.78	90.47	75.32
Dataset	FDB@0.5%			Epilepsy@1%			Gesture@10%			SleepEDF@0.5%		
Metric	ACC	ROC	PRC	ACC	ROC	PRC	ACC	ROC	PRC	ACC	ROC	PRC
PatchTST (s)	52.99	54.35	38.04	92.79	92.8	96.54	43.21	81.04	49.29	64.06	65.54	33.83
PatchTST (p)	60.15	55.91	39.59	93.34	93.81	97.09	49.6	83.79	56.36	71.05	67.8	36.33
bioFAME (s)	52.07	53.63	37.13	92.46	96.17	98.72	30.36	74.46	39.54	70.62	68.04	37.45
bioFAME (mp)	56.71	56.64	39.74	92.77	96.93	99.01	36.38	79.27	46.28	70.51	68.1	37.91
NLPatchTST (s)	58.61	56.81	39.68	84.03	78.38	88.43	43.35	82.06	50.31	58.59	64.53	33.79
NLPatchTST (p)	63.64	57.28	40.24	92.64	92.41	96.15	48.13	84.6	56.8	**74.08**	68.16	36.74
NLPatchTST (mp)	65.01	58.2	41.03	92.77	92.33	96.23	51.88	84.2	57.39	66.98	66.8	34.94
SimMTM (s)	43.26	52.69	36.45	93.35	55.15	86.21	53.26	84.7	**63.2**			
SimMTM (p)	68.89	60.59	44.66	93.21	55.03	85.63	**56.83**	**85.43**	63.0			
Ci (s)	74.26	65.26	50.37	93.75	97.74	99.31	35.09	75.98	43.72	69.02	**69.03**	**38.84**
Ci (p)	72.72	64.45	48.88	94.37	**97.93**	**99.36**	40.31	81.15	49.04	70.48	68.58	38.57
CiTrus (s)	72.34	63.87	47.85	93.2	97.68	99.17	42.77	81.6	51.08	67.61	67.75	36.41
CiTrus (p)	75.48	63.52	47.69	94.08	95.29	97.92	40.71	80.73	50.63	73.87	68.85	37.96
CiTrus (fp)	**79.78**	**66.63**	**52.35**	93.41	95.83	98.26	51.34	82.28	55.81	48.69	58.11	27.52
CiTrus (mp)	73.55	63.63	47.77	**94.53**	94.67	97.34	47.99	83.24	55.99	69.26	67.16	35.84

Table 1: A comparison of the different model architectures. We use the same evaluation method for each model, for EMG and FD-B we interpolate the data for fine-tuning, and for the other datasets we use our sliding window approach. The letters in brackets refer to how the model is trained; (s) is trained from scratch, (p) means it is pre-trained, (mp) means it uses multi-modal pre-training, and (fp) means it uses frequency pre-training. ACC, ROC, and PRC refer to the accuracy, area under the receiver operating characteristic, and the area under the precision recall curve, respectively. The best result for each metric is shown in bold, the second best result is double-underlined, and the third best result is single-underlined.

Sedghi, and Zhang 2020). Since time-frequency features are important in the analysis of bio-signals, such as EEG (Durongbhan et al. 2019), ECG (Odinaka et al. 2010), and EMG (Weiderpass et al. 2013), learning features that are consistent across time and frequency space can be helpful for downstream predictions. Time-frequency consistency (TFC) (Zhang et al. 2022) is a method that extracts features from the time and frequency domains and trains them to be similar. To do this, they use contrastive learning and introduce new frequency-based augmentations. Alternatively, bioFAME (Liu et al. 2023), introduces a Fourier neural operator (FNO)-based encoder (Li et al. 2020; Guibas et al. 2021) to directly learn features from the frequency space. Additionally, they use a frequency-aware masked autoencoder for multimodal pre-training and introduce PatchTST (Nie et al. 2022) as a baseline. Recently, SimMTM (Dong et al. 2024) has improved on TFC with the same model architecture by relating masked auto-encoding to manifold learning. They introduce a new pre-training task where the model needs to reconstruct the original timeseries with a set of masked timeseries outside the manifold. These previous works motivated us to develop a convolutional-transformer hybrid model (CiTrus), with two pre-training techniques, and a better way of transferring a model from one bio-signal dataset to another.

3 Method

Given the potential importance of the frequency representation in bio-signal transfer learning, it is a natural choice to look at convolutions. Convolutional networks have learn-

Dataset	EMG			ECG			PPG			HAR		
Data percentage	20%	50%	80%	0.5%	1%	2%	10%	20%	50%	10%	20%	50%
PatchTST (p)	**+18.7**	-12.4	-11.1	+2.8	**+3.5**	+1.9	+1.1	**+1.5**	+1.3	-7.7	-6.2	**-2.5**
bioFAME (mp)	**+49.0**	+27.7	+10.0	+5.0	+4.4	**+5.2**	**+3.3**	-0.6	+2.0	+10.5	+7.9	**+15.3**
NLPatchTST (p)	**+12.5**	+1.9	+0.1	**+3.4**	-0.1	-0.1	**+6.4**	+2.5	-0.9	-2.9	-3.7	**-0.8**
NLPatchTST (mp)	**+16.7**	+1.0	-1.4	**+5.6**	+1.5	+1.1	**+5.2**	+1.8	+0.3	-7.8	-7.4	**-1.7**
SimMTM (p)	-16.1	-3.2	**-0.2**	**+3.5**	-7.5	-3.5	-4.0	**+0.2**	-1.6	**+2.0**	+0.6	+0.5
Ci (p)	-1.1	-0.3	**-0.2**	-0.8	-1.4	**+1.9**	**+1.9**	-0.2	+0.3	-8.5	-3.8	**-0.3**
CiTrus (p)	-4.4	-0.7	**-0.2**	-3.5	-1.2	**+0.1**	**+6.5**	+2.6	-0.3	+2.0	**+2.3**	-0.6
CiTrus (fp)	**+0.4**	-0.9	+0.3	+8.3	**+8.9**	+7.2	**+12.9**	+12.7	+9.0	-6.5	-4.2	**-0.3**
CiTrus (mp)	-2.0	-2.0	**-0.2**	-0.8	+0.5	**+1.8**	**+7.1**	+3.3	-1.4	+1.0	**+2.4**	-1.3
Dataset	FDB			Epilepsy			Gesture			SleepEDF		
Data percentage	0.5%	1%	2%	1%	5%	10%	10%	20%	50%	0.5%	1%	2%
PatchTST (p)	**+7.0**	+6.9	-1.8	**+0.8**	-0.1	+0.0	**+12.2**	+8.7	+4.8	**+7.3**	+0.3	+1.3
bioFAME (mp)	+7.3	**+8.9**	+8.4	**+0.5**	+0.0	+0.1	**+16.8**	+11.8	+10.0	+0.4	**+0.6**	-1.2
NLPatchTST (p)	**+4.1**	+1.7	-3.1	+13.4	**+15.0**	+4.5	**+11.4**	+7.3	+4.4	**+13.7**	+4.4	+7.1
NLPatchTST (mp)	**+6.1**	+3.6	-1.1	+13.4	**+15.6**	+4.4	**+15.1**	+9.1	+5.4	**+7.2**	+1.8	+4.3
SimMTM (p)	**+33.5**	+19.2	+5.1	-0.3	-0.3	**-0.0**	**+3.0**	+1.2	+0.5			
Ci (p)	-1.9	-1.8	**-0.7**	**+0.3**	+0.3	+0.2	**+13.7**	-6.5	-0.5	+0.3	+0.0	**+1.7**
CiTrus (p)	**+2.1**	-3.3	-2.2	-0.9	-0.8	**-0.1**	-1.3	-6.1	**+0.4**	**+5.1**	-1.0	+1.5
CiTrus (fp)	**+8.8**	+1.1	+1.3	-0.8	**+0.1**	-0.0	**+11.8**	+1.2	+5.6	-22.1	-18.9	**-9.3**
CiTrus (mp)	**+1.1**	-4.8	-2.1	-1.2	-0.4	**-0.3**	**+9.1**	+3.0	-0.4	**+0.1**	-4.0	-1.8

Table 2: A comparison between a pre-trained and fine-tuned version of each architecture, and the same architecture trained from scratch on the downstream datasets; (p) means it is pre-trained, (mp) means it uses multi-modal pre-training, and (fp) means it uses frequency pre-training. Each value is the average percentage improvement (across all three metrics) of pre-training compared to training from scratch. Values are made bold for the data regime within a dataset where pre-training and fine-tuning most increases the performance to show the effect of data availability on pre-training improvements.

able filters that can learn representations in the frequency domain and are good at capturing local features in the time series. We propose a model that combines the best of both worlds that we call CiTrus, a [C]onvolution-[Tr]ansformer hybrid, as shown in Figure 1b. The encoder in our model consists of 3 residual blocks (a more detailed description can be found in Appendix I in the paper's extended version), and the transformer is a PatchTST (Nie et al. 2022) transformer. Our reason for choosing a PatchTST transformer, similar to bioFAME, is that it exhibits especially good performance for masked auto-encoding and is channel-independent. To create channel-independence for the convolutional encoder, we concatenate the number of channels along the batch, and both pre-train and fine-tune the model with a single channel as input. This both increases the parameter efficiency and increases the number of samples the weights in the convolutional encoder are trained with, but also allows the model to easily transfer to data with a different number of channels. Specifically, without making the convolutional encoder channel-independent we can't transfer a model trained with 3 channels to a downstream dataset with a single channel.

Pre-training (p) and frequency-pretraining (fp) In conventional masked-autoencoding (MAE), the timeseries first needs to be segmented into non-overlapping patches of equal length. Formally, let $x \in \mathbb{R}^T$ be a bio-signal with T timesteps, a patch size S, and assume T%S$=0$. The time-series is then split into T/S = P patches, to obtain a signal $x_p \in \mathbb{R}^{P \times S}$. The patched signal is masked by replacing X% of the patches with a mask token and a transformer is trained to reconstruct the masked patches using the information from patches that are not masked. Recent work has shown that it is beneficial to encode the signal into a latent space before patching and masking. Therefore, in our model, we use a CNN as the encoder before patching. Formally, let $x \in \mathbb{R}^{1 \times T}$, then CNN(x) $\in \mathbb{R}^{D \times P}$, with D the number of output channels, and P the number of patches. The stride of the CNN determines the stride of the patches, and the receptive field determines the "effective" patch length. The inputs to the transformer are thus P patches with D dimensions. Since the receptive fields for each patch overlap in the data space, typical masking won't work (neighboring patches have too much information about each other). Thus, for our models' pre-training we mask consecutive patches and call the number of consecutive masked patches the block mask size. The pre-train head used for this type of pre-training is a flipped version of the convolutional encoder. To ablate the impact of the convolutional encoder, we develop a new baseline; a 3-layer multi-layer perceptron (MLP) to encode and decode each patch. We call this model NLPatchTST (Non-Linear PatchTST) to differentiate it from the original PatchTST model that uses a linear embedding layer.

Besides predicting the original input time-series signal, we can also push the model to explicitly predict frequency representations of the original signal by predicting the spectrogram that corresponds to the masked patches instead of the original signal. The same low frequencies can be repeated across patches. Thus, to make the prediction of the

Dataset	EMG			ECG			PPG			HAR		
Data percentage	20%	50%	80%	0.5%	1%	2%	10%	20%	50%	10%	20%	50%
NLPatchTST	**+17.3**	**+7.5**	**+6.0**	**+2.2**	**+3.6**	**+1.9**	-0.5	-0.3	**+1.6**	-4.9	-3.7	-0.9
CiTrus	**+6.3**	**+8.1**	**+3.1**	**+6.0**	**+3.8**	**+2.4**	**+1.1**	**+1.5**	-0.6	-0.8	**+0.2**	-0.7
Dataset	FDB			Epilepsy			Gesture			SleepEDF		
Data percentage	0.5%	1%	2%	1%	5%	10%	10%	20%	50%	0.5%	1%	2%
NLPatchTST	-2.0	**+0.1**	-0.9	**+0.1**	**+0.5**	-0.1	**+3.8**	**+1.9**	**+1.3**	-5.5	-2.5	-2.5
CiTrus	-0.5	**+0.5**	-0.1	-0.2	**+0.4**	-0.1	**+12.7**	**+11.1**	-0.3	-4.7	-3.0	-3.2

Table 3: A comparison of multimodal pre-training to unimodal pre-training, where each value is the average improvement (across all three metrics) in percentages. Values that are larger than 0, indicating performance improvement, are made bold.

spectrogram more challenging, we z-score the spectrograms along the time dimension. This new pre-training method is visualized in Figure 1b. The exact frequency pre-training settings and implementation are described in Appendix C of the paper's extended version.

Multimodal pre-training Given that the utility of multimodal pre-training was verified for the bioFAME model, we explore how well multimodal pre-training works for CiTrus. To pre-train CiTrus with multimodal data, each modality is independently passed through the convolutional encoder and transformer. This ensures that the convolutional encoder and transformer see data from both modalities and can learn to adapt their weights to both. In our work, we follow bioFAME and use two modalities, EEG and EOG, during pre-training, with a separate decoder for each modality. Specifically, the EEG and EOG signals are concatenated along the batch dimension in the convolutional encoder and transformer. After the transformer, the EEG and EOG context features are separated and concatenated along the latent dimension, see Figure 1c. The context features are then used as input for each decoder to predict the masked patches in each modality. To encourage cross-modality learning, we mask patches independently for each modality, allowing information for a masked patch in one modality to be available from the other modality.

Transfer learning to new datasets Previous works, such as TFC and bioFAME transfer a pre-trained model to new downstream datasets with different temporal lengths by interpolating the data to match the signal length of the pre-training dataset. This causes a mismatch in sampling frequencies between the downstream fine-tuning datasets and the pre-train dataset. The potential frequency representation learned by the encoder and the temporal relationships learned by the transformer on the pre-training dataset are now potentially irrelevant because they act on a different frequency range and timescale. We therefore propose a new strategy where the fine-tuning dataset is resampled to match the frequency of the pre-training dataset. This allows the model to attend to similar frequency spectra in the fine-tuning data as the pre-training data. If the length of the re-sampled fine-tuning data is now longer than the temporal length of the pre-training signal, we use an overlapping sliding window approach, and average the embeddings across the windows. If the length of the re-sampled fine-tuning data is shorter, we pad the signal with zeros.

4 Experiments

Similar to previous work (Zhang et al. 2022), we use SleepEDF, a large sleep EEG dataset (Kemp et al. 2000), for pre-training. The dataset is split into 30 second windows of EEG data sampled at 100Hz. For the uni-modal pre-training, we use the two EEG channels that are captured. For multi-modal pre-training, we use two EEG channels and (single-channel) EOG data from the same dataset. As downstream datasets, we use 4 bio-signal datasets that were used to evaluate the TFC, bioFAME, and SimMTM models:

- An electromyography (EMG) dataset (Goldberger et al. 2000) with 375ms windows, sampled at 4KHz, and a single channel.
- A gesture recognition dataset (Liu et al. 2009) with 3.15s windows, sampled at 100Hz, and 3 channels.
- An EEG Epilepsy dataset with 1.02s windows, sampled at 174Hz, and a single channel.
- An electromotor fault-detection (FD-B) (Lessmeier et al. 2016) dataset with 80ms windows, sampled at 64KHz, and a single channel.

Additionally, to increase the diversity of downstream tasks and modalities we test, we also add the following datasets:

- A photoplethysmography (PPG) dataset (Schmidt et al. 2018) with 60s windows, sampled at 64Hz, and a single channel.
- The HAR dataset (Reyes-Ortiz et al. 2015) with 2.56s windows length, sampled at 50Hz, and 6 channels.
- An electrocardiogram (ECG) dataset (Moody 1983) with 10s windows, sampled at 250Hz, and 2 channels.
- The SleepEDF test set

More information about the datasets can be found in Appendix H of the paper's extended version.

The training, validation, and testing data splits were initially defined in the TFC work for the EMG, Gesture, FD-B, and Epilepsy datasets. These splits were also used for bioFAME and SimMTM. Although we evaluate our proposed model on those previously defined data splits, we find that the variance in performance across test folds is much larger than across random seeds. The results of these analyses are discussed in Appendix B and Table 6 in Appendix D in the paper's extended version, respectively. Thus, to more comprehensively evaluate models used for transfer learning, we advocate for and implement a cross-validation procedure

Dataset	ECG			PPG			HAR			Gesture		
Data percentage	0.5%	1%	2%	10%	20%	50%	10%	20%	50%	10%	20%	50%
PatchTST (s)	**+4.5**	-0.3	-0.7	**+58.9**	**+60.0**	**+58.9**	**+23.9**	**+22.1**	**+19.7**	**+16.0**	**+18.5**	**+9.5**
PatchTST (p)	**+6.7**	**+10.1**	**+3.5**	**+52.9**	**+58.9**	**+57.8**	**+14.8**	**+17.2**	**+18.8**	**+3.7**	**+5.6**	**+4.2**
bioFAME (s)	**+5.2**	**+2.9**	**+2.6**	**+60.7**	**+55.9**	**+47.7**	**+16.7**	**+15.3**	**+16.0**	**+7.8**	**+10.9**	**+9.5**
bioFAME (mp)	**+15.4**	**+11.4**	**+9.8**	**+64.1**	**+47.8**	**+56.2**	**+14.6**	**+16.6**	**+26.5**	-0.3	**+2.1**	**+0.1**
SimMTM (s)	**+4.4**	**+15.3**	**+8.1**	**+30.1**	**+29.9**	**+51.9**	**+0.2**	-0.1	-0.4	-3.8	**+0.0**	-1.7
SimMTM (p)	**+1.4**	**+1.7**	-1.8	**+24.3**	**+26.9**	**+28.4**	-0.3	-0.6	-0.1	-0.1	**+0.6**	-0.7
NLPatchTST (s)	-0.5	-2.0	-2.6	**+49.6**	**+50.4**	**+64.3**	**+20.9**	**+20.7**	**+15.8**	**+8.8**	**+14.2**	**+3.0**
NLPatchTST (p)	**+3.3**	**+3.5**	-0.5	**+58.6**	**+55.9**	**+61.8**	**+17.1**	**+20.1**	**+18.6**	-0.6	**+3.8**	**+3.5**
NLPatchTST (mp)	**+5.5**	**+8.0**	-0.2	**+57.6**	**+52.5**	**+64.9**	**+10.6**	**+14.7**	**+18.8**	**+3.3**	**+6.4**	**+4.9**
Ci (s)	**+0.2**	**+3.8**	**+2.6**	**+29.1**	**+30.4**	**+20.4**	**+22.7**	**+17.8**	**+13.2**	**+8.2**	**+4.9**	**+2.7**
Ci (p)	-1.2	**+1.6**	**+2.9**	**+39.1**	**+41.7**	**+31.9**	**+18.5**	**+20.4**	**+11.7**	**+0.8**	**+3.9**	**+4.1**
CiTrus (s)	**+0.4**	**+6.0**	**+1.0**	**+32.1**	**+24.8**	**+21.1**	**+16.7**	**+18.7**	**+17.6**	**+21.1**	**+8.8**	**+3.7**
CiTrus (p)	**+6.7**	**+6.7**	**+8.6**	**+49.4**	**+51.0**	**+47.3**	**+19.3**	**+17.7**	**+14.3**	-0.9	**+0.5**	**+2.6**
CiTrus (fp)	**+0.5**	**+4.5**	**+3.1**	**+50.3**	**+48.1**	**+30.8**	**+11.5**	**+14.8**	**+17.7**	**+9.0**	**+2.8**	**+3.4**
CiTrus (mp)	**+11.0**	**+5.9**	**+8.0**	**+51.6**	**+49.7**	**+44.2**	**+17.4**	**+16.9**	**+14.2**	**+0.8**	**+6.9**	**+1.7**

Table 4: A comparison between the previously used fine-tuning technique (temporal interpolation), and our proposed fine-tuning technique; (s) is trained from scratch, (p) means it is pre-trained, (mp) means it uses multi-modal pre-training, and (fp) means it uses frequency pre-training. Each value indicates the average performance improvement (across all three metrics) in percentages. Values larger than 0 are made bold.

that averages test performance across 10-fold test splits. The training and validation data (the remaining 9 folds) is then reduced to the specific data-regime percentage. This leftover percentage of training and validation data is then split into 75% training and 25% validation data. Averaging the performance of the models across random seeds and test splits helps average out some of the performance's randomness. We discuss the exact protocol we use to create the test splits in Appendix A of the paper's extended version. For all the downstream tasks (except the SleepEDF test set), we follow the TFC work and split the data into 2 second windows before pre-training because the downstream data is often very short. For the SleepEDF evaluation, we keep the original 30s for pre-training. We also compare 30 vs 2 second window pre-training in Appendix G of the paper's extended version. To understand the relationship between transfer learning and the amount of data available in the downstream datasets, we evaluate each model across a range of data percentages during fine-tuning.

Experimental settings The implementations for the bioFAME, SimMTM, and PatchTST models are taken from their respective official implementations. The following are the model hyperparameters for all the models, except SimMTM, for which we use the official implementation's hyperparameters. The hyperparameters are matched to the bioFAME paper as much as possible to make the comparisons as fair as possible. We use a 4-layer transformer (with 64 latent dimensions, 128 feed-forward dimension, and 8 heads), a 3-layer convolutional network with 32 channels in the first residual convolution layer that double every layer (only applicable for CiTrus), a patch size of 20, a 0.5 masking ratio, and a block masking size of 5 (only applicable for CiTrus). All models are pre-trained for 200 epochs, a batch size of 128, and with 0.0001 as the learning rate. All models are fine-tuned for 100 epochs with a batch size of 64, without any augmentations, and the same learning rate as during pre-training. We use the last pre-training model checkpoint for fine-tuning, and evaluate the best (based on the validation set) fine-tuning checkpoint on the test set. Our models are pre-trained and fine-tuned on an AWS instance with 4 NVIDIA A10 GPUs, with $[42, 1337, 1212, 9999]$ as the model seeds, and 42 as the seed for data randomization and fold generation. Given the 10 data folds, there are 40 runs per model, per data regime, and per dataset. A more thorough description of the model settings can be found in Appendix I of the paper's extended version. Since the evaluation is done on classification tasks, we report the accuracy (ACC), area under the curve of the receiver operating characteristic (ROC), and area under the precision-recall curve (PRC) to get a variety of classification metrics. Lastly, standard deviations and Wilcoxon signed-rank test outcomes for each experiment are discussed in Appendix F of the paper's extended version.

Model architecture comparisons The PatchTST, bioFAME, and SimMTM models are evaluated and compared to our NLPatchTST and CiTrus models for the transfer-learning task in Table 1. To understand the effect the transformer in CiTrus has on the performance, we compare Ci, which uses the encoded features, to CiTrus, which uses the context features. For each dataset, except the EMG and FD-B datasets, we use our proposed sliding window approach during fine-tuning. The EMG and FD-B datasets have windows for predictions that are too short for our sliding window technique, so we interpolate the data to 200 timesteps as described in the TFC work. Moreover, to limit the size of the table, we report the hardest data regime; the lowest amount of data available for each dataset. The percentage of training + validation data that is available is mentioned after

the name of the dataset in Table 1. Note, since the convolutional encoder used in the SimMTM model only works with inputs that are 200 samples in length, it is not compatible with the SleepEDF test set, so we do not report the results. The results in Table 1 indicate that convolution-based models (SimMTM, Ci, and CiTrus) almost always achieve the best performance. For ECG, PPG, and FD-B, using the context features (CiTrus) performs better than using the encoded features (Ci). The CiTrus model with pre-training and fine-tuning performs the best on those datasets as well. For every dataset, except the Gesture dataset, the CiTrus model is at least in the top three best models. The fact that both the HAR and Gesture datasets are accelerometer-based datasets could explain why CiTrus exhibits reduced performance on both datasets. Our frequency-based pre-training model performs the best on average, as described in the paper's extended version Appendix E of the paper's extended version.

Training from scratch vs pre-training To understand the effect of pre-training on downstream performance, we calculate how much better each model performs after pre-training. Since we use three metrics in Table 1 to evaluate each approach, we calculate the average percentage pre-train improvement across the three metrics over training the model from scratch. We also bold the largest improvement across the data regimes for each model and each dataset in Table 2. For most settings in Table 2, pre-training improves downstream performance, but this improvement depends on the dataset and the underlying structure of the model. We can see dramatic increases in performance for transformer-based models across all data regimes on the EMG dataset. However, gains for the Ci model, which is fully convolutional, are marginal. Moreover, pre-training improves performance the most for the lowest data regime on the Gesture, PPG, and SleepEDF datasets.

Multimodal vs unimodal pre-training For both the CiTrus and the NLPatchTST model we investigate whether multimodal pre-training further improves performance over unimodal pre-training. Similar to the pre-training results, we compute the average percentage improvement across the three metrics with respect to uni-modal pre-training. Results that improve performance over uni-modal pre-training are made bold in Table 3. There are a few datasets where multimodal pre-training almost always performs better: the EMG, ECG, and Gesture datasets. These show clear improvements for multimodal pre-training. Moreover, the CiTrus model generally benefits more from multi-modal pre-training than the NLPatchTST model. This could be due to the inherent ability of convolutional networks to learn local feature representations jointly across multiple modalities. Interestingly, for the SleepEDF test set, multimodal pre-training degrades performance, potentially because only EEG data is available during fine-tuning.

Resampling-adaptive fine-tuning To verify how well our proposed fine-tuning approach works for new datasets with different temporal lengths, sampling frequencies, and modalities, we compare it to the interpolation method used in the TFC, bioFAME, and SimMTM works. We compute the average percentage improvement with our proposed fine-tuning technique over fine-tuning with temporal interpolation for each model. We select datasets that have different sampling frequencies from the original pre-training dataset. We exclude the EMG and FD-B datasets, since their lengths are extremely short (< 400ms), and lead to very small temporal samples (< 40) at 100Hz. In Table 4 the average percentage increase in metrics shows that in almost all cases our new fine-tuning approach markedly improves performance. Especially on the PPG dataset, the new approach reaches improvements of 64%. Moreover, improvements are achieved for both convolutional and transformer-based models. Pre-trained models benefit the most from the new fine-tuning approach.

5 Discussion

In this work, we assessed the transferability of neural networks pre-trained on a large EEG dataset and fine-tuned on low-data downstream bio-signal datasets. First, we generally find that models with convolutions (SimMTM, Ci, and CiTrus) perform the best. We suspect this is largely due to their inductive bias and parameter efficiency. Convolutions share many of their weights when processing the timeseries, and given that convolutions are learnable frequency filters, they transfer well to datasets where features come from similar parts of the frequency spectrum. Second, the idea that the convolutions' parameter efficiency and inductive bias are key in low-data settings is further supported by the fact that transformer-based models improve the most with pre-training, especially in low-data settings, which indicates that they require more data to learn the important temporal relationships required for predictions. This is also true for other types of data, such as images, where the vision transformer (ViT) (Dosovitskiy et al. 2020) starts outperforming convolutions when pre-trained with a large dataset. Third, multimodal pre-training often improves downstream fine-tuning performance even more. Fourth, we find that pre-training is not necessary for the EMG and FD-B datasets. Although the TFC, bioFAME, and SimMTM models found models pre-trained on bio-signal data to transfer well to the FD-B electromotor dataset, we find that our Ci model trained from scratch outperforms all other models. Additionally, datasets where performance is already high, like the Epilepsy dataset, do not benefit much from pre-training either. Fifth, CiTrus with frequency-based pre-training performs the best on average across datasets for the lowest and highest data regimes we tested. Sixth, we find that variance across test folds is much higher (up to 4609%) than across random seeds, and thus strongly recommend evaluating transfer learning models across different test sets to more robustly evaluate their performance. Lastly, our fine-tuning approach is better in essentially all cases for all models and improves performance up to 60%. Our approach improves performance the most for pre-trained models, which we believe is because the temporal and frequency relationships the encoder and transformer learn are now aligned between the pre-training and downstream datasets.

References

Casselman, J.; Onopa, N.; and Khansa, L. 2017. Wearable healthcare: Lessons from the past and a peek into the future. *Telematics and Informatics*, 34(7): 1011–1023.

Charlton, P. H.; Celka, P.; Farukh, B.; Chowienczyk, P.; and Alastruey, J. 2018. Assessing mental stress from the photoplethysmogram: a numerical study. *Physiological measurement*, 39(5): 054001.

Chien, H.-Y. S.; Goh, H.; Sandino, C. M.; and Cheng, J. Y. 2022. Maeeg: Masked auto-encoder for eeg representation learning. *arXiv preprint arXiv:2211.02625*.

Deldari, S.; Spathis, D.; Malekzadeh, M.; Kawsar, F.; Salim, F.; and Mathur, A. 2023. Latent Masking for Multimodal Self-supervised Learning in Health Timeseries. *arXiv preprint arXiv:2307.16847*.

Dong, J.; Wu, H.; Zhang, H.; Zhang, L.; Wang, J.; and Long, M. 2024. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Durongbhan, P.; Zhao, Y.; Chen, L.; Zis, P.; De Marco, M.; Unwin, Z. C.; Venneri, A.; He, X.; Li, S.; Zhao, Y.; et al. 2019. A dementia classification framework using frequency and time-frequency features based on EEG signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(5): 826–835.

Eldele, E.; Ragab, M.; Chen, Z.; Wu, M.; Kwoh, C.-K.; Li, X.; and Guan, C. 2023. Self-supervised contrastive representation learning for semi-supervised time-series classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Foumani, N. M.; Mackellar, G.; Ghane, S.; Irtza, S.; Nguyen, N.; and Salehi, M. 2024. Eeg2rep: enhancing self-supervised EEG representation through informative masked inputs. *arXiv preprint arXiv:2402.17772*.

Goldberger, A. L.; Amaral, L. A.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.-K.; and Stanley, H. E. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220.

Gong, L.; Li, M.; Zhang, T.; and Chen, W. 2023. EEG emotion recognition using attention-based convolutional transformer neural network. *Biomedical Signal Processing and Control*, 84: 104835.

Guibas, J.; Mardani, M.; Li, Z.; Tao, A.; Anandkumar, A.; and Catanzaro, B. 2021. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

Kemp, B.; Zwinderman, A. H.; Tuk, B.; Kamphuisen, H. A.; and Oberye, J. J. 2000. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*, 47(9): 1185–1194.

Lessmeier, C.; Kimotho, J. K.; Zimmer, D.; and Sextro, W. 2016. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In *PHM Society European Conference*, volume 3.

Li, Y.; Li, X.; Ratcliffe, M.; Liu, L.; Qi, Y.; and Liu, Q. 2011. A real-time EEG-based BCI system for attention recognition in ubiquitous environment. In *Proceedings of 2011 international workshop on Ubiquitous affective awareness and intelligent interaction*, 33–40.

Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; and Anandkumar, A. 2020. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.

Liu, J.; Zhong, L.; Wickramasuriya, J.; and Vasudevan, V. 2009. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6): 657–675.

Liu, R.; Zippi, E. L.; Pouransari, H.; Sandino, C.; Nie, J.; Goh, H.; Azemi, E.; and Moin, A. 2023. Frequency-aware masked autoencoders for multimodal pretraining on biosignals. *arXiv preprint arXiv:2309.05927*.

Miltiadous, A.; Gionanidis, E.; Tzimourta, K. D.; Giannakeas, N.; and Tzallas, A. T. 2023. DICE-net: a novel convolution-transformer architecture for Alzheimer detection in EEG signals. *IEEE Access*.

Moody, G. 1983. A new method for detecting atrial fibrillation using RR intervals. *Proc. Comput. Cardiol.*, 10: 227–230.

Neyshabur, B.; Sedghi, H.; and Zhang, C. 2020. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33: 512–523.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

Odinaka, I.; Lai, P.-H.; Kaplan, A. D.; O'Sullivan, J. A.; Sirevaag, E. J.; Kristjansson, S. D.; Sheffield, A. K.; and Rohrbaugh, J. W. 2010. ECG biometrics: A robust short-time frequency analysis. In *2010 IEEE International Workshop on Information Forensics and Security*, 1–6. IEEE.

Peh, W. Y.; Yao, Y.; and Dauwels, J. 2022. Transformer convolutional neural networks for automated artifact detection in scalp EEG. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 3599–3602. IEEE.

Reyes-Ortiz, J.; Anguita, D.; Oneto, L.; and Parra, X. 2015. Smartphone-based recognition of human activities and postural transitions data set. *UCI Machine Learning Repository. School Inf. Comput. Sci. Univ. California at Irvine, Irvine, CA, USA, available online: http://archive. ics. uci.*

edu/ml/datasets/Smartphone-Based+ Recognition+ of+ Human+ Activities+ and+ Postural+ Transitions.

Schmidt, P.; Reiss, A.; Duerichen, R.; Marberger, C.; and Van Laerhoven, K. 2018. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM international conference on multimodal interaction*, 400–408.

Song, Y.; Zheng, Q.; Liu, B.; and Gao, X. 2022. EEG conformer: Convolutional transformer for EEG decoding and visualization. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31: 710–719.

Weiderpass, H.; Pachi, C.; Yamamoto, J.; Hamamoto, A.; Onodera, A.; and Sacco, I. 2013. Time-frequency analysis methods for detecting effects of diabetic neuropathy. *International Journal for Numerical Methods in Biomedical Engineering*, 29(9): 1000–1010.

Yang, C.; Westover, M.; and Sun, J. 2024. Biot: Biosignal transformer for cross-data learning in the wild. *Advances in Neural Information Processing Systems*, 36.

Zhang, X.; Zhao, Z.; Tsiligkaridis, T.; and Zitnik, M. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35: 3988–4003.