Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25–29 May 2026, Vancouver, BC

ADAPTive Input Training for Many-to-One Pre-Training on Time-Series Classification

Paul Quinlan[†,◇,*], Qingguo Li[‡,◇], Xiaodan Zhu[†,◇]
[†] Electrical and Computer Engineering, Queen's University
[‡] Mechanical and Materials Engineering, Queen's University
[◇] Ingenuity Labs Research Institute, Queen's University

Abstract

Recent work on time-series models has leveraged self-supervised training to learn meaningful features and patterns in order to improve performance on downstream tasks and generalize to unseen modalities. While these pretraining methods have shown great promise in one-to-many scenarios, where a model is pre-trained on one dataset and fine-tuned on a downstream dataset, they have struggled to generalize to new datasets when more datasets are added during pre-training. This is a fundamental challenge in building foundation models for time-series data, as it limits the ability to develop models that can learn from a large variety of diverse datasets available. To address this challenge, we present a new pre-training paradigm for time-series data called `ADAPT`, which can efficiently align the physical properties of data in the time-series domain, enabling mixed-batch pre-training despite the extreme discrepancies in the input sizes and channel dimensions of pre-training data. We trained on 162 time-series classification datasets and set new state-of-the-art performance for classification benchmarks. We successfully train a model within the time-series domain on a wide range of datasets simultaneously, which is a major building block for building generalist foundation models in time-series domains.

Keywords: time-series, foundation models, pretraining, self-supervised Learning.

1. Introduction

Analysis of time-series data is critical in many real-life applications, including those in the medical [1], financial [2], industrial [3], agricultural [4] and environmental domains [5], among others [6, 7]. Pre-training and transfer learning have allowed for the application of large models to diverse tasks even when there is limited task-specific data available. However, the unique characteristics and variability of time-series data often make it challenging to develop generalist models that can be successfully adapted to different downstream modalities after pretraining. Previous research has mainly focused on custom modality-specific designs, pre-trained on a singular dataset, to improve the inductive bias of model training. This approach is in contrast to the dominant strategies used in other domains, such as natural language processing (NLP) and computer vision (CV), which focus on training a single model on many large datasets and has led to the creation of *foundation models* such as GPT4 [8], Mixtral [9], LaMDA [10], wav2vec2.0 [11], DALLE-2 [12], T5 [13], among others.

These models have shown remarkable performance and ability to generalize to unseen data and tasks. While pre-trained models have been applied to differing downstream modalities via fine-tuning (one-to-many), no work has been able to successfully pre-train a time-series model over a wide range of time-series types and modalities. The work performed in [14] found that adding datasets during pre-training inhibited learning and rapidly degrade model performance (a roughly 25% decrease in performance between the one-to-one and four-to-one scenario). Each of the above foundation models illustrates a relatively simple correlation between volume of data, model size, and performance; by training larger models on large datasets, we can train powerful models with exceptional transfer learning capabilities.

* 15pwq@queensu.ca

Many-to-one training is very desirable as it may allow us to scale up the amount of training data, model size, ease transferring to new domains and modalities and allow us to train one large model for application on unseen time-series. Furthermore, there are many negative future implications in failing to pre-train models in a many-to-one or many-to-many setting. Refer to the Analysis and Limitations section for more details.

To address this challenge, we propose a new framework that achieves state-of-the-art performance on time-series classification benchmarks. Our framework trains a single model on 162 time-series datasets for classification, each with varying length, channel dimension, and modality. To create an input-agnostic model, we propose the use of average adaptive pooling during the data loading process, which facilitates mixed batch training for time-series data. Our framework is also designed to be completely model agnostic, which allows us to leverage future improvements in time-series-specific model architectures. Our proposed framework addresses several key challenges in building foundation models for time-series analysis: (i) The models must be able to process inputs of any dimension, modality, or channel sizes. (ii) They need to be trained using modern parallel computing strategies. Primarily we would like to train the models with large batch sizes in mixed batch training, requiring all data be aligned in a way that it can be mixed within batches with the same input length and modality size. (iii) A great deal of research has gone into creating powerful time-series specific model architectures. We would like our training strategy to be completely model agnostic to leverage possible future improvements in this research area.

By introducing the `ADAPT` framework, we brought forward the state-of-the-art performance of pretrained models on diverse time-series sensor data, which will in turn contribute to various real-life downstream tasks.

2. Related Work

Foundation Models in Time-Series. Foundation models have been trained on vast datasets for broad applicability across various tasks, as defined in [15]. However, some models, either in full or in part, claim this designation in the time-series domain, such as those mentioned in recent studies [16, 17], do not meet the CRFM's criteria, not just because they lack the extensive training datasets but also the appropriate pretrianing architectures that can help counteract and solve the problem. These models' primary limitation is the relative scarcity of training data. The availability of diverse time-series data suggests that adopting a many-to-one pre-training approach may overcome this hurdle, enabling effective adaptation to different time-series applications.

Pre-Training Strategies. Foundation models have gained a significant status in the fields of NLP and computer vision, exhibiting remarkable success in solving a wide range of problems [18]. These models largely owe their success to self-supervised pretraining strategies that utilize modern parallel computing techniques to train on massive amounts of data. Recent research, such as [19], has found that, besides advancements in model architecture, training on more data for longer periods is critical in building powerful foundation models. Adding in more datasets for these models is simple since the underlying data in each of the commonly studied domains (i.e., text or image) is consistent across datasets.

Some recent works have focused on skipping the pre-training phase of time-series models by simply adopting pre-trained models from other domains [16, 20, 21]. The purpose is to take large pre-trained language or vision models and finetune them for downstream tasks. While we think this work is important, it still does not provide us with a solution for training and scaling models on diverse time-series. Investigating time-series specific pre-training strategies play a fundamental role in both building time-series specific models but also as components in future multi-modality time-series models, for example in time-series specific variants of CLIP [22].

Time-series are unique in terms of modality types, input lengths, modality dimensions, sampling rates, and other factors. As a result, recent works have focused on learning universal features for the time-series domain that transfer between time-series [23],[14]. One common feature of these works is the emphasis on learning strategies in which the model only has access to one dataset during pretraining and is fine-tuned on downstream domain data. Since there is no effective solution for aligning different data during pretraining, this has been a realistic scenario as the time-series domain comprises of a wide range of data modalities with varying characteristics. Some recent examples of self-supervised frameworks for pre-training within the time-series domain include ts2vec [23], TS-TCC [24], COST [25], CLOCS [26], CLUDA [27] and TNC [28]. Each of these models represents important contributions to acquiring universal representations from time-series datasets in the *one-to-many* scenario using self-supervised learning.

Many-to-one Pre-training. Previous work in training models in the many-to-one setting for time-series is limited, only explored very recently in TF-C [14]. This method focuses on a contrastive objective between the time and frequency components of the input signal to improve generalization from the pretraining dataset to the target modality. TF-C sets a state-of-the-art performance for time-series classification; however when trained in the many-to-one scenario, model performance degrades rapidly. In order to allow for many-to-one training they merge four pre-training datasets into one (SleepEEG, FD-A, HAR, and ECG). To address mismatch in dimensionalities, they limited each dataset to only one channel dimension and truncated or padded each dataset 1,500 observations. Their experiment demonstrated a 25% decrease in performance compared to pre-training on a singular dataset.

A comparable solution for implementing mixed-batch training across multiple modalities is presented in GATO [29], which is a large foundation model trained on controls, text, and image data. One key contribution is their embedding scheme which allows for proprioception data, image, text, continuous actions and discrete agent actions to be represented in a mixed batch form. They performed this by applying a separate embedding structure for each modality and each embedding strategy matches the channel dimensions between the modalities. While it may seem like this structure could work for the time-series domain, there are far too many modalities to adapt this strategy (in our setup, it would require over 150 different embedding strategies). Secondly, in the time-series domain, embedding is incorporated into the structure of the model (usually as linear or convolutional layers), meaning that in order to separate the model and embedding, we would first need to train a small embedding model for each dataset in the database. Finally, while the relative lengths of the input sequences in GATO were similar, data within the time-series domain significantly differs in input length, meaning that aligning the channel dimensions is insufficient. Our proposed solution, `ADAPT`, overcomes all these challenges and provides a basis for helping construct foundation models in the time-series domains.

3. Our Approach

3.1. Model Overview

Within the time-series domains, it has been standard practice to pre-train models using only a singular type of data and fine-tune them on a specific target task, which we call the *one-to-one* setup. In general, the models for diverse time series yet struggle to learn high-quality and transferable representations, under the *many-to-one* setting (refer to the related work section for more discussion), when the properties of the pretraining database are diverse or heterogeneous. Unlike the well-studied models that focus on homogeneous data (e.g., the large language models), our approach targets the time-series data that are made up of many diverse modalities and from different sources, such as smart home, human motion, healthcare, and fault detection data. The diverse time-series data have varying

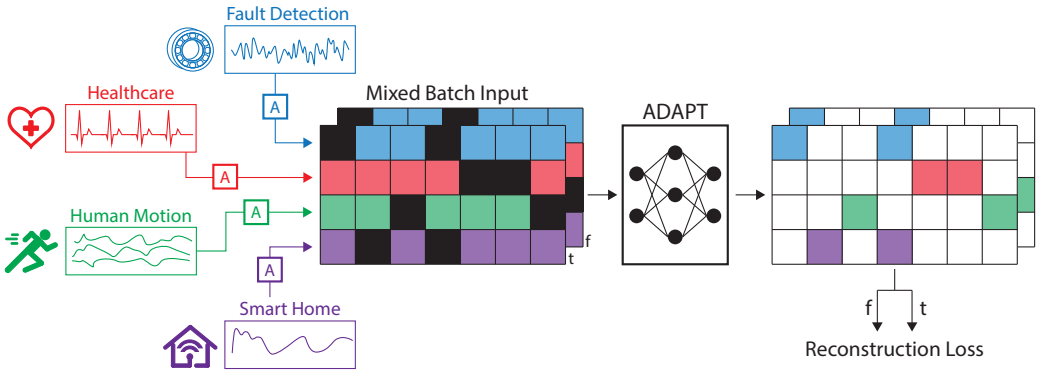

Figure 1. An overview of the adapt process and time-frequency masking algorithm. Each time-series is adaptive-pooled (in the [A] boxes) to a universal representation space and enable mixed-batch training among different modality types. At training time we add noise to the joint representation space in both the time and frequency space.

physical properties and channel dimensions, and no research yet has found an efficient way of unifying these properties for pretraining. The lack of a universal method for performing alignment across datasets prevents models in the time-series domain from building better representations.

The overall architecture of the proposed `ADAPT` approach is shown in Figure 1, which aims to design an effective method for building a unified representation space in the many-to-one training scenario by aligning physical properties for training and enabling different combination of time-series datasets and modalities during pretraining and down-stream finetuning. We believe that domain shift and adaptation can be mitigated by training a model on large volumes of increasingly heterogeneous data. If successful, transferring knowledge between domains should be easier as the model will have likely seen some related data during training.

3.2. **Adaptive Input Training**

Time-Series data consists of widely different input lengths and channel dimensions. This means that truncation or padding will may seriously degrade or dilute the inputs when there are large discrepancies between the maximum and minimum input size. Difficulties in aligning channel dimensions also put constraints on the model architecture requirements for processing varying channel dimensions. The physical constraints mean that allowing mixed batch training for time-series based models without seriously degrading model performance was an open problem. We propose the use of average adaptive pooling applied during data-loading in order to transform each input to the same representation space regardless of size.

We recognize that adaptively pooling the data risks losing fine-grained information in the data and may cause temporal distortions. However, the dimensional size of many popular datasets and data types are often smaller than that of the target dimension for embedding. This means that we are often up-sampling the data. We will show that any accuracy losses caused by the temporal distortions are mitigated by the increase in model performance using `ADAPT`.

Each input is adaptively pooled before batching to enable mixed batch pre-training. Refer to Algorithm 2 for a summary of our data processing strategy. While adaptive pooling could be applied for any pre-training strategy and any model architecture, the stability of the model is not guaranteed. We have created a masked-language-modeling-inspired pretraining

strategy using span-masking which also considers the frequency components of the input signal. Both the time and frequency input domains are embedded via separate time and frequency encoders and then fed to a stacked transformer encoder model with two separate MLP layers applied to the output to provide reconstruction loss in the time and frequency domain. Motivated by [30], which found that noise can emulate realistic data shifts in EEG data, we add random Gaussian noise to each sample during training.

Algorithm 1 Data processing for diverse time series. Data is loaded, normalized, and converted to frequency representation via FFT before adaptive pooling. The processed data from all datasets is then combined. Samples from the combined data undergo noise addition (ADDNOISE) and span-masking (SPANMASK) for batching.

Initialize $TrainingDataset \leftarrow \emptyset$
for each *dataset* **do**
    $data[time] \leftarrow$ LoadData($dataset$)
    $data[freq] \leftarrow$ FFT($data[time]$)
    AdaptPool(data[time], data[freq])
    Append *data* to $TrainingDataset$
end for
while $length(batch) < batchSize$ **do**
    $sample \leftarrow$ Extract sample from $TrainingDataset$
    $sample \leftarrow$ AddNoise($sample$)
    $sample \leftarrow$ SpanMask($sample$)
    Add current *sample* to *batch*
end while

Algorithm 2 Adaptive Pooling Procedure

$x \leftarrow$ input, output, D, H_{in}, W_{in}, H_{out}, W_{out}, Stride_D, Stride_H, Stride_W
function $Start\_Index(idx, O, I)$
    return $floor((idx * I)/O)$
end function
function $End\_Index(idx, O, I)$
    return $ceil(((idx + 1) * I)/O)$
end function
function AverageAdaptivePool2d(x)
    for $d \leftarrow 0$ to D **do**
        for $h \leftarrow 0$ to H_{out} **do**
            $H_{start} = Start\_Index(h, H_{out}, H_{in})$
            $H_{end} = End\_Index(h, H_{out}, H_{in})$
            $k_H = H_{end} - H_{start}$
            for $w \leftarrow 0$ to W_{out} **do**
                $W_{start} = Start\_Index(w, W_{out}, W_{in})$
                $W_{end} = End\_Index(w, W_{out}, W_{in})$
                $k_W = W_{end} - W_{start}$
                $output[d, h, w] = mean(input[d, h : h + k_H, w + k_W])$
            end for
        end for
    end for
end function

3.3. Masked Model Design

To properly implement masked language modeling for time-series data we use span masking, first ued in [31] for text and then adapted by [32] for sensor data, to prevent the

trivialization of the MLM objective. One major concern is that when randomly masking input values as typically performed in NLP, the model may be able to make accurate predictions with trivial strategies such as simply taking the closest unmasked value or the average of the unmasked values on either side, inhibiting the model's ability to learn. Span masking solves this by masking continuous spans within the input sequence. The model then needs to learn how to predict entire spans of the input data, and the above strategies are unlikely to provide accurate results.

Formally, span masking works by sampling the lengths, l, of each span mask from a geometric distribution which is restricted to a maximum sequence length, l_{max} of 10. We skew the geometric distribution by a factor p as shown in Equation 1. When the p value is lowered, the sequence lengths trend towards l_{max}.

$$P(l = k) = p(1-p)^{k-1}\epsilon(1, l_{max}) \tag{3.1}$$

We select random starting points and apply the span masks of length l until we have reached the desired masking ratio. We want our final model to perform well in pretraining with the masked data, but also during supervised training on the downstream tasks. To train the model to accept both masked and unmasked data we stipulate the masking is successful with a probability of p_m and that the masks are replaced by random values with a probability of p_r. If the input is masked, all masked values are replaced by zeros.

3.4. **Training Objective**

Given an input sample time series $s \in \mathbb{R}^{i \times j}$, where i and j are positive real numbers, we define the tuple $x = (A(s), A(FFT(s)))$, where A represents the adaptive pooling algorithm and FFT is the normalized Fast-Fourier-Transform function. We apply span masking as described above and then the time and frequency components of x are projected separately to transformer dimension d using two fully-connected neural networks L_t and L_f:

$$E_i = L_t(span(A(s))) + L_f(span(A(FFT(s)))) \tag{3.2}$$

We then pass our input embeddings E_i to our stacked transformer encoder model M to obtain output embeddings E_o. To reconstruct the masked portions of each input signal, T_p and F_p, we apply two fully-connected neural networks. The reconstruction loss is then given by:

$$L = \frac{1}{n}\sum_{i=q_1}^{q_n}(T_{p_i} - T_{m_i})^2 + \frac{1}{n}\sum_{i=q_1}^{q_n}(F_{p_i} - F_{m_i})^2 \tag{3.3}$$

where T_m and F_m are the original inputs, and $q_i \in Q$ are the masked indices of the sequence.

4. **Experiment Setup**

Datasets. `ADAPT` is trained on 162 different datasets used for classification, their properties are summarized in Table 2. Of the total 162 datasets, 158 come from the UEA and UCR time-series archive [33]. We also include time-series datasets SleepEEG [34], FD-A [35], HAR [36] and ECG [37]. The data was split by [14]. All data is normalized the data at the channel level by subtracting the mean and dividing by the standard deviation. In total our training datasets consist of almost 550,000 samples.

We test `ADAPT` on several popular classification benchmarks within the time series domain. These datasets are described in Table 1. Each dataset has a small amount of data for training and validation in order to challenge the limits of each self-supervised learning method.
Baselines. We compare our model with the state-of-the-art models: TS-SD [38], TS2vec [23], CLOCS [26], MIXING-UP [39], TS-TCC [24], SimCLR [40] and TF-C [14]. Each of these methods is pre-trained on the SleepEEG dataset as it presents complicated temporal dynamics and the largest dataset for pre-training. We report baselines results from [14] for comparison.

Table 1. Testing datasets and their respective data size.

Dataset	Len.	Train/Val/Test	Ch.	Cls.
Epilepsy	178	60 / 20 / 11 420	1	2
FD-B	5 120	60 / 21 / 13 559	1	3
Gesture	315	320 / 120 / 120	3	8
EMG	1 500	122 / 41 / 41	1	3

Table 2. Datasets used during pre-training.

Dataset	Len.	Samples	Ch.	Cls.
UEA/UCR	8–17 984	12–30 000	1–1345	2–60
SleepEEG	200	371 055	1	5
FD-A	5 120	8 184	1	3
HAR	128	10 299	9	6
ECG	1 500	43 673	1	4

Each model is fine-tuned five times with varying random seeds and we take the average across all trails.

To disentangle the performance gains due to improvements in our pre-training algorithm via the time and frequency span masking, compared to the benefits of mixed-batch, many-to-one pre-training, we train a model called `ADAPT(EEG)`. This model is identical to ADAPT, including the use of adaptive pooling to the same input dimensions, except that it is trained on only the SleepEEG dataset (as with the state-of-the-art baselines).

Implementation Details. Our model is trained for 1000 epochs with a batch size of 1024 on two NVIDIA A40 GPU's. We use a base learning rate of $5e-4$ following a cosine loss schedule. Our chosen optimizer is AdamW [41] using $\beta_1 = 0.9, \beta_2 = 0.999$ and warmup for 40 epochs. We clip the gradients at $max\_norm = 1$.

Our core architecture consists of six stacked transformer encoders [42] with a hidden dimensional size of 128. For the span-masking algorithm applied to the time and frequency representations we use $l_{max} = 10$, $p = 0.2$, $p_m = 0.8$ and $p_r = 0.2$.

5. Experimental Results

5.1. Classification Performance

For comparison we train two versions of the `ADAPT` architecture, one that is trained on all of the datasets in Table 2 and one that is only trained on the SleepEEG dataset as with the other baselines. The performances of `ADAPT` and other models on the classification benchmarks are shown in Table 3. As noted in previous work [14], pretraining time-series models on many datasets can degrade the quality of the model. We compare the performance of our model to other state-of-the-art models on in-domain pretraining on the Epilepsy dataset. For in-domain classification `ADAPT` is competitive with the other state of the art base-lines. Interestingly, `ADAPT` outperforms `ADAPT(EEG)` which indicates that our pretraining architecture benefits from mixed batch pretraining and that including datasets from outside the target domain can increase the performance of our model.

As previously stated, the focus within the time-series domain has been on training models which can generalize to unseen domains. This importance has been exacerbated by the limitations of pre-trained time-series models to singular datasets. `ADAPT` outperforms other state-of-the-art models across the generalization datasets on average, particularly within the FD-B and EMG datasets where both `ADAPT` models greatly improve the state-of-the-art performance. Interestingly the `ADAPT(EEG)` model outperforms `ADAPT` on the FD-B benchmark. Considering that `ADAPT(EEG)` provided worse performance than `ADAPT` on the Epilepsy (in-domain) benchmark, it is not clear why it it achieves close to perfect classification accuracy on this benchmark. We believe this may be a result of vastly increasing the diversity of the training data between the two models without proportionally increasing the amount of training data, resulting in some instabilities in model performance. `ADAPT` outperforms `ADAPT(EEG)` on all the other generalization benchmarks.

Table 3. Overall classification accuracy of ADAPT compared to other state-of-the-art pretrained models for time-series classification. Baseline implementations are from [14] and are pretrained on the SleepEEG dataset. We pre-train two models, ADAPT(EEG) which is only pre-trained on the SleepEEG datasets, and ADAPT which is trained on all of the pretraining datasets in Section 5.2. The subscripts are standard deviations.

Dataset: Epilepsy					**Dataset: FD-B**				
Model	**Acc**	**Prec**	**Rec**	**F1**	**Model**	**Acc**	**Prec**	**Rec**	**F1**
TS-SD	$89.5_{5.2}$	$80.2_{22.4}$	$76.5_{14.9}$	$77.7_{18.6}$	TS-SD	$55.7_{2.1}$	$57.1_{5.4}$	$60.5_{2.7}$	$57.0_{3.3}$
TS2vec	$94.0_{0.4}$	$90.6_{1.2}$	$90.4_{1.2}$	$90.5_{0.7}$	TS2vec	$47.9_{1.1}$	$43.4_{0.9}$	$48.4_{2.0}$	$43.9_{1.1}$
CLOCS	$95.1_{0.3}$	$93.0_{0.7}$	$91.3_{1.7}$	$92.1_{0.7}$	CLOCS	$49.3_{3.1}$	$48.2_{3.2}$	$58.7_{3.9}$	$47.5_{4.9}$
Mixing-up	$80.2_{0.0}$	$40.1_{0.0}$	$50.0_{0.0}$	$44.5_{0.0}$	Mixing-up	$67.9_{2.5}$	$71.5_{3.4}$	$76.1_{2.0}$	$72.7_{2.3}$
TS-TCC	$92.5_{1.0}$	$94.5_{0.5}$	$81.8_{2.6}$	$86.3_{2.2}$	TS-TCC	$55.0_{2.2}$	$52.8_{2.9}$	$64.0_{1.8}$	$54.2_{3.4}$
SimCLR	$90.7_{3.4}$	$92.2_{1.7}$	$78.6_{10.7}$	$81.8_{10.0}$	SimCLR	$49.2_{4.4}$	$54.5_{10.2}$	$47.6_{8.9}$	$42.2_{11.4}$
TF-C	$94.9_{2.5}$	$94.6_{1.1}$	$89.1_{2.2}$	$91.5_{5.3}$	TF-C	$69.4_{2.3}$	$75.6_{3.5}$	$72.0_{2.6}$	$74.9_{2.7}$
ADAPT(EEG)	$88.6_{3.5}$	$82.2_{4.2}$	$91.1_{1.6}$	$84.8_{3.9}$	ADAPT(EEG)	$97.3_{2.8}$	$98.2_{1.8}$	$98.0_{2.0}$	$98.0_{2.0}$
ADAPT	$93.6_{2.9}$	$90.1_{4.9}$	$91.7_{0.9}$	$88.5_{4.6}$	ADAPT	$91.2_{4.2}$	$91.0_{4.1}$	$91.8_{4.2}$	$88.8_{5.7}$
Dataset: Gesture					**Dataset: EMG**				
Model	**Acc**	**Prec**	**Rec**	**F1**	**Model**	**Acc**	**Prec**	**Rec**	**F1**
TS-SD	$69.2_{4.4}$	$67.0_{4.7}$	$68.7_{4.9}$	$66.6_{4.4}$	TS-SD	$46.1_{0.0}$	$15.5_{0.0}$	$33.3_{0.0}$	$21.1_{0.0}$
TS2vec	$69.2_{3.3}$	$65.5_{3.6}$	$68.5_{3.5}$	$65.7_{3.9}$	TS2vec	$78.5_{3.2}$	$80.4_{7.5}$	$67.9_{4.0}$	$67.7_{5.0}$
CLOCS	$44.3_{5.2}$	$42.4_{7.9}$	$44.3_{5.2}$	$40.1_{6.0}$	CLOCS	$69.9_{3.2}$	$53.1_{7.5}$	$53.5_{2.9}$	$51.4_{4.1}$
Mixing-up	$69.3_{2.3}$	$67.2_{2.3}$	$69.3_{2.3}$	$65.0_{3.1}$	Mixing-up	$30.2_{5.3}$	$11.0_{1.3}$	$25.8_{4.6}$	$15.4_{2.0}$
TS-TCC	$71.9_{3.5}$	$71.4_{3.5}$	$71.7_{3.7}$	$69.8_{3.6}$	TS-TCC	$78.9_{1.9}$	$58.5_{9.7}$	$63.1_{9.9}$	$59.0_{9.5}$
SimCLR	$48.0_{5.9}$	$59.5_{16.2}$	$54.1_{19.5}$	$49.6_{18.7}$	SimCLR	$61.5_{5.8}$	$53.6_{17.2}$	$49.9_{12.1}$	$47.1_{14.9}$
TF-C	$76.4_{2.0}$	$77.3_{3.6}$	$74.3_{2.7}$	$75.7_{3.1}$	TF-C	$81.7_{2.9}$	$72.7_{3.5}$	$81.6_{2.9}$	$76.8_{3.1}$
ADAPT(EEG)	$72.5_{1.2}$	$70.8_{0.8}$	$72.5_{1.2}$	$70.7_{0.7}$	ADAPT(EEG)	$96.6_{4.5}$	$94.4_{6.1}$	$97.3_{3.6}$	$95.0_{6.2}$
ADAPT	$77.0_{2.5}$	$74.9_{3.7}$	$77.0_{2.5}$	$75.1_{2.9}$	ADAPT	$98.5_{1.2}$	$96.7_{2.7}$	$98.8_{1.0}$	$97.6_{2.0}$

A key contribution is that contrary to what has been observed in previous literature, **ADAPT does not rapidly regrade performance in the many-to-one scenario**. ADAPT sets several new state-of-the-art performances for domain adaptation and generalization. It is also easily amenable to any model architecture. Our results justify both the effectiveness of mixed-batch pretraining in certain key downstream generalization scenarios and also the effectiveness of our time-frequency masking strategy during pretraining.

5.2. Performance Compared to Dataset Properties

So far we have demonstrated that ADAPT is an effective pretraining method with allows many-to-one pretraining without degrading the accuracy of the pre-trained model as noted in previous works [14]. We compare the model performance across the UCR and UEA archives with several key dataset properties in order to elucidate some insights on the effectiveness of adaptive input training. We compare the accuracy with the number of channel dimensions, input length, number of classes and the ratio of training and testing data available during finetuning in Table 2. Along with each comparison we show the Pearson correlation between accuracy and the corresponding dataset property. Interestingly, the correlations between the length and channel complexity of the input and the accuracy of the model are extremely weak. We further explore the performance of ADAPT with respect to total dimensional size. Total dimensional size is given by dataset length multiplied by the number of channels. We chose a model embedding size of (256,32). Any dataset that is significantly below this total dimensional size will be up-sampled and those above this threshold will be down-sampled in order to meet these requirements. These results indicate that the model accuracy is not strongly correlated with the up-sampling or down-sampling of the dataset. This leads us to the conclusion that the model accuracy is more dependant on the specific task type rather

than a specific dataset property. The accuracy was slightly correlated to the number of distinct classes within the dataset which futher supports this claim. These findings are highly encouraging as it shows that ADAPT can be a general embedding strategy for time-series data and can be used universally independent of dataset properties. For details regarding the model accuracy on each of the 158 datasets in the UCR and UEA archives.

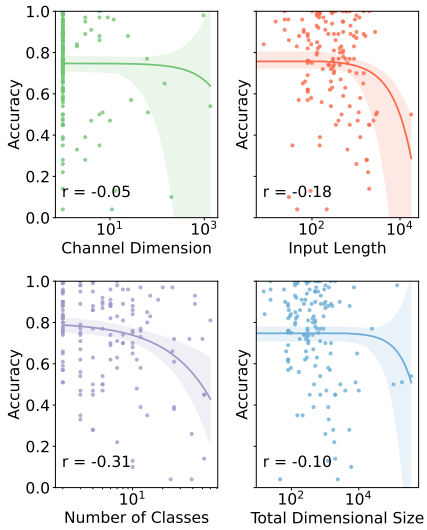

Figure 2. Model accuracy on UCR and UEA datasets compared to basic dataset properties. The "Length" and "Channel Dimension" corresponds to the dimensionality of the data or the total length of the input and number of channels at each time period. "Number of Classes" corresponds to the number of classes for classification in the downstream dataset, and "Total Dimensional Size" refers to length of the input multiplied by the number of channel dimensions. Overall, we can see that the performance is not strongly linked to the physical properties of the datasets. The number of classes of a dataset could represent an increase in dataset complexity and the model has a weak negative correlation as expected.

5.3. **Representation Visualization for Classification**

One potential concern is that adaptive pooling could significantly distort the input space. We use T-SNE [43] to visualize the change in put in both the time and frequency domain caused by adaptive pooling. Figure 3 shows the T-SNE analysis for the Gesture dataset, chosen for its diverse number of classes and channels. While there are small differences between the raw and adaptive inputs, the clustering of the classes and their relation to one-another remains largely intact.

6. **Conclusion**

In this paper we introduce ADAPT to investigate many-to-one pre-training strategies for time-series analysis. Our adaptive pooling embedding strategy allows for training a time-series model using over 150 datasets and establishing a new state of the art on different classification benchmarks. We demonstrate that adaptive pooling can be applied as a time-series embedding strategies which can handle data over a wide range of physical properties, input lengths and modalities, as well as dataset complexity with diverse classes. ADAPT only changes the physical properties of the dataset and does not make any special considerations for the modality, showing that it was the diversity of the physical dataset properties (input

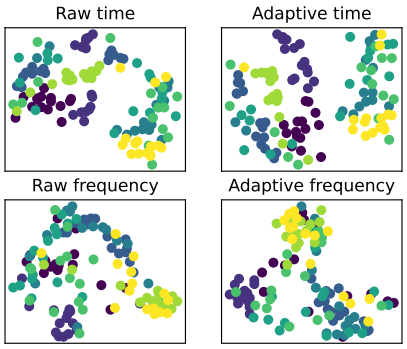

Figure 3. T-SNE visualisation of the time and frequency inputs before and after adaptive pooling on the Gesture dataset. We note that there is no discernible drop in the quality of inputs after they are transformed. The relative groupings and distributions between the 8 different classes remain largely intact.

length and channel dimension) that have prevent time-series models from learning quality representations in the many-to-one training scenario. We hope our work will help enable the creation of capable foundation models for the time-series and sensor data and bring the advancements in these areas in line with other domains.

Acknowledgements

We would like to acknowledge the support and funding from Ingenuity Labs Research Institute at Queen's University and Ingenuity Labs Seed Funding. This research was enabled in part by support provided by Calcul Québec and the Digital Research Alliance of Canada. This Research is partially supported by NSERC Discovery Grants.

References

[1] J. N. Acosta, G. J. Falcone, P. Rajpurkar, and E. J. Topol. "Multimodal biomedical AI". In: *Nature Medicine* 28 (9 Sept. 2022), pp. 1773–1784. ISSN: 1078-8956. DOI: 10.1038/s41591-022-01981-2.

[2] L. Cao. "AI in Finance: Challenges, Techniques, and Opportunities". In: *ACM Comput. Surv.* 55.3 (2022). ISSN: 0360-0300. DOI: 10.1145/3502289. URL: https://doi.org/10.1145/3502289.

[3] N. Kashpruk, C. Piskor-Ignatowicz, and J. Baranowski. "Time Series Prediction in Industry 4.0: A Comprehensive Review and Prospects for Future Advancements". In: *Applied Sciences* 13.22 (2023). ISSN: 2076-3417. DOI: 10.3390/app132212374. URL: https://www.mdpi.com/2076-3417/13/22/12374.

[4] K. Amankulova, N. Farmonov, and L. Mucsi. "Time-series analysis of Sentinel-2 satellite images for sunflower yield estimation". In: *Smart Agricultural Technology* 3 (Feb. 2023), p. 100098. ISSN: 27723755. DOI: 10.1016/j.atech.2022.100098.

[5] P. Amoatey, N. J. Osborne, D. Darssan, Z. Xu, Q.-V. Doan, and D. Phung. "The effects of diurnal temperature range on mortality and emergency department presentations in Victoria state of Australia: A time-series analysis". In: *Environmental Research* 240 (2024), p. 117397.

[6] Y. Liu, P. Ramin, X. Flores-Alsina, and K. V. Gernaey. "Transforming data into actionable knowledge for fault detection, diagnosis and prognosis in urban wastewater systems with AI techniques: A mini-review". In: *Process Safety and Environmental Protection* 172 (2023), pp. 501–512. ISSN: 0957-5820. DOI: https://doi.org/10.1016/j.psep.2023.02.043. URL: https://www.sciencedirect.com/science/article/pii/S0957582023001428.

[7] M. H. M. Ghazali and W. Rahiman. "Vibration-Based Fault Detection in Drone Using Artificial Intelligence". In: *IEEE Sensors Journal* 22.9 (2022), pp. 8439–8448. DOI: 10.1109/JSEN.2022.3163401.

[8] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[9] A. Q. Jiang et al. *Mixtral of Experts*. 2024. arXiv: 2401.04088 [cs.LG].

[10] R. Thoppilan et al. *LaMDA: Language Models for Dialog Applications*. 2022. arXiv: 2201.08239 [cs.CL].

[11] A. Baevski, H. Zhou, A. Mohamed, and M. Auli. "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS'20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[12] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022. arXiv: 2204.06125 [cs.CV].

[13] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[14] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik. "Self-Supervised Contrastive Pre-Training For Time Series via Time-Frequency Consistency". In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. 2022. URL: https://openreview.net/forum?id=OJ4mMfGKLN.

[15] Stanford CRFM. *Stanford Center for Research on Foundation Models (CRFM)*. Online. Accessed: 2024-01-30. 2021. URL: https://crfm.stanford.edu/.

[16] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin. *One Fits All:Power General Time Series Analysis by Pretrained LM*. 2023. arXiv: 2302.11939 [cs.LG].

[17] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long. "TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=ju_Uqw384Oq.

[18] C. Zhou et al. *A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT*. 2023. arXiv: 2302.09419 [cs.AI].

[19] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].

[20] Z. Li, S. Li, and X. Yan. "Time Series as Images: Vision Transformer for Irregularly Sampled Time Series". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=ZmeAoWQqe0.

[21] C. Chang, W.-Y. Wang, W.-C. Peng, and T.-F. Chen. *LLM4TS: Aligning Pre-Trained LLMs as Data-Efficient Time-Series Forecasters*. 2024. arXiv: 2308.08469 [cs.LG].

[22] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[23] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu. "TS2Vec: Towards Universal Representation of Time Series". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.8 (2022), pp. 8980–8987. DOI: 10.1609/aaai.v36i8.20881. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20881.

[24] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan. "Time-Series Representation Learning via Temporal and Contextual Contrasting". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. Ed. by Z.-H. Zhou. Main Track. International Joint Conferences on Artificial Intelligence Organization, Aug. 2021, pp. 2352–2359. DOI: 10.24963/ijcai.2021/324. URL: https://doi.org/10.24963/ijcai.2021/324.

[25] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi. "CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=PilZY3omXV2.

[26] D. Kiyasseh, T. Zhu, and D. A. Clifton. "CLOCS: Contrastive Learning of Cardiac Signals Across Space, Time, and Patients". In: *International Conference on Machine Learning*. 2020.

[27] Y. Ozyurt, S. Feuerriegel, and C. Zhang. "Contrastive Learning for Unsupervised Domain Adaptation of Time Series". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=xPkJYRsQGM.

[28] S. Tonekaboni, D. Eytan, and A. Goldenberg. "Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=8qDwejCuCN.

[29] S. Reed et al. *A Generalist Agent*. 2022. arXiv: 2205.06175 [cs.AI].

[30] N. Wagh, J. Wei, S. Rawal, B. M. Berry, and Y. Varatharajah. "Evaluating Latent Space Robustness and Uncertainty of EEG-ML Models under Realistic Distribution Shifts". In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. 2022. URL: https://openreview.net/forum?id=KRk0lBRPpOC.

[31] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. "SpanBERT: Improving Pre-training by Representing and Predicting Spans". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 64–77. DOI: 10.1162/tacl_a_00300. URL: https://aclanthology.org/2020.tacl-1.5.

[32] H. Xu, P. Zhou, R. Tan, M. Li, and G. Shen. "LIMU-BERT: Unleashing the Potential of Unlabeled Data for IMU Sensing Applications". In: *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. SenSys '21. Coimbra, Portugal: Association for Computing Machinery, 2021, 220–233. ISBN: 9781450390972. DOI: 10.1145/3485730.3485937. URL: https://doi.org/10.1145/3485730.3485937.

[33] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. *The UCR Time Series Archive*. 2019. arXiv: 1810.07758 [cs.LG].

[34] B. Kemp, A. Zwinderman, B. Tuk, H. Kamphuisen, and J. Oberye. "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG". In: *IEEE Transactions on Biomedical Engineering* 47.9 (2000), pp. 1185–1194. DOI: 10.1109/10.867928.

[35] C. Lessmeier, J. Kimotho, D. Zimmer, and W. Sextro. "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: July 2016.

[36] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra. *Human Activity Recognition Using Smartphones*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C54S4K. 2012.

[37] G. Clifford, C. Liu, B. Moody, L. wei Lehman, I. Silva, Q. Li, A. Johnson, and R. Mark. "AF Classification from a Short Single Lead ECG Recording: the Physionet Computing in Cardiology Challenge 2017". In: Sept. 2017. DOI: 10.22489/CinC.2017.065-469.

[38] J. Shi, W. Ye, and Z. Qin. "Self-Supervised Pre-training for Time Series Classification". In: July 2021, pp. 1–8. DOI: 10.1109/IJCNN52387.2021.9533426.

[39] K. Wickstrøm, M. Kampffmeyer, K. O. Mikalsen, and R. Jenssen. "Mixing up Contrastive Learning: Self-Supervised Representation Learning for Time Series". In: *Pattern Recogn. Lett.* 155.C (2022), 54–61. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2022.02.007. URL: https://doi.org/10.1016/j.patrec.2022.02.007.

[40] C. I. Tang, I. Perez-Pozuelo, D. Spathis, and C. Mascolo. *Exploring Contrastive Learning in Human Activity Recognition for Healthcare*. 2021. arXiv: 2011.11542 [cs.LG].

[41] I. Loshchilov and F. Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[43] L. van der Maaten and G. Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.