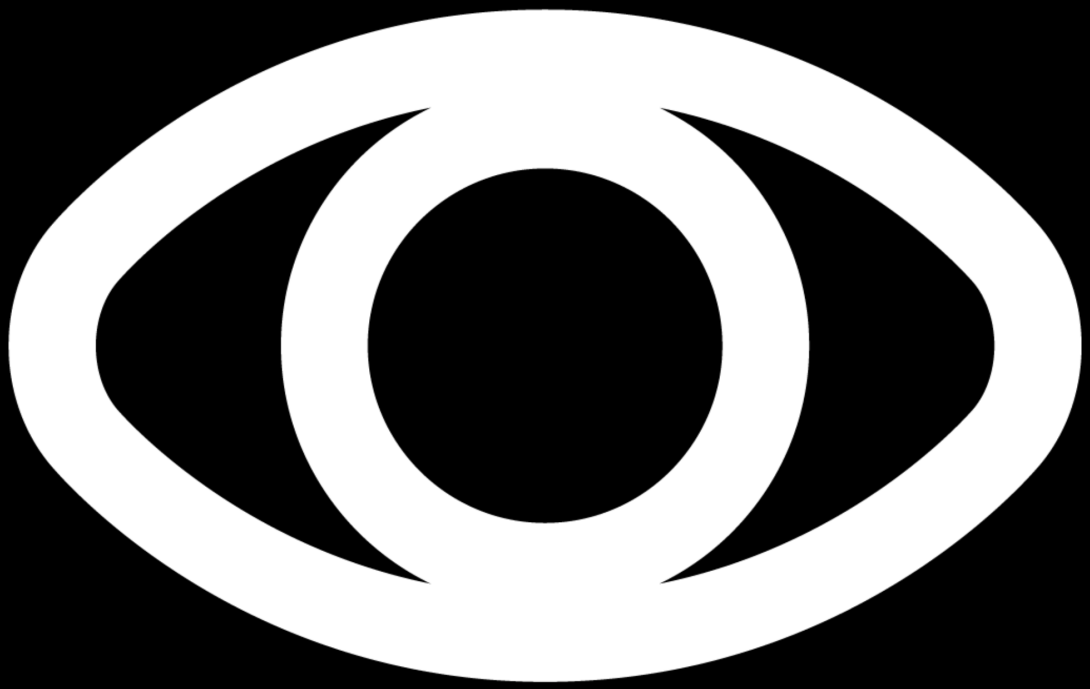
Self-Supervised Human Activity Recognition

Self-supervised Transformer utilizing Time Series Eye Gaze data on HAR

CESE5020: Research Project

Tongyun Yang

Delft University of Technology



TU Delft

Self-Supervised Human Activity Recognition

Self-supervised Transformer utilizing Time Series Eye Gaze data on HAR

by

Tongyun Yang

Student Name	Student Number
Tongyun Yang	5651794

Instructor:	Prof. Guohao Lan & Lingyu Du
Project Duration:	Sep., 2023 - Jan., 2024
Faculty:	EEMCS, Delft

Cover: Eye image downloaded from pngfind.com, and cover made by the author Tongyun Yang

TU Delft

Contents

Abstract ii

Nomenclature iii

1 Introduction 1

2 Related Work 2
- 2.1 Cognitive Context Recognition with Eye Movement . . . 2
- 2.2 State-Of-The-Art Machine Learning . . . 3
- 2.3 Unsupervised Learning on Ubiquitous Computing . . . 3

3 Methods 4
- 3.1 Model Architecture . . . 4
- 3.2 MLM Pretrain . . . 4
- 3.3 Input Projection & Reconstruction . . . 7
- 3.4 Classification . . . 7

4 Results 9
- 4.1 Datasets and Signal Preprocessing . . . 9
- 4.2 Experimental Design . . . 9
- 4.3 Input Projection & Reconstruction Comparison . . . 10
- 4.4 Experimental Results . . . 11
  - 4.4.1 DesktopActivity Results . . . 12
  - 4.4.2 ReadingActivity Results . . . 12

5 Conclusion & Recommendations 14
- 5.1 Conclusion . . . 14
- 5.2 Recommendations for Future Research . . . 14
  - 5.2.1 Further Investigation of Reconstruction Phenomena . . . 14
  - 5.2.2 Model Architecture Optimization . . . 15
  - 5.2.3 Cross-Activity Transfer Learning . . . 15
  - 5.2.4 Bridging the Gap with Hand-Crafted Features . . . 15

References 16

6 Table & Figures 19

Abstract

This study explores the application of self-supervised learning techniques, specifically Masked Language Modeling (MLM) pretraining, to Human Activity Recognition (HAR) using eye gaze data. The research focuses on two datasets: DesktopActivity and ReadingActivity, utilizing a Transformer-based architecture adapted for time series data.

The study compares the performance of MLM-pretrained models against fully supervised models across various fine-tuning scenarios, with window sizes of 30 and 60 seconds. Results demonstrate that MLM pretraining generally outperforms fully supervised approaches, particularly when limited labeled data is available. The research also investigates the impact of different reconstruction techniques in the pretraining phase, finding that convolutional layers offer superior performance and efficiency compared to linear layers.

Key findings include the effectiveness of MLM pretraining in learning generalizable features from eye gaze data and the positive impact of incorporating diverse data sources during fine-tuning. However, the study also reveals challenges in matching the performance of methods using hand-crafted features and inconsistencies in results between different window sizes.

The research concludes by proposing several directions for future work, including further investigation of reconstruction phenomena, exploration of cross-activity transfer learning, and potential integration of insights from hand-crafted feature approaches into deep learning architectures for eye gaze analysis.

Nomenclature

Abbreviations

Abbreviation	Definition
AR	Augmented Reality
VR	Virtual Reality
HAR	Human Activity Recognition
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
MLM	Masked Language Modeling
NSP	Next Sentence Prediction
EOG	Electrooculographic
LFI	Laser Feedback Interferometry
mRMR	Maximum Relevance – Minimum Redundancy
SVM	Support Vector Machine
LDA	Latent Dirichlet Allocation
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
MVTS	Multivariate Time Series Transformer
NLP	Natural Language Processing
MSE	Mean Sqaured Error

1 Introduction

Recently, with the development of AR and VR, HMDs are becoming popular. HMDs are considered to be the next ubiquitous wearable devices following smart watches. Among all the techniques used on applications of wearable devices, HAR is one of the most crucial. The purpose of HAR is to classify user's activity and benefit applications, e.g., health monitoring [20], sports performance monitoring [12], gaming [13] and smart homes [3].

In general, there are two types of scenarios when it comes to methods of HAR, inertial-sensor-based and video-based [30]. Smart watches and other wearable devices usually leverage the inertial-sensor-based methods, however, due to differences in nature of wearing HMDs, they usually combine the two scenarios. HMDs are usually equipped with both inertial sensors and cameras that captures user's eye movement. In this project, the original data are video-based eye movement, but the data used for HAR are inertial-sensor-based-like time series which were extracted from the captured videos.

Though some researches have been conducted by biologists and psychologists showing that eye movement are related to everyday activities [10, 17, 18] and sports performance [34], no solid conclusion could be drawn revealing the correlation. This is due to the lack of studies in cognitive context related to eye movements in the past.

The aim of this project is to explore whether state-of-the-art machine learning methods could bridge the gap of HAR in eye movement without apriori knowledge. The first approach is to leverage the attention mechanism [29] and verify the performance of a light-weight transformer model. The second approach is to introduce an MLM pretraining process from BERT [7], it enhances the model's capability of learning the hidden pattern beneath the eye movement. At last, the gain in performance and the decrease in the amount of required samples utilizing the pretraining process is verified.

The specific contributions of this project are: (1) the exploration of a light-weight transformer on HAR with eye movement data; (2) the performance gain and (3) the decrease in the amount of required samples with an MLM pretrain process in addition to the light-weight transformer. Though these contribution are no solid conclusion to the hidden pattern in eye movement, they reveal more about the correlation between eye movement and different activities.

2

Related Work

The work in this project is related to previous works on: (1) recognition of user's cognitive context with eye movement, (2) state-of-the-art machine learning algorithms and (3) unsupervised learning mechanisms applied on ubiquitous computing.

2.1. Cognitive Context Recognition with Eye Movement

Eye movement analysis has a long history which dates back more than a hundred years. Humans' eyes are constantly in motion, e.g., saccades, smooth pursuit and vergence movements, however, humans still maintain a stable and continuous visual world. Understanding how visual information is processed, and how eye movement impacts the visual perception is the key to recognizing the cognitive context.

During the early years, in 1900, Dodge [8] discussed the visual perception during eye movement while reading, he concluded that a smooth visual transition from word to word or line to line is important to maintain efficiency and accuracy. In 1924, during the change from vertical text alignment to horizontal in China, Shen [26] measured the number and duration of fixation-pauses made reading the same text in both manners using photographic technique. Shen [26] warned about the trend by concluding that reading horizontally is less efficient than vertically. It was not until the late 20^{th} century when researches on the relationship between eye movement and cognitive context were conducted. In 1998, Rayner [24] categorized and summarized studies conducted during the past 20 years revealing the cognitive context of eye movement. E.g., movement performing 'reading'[1] and 'searching'[2] are similar, but have important differences; stuttering children make more fixations[3] and regressions[4] than non-stutterers; reading contrapuntal music tends to generate a horizontal series of saccades followed by a vertical movement, while reading homophonic music results in a vertical fixation sequence alternating from one half to the other, etc. These are early evidences of eye movement revealing the human's cognitive context.

Nowadays, as data are more achievable with technologies, researches revealing the cognitive context beneath eye movement are conducted utilizing EOG, eye gaze, IMU, LFI data, etc. Bulling et al. [4] devised hand-crafted features on eye movement using a wearable EOG systems. 70.5% overall accuracy was achieved with a 30 seconds sliding window and 0.25 seconds step size using mRMR and SVM identify 5 commonly performed working activities. Kunze et al. [15] conducted a research identifying different types of documents reading based purely on extracted features from eye gaze data. They measured an accuracy of 74% with a 1 minute sliding window using J48 decision tree classifier. Steil and Bulling [28] proposed a method focusing on long-term activity recognition in an unsupervised manner. The method encodes recorded eye movement in bag-of-words representation and classifies via an LDA topic model. The result shows that the achieved accuracy is competitive of supervised SVM method. With the rise of deep learning mechanisms, Meyer et al. [22] introduced 1D-CNN to HAR with

[1] normal reading behaviour, left to right, up to down
[2] searching for target word/letter in less meaningful text, e.g., letter strings
[3] eyes remain relatively still for about 200-300 ms
[4] right-to-left movements along the line in English reading or movements back to previously read lines are regressions

LFI eye movement data combined with IMU head movment data. They achieved an 88.15% accuracy with transfer learning and 80.98% without.

2.2. State-Of-The-Art Machine Learning

As Transformer was introduced by Vaswani et al. [29], it outperformed all the predecessors, e.g., RNNs and CNNs, in the tasks of achieving the highest scores in English-to-German and English-to-French translation. With the use of attention mechanism introduced by Bahdanau, Cho, and Bengio [1], Transformer not only solved the limitations in retaining information between two pieces of information that are far apart. It also utilized GPU resources better, leading to shorter model training time.

It was later proven that the Transformer architecture performed well on other tasks besides translation. A lot of other researches based on Transformer achieved extremely high accuracy on tasks in other fields, i.e., BERT [7] in natural language understanding, ViT [9] in image recognition, TimeSformer [2] in video classification, DETR [5] in object detection, ViLT [14] in language-vision multimodality downstream tasks, etc.

Given that Transformer constitutes the state of the art performance for various tasks, applying it to HAR utilizing eye movement data in this project is naturally thought of. Classification of time series tasks are dominant by non-deep learning methods, e.g., HIVE-COTE [23], ROCKET [6] and TS-CHIEF [27]. However, there are also Transformer-based models that showed superior performance in time series forecasting and imputation [19, 21]. Zerveas et al. [33] proposed a Transformer-based framework which outperformed most of the existing methods for classification on multivariate time series. The framework is used in this project.

2.3. Unsupervised Learning on Ubiquitous Computing

As mentioned previously, BERT is a Transformer-based model proposed by Devlin et al. [7] which achieved state-of-the-art performance on natural language understanding. Leveraging two unsupervised pretraining tasks is a reason of it achieving superior performances. The two tasks are masked language modeling (MLM) and next sentence prediction (NSP). The earlier task masks certain amount of words in a sentence and the model would predict what the missing words originally are, while the later one predicts whether the two sentences come one after another. MLM enhances the model's ability to learn the relation between words, and NSP enables the model to understand longer-term dependencies across sentences.

Xu et al. [32] employed the idea of BERT in the task of HAR with mobile IMU data and proposed LIMU-BERT. In order to adapt the multivariate time series IMU data, several modifications are made to the original BERT architecture. The adjustments include normalization of the original data, loss function and the masking mechanism of the MLM pretrain task. LIMU-BERT only adopted MLM as the pretraining task, as the tasks are classifications rather than forecasting, and the temporal relations and feature distributions in IMU data is the prior learning task for the model. The result shows that LIMU-BERT can not only outperform existing approaches in most of the HAR tasks with IMU data, but also achieve a much higher accuracy compared to others when there are limited amount of labeled data. The idea of MLM pretrain task is also introduced in this project, as it is one of the reasons LIMU-BERT remains robust when labeled data are limited.

3 Methods

3.1. Model Architecture

The main architecture of the model is based on the multivariate time series transformer framework (mvts) [33], adjustments to the architecture are mentioned in detail when necessary. The mvts shares a very similar structure as the original Transformer without the decoder [29], as shown in Figure 3.1. During the fine-tuning phase following MLM pretraining, we replace the 'Reconstruction' linear layer with a classifier and switch the loss function from MSE to cross-entropy loss. Figure 3.2 shows the generic part of the model, which is common across pretraining and fine-tuning.

In particular, each sample $X \in \mathbb{R}^{w \times m}$, with sequence length w and m feature dimensions, consists of a sequence of w feature vectors $x_t \in \mathbb{R}^m = [x_1, x_2, ..., x_w]$. Initially, the feature vectors are standardized. Values from each dimension are subtracted by the mean and divided by the variance computed across all data. The vectors x_t are then each linearly projected onto a d-dimensional space, where d is the dimension of the representation of each time step in the transformer model. The d-dimensional representation space is referred as model dimension:

$$u_t = W_p x_t + b_p \tag{3.1}$$

where $W_p \in \mathbb{R}^{d \times m}$, $b_p \in \mathbb{R}^d$ are parameters and $u_t \in \mathbb{R}^d, t = 0, ..., w$ are the feature vectors, which correspond to word vectors in the original Transformer [29]. Positional encoding, which compensates for the Transformer architecture's insensitivity to the order of input, is added to these feature vectors. Unlike the original Transformer that uses a deterministic sinusoidal pattern [29], this work employs a fully learnable positional encoding, which has shown improved performance in empirical studies [33].

Despite conventional use of layer normalization in NLP tasks due to its ability to handle the variability of input sequence length, and preserve the independence among data points. In this work, batch normalization is employed following experiments [33] that suggest it provides a substantial performance advantage by mitigating the effect of outliers. In the work of mvts [33], the placement of normalization is not specified. It has been found out from the source code that the normalization is directly placed in a post-norm fashion, as shown in Figure 3.3. However, several empirical [11, 25] and theoretical [31] studies advocate for a pre-norm design in Transformer architectures. This earlier placement of normalization layers has been shown to stabilize the training process and potentially enhance performance.

3.2. MLM Pretrain

In the context of NLP, MLM involves randomly masking some words in the input sentences and training the model to predict these masked words based on the surrounding context. Similarly, for time series data, MLM is adapted by masking out certain time steps (values) and training the model to predict these masked values from the surrounding data points. This method helps the model learn the underlying patterns and dependencies within the time series data. The corresponding setup is depicted in Figure 3.4.

The masking process begins when loading the data. A binary noise mask $M \in \mathbb{R}^{w \times m}$ is created independently for each training sample, where 0 indicates a masked value and 1 indicates an unmasked

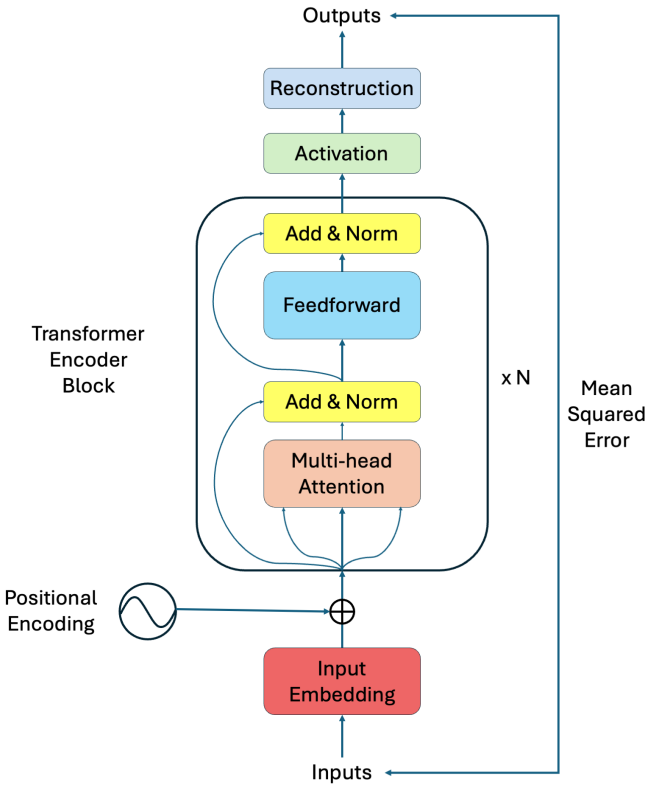


Figure 3.1: Architecture of mvts MLM pretraining task

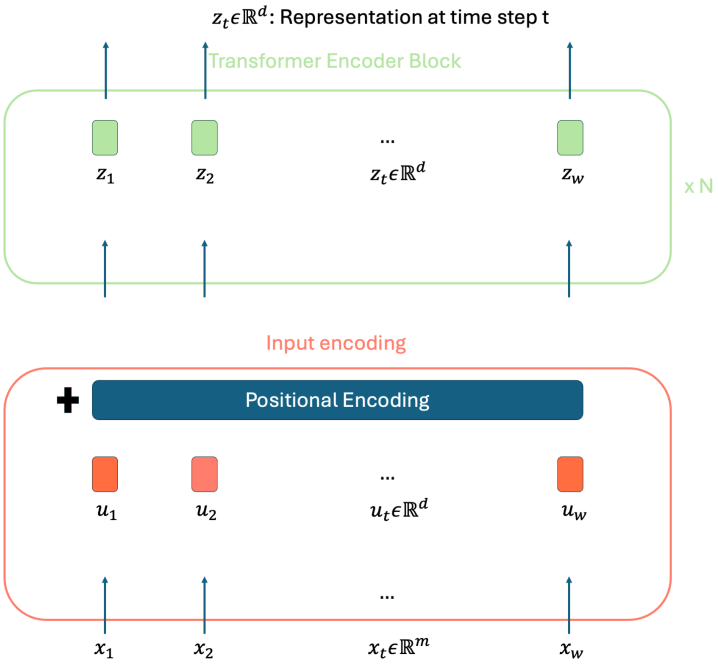


Figure 3.2: Generic model architecture overview

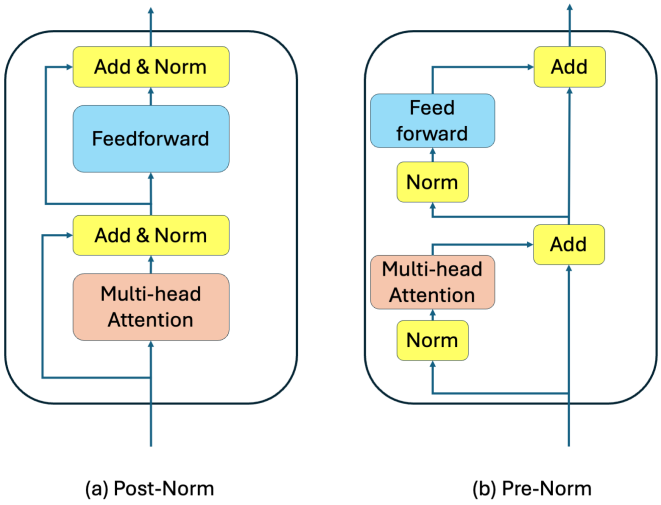


Figure 3.3: Post-norm (a), and Pre-norm (b). Different placement of the normalization layer in the Transformer encoder block

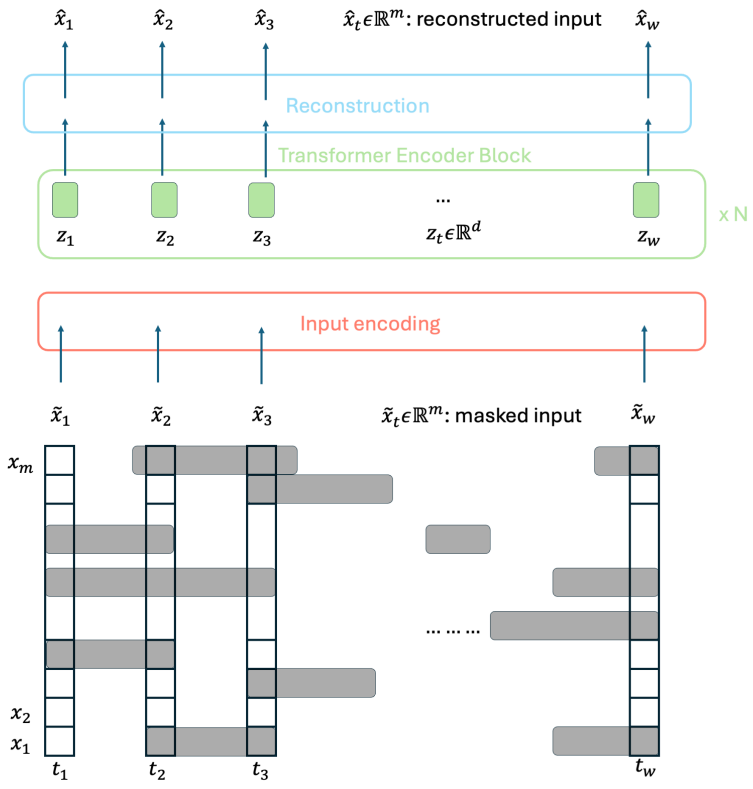


Figure 3.4: Setup of MLM unsupervised pretraining

value. The masked input is then created through an element-wise multiplication: $\tilde{X} = X \odot M$. The noise mask is generated independently for each feature, masking on average a proportion r of each feature dimension over length w. The generation of masked segments for each feature follows a geometric distribution with a mean length l_m, leaving unmasked segments with mean length $l_u = \frac{1-r}{r} l_m$. Mean masked length $l_m = 3$, and masking ratio $r = 0.10$ is chosen for all the experiments in this work, slightly differ from the settings in mvts where $r = 0.15$ [33]. The benefit of generating the noise mask with geometric distribution, rather than a Bernoulli distribution for each time step, avoids the problem of easily predictable short masked sequences. A good approximation could be achieved predicting short masked sequences by replicating surrounding values. In order to obtain long masked sequences with a Bernoulli distribution, a high r is required, making the task excessively difficult. According to the findings in mvts [33], using a geometric distribution for the masking scheme proves more effective for denoising and encourages the model to attend both to preceding and succeeding segments, as well as to contemporary values of other variables in the series, thereby enhancing its ability to model inter-dependencies between variables.

A linear reconstruction layer with parameters $W_o \in \mathbb{R}^{m \times d}$, $b_o \in \mathbb{R}^d$ is used to obtain the estimation $\hat{x}_t$ of input x_t at each time step from the representation $z_t \in \mathbb{R}^d$ in the model dimension space. However, only predictions for the masked values (with indices in the set $M_{(t,i)} \equiv \{(t,i) : m_{t,i} = 0\}$, where $m_{t,i}$ are elements of the generated binary mask) are considered in the MSE loss calculation:

$$\hat{x_t} = W_o z_t + b_o \tag{3.2}$$

$$\mathcal{L}_{MSE} = \frac{1}{|M|} + \sum_{(t,i) \in M} (\hat{x}_{(t,i)} - x_{(t,i)})^2 \tag{3.3}$$

3.3. Input Projection & Reconstruction

The model architecture described in Section 3.1 initially employed a linear projection to project each sample $X \in \mathbb{R}^{w \times m}$ (with sequence length w and m feature dimensions) onto a d-dimensional space. Correspondingly, during the MLM pretraining, a linear reconstruction layer was used to project each sample back onto the m-dimensional space.

However, empirical evidence presented in Section 4.3 suggests that this combination of linear projection and reconstruction layers fails to effectively capture the underlying data patterns during pretraining. Instead, it tends to learn only the mean differences among data points within each sample.

To address this limitation, we propose replacing the linear projection layer with a 1D convolution layer. This change allows for the extraction of more meaningful representations of low-dimensional features. For reconstruction, we implement a 1D transposed convolution operator followed by a linear layer. This approach better preserves the spatial relationships in the data. A clear comparison of the reconstructed data is presented in Section 4.3.

However, a consequence of using 1D convolution is the alteration of sequence length. Moreover, the 1D transposed convolution with identical configurations does not always reconstruct samples to their original sequence length. The sequence lengths after convolution (L_{conv}) and transposed convolution (L_{tconv}) are governed by the following equations:

$$L_{\text{conv}} = \left\lfloor \frac{L_{\text{in}} + 2P - D(K-1) - 1}{S} + 1 \right\rfloor \tag{3.4}$$

$$L_{\text{tconv}} = (L_{\text{conv}} - 1)S - 2P + D(K-1) + 1 \tag{3.5}$$

Where L_{in} is the input sequence length, P is padding, D is dilation, K is kernel size, and S is stride. To ensure consistency between the reconstructed and input sequence lengths, we incorporate an additional linear layer following the transposed convolution. This revised architecture leverages the strengths of convolutional operations while maintaining the required dimensionality during the MLM pretraining task.

3.4. Classification

The model architecture presented in Section 3.1 and illustrated in Figure 3.1 can be adapted for classification tasks by replacing the final reconstruction layer with a classification linear layer. This modification allows the model to leverage the pretrained representations for downstream classification tasks.

In this configuration, the final feature vector $z_t \in \mathbb{R}^d$ corresponding to the sequence length l ($l = w$ if the input projected linearly, else calculated with Equation 3.4) is flattened into a single vector $\overline{z} \in \mathbb{R}^{l \times d}$. This flattened vector then serves as input to the linear classification layer with parameters $W_o \in \mathbb{R}^{n \times (d \times l)}$ and $b_o \in \mathbb{R}^n$, where n represents the number of classes:

$$\hat{y} = W_o \overline{z} + b_o \tag{3.6}$$

The predictions $\hat{y}$ are subsequently passed through a softmax function to obtain the probability distribution over classes. The cross-entropy between this distribution and the ground truth labels constitutes the sample loss for optimization.

In the mvts framework [33], it is possible to freeze all pretrained weights except for the classification layer. However, empirical studies presented in [33] demonstrate a trade-off between speed and performance. Specifically, freezing weights results in faster training speed but comes with lower performance, while fine-tuning all weights leads to improved performance at the cost of increased training time.

Considering this trade-off, our work adopts the approach of allowing all weights to be trained in both the pretraining and fine-tuning (classification) tasks. As the goal of our work prioritizes model performance over computational requirements.

4

Results

4.1. Datasets and Signal Preprocessing

This study utilizes two datasets for empirical evaluation: DesktopActivity [16] and ReadingActivity [15]. The DesktopActivity dataset consists of data collected from eight subjects at a sampling rate of 30 Hz while performing six common desktop activities: browsing, playing, reading, searching, watching, and writing. The ReadingActivity dataset, also sampled at 30 Hz, contains data from nine subjects engaged in reading six different document types: magazines, manga, newspapers, novels, scientific papers, and textbooks.

As the original works associated with these datasets did not fully specify their preprocessing methodologies, we detail our approach here. For both datasets, we employ a two-step preprocessing method. First, all data are standardized followed by normalization on a subject-activity-specific basis. This approach enhances robustness against sensor drift over time and variations in hardware calibration across subjects. Second, the preprocessed data is segmented into samples using a sliding window. Window sizes vary from 30 to 60 seconds, with a consistent overlap of 80% between adjacent windows. This preprocessing pipeline ensures data consistency and facilitates subsequent analysis and model training.

4.2. Experimental Design

Our experimental methodology comprises two distinct phases: hyperparameter optimization and model evaluation. This approach enables a comprehensive assessment of our proposed techniques while simulating realistic application scenarios.

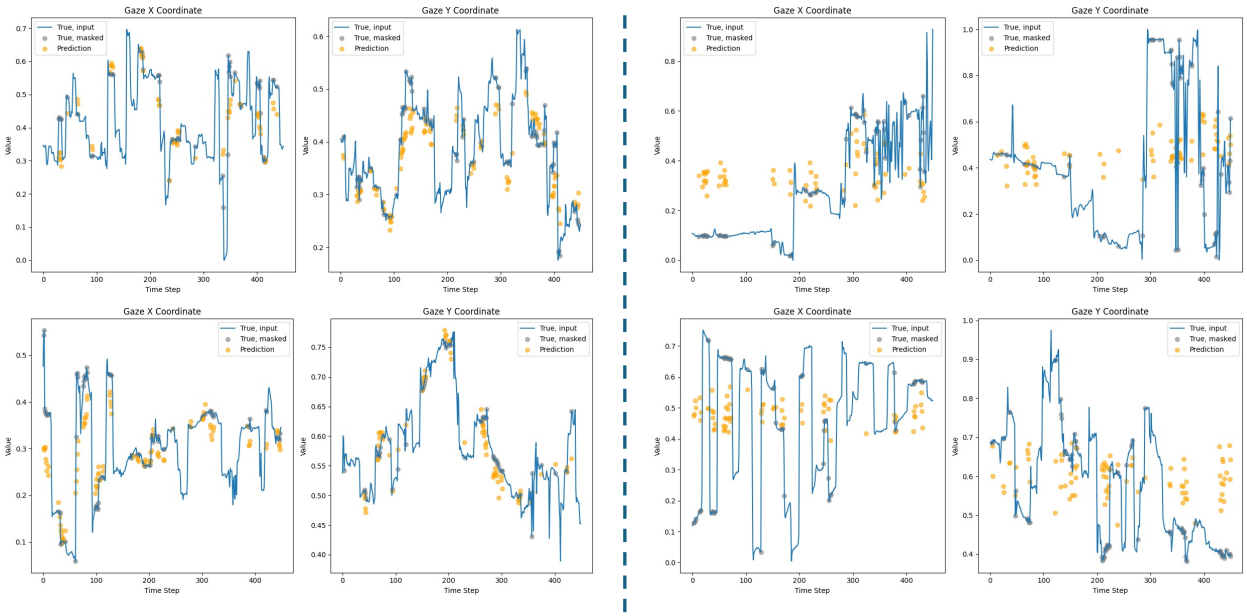
For hyperparameter optimization, we employ a fully supervised, leave-one-subject-out cross-validation strategy. During this phase, we omit the MLM pretraining to focus on the core model architecture. To mitigate overfitting, we always implement a random split of 80% training and 20% validation data within each fold of the cross-validation.

Following hyperparameter optimization, we train and evaluate two model variants weighted f1-score indicating respective performance:

MLM-pretrained model: This model undergoes unsupervised MLM pretraining using all available data, as this phase does not require labels. Subsequently, it is fine-tuned in a supervised manner using a subject-dependent strategy. The fine-tuning data consists of the initial X% of samples from the subject for fine-tuning, along with Y% of the labels from other subjects available during the MLM pretraining phase, where $X \in [0.1, 0.3]$ and $Y \in [0.0, 1.0]$.

Fully supervised model: This variant is trained solely on the labeled data available during the fine-tuning phase of the MLM-pretrained model. To compensate for the lack of pretraining, this model is trained for a greater number of epochs.

This evaluation approach is designed to align with potential real-world applications. In practice, personalized applications often require individual user data for specific activities. The fine-tuning data used in our experiments simulates this scenario by utilizing a continuous segment of personal data collected over a limited time period. This approach aims to recognize similar behaviors committed subsequently, mirroring the operational requirements of practical applications.



(a) Convolution Reconstructing Layer (b) Linear Reconstructing Layer

Figure 4.1: Comparison of reconstruction methods: (a) Convolution layer reconstruction, demonstrating improved accuracy in following the original data trend. (b) Linear layer reconstruction, showing a tendency to converge towards the mean value for masked continuous data points.

This comparative approach allows us to assess the impact of MLM pretraining on model performance under various data availability scenarios. For a comprehensive list of specific hyperparameters used in these experiments, please refer to Table 4.1.

Parameter	Value
activation	gelu
dropout	0.1
unsupervised learning rate	0.0001
supervised learning rate	0.001
batch size	64
warmup/total epochs ratio	1/9

Table 4.1: Hyperparameters used in the experiments

4.3. Input Projection & Reconstruction Comparison

Figure 4.1 presents a comparison of reconstruction results using different layer types in the MLM pretrain task. Panel (a) shows the superior reconstruction achieved by the convolution layer, which closely follows the original data trend. In contrast, panel (b) illustrates the results from a linear layer, where predictions for masked continuous data points tend to converge towards the mean value.

This behavior is similar to the problem of minimizing MSE between a constant value and a variable. While predicting the mean value of masked points can indeed minimize the MSE loss, it does not necessarily provide a good reconstruction of the original data points. The MSE loss for masked values is given by:

$$\mathcal{L}_{MSE} = \frac{1}{|M|} \sum_{(t,i) \in M} (\hat{x}_{(t,i)} - x_{(t,i)})^2 \tag{4.1}$$

where M is the set of masked indices, $\hat{x}_{(t,i)}$ is the predicted value, and $x_{(t,i)}$ is the true value.

It can be proven that if $\hat{x}_{(t,i)}$ is predicted as a constant value, the mean value of all masked points minimizes this loss. Let's define the mean of the masked values:

$$\bar{x} = \frac{1}{|M|} \sum_{(t,i) \in M} x_{(t,i)} \tag{4.2}$$

To prove that using $\bar{x}$ as the prediction for all masked values minimizes the MSE loss, we consider a general prediction $\hat{x}$ and show that the MSE is minimized when $\hat{x} = \bar{x}$. The MSE loss is:

$$\mathcal{L}_{MSE}(\hat{x}) = \frac{1}{|M|} \sum_{(t,i) \in M} (\hat{x} - x_{(t,i)})^2 \tag{4.3}$$

To find the minimum, we differentiate with respect to $\hat{x}$ and set it to zero:

$$\frac{\partial \mathcal{L}_{MSE}}{\partial \hat{x}} = \frac{2}{|M|} \sum_{(t,i) \in M} (\hat{x} - x_{(t,i)}) = 0 \tag{4.4}$$

Solving this equation yields:

$$\hat{x} = \frac{1}{|M|} \sum_{(t,i) \in M} x_{(t,i)} = \bar{x} \tag{4.5}$$

This proves that if $\hat{x}_{(t,i)}$ could only be predicted as a constant value, the mean value $\bar{x}$ minimizes the MSE loss. To confirm it's a minimum (not a maximum), we check the second derivative:

$$\frac{\partial^2 \mathcal{L}_{MSE}}{\partial \hat{x}^2} = \frac{2}{|M|} \sum_{(t,i) \in M} 1 = 2 > 0 \tag{4.6}$$

Since the second derivative is positive, this confirms that $\bar{x}$ gives the global minimum of the MSE loss for this constrained problem.

However, it is important to note that this is not the actual scenario in our model. The linear reconstruction layer with parameters $W_o \in \mathbb{R}^{m \times d}$ and $b_o \in \mathbb{R}^d$ can deliver predictions that vary among different time steps:

$$\hat{x}_t = W_o z_t + b_o \tag{4.7}$$

Furthermore, even if constant prediction as the mean of several masked data points could minimize the MSE loss under some circumstances, it fails to capture the temporal dynamics and local patterns present in the data, which are crucial for accurate reconstruction.

The reason for this issue is not immediately apparent, as the linear layer should theoretically be capable of reconstructing whatever the 1D convolution and transpose convolution could produce. One possible explanation is that convolution and transposed convolutions inherently account for more locality in the data. This property is particularly beneficial for eye gaze data, which consists of rich features that exist locally, such as saccades and fixations [15, 16].

In our work, we have found that using convolution and transposed convolution layers as the projection and reconstruction layers yields better results compared to using linear layers. This improvement is observed not only in reconstruction quality but also in computational efficiency. Specifically, training one epoch with linear layers takes approximately 12.84 seconds, while the convolution layers reduce this time to just 2.33 seconds, which is an increase of x5.5 in speed[1].

4.4. Experimental Results

We present the results of our experiments for both DesktopActivity and ReadingActivity datasets, focusing on a window size of 30 seconds. The results for the 60-second window will be included in the appendix. These results demonstrate the performance of our MLM-pretrained model compared to the fully supervised model across various fine-tuning scenarios.

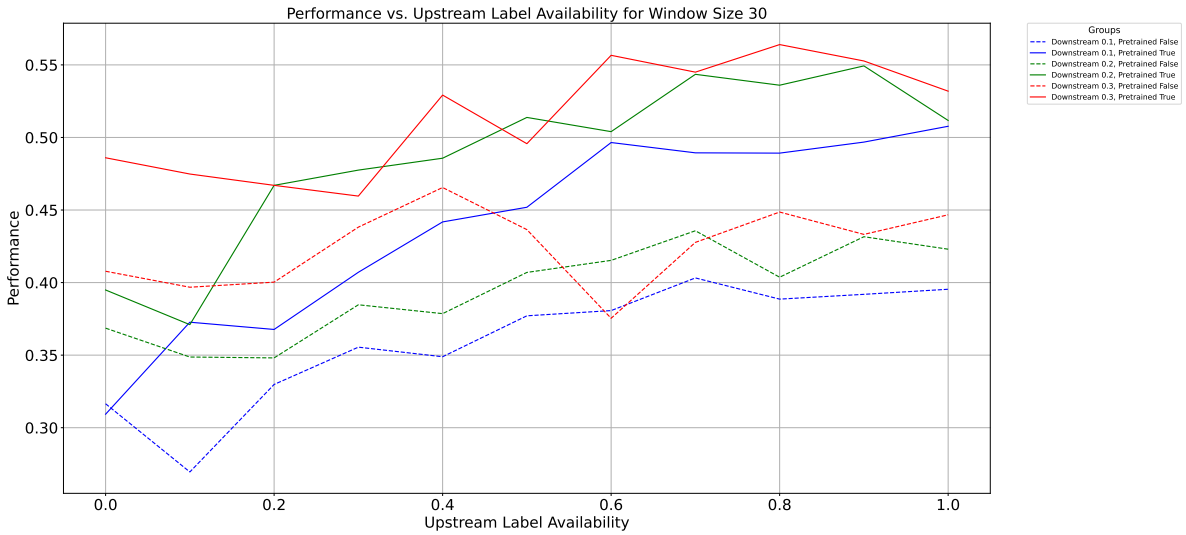


Figure 4.2: DesktopActivity: Performance comparison between MLM-pretrained and fully supervised models. Solid lines represent the performance with pretraining, and dotted lines represent performance without pretraining.

4.4.1. DesktopActivity Results

Figure 4.2 illustrates the performance comparison for the DesktopActivity dataset.

As observed in Figure 4.2, the MLM-pretrained model consistently outperforms the fully supervised model, with one exception: when upstream label availability is 10% and downstream label availability is 10%. We have also noticed two patterns: (1) increasing the percentage of data from other subjects during fine-tuning generally improves performance, and (2) the more data available from other subjects during pretraining also contributes to an increase in performance. However, these patterns observed in the 30-second window results do not necessarily apply when the window size is increased to 60 seconds. These differences will be discussed in Chapter 5.

4.4.2. ReadingActivity Results

Figure 4.3 presents the performance comparison for the ReadingActivity dataset.

The results for ReadingActivity, as shown in Figure 4.3, reveal some interesting patterns. Similar to the DesktopActivity results, the MLM-pretrained model generally outperforms the fully supervised model, with a few exceptions. For the 30-second window (Figure 4.3), we observe that the performance improvement from incorporating more data from other subjects is less pronounced compared to the DesktopActivity results. This suggests that reading activities might have more individual-specific patterns that are less transferable across subjects. This might also explain the decrease in performance when downstream available data is 20% and 30%, as the upstream data availability increases. The reason for this phenomenon remains unknown and requires further investigation.

In both datasets and across both window sizes, we observe that the MLM-pretrained model's performance is more stable and generally superior, highlighting the effectiveness of our proposed approach in leveraging unlabeled data and enhancing model generalization.

[1] timing measurements were obtained by pretraining the same model with samples of 15-second window size on a laptop with an Intel i7-11800H CPU and an NVIDIA RTX3060 Mobile GPU.

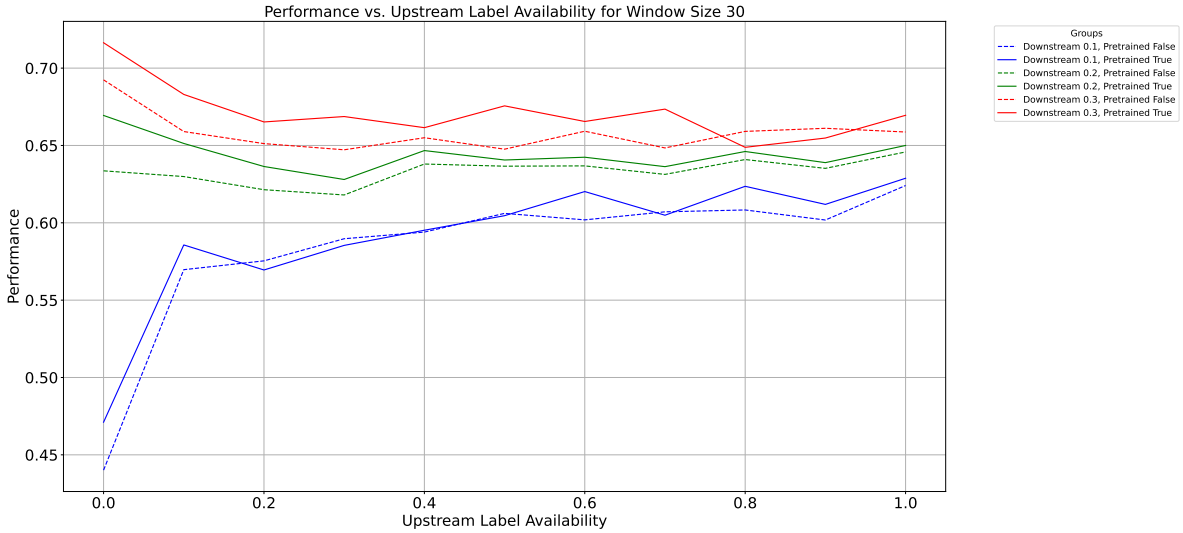


Figure 4.3: ReadingActivity: Performance comparison between MLM-pretrained and fully supervised models. Solid lines represent the performance with pretraining, and dotted lines represent performance without pretraining.

5

Conclusion & Recommendations

5.1. Conclusion

Our study on MLM pretraining for eye gaze data analysis has yielded several important findings:
MLM Pretraining Effectiveness: In general, the MLM-pretrained model performs better than the fully supervised model for both DesktopActivity and ReadingActivity datasets. This suggests that MLM pretraining is effective in learning generalizable features from eye gaze data.
Data Availability Impact: Increasing the percentage of data from other subjects during fine-tuning generally improved model performance. This indicates that incorporating diverse data sources can enhance the model's ability to generalize across subjects.

These findings demonstrate the potential of MLM pretraining in improving eye gaze data analysis. However, it is worth noting that none of these performances have reached the level claimed in other studies where hand-crafted features were utilized to perform the same classification tasks. This gap highlights the need for further research and refinement of our approach.

5.2. Recommendations for Future Research

Based on our results and the limitations identified in this study, we propose the following suggestions for further investigation:

5.2.1. Further Investigation of Reconstruction Phenomena

While our current study has yielded significant insights into the behavior of the MLM pretraining process for eye gaze data, several avenues for future research have emerged. We propose the following investigations to further enhance our understanding and improve the model's performance:
Extended Masking Experiments: Future work should explore the impact of varying mask lengths on reconstruction quality. Our preliminary observations suggest that shorter masks lead to better reconstructions, while longer masks tend to result in mean-value predictions. A research of this phenomenon could provide valuable insights into optimal masking strategies for eye gaze data.
Feature Analysis: An in-depth analysis of which features are most accurately reconstructed versus those that default to mean-value predictions could offer crucial insights. This investigation may reveal which aspects of eye gaze data are more predictable from context, potentially informing both model design and our understanding of eye movement patterns.
Reconstruction Techniques: Given our findings that the linear reconstruction layer, while theoretically capable of complex reconstructions, tends to favor simpler mean-value predictions for longer masked sequences, future work should explore more sophisticated reconstruction methods. Potential approaches include developing more advanced decoder architectures tailored to the specific characteristics of eye gaze data, incorporating tasks that explicitly encourage the preservation of temporal patterns in the reconstructed data, and investigating alternative loss functions or additional constraints that penalize mean-value predictions and reward reconstructions that better preserve the temporal dynamics of the original data.

These proposed investigations aim to address the current limitations in our model's reconstruction capabilities, particularly for longer masked sequences. By pursuing these research directions, we hope

to develop more accurate and robust models for eye gaze data analysis, ultimately leading to improved performance in downstream tasks and a deeper understanding of eye movement patterns. Based on engineering so far, where to go next.

5.2.2. Model Architecture Optimization

The observed differences between 30-second (Section 4) and 60-second (Section 6) window results raises requirements for further studies. While the 30-second window shows results that follow our expectations, the 60-second window not only deviates from expectations but also shows significantly poorer performance. This unexpected outcome raises questions about the suitability of our current model architecture for longer sequences. Future research should aim to investigate the causes of performance decrease in longer window sizes and explore modifications to the Transformer architecture to better handle longer temporal dependencies in eye gaze data. Additionally, consideration should be given to alternative architectures that might be more suitable for capturing long-range patterns in time series data. Simpler architectures such as temporal convolutional networks may deliver better performances.

5.2.3. Cross-Activity Transfer Learning

Given the differences observed between DesktopActivity and ReadingActivity, exploring the transferability of pretrained models across different types of activities could lead to more general and robust models for eye gaze analysis. However, before diving deep into transferring learned features, we suggest conducting a thorough analysis of the current model's decision-making process. This should include investigating which features the model extracts and relies on for classification, as well as interpreting what aspects of the eye gaze data are most influential in the model's predictions. This deeper understanding of the model's behavior will inform more effective transfer learning strategies and potentially reveal insights into the fundamental differences between eye movement patterns in various activities.

5.2.4. Bridging the Gap with Hand-Crafted Features

Given that our current approach has not yet matched the performance of methods using hand-crafted features, future work should focus on analyzing successful hand-crafted features to understand what information they capture. This analysis could provide inspiration for designing a new type of deep learning architecture. The goal would be to develop a neural network architecture specifically dedicated to eye gaze analysis, potentially incorporating insights from traditional feature engineering approaches. By combining the strengths of hand-crafted features with the flexibility and power of deep learning, we may be able to create more effective and specialized models for eye gaze data processing and classification.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL].

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. *Is Space-Time Attention All You Need for Video Understanding?* 2021. arXiv: 2102.05095 [cs.CV].

[3] Valentina Bianchi et al. "IoT Wearable Sensor and Deep Learning: An Integrated Approach for Personalized Human Activity Recognition in a Smart Home Environment". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 8553–8562. DOI: 10.1109/JIOT.2019.2920283.

[4] Andreas Bulling et al. "Eye movement analysis for activity recognition". In: *Proceedings of the 11th International Conference on Ubiquitous Computing*. UbiComp '09. Orlando, Florida, USA: Association for Computing Machinery, 2009, pp. 41–50. ISBN: 9781605584317. DOI: 10.1145/1620545.1620552. URL: https://doi.org/10.1145/1620545.1620552.

[5] Nicolas Carion et al. *End-to-End Object Detection with Transformers*. 2020. arXiv: 2005.12872 [cs.CV].

[6] Angus Dempster, François Petitjean, and Geoffrey I. Webb. "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels". In: *Data Mining and Knowledge Discovery* 34.5 (July 2020), pp. 1454–1495. ISSN: 1573-756X. DOI: 10.1007/s10618-020-00701-z. URL: http://dx.doi.org/10.1007/s10618-020-00701-z.

[7] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[8] Raymond Dodge. "Visual perception during eye movement." In: *Psychological Review* 7.5 (1900), p. 454.

[9] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[10] Mary Hayhoe. "Vision using routines: A functional account of vision". In: *Visual Cognition* 7.1-3 (2000). Cited by: 224; All Open Access, Green Open Access, pp. 43–64. DOI: 10.1080/135062800394676. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0034041961&doi=10.1080%2f135062800394676&partnerID=40&md5=99411a49abd7089eddb70f57511f2aa0.

[11] Ruining He et al. *RealFormer: Transformer Likes Residual Attention*. 2021. arXiv: 2012.11747 [cs.LG]. URL: https://arxiv.org/abs/2012.11747.

[12] Alexander Hoelzemann et al. "Hang-Time HAR: A Benchmark Dataset for Basketball Activity Recognition Using Wrist-Worn Inertial Sensors". In: *Sensors* 23.13 (2023). ISSN: 1424-8220. DOI: 10.3390/s23135879. URL: https://www.mdpi.com/1424-8220/23/13/5879.

[13] Shian-Ru Ke et al. "A Review on Video-Based Human Activity Recognition". In: *Computers* 2.2 (2013), pp. 88–131. ISSN: 2073-431X. DOI: 10.3390/computers2020088. URL: https://www.mdpi.com/2073-431X/2/2/88.

[14] Wonjae Kim, Bokyung Son, and Ildoo Kim. *ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision*. 2021. arXiv: 2102.03334 [stat.ML].

[15] Kai Kunze et al. "I know what you are reading: recognition of document types using mobile eye tracking". In: *Proceedings of the 2013 International Symposium on Wearable Computers*. ISWC '13. Zurich, Switzerland: Association for Computing Machinery, 2013, pp. 113–116. ISBN: 9781450321273. DOI: 10.1145/2493988.2494354. URL: https://doi.org/10.1145/2493988.2494354.

[16] Guohao Lan et al. “GazeGraph: graph-based few-shot cognitive context sensing from human visual behavior”. In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. SenSys ’20. Virtual Event, Japan: Association for Computing Machinery, 2020, pp. 422–435. ISBN: 9781450375900. DOI: 10.1145/3384419.3430774. URL: https://doi.org/10.1145/3384419.3430774.

[17] Michael Land, Neil Mennie, and Jennifer Rusted. “The Roles of Vision and Eye Movements in the Control of Activities of Daily Living”. In: *Perception* 28.11 (1999). PMID: 10755142, pp. 1311–1328. DOI: 10.1068/p2935. eprint: https://doi.org/10.1068/p2935. URL: https://doi.org/10.1068/p2935.

[18] Michael F. Land and Mary Hayhoe. “In what ways do eye movements contribute to everyday activities?” In: *Vision Research* 41.25 (2001), pp. 3559–3565. ISSN: 0042-6989. DOI: https://doi.org/10.1016/S0042-6989(01)00102-X. URL: https://www.sciencedirect.com/science/article/pii/S004269890100102X.

[19] Shiyang Li et al. *Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting*. 2020. arXiv: 1907.00235 [cs.LG].

[20] Rex Liu et al. “An Overview of Human Activity Recognition Using Wearable Sensors: Healthcare and Artificial Intelligence”. In: *Internet of Things – ICIOT 2021*. Ed. by Bedir Tekinerdogan, Yingwei Wang, and Liang-Jie Zhang. Cham: Springer International Publishing, 2022, pp. 1–14. ISBN: 978-3-030-96068-1.

[21] Jiawei Ma et al. *CDSA: Cross-Dimensional Self-Attention for Multivariate, Geo-tagged Time Series Imputation*. 2019. arXiv: 1905.09904 [cs.LG].

[22] Johannes Meyer et al. “A CNN-based Human Activity Recognition System Combining a Laser Feedback Interferometry Eye Movement Sensor and an IMU for Context-aware Smart Glasses”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5.4 (Dec. 2022). DOI: 10.1145/3494998. URL: https://doi.org/10.1145/3494998.

[23] Matthew Middlehurst et al. *HIVE-COTE 2.0: a new meta ensemble for time series classification*. 2021. arXiv: 2104.07551 [cs.LG].

[24] Keith Rayner. “Eye movements in reading and information processing: 20 years of research.” In: *Psychological bulletin* 124.3 (1998), p. 372.

[25] Oliver Richter and Roger Wattenhofer. *Normalized Attention Without Probability Cage*. 2020. arXiv: 2005.09561 [cs.LG]. URL: https://arxiv.org/abs/2005.09561.

[26] Eugene Shen. “An analysis of eye movements in the reading of Chinese.” In: *Journal of experimental psychology* 10.2 (1927), p. 158.

[27] Ahmed Shifaz et al. “TS-CHIEF: a scalable and accurate forest algorithm for time series classification”. In: *Data Mining and Knowledge Discovery* 34.3 (Mar. 2020), pp. 742–775. ISSN: 1573-756X. DOI: 10.1007/s10618-020-00679-8. URL: http://dx.doi.org/10.1007/s10618-020-00679-8.

[28] Julian Steil and Andreas Bulling. “Discovery of everyday human activities from long-term visual behaviour using topic models”. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp ’15. Osaka, Japan: Association for Computing Machinery, 2015, pp. 75–85. ISBN: 9781450335744. DOI: 10.1145/2750858.2807520. URL: https://doi.org/10.1145/2750858.2807520.

[29] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[30] Jindong Wang et al. “Deep learning for sensor-based activity recognition: A survey”. In: *Pattern Recognition Letters* 119 (Mar. 2019), pp. 3–11. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2018.02.010. URL: http://dx.doi.org/10.1016/j.patrec.2018.02.010.

[31] Ruibin Xiong et al. *On Layer Normalization in the Transformer Architecture*. 2020. arXiv: 2002.04745 [cs.LG]. URL: https://arxiv.org/abs/2002.04745.

[32] Huatao Xu et al. “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, pp. 220–233. ISBN: 9781450390972. DOI: 10.1145/3485730.3485937. URL: https://doi.org/10.1145/3485730.3485937.

[33] George Zerveas et al. *A Transformer-based Framework for Multivariate Time Series Representation Learning*. 2020. arXiv: 2010.02803 [cs.LG].

[34] Liye Zou et al. “Look into my eyes: What can eye-based measures tell us about the relationship between physical activity and cognitive performance?” In: *Journal of Sport and Health Science* 12.5 (2023), pp. 568–591. ISSN: 2095-2546. DOI: https://doi.org/10.1016/j.jshs.2023.04.003. URL: https://www.sciencedirect.com/science/article/pii/S209525462300042X.

6

Table & Figures

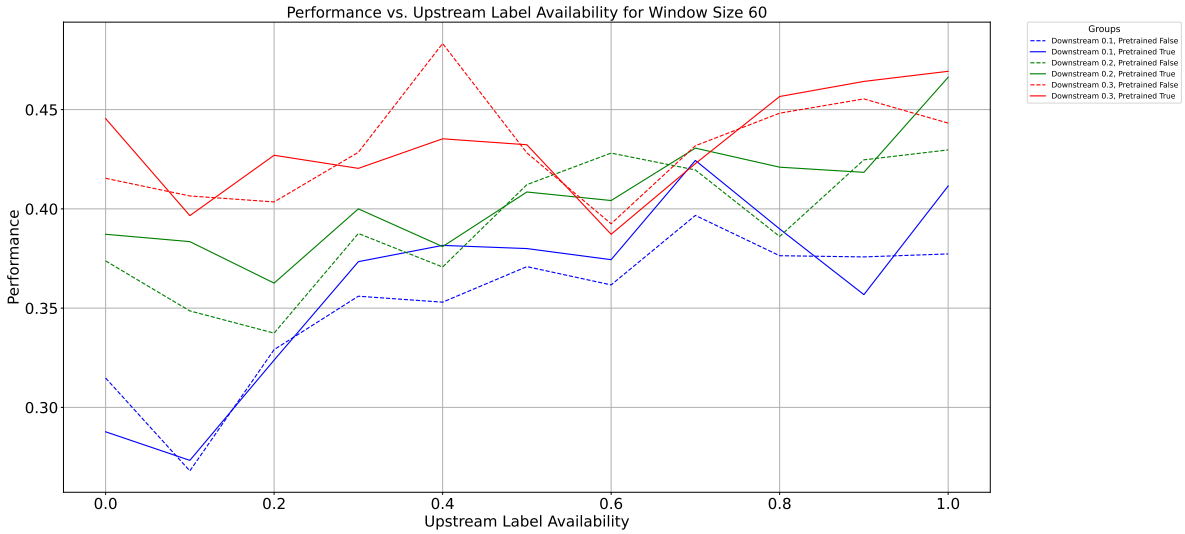


Figure 6.1: ReadingActivity (60sec window): Performance comparison between MLM-pretrained and fully supervised models. Solid lines represent the performance with pretraining, and dotted lines represent performance without pretraining.

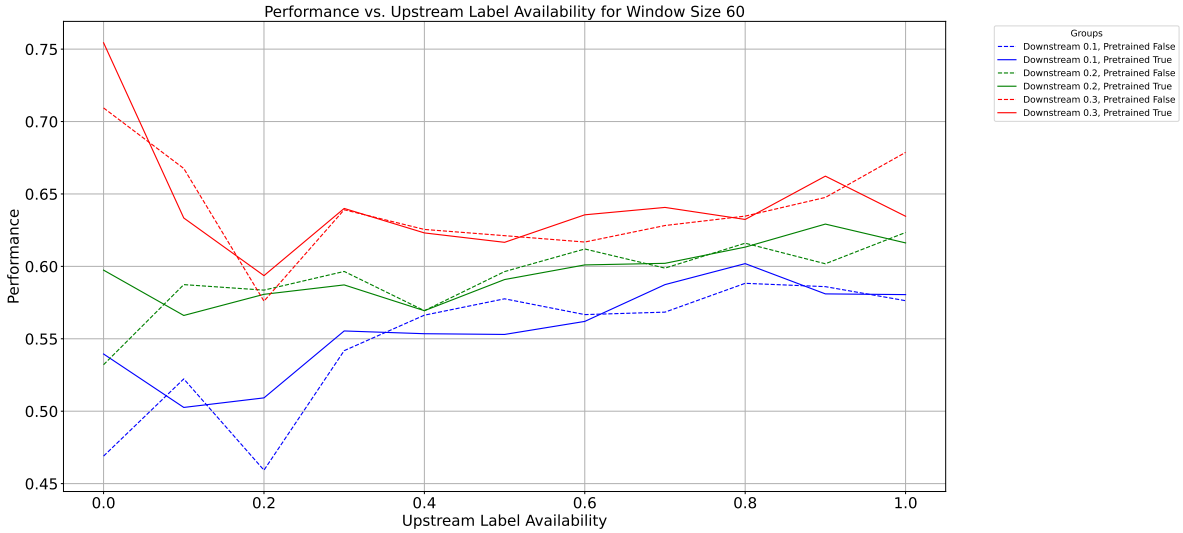


Figure 6.2: ReadingActivity (60sec window): Performance comparison between MLM-pretrained and fully supervised models. Solid lines represent the performance with pretraining, and dotted lines represent performance without pretraining.