
# Explainable Spatio-Temporal Video Transformer with Attention Supervision to Predict Left Ventricular Ejection Fraction in Echocardiography

**Student: Jeffin Siby**

**Supervisor: Qingjie Meng**

**Inspector: Wendy Yanez Pazmino**

**Academic Year: 2023-2024**

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

2023-24

# Abstract

Left Ventricular Ejection Fraction (LVEF) is calculated by clinicians to identify cardiac dysfunction, which is a leading cause of hospitalisation in the US [27]. However, making accurate estimations requires accurately locating the frames in which the heart is fully contracted and fully expanded from a video that possibly contains multiple heartbeat cycles. Then the volume of the left ventricle in both frames must be calculated. The low contrast of echocardiography makes this particularly challenging and as a result, high inter-observer variances ranging from 7.6% to 13.9% [27]have been reported. The challenging nature has given rise to the need for accurate automatic methods. In this paper we a propose novel loss specific for LVEF estimation, and show that GridMask data augmentation can improve results for this task. Using our proposed changes, we trained a spatio-temporal transformer which achieved an $R^2$ score of 0.72 on the EchoNet-Dynamic dataset and demonstrated it's superior performance to the baseline model that was trained without such changes. We also show that both the spatial and temporal information that guides the model to make a particular prediction can be explained via explainability methods. In particular, we provide a quantitative analysis of state-of-the-art gradient visual explainability methods and conclude that Eigen-CAM [24] provides the best activation maps to explain which parts of the input frames our model model focuses on. Further, we demonstrate that it is possible to identify poor predictions based on the activation maps generated.

# Acknowledgements

I would like to extend my heartfelt gratitude to my supervisor, Dr. Qingjie Meng, and to Hadrien Reynaud for their invaluable guidance and support throughout this project. Their knowledge and insights have been a tremendous help in guiding me in the right direction, enabling me to complete this work to the best of my ability.

# Abbreviations

| | |
|---|---|
| EF | Ejection Fraction |
| LV | Left Ventricle |
| LVEF | Left Ventricular Ejection Fraction |
| ES | End-systolic |
| ED | End-diastolic |
| ViT | Vision Transformer |
| MAE | Mean Absolute Error |
| RMSE | Root Mean Squared Error |
| mAP | Mean Average Precision |
| mIOU | Mean Intersection Over Union |
| GM | GridMask |
| DGT Loss | Distance-Guided Temporal Loss |
| SOTA | State-of-the-art |

# Contents

# List of Figures

# List of Tables

# CHAPTER 1

# Introduction

## 1.1 Aims and Objectives

Abnormalities in cardiac function can result in various issues such as "dyspnea, fatigue, exercise intolerance, fluid retention and greater risk of mortality" [27] and is a growing global health issue as noted by Ouyang et al. [27].

One of the most important indicators used to predict heart failure is the Left Ventricular Ejection Fraction (LVEF). It can be measured as the difference between the left ventricle (LV) volume at the end-diastolic (ED) and end-systolic (ES) frames divided by the ED volume, estimated using the apical four-chamber (A4C) and apical two-chamber (A2C) views. Echocardiography is the most used modality as it's cheap, ionising radiation-free and portable unlike other modalities such as MRI and it provides real-time visualisation [25]. However, the poor contrast of the modality makes accurate measurements difficult and requires expert operators with years of experience. Not only is this time-consuming, but an inter-observer variability of 7.6%-13.9% has been reported in clinical EF estimates [27], thus highlighting the need for accurate automatic methods.

We observe that most methods either ignore data augmentation or have reported limited success with standard augmentation techniques, despite it being a crucial step in ML frameworks for making the model robust and generalisable.

The decision making process of large neural networks is often seen as a black-box, it is often critical to understand the underlying assumptions a model makes or the features that are most relevant to its prediction, particularly in the medical field where verification from an expert clinician may be required. As a result, several explainability methods have been proposed. Although there is an emerging trend to provide model explainability in the field of deep learning for LVEF estimation, the explainability methods used in this domain are not state-of-the-art (SOTA) techniques.

In this study, we used a deep learning approach adapted from a successful method [22] for LVEF calculation from echocardiography. The model was chosen due to its focus on providing explainability and promising results. Since

explainability was a key aim of our study, we quantitatively compared the results of various SOTA explainability methods applied to our proposed model to highlight the need to explore more sophisticated methods and to provide guidance for future studies on which explainability method to select. To the best of our knowledge, most of the explainability methods we evaluated including those that resulted in the best performance have not been used in LVEF estimation papers and a quantitative analysis of the different methods has also not been performed in LVEF estimation papers.

We also conducted various experiments such as the use of a novel data augmentation technique not used before for LVEF estimation, and attention supervision via a novel attention loss function specific for LVEF estimation. Their effects have been reported to guide future research in this domain.

This project answers the following sub-questions:

- Can data augmentations be made to improve the performance of video transformers?
- Can attention supervision be performed to guide the model to focus more on the ES and ED frames which are required to predict LVEF and in turn improve model performance?
- Can explainability methods show that the model focuses on clinically relevant features in the input?
- How can explainability methods for LVEF estimation be evaluated?
- Which explainability methods provide the best results?
- Can poor performance be explained via explainability techniques?

to answer the broader research question: Can Data Augmentation, Attention Supervision, and Explainability Improve Spatio-Temporal Video Transformers for Predicting Left Ventricular Ejection Fraction in Echocardiography?

Our contributions are as follows:

- We show that explainability methods can highlight areas of the input that are most relevant for the model to make predictions, and show that the activation map can be inspected to identify cases where the model may make errors.
- We provide suggestions for the best explainability techniques by performing a quantitative analysis of SOTA explainability methods on our model, most of which have not been previously used in LVEF studies.
- We demonstrate that our novel loss function applied through attention supervision can help the model focus on the ES and ED frames and in turn improve performance.
- We demonstrate that GridMask data augmentation, which to the best of our knowledge has not been used previously in LVEF studies, can improve model performance and reduce the effects of overfitting.
- We apply various experimentation to inform future work using video transformers.

- We demonstrate that both the spatial and temporal modules of our model is explainable.

## 1.2 Paper Structure

The paper is structured as follows:

- **Section 2** contains the relevant background information including the method for manual LVEF calculation, vision transformers, explainability methods and the evaluation metrics used.
- **Section 3** discusses SOTA related work and identifies knowledge gaps.
- **Section 4** discusses the dataset used, model architecture, proposed experiments as well as unsuccessful experiments.
- **Section 5** sates the results of the proposed changes using the evaluation metrics mentioned in section 2.
- **Section 6** discusses and evaluates the results of the proposed changes and limitations of the project.
- **Section 7** concludes the paper with directions for future work.

# CHAPTER 2

# Background

## 2.1 Traditional methods for measuring LVEF

LVEF can be measured using various techniques and modalities that are both invasive and non-invasive. The most popular modality is the non-invasive echocardiography method on which the Modified Simpson method (biplane method of disks) is recommended to estimate LVEF by The American Society of Echocardiography [15]. Simply put, the method involves locating two orthoganal views such as the A4C and apical two-chamber (A2C) views, locating the ES and ED frames in each, splitting the LV into stacked slices, calculating and summing the volume of each to get the LV volume, and using the following formula to calculate LVEF:

$$\text{LVEF} = \left( \frac{\text{EDV} - \text{ESV}}{\text{EDV}} \right) \times 100\% \tag{2.1}$$

where EDV and ESV correspond to end-disatole volume and end-systole volumes respectively.

However, the challenges in accurately detecting the ES and ED frames; accurately calculating the volume of each slice with the poor detail of echocardiography, the possibility of the accumulation of errors with incorrect volumes calculated for each slice, the need for the availability of expert clinicians, and the time consuming nature of the task highlight the need for automatic methods.

## 2.2 Transformer Encoder

The transformer architecture was introduced by Vaswani et al. [36] which used the self-attention mechanism to learn relationships between inputs and outputs. The original transformer featured an encoder-decoder architecture consisting of multiple encoder blocks which received the entire input sequence and outputted embeddings that were fed to the autoregressive decoder blocks.

To keep the input size a fixed length, the input was tokenised after which a positional encoding was injected to capture the positional relationships.

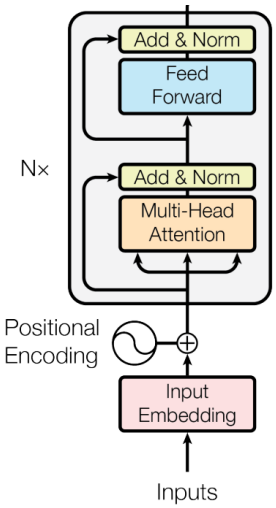


Figure 2.1: Figure from [36] showing the transformer encoder architecture and the input tokenisation process. Nx represents the encoder block is duplicated multiple times.

Each encoder block was a copy of the same architecture, connected in series, which consisted of a multi-head self-attention layer which allowed each token to capture the contextual information from the entire sequence, a feed-forward layer and residual connections followed by layer normalisation (See Figure 2.1).

## 2.3 Vision Transformers

Vision Transformers (ViTs) were introduced by Dosovitskiy et al. [10]. Inspired by the success of transformers in NLP, ViT's were introduced to work with images. More specifically, the method proposed that each image $\mathbf{x} \in R^{H \times W \times C}$ can be split into patches of resolution $P$x$P$ and the number of patches can be calculated as $N = HW/P^2$, where $H$ and $W$ are the height and width of the image and $C$ is the number of channels.

Once the image is tokenised to a fixed-sized vector, the authors proposed passing it through a transformer encoder similar to that proposed by Vaswani et al. [36]. Different to Vaswani et al.'s implementation, later papers such as Wang et al.'s [38] and Baevski et al's [3] applied layer normalisation before the self-attention and feed-forward layers as opposed to after, and reported improved results. As such, the ViT's encoder followed the latter architecture.

## 2.4 Explainability for Visual Data

With the increased complexity and depth of modern deep learning architectures, the ability to explain and understand a model's reasoning for an output is highly sought after, particularly in the medical field where model decisions may need

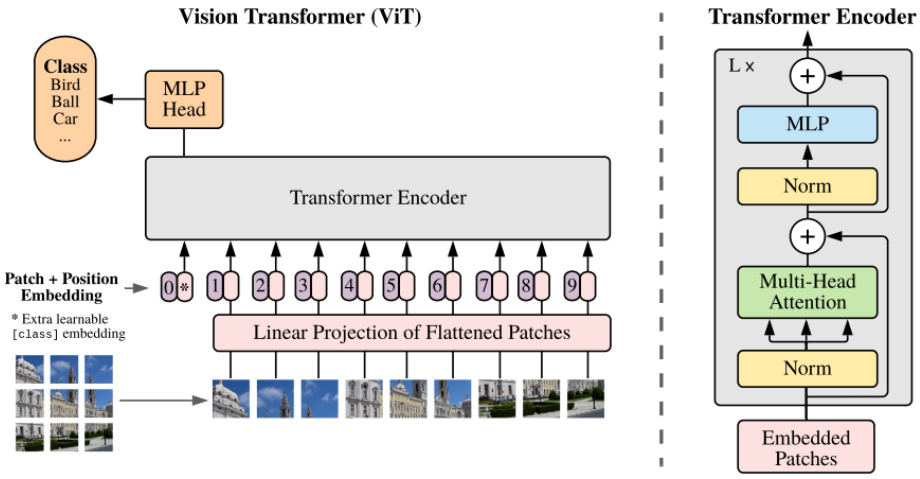


Figure 2.2: Figure from [10] showing the ViT architecture (left) and the transformer encoder architecture (right). Note that the only difference of the transformer encoder architecture from Figure 2.1 is that layer normalisation is applied before the self-attention and feed-forward layers as opposed to after.

to be verified by a medical practitioners. Human verification in this domain not only allows practitioners to verify their own diagnosis, but the transparency allows the model to be more trusted and adopted by both practitioners and patients who are the receiving the diagnosis [26]. In this paper we refer to explainability as the ability to reason why a model made a particular prediction in a human understandable manner.

For visual input data such as images, explanations for model predictions are generally achieved through visualisation techniques. Most visualisation techniques fall into two main categories [26]:

- **Visualisation by Perturbation**: methods by which the input is modified through occlusion or by adding noise and observing alterations in the model prediction. Local Interpretable Model-Agnostic Explanations (LIME) [30] is a popular method that falls into this category which works by altering data (e.g. pixels) in the original data to create a new dataset, training the black-box model on the new data, weighing the altered samples as a function of their similarity to the original data, and then fitting a simpler model (e.g. linear regression) using the weights. The idea behind this technique is that the simpler model, which learns the input and output relationship of the complex model, is easier to explain.

- **Visualisation by Gradient/Backpropagation**: methods which compute the relevance of features in the input for the model prediction using the backward pass. Most of the CAM methods discussed in this chapter fall into this category, as well as other popular methods such as LRP [2] and SHAP [20].

## 2.5 Attention Rollout

Since the success of transformer models is largely contributed by the attention mechanism, the attention scores given by a transformer model to an input is commonly used for providing a visual explanation to where the model is focusing. However for deep models with multiple layers, the attention signals can fade out as the number of layers increase. To address this issue for latter layers, Abnar et al. [1] suggested the recursive attention rollout algorithm. The attention matrix of the first layer is used as the base case, after which the attention rollout for each layer thereafter is calculated by peforming the matrix multiplication operation between the attention matrix of the current layer and the attention rollout of the previous layer as shown below [1]:

$$\widetilde{A}(l_i) = \begin{cases} A(l_i)\widetilde{A}(l_{i-1}) & \text{if } i > j \\ A(l_i) & \text{if } i = j \end{cases} \tag{2.2}$$

where $l_i$ represents layer $i$, $\widetilde{A}$ is attention rollout result of the respective layer and $A$ is the raw attention matrix of the respective layer.

## 2.6 Class Activation Map Techniques

CAM [42] was the first technique to produce class activation maps by removing the classification head of a CNN model, passing the output of the last convolutional layer to a global average pooling layer and computing a weighted average of extracted features [24]. Its success gave rise to SOTA explainability techniques such as Grad-CAM [32] and other variants. The subsections below provide an overview of a non-exhaustive list of recent, popular CAM methods.

### 2.6.1 Grad-CAM

Grad-CAM [32] is a more generalised version of CAM as Grad-CAM was designed for any CNN architecture, whereas CAM only works with specific architectures. From a high-level point of view, Grad-CAM works by passing in an input to the CNN model and calculating the gradient of the target value or class with respect to the the target feature maps in the network, which is usually the feature maps produced by the final convolutional layer. The feature maps are then weighted by the calculated gradients and summed up to create a class activation map. More formally, the Grad-CAM activation map $L^c_{\text{Grad-CAM}} \in R^{u \times v}$ for a class $c$, can be computed as the following ReLU activated weighted linear combination (as stated by Selvaraju et al. [32]):

$$\mathcal{L}^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k w^c_k A^k\right) \tag{2.3}$$

where $A^k$ are the feature maps of the target layer, $y^c$ is the raw score given by the model for class $c$, and $w^c_k$ weighs the importance of each feature map $k$ for a target class $c$ which is computed by performing a global average pooling of the gradients of the target with respect to each attention map [32]:

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{2.4}$$

This step is referred to as the gradient averaging step. $Z$ is a constant representing the number of pixels in the activation map.

It should be noted that since the ReLU activation function only produces a response for positive values, only pixels that cause the output to increase when its intensity is increased (i.e. pixels that have a positive influence) are highlighted in the generated map [32].

### 2.6.2 Grad-CAM++

Grad-CAM++ was introduced to improve Grad-CAM by providing better explanations for multiple instances of an object within an image [5]. Chattopadhay et al. [5] also claim that the localisations provided by Grad-CAM do not correspond to the entire object. They claim that since the partial derivatives in Eq. 2.4 are divided by a constant, responses that have a small area result in a smaller weight value compared to a response that takes up a larger area.

To address these issues, they proposed performing a weighted average using both the feature maps and gradients, instead of dividing Eq. 2.4 by $Z$:

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \cdot \text{ReLU}\left(\frac{\partial y^c}{\partial A_{ij}^k}\right) \tag{2.5}$$

Here, $\alpha_{ij}^{kc}$ represents the following:

$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{\frac{\partial^3 y^c}{(\partial A_{ij}^k)^3}\right\}} \tag{2.6}$$

where $(i, j)$ and $(a, b)$ are iterators over the same activation map [5].

### 2.6.3 HiResCAM

HiResCAM [11] is also another variant of Grad-CAM that alters the gradient averaging step (Eq. 2.4). The method does not average the gradients, instead the Hammard Product is used to multiply the feature map directly with the gradients in an element-wise manner, such that the output heatmap is given by:

$$\mathcal{L}_{\text{HiResCAM}}^c = \text{ReLU}\left(\sum_k \frac{\partial y^c}{\partial \mathcal{A}^k} \odot A^k\right) \tag{2.7}$$

The authors [11] state that by performing the element-wise multiplication, the output reflects the model's computations unlike Grad-CAM which they claim "blurs the effect of the gradients" due to the gradient averaging step. For example, if the gradients indicate that certain feature map elements should be scaled, HiResCAM [11] will alter the elements accordingly. The idea is that by doing so, the resulting heatmap will highlight areas that are more faithful to what the model uses for making its prediction.

### 2.6.4 LayerCAM

The authors of LayerCAM [14] aimed to improve both Grad-CAM and Grad-CAM++ because they observed that both techniques resulted in false positives when applied to feature maps in earlier layers. They claimed that this was because earlier layers usually capture fine-grained details, regardless of whether these belong to the object of interest or the background, and by using a weight $w_k^c$ that is applied equally to every location in a feature map $A_c^k$, noisy areas belonging to the background are not suppressed.

With the above finding, the authors proposed using a separate weight for each spatial location $(i, j)$ instead of a location independent $w_k^c$ such that:

$$w_{ij}^{kc} = \text{ReLU}\left(\frac{\partial y^c}{\partial A_{ij}^k}\right) \tag{2.8}$$

This weight is multiplied with the activation value at the corresponding location to get the following:

$$\hat{A}_{ij}^k = w_{ij}^{kc} \cdot A_{ij}^k \tag{2.9}$$

Finally, the activation map can be calculated as:

$$\mathcal{L}_{\text{LayerCAM}}^c = \text{ReLU}\left(\sum_k \hat{A}^k\right). \tag{2.10}$$

The authors demonstrated that the method outperformed both Grad-CAM and Grad-CAM++, as well as others such as ScoreCAM [37] for various tasks including object localisation, particularly in shallower layers.

### 2.6.5 Eigen-CAM

Eigen-CAM [24] proposed a very different approach from other CAM based methods which does not require the backpropogatation of any computations. Instead, the technique looks at principle components.

Assuming $W_{L=n}$ is a matrix which represents the combined weights of the first $k$ layers of size $mxn$, the technique involves projecting the input image $I$ to the last convolutional layer $k$

$$O_{L=k} = W_{L=k}^T I \tag{2.11}$$

upon which singular value decomposition is applied to obtain: an orthogonal matrix U of size $mxm$, an orthogonal matrix $V$ of size $nxn$ and a diagonal matrix $\mathbf{\Sigma}$ of size $mxn$ [24]

$$O_{L=k} = U\mathbf{\Sigma}V^T. \tag{2.12}$$

Finally, the activation map can be obtained by projecting $O_{L=k}$ onto the first eigenvector which represents the direction of maximum variance:

$$\mathcal{L}_{\text{Eigen-CAM}} = O_{L=k}V_1 \tag{2.13}$$

## 2.7 Evaluation Metrics

### 2.7.1 Mean Absolute Error

Mean absolute error (MAE) calculates the absolute difference between model prediction and the ground truth without taking into account the signs or direction:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.14}$$

where $y_i$ are the ground truth values, $\hat{y}_i$ are the predictions produced by the model and $N$ is the number of samples.

### 2.7.2 Root Mean Squared Error

Root mean sqaured error calculates the square root of the mean sqaured difference between the ground truth and predicted values:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2.15}$$

Compared to MAE, RMSE is generally less tolerant to outliers as it squares the error.

### 2.7.3 Coefficient of Determination

Originally proposed by Wright [39], the coefficient of determination, or $R^2$, is a popular method used for evaluating regression models. As explained by Chicco et al. [7], it measures the proportion of the variance in the dependent variable that can be explained from the independent variables:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{2.16}$$

where $\bar{y}$ is the average of the ground truth values. The maximum value for this metric is 1, which indicates that the variance of the dependent variable can be completely explained by the model. Higher values typically indicate a better performing regression model. On the other hand, a value of 0 can be achieved by a model that consistently predicts the mean of the ground truth values. This indicates that a model with a score of 0 does not explain any variability in the dependent variable. Note that the numerator in the fraction is simply the formula for the mean square error (MSE) of the model. Similarly, the denominator can be thought of as the MSE of a naive model that simply predicts the mean of the ground truth values. As such, a negative $R^2$ can also be predicted by a model that performs worse than the naive model.

### 2.7.4 F1

For all the positive instances that a model predicted, the precision metric penalises the model for wrongfully labelling a negative instance as positive:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.17}$$

Recall has a similar formula, but it focuses on the number of positive instances the model labelled as positive, over the total number of possible positives:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negative}} \tag{2.18}$$

The F1 score is a harmonic mean of both precision and recall because it combines both into a single metric:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.19}$$

As such, if either precision or recall is low, the F1 metric will also be low. This property is particularly useful in contexts where both false positives and false negatives are undesirable, such as models being used for the diagnosis of medical conditions.

### 2.7.5 Mean Average Precision

With precision and recall defined above, average precision (AP) can be calculated by finding the area under the precision-recall curve. Averaging the AP score for each query or class results in the mean average precision score (mAP).

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i \tag{2.20}$$

mAP is a popular metric used in the field of computer vision, particularly in the domain of object detection and segmentation for the comparison of different algorithms. A higher mAP score suggests a better performing model.

### 2.7.6 Mean Intersection Over Union

Assuming the model to be evaluated produces an area or boundary that can be compared to the ground truth area or boundary, the intersection over union IoU metric can be calculated as follows:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{2.21}$$

An IoU value of 1 is equivalent to perfect overlap with the ground truth, whereas an IoU value of 0 is equivalent to no overlap between the output and the ground truth. Again, this metric is popular in computer vision and the mean intersection over union (mIOU) can be calculated by taking an average over the IoU values for each query or class.

# CHAPTER 3

# Related Work

## 3.1 LVEF Estimation with CNNs

Early studies on EF estimation have relied on an initial left ventricle (LV) segmentation before regressing the LVEF value. For example, Ouyang et al. [27] used a CNN model to extract the LV and a 3D CNN model to regress the LVEF. However, this method performed the segmentation on each frame independently of the other, thus failing to utilise the temporal information for better segmentation consistency within each clip. Following this, methods have attempted to utilise the temporal information for LV segmentation, such as Thomas et al. [35] who used a CNN to encode the video into a lower dimensional representation, and used a graph convolutional network for segmentation, and regression head for calculating the LVEF. However, these rely on accurate segmentation which fails in challenging cases [21].

As a result, the SOTA methods focus on directly regressing the LVEF. Although early works focused on CNNs, vision transformers quickly gained popularity through their attention mechanism whereby they have the entire field of view, unlike CNNs which have a small receptive field in the initial layers [25].

## 3.2 LVEF Estimation with Transformers

Reynaud et al.'s work [29] was one of the first to introduce transformers successfully for the task of LVEF estimation, and they outperformed most previous CNN work. They proposed that the echocardiogram video is similar to text in NLP as both vary in length and the current frame or word is affected by the previous frames or words respectively. As such, they initially proposed an encoder to reduce the dimensionality of the data which was passed into a BERT-based module for capturing spatio-temporal relationships. They implemented a regression head estimating the LVEF, and a separate regression head labelling the ES and ED frame indices.

Following their success, others adapted other variations of vision transformer models. Fazry et al. [12] adapted the video swin transformer [19] which builds on top of the swin transformer [18] used for images, but extends the scope of local attention from the spatial domain to the spatio-temporal domain. The reduced computational complexity from using shifted window attention instead of global attention allows the model to handle larger images whilst being more efficient. They then added a regression head to estimate the LVEF.

More recently, motivated by the lack of explainability of previous works, Mokhtari et al. [22] proposed a method encompassing Vision Transformers (ViTs) in 3 hierarchical modules that first extracted the spatial relationship, followed by the temporal relationships and finally video-level information that captured relationships in clips of the same patient across different views (for example the A4C and A2C views of a patient). Their novel method provided explainability by visualising the learnt attention assigned by the spatial module to individual images and achieved remarkable results for LVEF estimation.

Others [21, 25] have combined CNNs with transformer models to utilise the advantages of both and have achieved remarkable results. CoReEcho [21] achieved SOTA results for LVEF by using a UniFormer [17] architecture, originally proposed for LVEF in EchoCoTr [25]. However, it should be noted that simply comparing metrics such as RSME to pick the best architecture is not reliable, because most methods vary in key aspects such as the sampling strategy used i.e. the number of frames in the train and test sets of different works vary substantially, which leads to varying performance. For example, the EchoCoTr [25] model that achieved an MAE of 3.947 with a frame frequency of 4 achieved an MAE of 4.168 when the frequency was set to 2, which is a greater error than reported for the GEMTrans [22] model.

## 3.3 Explainability in LVEF estimation papers

More recent papers have begun to explore the concept of explainability for the task of LVEF estimation. The authors of EchoGNN [23] implemented a model using graph neural networks (GNNs) and used the edge weights to provide explainability. They proposed that when graphically plotting the learnt edge weights, larger weights given to frames and edges that are in between ES and ED phases indicate that the model is able to accurately locate the respective frames for LVEF estimation. On the other hand, a more even distribution of weights indicates that the model is uncertain and that human intervention is required.

As previously mentioned, Mokhtari et al. [22] visualised areas of the input their proposed model focused on by performing attention rollout on the CLS (or class) token of their spatial module. They were able to demonstrate that their model located the LV before making the LVEF prediction.

Maani et al. [21] used Grad-CAM across a series of frames in their 2024 paper to show that their model focused on the LV region to make the LVEF estimation, but did not provide any further detail or analysis.

## 3.4 Knowledge Gaps

A major limitation in most works is the lack explainability. The EchoGNN model can indicate when human intervention may be required, but it does not identify which regions in the input the model focuses on or is unsure about, making it difficult to diagnose where the model is failing.

Although Mokhtari et al.'s [22] attempted to address explainability in their GEMTrans paper by visualising the learnt attention assigned by their spatial module, they provided no explanation of results from the temporal module and no other explainability techniques were explored.

Additionally, the explainability techniques used in the literature are not the current SOTA techniques, and we are not aware of any other paper in the domain of LVEF estimation that compare different SOTA explainability methods.

Another limitation in most techniques[29] is the lack of experimentation with data augmentation techniques. Although Fazry et al. [12] claimed that data augmentation led to worse performance due to the sensitivity of ultrasound video, others such as Leclerc et al. [16] indicated that non-frequent probe tilts, artefacts such as shadowed regions and low contrast can mislead a network and suggest that data augmentation can be beneficial.

Most also fail to express the clinical relevance of the model by using metrics such as F1 to evaluate how likely the model is to suggest false positive or false negative predictions for heart failure. This is important because a LVEF below 40% is a strong indicator of heart failure [23] and so it is clinically important for a model to accurately identify such cases.

# CHAPTER 4

# Methodology

## 4.1 Software and Tools

The following tools were used for this project:

- **Programming language:** Python 3.9.0
- **Framework**: Pytorch 1.12.1
- **Hyperparameter Tuning**: Weights and Biases.
- **GPU**: NVIDIA L4 GPU 64GB RAM (from lightning.ai).

## 4.2 Dataset

The EchoNet-Dynamic [27] dataset contains 10,030 echocardigraphy videos in the A4C view with labelled measurments including the LVEF and end-systole (ES) and end-diastole (ED) frame indices. Each video consists of grayscale frames of size 112x112 pixels. The videos were obtained from individuals who underwent imaging between 2016 and 2018 at Stanford University Hospital [27]. A segmentation of the LV at the ES and ED frames can also be obtained. It should be noted however that although a video can contain multiple heartbeat cycles, each video only has one frame labelled as the ES and one as ED.

| | Num. Videos | Num Frames |
|---|---|---|
| CardiacUDA | 992 | 102,796 |
| CAMUS | 500 | 10,000 |
| EchoNet-Dynamic | **10,030** | **1,755,250** |

Table 4.1: Table listing 2D echocardiography datasets, adapted from [40].

The dataset is pre-split into train, validation and test sets with 7465, 1288 and 1277 videos respectively which was not modified for this study.

Since it is the largest known echocardiogram video dataset most widely used for echocardiogram studies, it allows easy comparison to SOTA models and so this study also used the EchoNet-Dynamic dataset.

## 4.3 Sampling Strategy

Since the input to the model needed to be a fixed sized vector and using every frame in the video was not feasible due to the computational expense and due to the redundancy in neighbouring frames, we performed the common practice of sampling to the data. Following the implementation of Mokhtari et al. [22], both the training, validation and test sets underwent uniform sampling, whereby 32 equally spaced frames were selected. Note that the the uniform sampling technique does not take into account the position of the ES and ED frames, hence these frames are not guaranteed to be present in the training, validation or test data.

## 4.4 Model Architecture

The same architecture proposed in the GEMTrans model [22] was implemented with the following alterations. Firstly, since the EchoNet-Dynamic [27] dataset only contains a single A4C view of each patient, the video model which captured the relationship between multiple videos was removed. Thus, the output from the CLS token of the temporal module was passed to the regression head for predicting the LVEF. Finally, the input frames were scaled to 112x112, instead of 224x224, as this was the original size of the dataset. See Figure 4.2 for reference. As our paper does not propose any architectural changes to improve performance, the architecture stated in this section is used for all of our proposed models (except for the change in the input size of our final model).

### 4.4.1 Spatial Tokeniser

Each input image was split into equally sized non-overlapping patches of size 16x16 by the Spatial Tokensier (ST). Following this, the ST flattened and linearly projected each patch into a d-dimensional embedding which was fed into the following transformer encoder network.

### 4.4.2 Spatial Transformer Encoder

The Spatial Transformer Encoder (STE) followed the ViT architecture proposed by Dosovitskiy et al. [10]. The tokens from the ST were combined with learnable positional embeddings to retain positional information and then fed into a transformer encoder architecture. A learnable embedding, similar to BERT's [class] or cls token, was also prepended to the sequence of patch embeddings as per the ViT architecture [10].

The STE architecture can be described as the following [10, 22]:

$$h_t^0 = [\text{cls}_{\text{spatial}}; x_t'] + E_{\text{pos}}; \tag{4.1}$$

$$h_t^l = \text{MHA}(\text{LN}(h_t^{l-1})) + h_t^{l-1}, \quad l \in [1, \ldots, L]; \tag{4.2}$$

$$h_t^l = \text{MLP}(\text{LN}(h_t^l)) + h_t^l, \quad l \in [1, \ldots, L]; \tag{4.3}$$

$$z_t = \text{LN}(h_{t,0}^L), \tag{4.4}$$

where $x_t'$ represents the linearly projected patch from the STE, $E_{\text{pos}} \in R^d$ is a learnble positional embedding, $\text{cls}_{\text{spatial}} \in R^d$ represents the cls token, MHA is a multi-head attention network, LN is LayerNorm, and MLP is a multi-layer perceptron [22].

### 4.4.3 Temporal Transformer Encoder

The temporal module consisted of a transformer encoder performing similar operations outlined in Eqs. (4.1) to (4.4) on the output embeddings from the STE module. The final output from the part of the network corresponding to the CLS token was passed into the regression head.

### 4.4.4 Regression Head

The architecture of the regression head (shown in Figure 4.1) was also adapted from the GEMTrans model. However, since the latent embedding input for the regression head in our implementation came from the TTE instead of the video module, the number of input features for the first linear layer was changed from 200 to 600 and the number of output features were halved for every linear layer as also performed in GEMTrans.

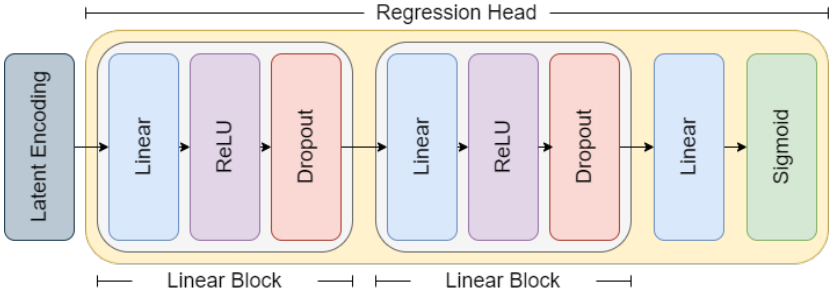


Figure 4.1: Diagram of the regression head used to regress the value of the LVEF. Latent embeddings from the TTE module is used as input

## 4.5 Pre-training of the Spatial Module

Since transformers require significant amounts of data for training, transfer learning was performed on the ViT module to provide a better initialisation and faster convergence. Like Mokhtari et al. [22], the weights of the B_16 ViT model pretrained on ImageNet-21k [31] were used to initialise our STE module.

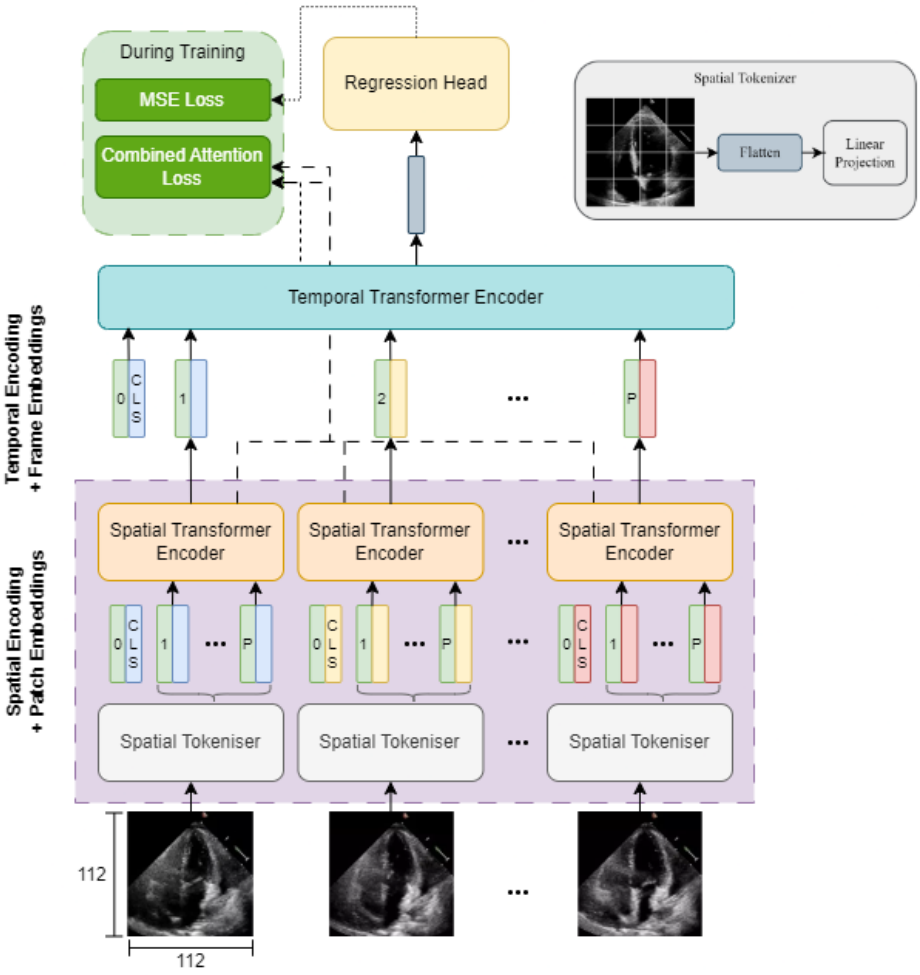


Figure 4.2: Diagram of our proposed model architecture adapted from GEMTrans [22]. Unlike GEMTrans, it does not have a video module.

The only difference is that in our work, the input image was of size 112x112 instead of 224x224 so the B_16 positional embedding weights were loaded and resized from size 1x197x768 to 1x50x768. Also following their [22] approach, no layers were frozen.

## 4.6 Hyperparameter Tuning

Due to computational limitations, hyperparameter tuning on the entire training split was not possible. For the same reason, methods such as Grid Search and cross-validation were also not appropriate. Instead, the entire baseline model architecture was trained on 10% of the training data for 10 epochs and evaluated on the hold-out validation set to filter out the bad runs. This was then repeated for 40% of the data using the best hyperparameters. The idea behind this approach was that the hyperparameters that enable the model to achieve the highest score would provide a good estimate of the best hyperparameters on the entire dataset for more epochs.

An alternative approach would have been to use the whole dataset but on a smaller model, however, for an NLP task, Raffel et al. [28] concluded that training a smaller model on lots of data often gave worse results than training a larger model for fewer steps (which is similar to our approach since the entire dataset is not explored in both techniques). Nevertheless, it should be noted that the performance of the hyperparamters can vary substantially with a change in data and the hyperparameters chosen using this method are unlikely to be optimal.

A Weights & Biases sweep [4] was performed to find the best hyperparameter values, i.e. values that allowed the model to achieve the highest $R^2$ score on the validation set. The $R^2$ score was chosen because this is the key metric used to compare LVEF methods. The learning rate, weight decay term, constant term for the temporal attention loss (see $\lambda_{temporal}$ below), and the constant term for the spatial attention loss (see $\lambda_{spatial}$ below) were optimised using Bayesian Hyperparameter Optimisation [34]. Unlike Grid Search which iterates through all possible combinations in a brute-force manner and random search which randomly selects combinations of hyperparameters to tune, the Bayesian approach takes advantage of the previous hyperparameter combination and its result to probabilistically select the next hyperparameter combination.

More formally, the technique builds a probabilistic model for the objective function, in this case maximising the $R^2$ score on the validation set, represented as

$$P(R^2 \textit{ is maximised}|\textit{hyperparameter combination}) = P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)} \tag{4.5}$$

which is easier to optimise than the objective function itself [4]. By taking a probabilistic approach whose hyperparameter combination selection improves with more runs, we were more likely to find the optimal hyperparameters with fewer iterations than a brute force approach, thus making it more computationally efficient.

Architectural hyperparameters such as the number of hidden layers were not optimised as the architecture closely resembles the original GEMTrans [22]

| | Spatial Attn. Loss $\lambda$ | Frame Loss $\lambda$ | LR | Weight Decay | Val. $R^2$ on 10% | Val. $R^2$ on 40% |
|---|---|---|---|---|---|---|
| **Run 1** | 0.87267 | 0.063637 | 0.00001 | 0.000001 | **0.33194** | 0.34443 |
| **Run 2** | 0.55060 | 0.063547 | 0.00001 | 0.00001 | 0.31018 | 0.37789 |
| **Run 3** | 0.03629 | 0.23223 | 0.00001 | 0.00001 | 0.13088 | **0.40067** |

Table 4.2: Table showing the top 3 configurations that resulted in the highest $R^2$ score on the validation set when trained with 10% of the training data for 10 epochs. The corresponding results when trained on 40% of the training data are also shown.

model which was already optimised for the dataset, and it would have been too computationally expensive to do so. We also manually chose a batch size of 8 because this provided a suitable balance between the time taken to compute the input and the compute capacity available.

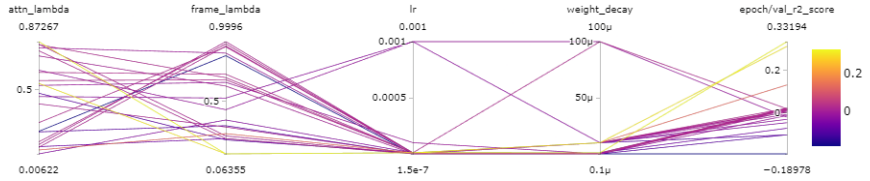


Figure 4.3: Figure showing the hyperparamters $\lambda_{spatial}$ (attn_lambda), $\lambda_{temporal}$ (frame_lambda), learning rate (lr), weight decay term which were tuned on 10% of the data and the resulting $R^2$ score achieved on the validation set.

To help select the final hyperparameter combination, the top three combinations that gave the highest score on the validation set were chosen and they were used to train the model on 40% of the total training data to see if the chosen hyperparameters were suitable with more data (see Figure 4.4 for the results).

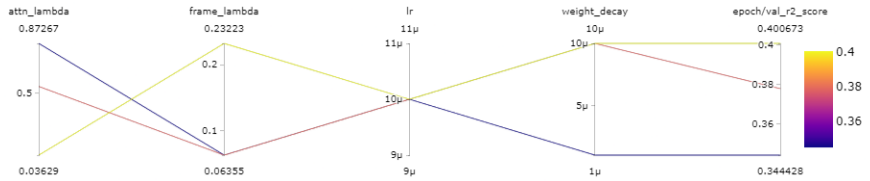


Figure 4.4: Figure showing the hyperparamters $\lambda_{spatial}$ (attn_lambda), $\lambda_{temporal}$ (frame_lambda), learning rate (lr), weight decay term which were tuned on 40% of the data and the resulting $R^2$ score achieved on the validation set.

Although both the runs that gave a high weighting to $\lambda_{spatial}$ and low weighting to $\lambda_{temporal}$ achieved significantly higher scores when trained on 10% of the data (Figure 4.3), they improved very little when trained on 40% of the data. In contrast, Run 3 which gave a high weighting to $\lambda_{temporal}$ and low weighting to $\lambda_{spatial}$ significantly improved in performance with more data and achieved the highest score out of those trained on 40% of the training data. As a result of its superior performance and potential to excel with more data, the hyperparameters used for Run 3 were used to train our baseline model.

## 4.7 Distance Guided Temporal Attention Supervision

By definition, calculating the LVEF requires identifying the ES and ED frames accurately, which itself is a challenging task. Since the EchoNet-Dynamic [27] dataset contains labels for the index of a single ES and ED frame within each video, Mokhtari et al. [22] proposed to supervise the attention given by the CLS token in the temporal module and guide it to increase the attention given to the ED and ES frames as follows:

$$L_{\text{attn},t}^{\text{temporal}} = \begin{cases} (\text{ATTN}_{\text{cls},s}^{\text{temporal}} - 1)^2, & \text{if } t \in [\text{ED, ES}] \\ 0, & \text{otherwise;} \end{cases} \tag{4.6}$$

$$y'_{\text{seg}} = \text{OR}(y_{\text{seg}}^{\text{ed}}, y_{\text{seg}}^{\text{es}}); \tag{4.7}$$

$$L_{\text{attn},s}^{\text{spatial}} = \begin{cases} (\text{ATTN}_{\text{cls},s}^{\text{spatial}} - 0)^2, & \text{if } y'_{\text{seg},s} = 0 \text{ (outside LV)} \\ 0, & \text{otherwise;} \end{cases} \tag{4.8}$$

$$L_{\text{attn}} = \lambda_{\text{temporal}} \sum_{t=1}^{T} L_{\text{attn},t}^{\text{temporal}} + \lambda_{\text{spatial}} \sum_{s=1}^{HW} \frac{1}{p^2} L_{\text{attn},s}^{\text{spatial}}, \tag{4.9}$$

where, as originally stated by Mokhtari et al. [22], "$y'^{ed}_{seg}, y'^{es}_{seg} \in \{0,1\}^{\frac{HW}{p^2}}$ are the coarsened versions (to match patch size) of $y_{seg}$ at the ED and ES locations, OR is the bit-wise logical *or* function", *ed* and *es* indicate the temporal embedding indices at the ED and ES frames respectively, and $\lambda_{spatial}, \lambda_{temporal} \in [0,1]$ coefficients weigh the contribution of the spatial and temporal losses respectively on the overall attention loss.

However, this loss does not penalise the attention wrongly given to frame embeddings that are not ES or ED, thus limiting the extent of supervision provided. Instead, we propose the following novel function to penalise the model for giving attention to frames based on their Euclidean distance from the ES and ED frames:

$$L_{\text{attn,t}}^{\text{temporal\_1}} = \begin{cases} \left(\text{ATTN}_{\text{cls,s}}^{\text{temporal}}\right)^2 & \text{if } t > \max(\text{ES}, \text{ED}) + \frac{|\text{ES}-\text{ED}|}{2} \\ \left(\text{ATTN}_{\text{cls,s}}^{\text{temporal}}\right)^2 & \text{if } t < \min(\text{ES}, \text{ED}) - \frac{|\text{ES}-\text{ED}|}{2} \\ \left(\text{ATTN}_{\text{cls,s}}^{\text{temporal}} \cdot \sin\left(\frac{\pi(t-\text{ED})}{\text{ES}-\text{ED}}\right)\right)^2 & \text{otherwise} \end{cases} \tag{4.10}$$

The intuition behind this is that frames directly before and after, say the ES frame, are a closer approximation to the actual volume of the LV at the ES frame than frames further away. Therefore attention given to closer frames is more favourable than attention given to frames further away.

However, this loss alone can result in the attention for the ED and ES frames being set to 0 as well, because doing so will reduce the loss. To avoid this, we propose adding the temporal loss component suggested by Mokhtari et al. [22] (Eq. 4.11) to our loss, which encourages the model to give more attention to the ED and ES frames (see Figure 4.5 for a visualisation).

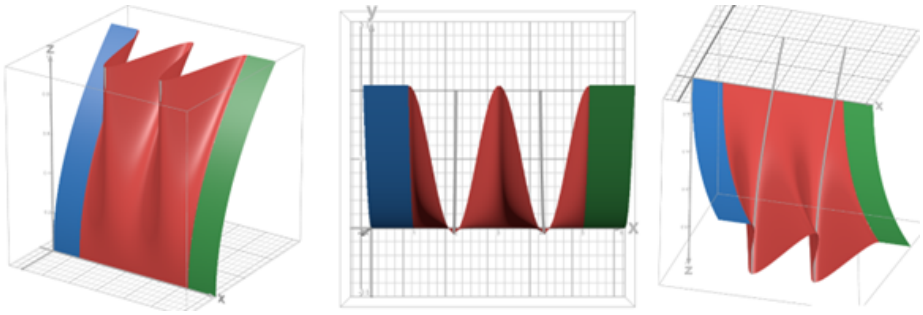

Figure 4.5: Visualisation of the proposed component of the temporal attention loss ($L_{\text{attn,t}}^{\text{temporal\_1}}$), i.e. DGT loss, in 3D when ES=2 and ED=4. x-axis represents each frame t, y-axis gives the value of the loss and z-axis represents the attention scores. The grey lines indicate the original temporal loss function ($L_{\text{attn,t}}^{\text{temporal\_2}}$) at the ES and ED frames.

$$L_{\text{attn},t}^{\text{temporal\_2}} = \begin{cases} (\text{ATTN}_{\text{cls},s}^{\text{temporal}} - 1)^2, & \text{if } t \in [\text{ED, ES}] \\ 0, & \text{otherwise;} \end{cases} \tag{4.11}$$

With both losses combined, we overall temporal attention loss can be written as:

$$L_{\text{attn},t}^{\text{temporal}} = \frac{L_{\text{attn},t}^{\text{temporal}_1} + L_{\text{attn},t}^{\text{temporal}_2}}{2} \tag{4.12}$$

Although this new loss function works with a single pair of labelled ES and ED frames, we presumed it may have a negative effect on the EchoNet-Dynaimic [27] dataset because each input video could contain multiple heartbeat cycles despite there being only one pair of ES and ED locations labelled. As such, the initially proposed loss function would penalise the attention given to frames in unlabelled parts of a different heartbeat that closely resemble the ES or ED volumes the same way it penalises irrelevant frames. This was thought to be undesired because such frames could have otherwise helped provide a more accurate estimate to the LVEF value.

To overcome this, we attempted to simplify the loss function to the following so that only the local region surrounding the labelled ES and ED frames was affected by the loss:

$$L_{\text{attn,t}}^{\text{temporal\_1}} = \begin{cases} 0, & \text{if } t > \max(\text{ES}, \text{ED}) + \frac{|\text{ES}-\text{ED}|}{2} \\ 0, & \text{if } t < \min(\text{ES}, \text{ED}) - \frac{|\text{ES}-\text{ED}|}{2} \\ \left(\text{ATTN}_{\text{cls,s}}^{\text{temporal}} \cdot \sin\left(\frac{\pi(t-\text{ED})}{\text{ES}-\text{ED}}\right)\right)^2 & \text{otherwise} \end{cases} \tag{4.13}$$

Define the number of frames between the ES and ED frames as $dist(ES, ED)$. This loss assumes that the number of frames between the ES and ED frames of

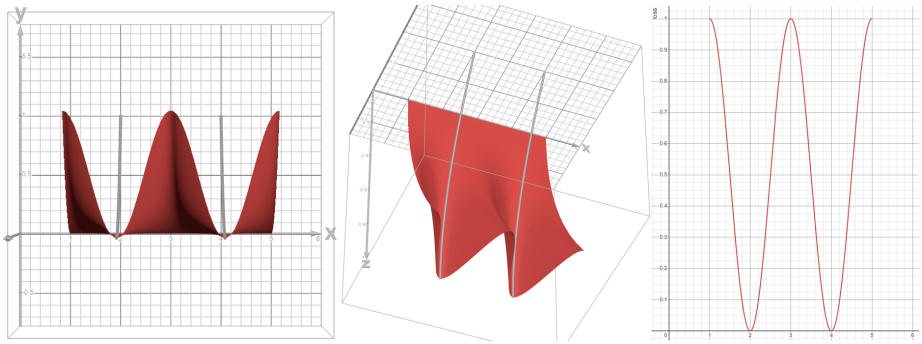

Figure 4.6: Visualisation of the proposed component of the temporal attention loss ($L_{\text{attn,t}}^{\text{temporal\_1}}$) in 3D and 2D when ES=2 and ED=4. x axis represents each frame t, y-axis gives the value of the loss and z axis represents the attention scores. The grey lines indicate the original temporal loss function ($L_{\text{attn,t}}^{\text{temporal\_2}}$) at the ES and ED frames.

each heartbeat cycle in a single video is roughly similar such that frames a distance of $dist(ES, ED)/2$ frames away from the labelled ES and ED frames are least relevant to calculating the LVEF because these frames capture the volume of the heart when it's roughly halfway through it's heartbeat cycle. Although not perfect, this is a reasonable assumption because the heartbeat variability is generally of a small magnitude. It should be noted that by simplifying the loss, the attention given by the CLS token of the temporal module is supervised for fewer embeddings.

However, when experimenting with both loss functions, Eq. 4.10 demonstrated better performance than Eq. 4.13, possibly because Eq. 4.10 supervises more encodings, thus leading to better results. Therefore here onwards, 'DGT loss' refers to the loss function that uses Eq. 4.10. To explore the effects of the proposed loss function, a second model was trained with the DGT loss applied and everything else including model parameters and architecture was kept the same as our baseline for a fair comparison. The second model shall be referred to as the 'DGT Loss' model.

## 4.8 Data Augmentation

Most image data augmentation techniques fall into the categories of spatial transformation (e.g. random flips and rotations), colour distortion (e.g. alterations in hue and contrast) and information dropping [6]. The latter has gained more attention in recent years in the field of computer vision, motivated by the idea that removing a small amount of information could help the model learn more important information, resulting in a more robust model.

### 4.8.1 Standard Data Augmentations

Our baseline model used the same data augmentations applied by Mokhtari et al. [22] as this was shown to give them promising results. These were the data augmentations applied:

- Convert the frames to grayscale.
- Resize the frames to be processed by the model.
- Normalisation with the mean and standard deviation of the EchoNet-Dynamic dataset [27] (i.e. 0.12922 and 0.19023 respectively to 5 decimal places).
- Random horizontal flips with 30% probability.

### 4.8.2 GridMask

One of the key challenges with echocardiography is occlusion and regions of the heart falling outside the scan view. Inspired by this, we experimented with information dropping data augmentation to help improve the mdoel's robustness to such challenges and to improve its interpolation ability.

More specifically, although several information dropping techniques such as random erasing [41], cutout [9] and hide-and-seek [33] exist, we experiment with GridMask [6] because it avoids excessive deletion of continous regions and was shown to outperform the alternative methods previously stated for a variety of tasks such as object detection and semantic segmentation [6].

The GridMask [6] data augmentation technique works by overlaying the input image with a grid of uniformly shaped and uniformly spaced black squares. The size and spacing of the squares can be tuned using the four variables shown in Figures 4.7 (see Figure 4.8 for an example).

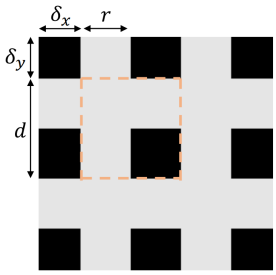


Figure 4.7: Figure from [6] which shows the parameters needed to create the grid in the GridMask data augmentation technique.

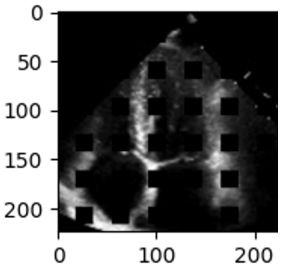


Figure 4.8: Figure showing the GridMask data augmentation applied on a sample frame of size 224x224 pixels.

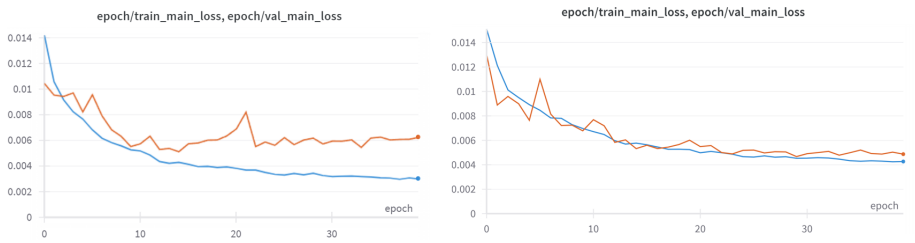


Figure 4.9: Figure showing the training and validation losses of the DGT Loss model (left) and the DGT Loss + GM model (right).

Observing the training and validation losses for the 'DGT Loss' model, it was clear that the model was overfitting (see Figure 4.9). To reduce its effect and increase the model's robustness, GridMask was applied to the training data of a third model. As echocardiography scans have poor detail, parameters were chosen by manual inspection to prevent excessive information deletion and preserve enough detail to approximate the shape of the LV. Hereafter, the third model which was trained and tested under the same settings as the 'DGT Loss model' aside from the addition of the GridMask data augmentation will be referred to as the 'DGT Loss + GM' model.

## 4.9 Summary of Proposed models

To summarise, the following models were trained and tested:

- **Baseline**. Model that had the same architecture shown in Figure 4.2, trained with the standard data augmentations used by Mokhtari et al. [22] and used the hyperparameters found from hyperparamter tuning.
- **DGT Loss model**. Model that had the same architecture, and used the same data augmentations and hyperparamters as Baseline for a fair comparison, but uses the proposed DGT loss (Eq. 4.10).
- **DGT Loss + GM model (112)**. Model that was trained with the same settings as the DGT Loss model but also had GridMask data augmentation applied to the training data.
- **DGT Loss + GM model (224)**. Model that was trained with the same settings as above, but the frames in the input were upscaled to a height and width of 224 pixels. The batch size was also reduced from 8 to 2 as it was not computationally feasible to do so otherwise.

All proposed models were trained for a maximum of 40 epochs as we observed that the validation scores did not improve after this.

## 4.10 Explainability through Visualisations

### 4.10.1 Visualisation of the Spatial Module

To help provide qualitative explanations as to which parts of the input the model focused on to generate predictions, we employed and evaluated various visualisation techniques. As originally reported in the GEMTrans [22] paper, we applied attention rollout on the learned attention of the STE module for each input. In addition to this, we also explored the following CAM-based visualisation techniques: Grad-CAM, Grad-CAM++, HiResCAM, LayerCAM and Eigen-CAM. These techniques were chosen because they are popular for providing model explainaibility through visualisation for visual data, and they are less computation demanding unlike other techniques such as SHAP or LIME. It should also be noted that the previously stated techniques are not an exhaustive list of CAM techniques. Those specific techniques were chosen because they were easy to implement using the the pytorch-grad-cam [13] library, were shown to have promising results in various papers, and were computationally more efficient than other techniques such as Score-CAM [37] which measures how much the output changes when the image is perturbed, and Ablation-CAM [8] which requires setting activation map values to 0 to also monitor the change in output.

Since the CAM-based techniques were originally designed for the feature maps of CNN models, the output from after the first normalisation of the last block of the ViT was reshaped to an image with a height and width of 7 and 768 channels to represent the feature maps of a CNN, as is standard practice [13].

A layer before the final output was chosen because the gradient of the output with respect to the last layer would be 0 [13] since the regression head only takes input from the CLS token in the last block as opposed to the full output.

The reason for performing heatmap visualisations within the STE module instead of the TTE module is that the STE module is most relevant for capturing the spatial information within the input.

### 4.10.2 Visualisation of the Temporal Module

Similar to what was performed in the STE module, attention rollout was also performed on the TTE module to help provide a qualitative analysis of the effects the proposed changes in the temporal domain i.e. to visualise the attention given to each of the 32 frames.

## 4.11 Evaluation Metrics

### 4.11.1 Model Performance

To evaluate the performance of our model for the proposed changes and against SOTA methods, we followed the standard practice and used root mean square error (RMSE), mean absolute error (MAE), $R^2$ score and also included the F1<40% score due to its clinical relevance. Of these, the $R^2$ score was used to optimise the model during training as this is the primary metric used for comparisons in this research area. F1<40% refers to the F1 score where LVEF

scores of below 0.4 or 40% are considered as positive instances. This metric helps to evaluate the model's clinical suitability as LVEF values below 40% are a strong indicator of heart failure [23] and we prefer a model that reports minimal false positives and false negatives.

### 4.11.2 Explainability

We evaluated the different explainability methods by quantitatively assessing the quality of the heatmaps produced by each of the visualisation techniques stated previously. Since EchoNet-Dynamic provided the binary segmentation masks of the left ventricle for each video, we used this as the ground truth and used the mIOU and mAP metrics on all examples in the test set that contained both an ES and ED frame after uniform sampling. A higher mIOU and mAP score was interpreted to indicate a better performing model. It should be noted that this interpretation requires the underlying assumption that the model looks solely and entirely at the left ventricle segmentation to calculate the LVEF, which may not necessarily be the case as discussed in the Evaluation chapter.

Alternatively, it would have been possible to evaluate the methods using perturbation-based metrics such as the drop in confidence of the model when parts of the heatmap with the highest response are removed and to complement it, the increase in confidence of the model when only providing the region of maximum response as input. However, this is computationally expensive as a second forward pass would have to be made once the input has been modified (more forward passes if the perturbation step is an iterative process).

## 4.12 Other Experimentation

Various other experiments were carried out to improve performance, but the specific results of these models will **not** be reported in the Results section as they were stopped early due to their poor results and are not the main focus of this paper. This subsection outlines such experiments.

One of the main challenges with the uniform sampling technique was that the ES and ED frames were not guaranteed to be in a sampled clip, which meant that accurately calculating the LVEF was not possible. It also meant that our loss could not be applied to all instances of the training data. To address this, inspired by Reynaud et al. [29], we performed the following for each video in the training set:

1. The labelled ES and ED frames were located and a sequence $S$ was formed where $S = [f_{ES}, f_{T_1}, ..., f_{T_N}, f_{ED}]$ [29]. Each $f_{T_i}$ represents the $i$th frame between the ES and ED frames exclusive.

2. $S$ was duplicated to form $\bar{S}$.

3. The ES and ED frames were removed from $\bar{S}$ and the sequence was reversed to form $\tilde{S} = [f_{T_N}, ..., f_{T_1}]$.

4. $\tilde{S}$ and $\bar{S}$ were iteratively appended to S until the sequence $S' = [f_{ES}, f_{T_1}, ..., f_{T_N}, f_{ED}, f_{T_N}, ..., f_{T_1}, f_{ES}, ...]$ was formed such that it had a length less than or equal to 32.

5. A random number was generated between 0 and the difference between the length of $S'$ and 32 and the resulting number of frames were prepended to $S'$.

6. The remaining number of frames needed to make the length of the sequence 32 was then appended to the sequence.

This process ensured that:

- Each video contained an ES and ED frame.
- The ES and ED frames were located in different indexes across the videos.
- All ES and ED frames were accurately labelled.

The temporal loss function was also updated to be a continuous sinusoidal loss with its minima at each ES and ED position to account for the multiple ES and ED frame pairs located at a uniform distance (see Eq. 4.14).

$$L_{\text{attn,t}}^{\text{temporal\_1}} = \left( \text{ATTN}_{\text{cls,s}}^{\text{temporal}} \cdot \sin\left( \frac{\pi(t - \text{ED})}{\text{ES} - \text{ED}} \right) \right)^2 \tag{4.14}$$

However, the trained model reported worse results than our baseline model. This was thought to be a result of the clips in the training set containing consecutive frames with minimal motion difference unlike the validation and test sets which contained a greater change in movement of the heart with each consecutive frame as the entire clip was uniformly sampled. To further improve the results by reducing the redundancy of consecutive frames, each sub-sequence $f_{T_1}, ..., f_{T_N}$ in $S'$ was uniformly sampled, resulting in shorter length heartbeats. The uniform sampling was performed with a frequency of 2,4, and 8. Although this improved results from the previous attempt with a frequency of 8 providing the best results, the results against the validation set were still worse than our baseline model, so early stopping was performed. The poor performance may have occurred because the training data was of a different structure to the validation data in terms of the number of duplicate frames and more importantly, the existence of multiple valid ES and ED frames in the training data which did not occur in the validation or test data due to the uniform sampling performed on those sets. As a result, we hypothesise the model may have forcefully tried to identify multiple ES and ED frames in the validation set which did not exist. This is supported by the peaks in the attention scores given by the CLS token of the TTE on example inputs from the validation set as shown in Figure 4.10, which suggests that the model is falsely identifying multiple ES and ED frames, leading to poor performance.

To only retain one ES and ED frame in the training data, but also perform the mirrored sampling, the same steps as above were carried out, however, the following changes were made:

- $\tilde{S} = [f_{T_N}, ..., f_{T_2}]$
- $\bar{S} = [f_{T_1}, ..., f_{T_{N-1}}]$

to form the following:
$S' = [f_{ES}, f_{T_1}, ..., f_{T_N}, f_{ED}, f_{T_N}, ..., f_{T_1}, f_{T_2}, ..., f_{T_N}, f_{T_{N-1}}, ...]$. This also allowed the use of the original temporal loss we proposed (Eq. 4.10) instead

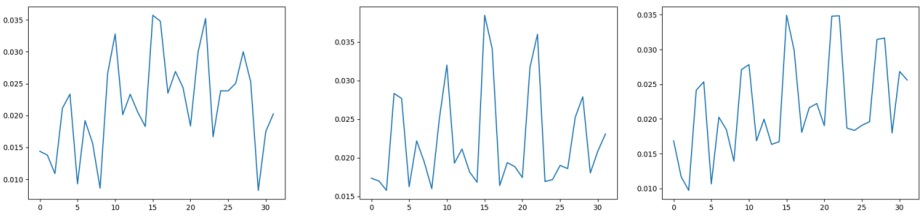

Figure 4.10: Figure showing the attention scores (y-axis) given by the CLS token of the TTE of the model trained with mirror sampling to each of the 32 frames (x-axis) for 3 example inputs in the validation set.

of Eq. 4.14. However, this resulted in poor performance as well, again possibly due to a combination of consecutive frames with minimal change and duplicate frames across each heartbeat, and was discontinued.

Experimentations with freezing layers of the pre-trained ViT was also performed to speed up training, however, it was observed that freezing layers dramatically reduced the performance. This most liekly occurred because the information learnt by the ViT in the early layers from ImageNet-21K were vastly different from the features in the EchoNet-Dynamic dataset.

# CHAPTER 5

# Results

## 5.1 Proposed Models

Although several experimentations were carried out, the results for only 4 models are reported as these were the core focus of this project.

The baseline model achieved an $R^2$ score of 0.59. In comparison, the DGT Loss model achieved an $R^2$ score of 0.64 (0.05 higher) and the DGT Loss + GM model (112) achieved an even higher $R^2$ score of 0.65 (see Table 5.1). Finally, our DGT Loss + GM model (224) model, which only differed from the DGT Loss + GM model (112) model in terms of the batch size used and the input video size, achieved the highest score for all metrics, particularly the $R^2$ score on the test set which was 0.13 higher than the baseline model.

## 5.2 Results against SOTA

The results of our best model, which achieved an $R^2$ score of 0.72, is compared to SOTA methods in table 5.2, the best of which achieved an$R^2$ score of 0.82 [21]. Our model achieved higher results than Reynaud et al.'s model [29] and achieved scores most similar to EchoGNN [23], particularly for the $F1 < 40\%$ metric which was only 0.02 lower.

| Models | Train | | | | Validation | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $R^2$ | MAE | RMSE | F1<40% | $R^2$ | MAE | RMSE | F1<40% | $R^2$ | MAE | RMSE | F1<40% |
| Baseline | 0.63 | 5.54 | 7.58 | 0.70 | 0.60 | 5.64 | 7.79 | 0.68 | 0.59 | 5.71 | 7.82 | 0.70 |
| DGT Loss | 0.72 | 4.87 | 6.58 | 0.76 | 0.66 | 5.21 | 7.18 | 0.72 | 0.64 | 5.31 | 7.32 | 0.69 |
| DGT Loss + GM (112) | 0.71 | 4.97 | 6.73 | 0.75 | 0.69 | 5.06 | 6.83 | 0.73 | 0.65 | 5.27 | 7.22 | 0.69 |
| DGT Loss + GM (224) | **0.73** | **4.68** | **6.40** | **0.78** | **0.74** | **4.63** | **6.33** | **0.77** | **0.72** | **4.81** | **6.53** | **0.76** |

Table 5.1: Table showing the results of our 3 models evaluated on the train, validation and test sets. The $R^2$, MAE, RMSE and F1<40% scores are reported. DGT Loss + GM (112) represents the model trained with frames of height and width 112 pixels whereas DGT Loss + GM (224) represents the model trained with frames of height and width 224 pixels.

| Models | $R^2$ | MAE | RMSE | F1<40% |
|---|---|---|---|---|
| Reynaud et al. [29] | 0.52 | 5.95 | 8.38 | - |
| Ours | 0.72 | 4.81 | 6.53 | 0.76 |
| EchoGNN [23] | 0.76 | 4.45 | - | 0.78 |
| GEMTrans [22] | 0.79 | 4.15 | - | - |
| Ouyang et al. [27] | 0.81 | 4.05 | 5.32 | - |
| EhoCoTr[25] | 0.81 | 3.98 | 5.34 | - |
| CoReEcho [21] | **0.82** | **3.90** | **5.13** | - |

Table 5.2: Table showing a performance comparison of our model against different SOTA models. The $R^2$, MAE, RMSE and F1<40% scores are reported where possible. 'Ours' represents the model trained on data of height and width 224 pixels with the proposed DGT Loss and GridMask data augmentation applied (DGT Loss + GM (224) model).

## 5.3 Quantitative Results of Explainability Methods

| Method | mIoU | mAP |
|---|---|---|
| Eigen-CAM | **0.17** | **0.22** |
| Grad-CAM++ | 0.16 | 0.16 |
| Attention Rollout | 0.16 | 0.16 |
| LayerCAM | 0.10 | 0.17 |
| HiResCAM | 0.08 | 0.15 |
| Grad-CAM | 0.03 | 0.11 |

Table 5.3: Comparison of explainability methods based on mIoU and mAP against the ground truth LV segmentations of the ES and ED frames.

The mIoU and mAP scores suggest that Eigen-CAM is the most reliable explainability method for this model for this particular task as it achieved the highest scores of 0.17 and 0.22. Grad-CAM++ and attention rollout achieved very similar scores to each other and was similar to Eigen-CAM in terms of mIoU. Grad-CAM resulted in the worst performance for both metrics, achieving an mIoU score of 0.03 and mAP score of 0.11.

# CHAPTER 6

# Discussion and Evaluation

## 6.1 Model Performance

Our results showed that our DGT loss significantly improved the model performance and the GridMask data augmentation had a positive impact in reducing the effects of overfitting and also further improving results. The improved performance using GridMask is worthy of mention because others such as Fazry et al. [12] have claimed that their data augmentation led to poorer performance due to the sensitivity of ultrasound. We believe that the GridMask augmentation retained just enough information to allow the model to interpolate the full outline of the LV and drop excess noise in the image, which together facilitated the training of a more robust model. We also observed that when our model was augmented with an additional random colour jitter and Gaussian blur data augmentation after GridMask was applied, the performance severely dropped, causing the model's $R^2$ score to drop 0.1 lower than the model without the additional augmentation on the validation set. As a result, recognise that data augmentation techniques should be applied carefully and in moderation for echocardiography tasks.

We also observed that the performance significantly improved after the input height and width were resized to 224 pixels. We propose that although the original height and width of the frames in the EchoNet-Dynamic dataset was 112 pixels, the pre-trained ViT model architecture was optimised for images of size 224x224 which was why the model achieved superior performance once the input was resized.

The $F1 < 40\%$ score for most methods have not been reported, making comparison difficult. Our model's $F1 < 40\%$ score is only 0.02 less than EchoGNN, which implies that our model makes a similar number of false positives and false negatives. Therefore, it has a very similar capability to EchoGNN at identifying cases that are at a high risk of heart failure, highlighting its clinical feasibility.

Although our best model achieved competitive results with some of the other papers in the field, the $R^2$ score was lower than the best model and also the

original GEMTrans model. One of the key reasons for this was because our hyperparameters were not tuned. Due to limited GPU compute, we tuned a selected number of hyperparaemter values on a small subset of the data and these parameters were used throughout for all our models, including our best performing model. This means that the hyperparameters used for all our models are very unlikely to be optimal and thus would have resulted in non-optimal results.

## 6.2 Explainability Methods

As shown in Table 5.3, Eigen-CAM produced the best results quantitatively. Visually inspecting the example activation maps produced on a sample input in Figure 6.3, it is clear that Eigen-CAM provided better localisations around and within the LV region compared to techniques such as Grad-CAM which suggested that the model was paying attention to areas outside the ultrasound scan region. As a result, it achieved the lowest mIoU and mAP scores.

Although the mIoU value of 0.17 was similar to that of some of the other top-performing methods, Eigen-CAM's mAP value was significantly higher. This suggests that a smaller proportion of Eigen-CAM's activation map's areas of maximum response fell outside the LV region (i.e. fewer false positives) compared to other explainability techniques, but the areas overlapped with a similar proportion of the LV segmentation mask as other top-performing methods.

Despite the competitive performance of Grad-CAM++, the activation maps produced were noisy upon visual inspection. The results may have lowered if the threshold to create a binary mask of the activation map was lowered as a lot of additional noise from the method would be retained.

Grad-CAM's poor performance may be explained by the gradient averaging step it performs in Eq. 2.4 as discussed in the Background chapter, which may have caused the model to focus more on larger regions, leading to less precise localisation. Consequently, methods such as Grad-CAM++ and HiResCAM which addressed this issue using a different gradient averaging step achieved higher results, likely due to its better ability to localise on smaller features that are important for prediction. Eigen-CAM, as again discussed in the Background chapter, calculates the direction of maximum variance. As a result, Eigen-CAM may have been able to focus on highlighting the LV which was key to making the prediction and filter out noise or less important features.

Comparing the Eigen-CAM map of the DGT Loss + GM model (112) in Figure 6.2 and the corresponding map of the model with the up-scaled input in Figure 6.3, it can be seen that the latter model focused more on the area within the LV region (both the upper and lower half of the LV), whilst the former model appears to have focused more on the surrounding walls of the LV, which could explain why the predictions of the former were less accurate.

It should be noted that the quantitative analysis made comes under the assumption that the model focuses on the LV region to make the predictions. Although the LV segmentation was used because by definition the LVEF is calculated by measuring the volume of the LV region, the question of whether the model needs to pay attention to the *entire* LV segmentation arises. For example, looking at the results in Figure 6.3, it can be seen that although most instances suggest that the model looked within the LV region, the activation

map on other frames (e.g. row 2 and 3 of LayerCAM) indicate that the model sometimes focused on the tissue around the LV, which in turn can lead to low IOU scores when compared against the entire LV segment. Moreover, for instances where the model did focus on the LV, the activation map, e.g. for Eigen-CAM, does not encompass the entire region. This could explain why the mIOU scores across all metrics were low and it suggests that our assumption has room for improvement. Since the movement of the LV is not isolated from the rest of the heart, it is also possible that the model looked at deformation in other surrounding areas as well to calculate the LVEF score, which could also lead to lower mIOU and mAP scores. This phenomenon is more evident for the DGT Loss + GM model (112) as previously discussed (Figure 6.2).

Further, to demonstrate that the activation maps can be analysed to identify cases where the model prediction was inaccurate, we present an input on which all proposed methods performed poorly. As shown in Figure 6.1, the LV in the frame has poor contrast compared to the background, making the task of predicting the LVEF difficult. From all 3 activation maps, particularly the map produced by the attention rollout method and Grad-CAM++, it is clear that the baseline model focused on areas outside the LV region, which explains why it resulted in the worst performance of the 3 models. The other two models, which had a lower RMSE on this example, focused less on areas outside of the ultrasound scan to make the prediction. As shown in the Eigen-CAM activation map, the DGT Loss + GM (112) model has a larger overlap with the LV segmentation mask and the model did not focus on the irrelevant areas on the bottom left of the frames unlike the DGT Loss + GM (224) model, which could explain why the DGT Loss + GM (112) model performed the best on this particular input. From this example, we hypothesise that it is possible to identify poor predictions made by the model by observing if the activation map suggests that the model focused on areas *outside* the LV region (note that the activation map of good prediction does not necessarily have to encapsulate the entire LV region as seen in many examples).

Finally, inspecting the attention given by the CLS token of the temporal module, it can be seen that in the example shown in Figure 6.4, the baseline model gave more attention to frames that are not labelled as ES or ED, unlike the model trained with our proposed DGT loss which gave more attention to the labelled ES and ED frames, whilst also reducing the attention given to other frames. Since the clip contained multiple heartbeat cycles, other unlabeled ED and ES frames of a similar volume to the labelled frames may have existed, which could explain why other frames also received high attention scores. If this was the case, this example suggests that our proposed loss allowed the model to give the most attention to the true ES and ED frames that contain the greatest change in volume, whilst also recognising and giving sufficient attention to other ES and ED frames. It should be noted however that although this example supports that our loss function works as intended, the results of this single example may not necessarily hold for all cases.

## 6.3 Limitations

One of our key assumptions was that the ground-truth ES and ED frames labelled by experts represent the true ES and ED frames. However, as noted by

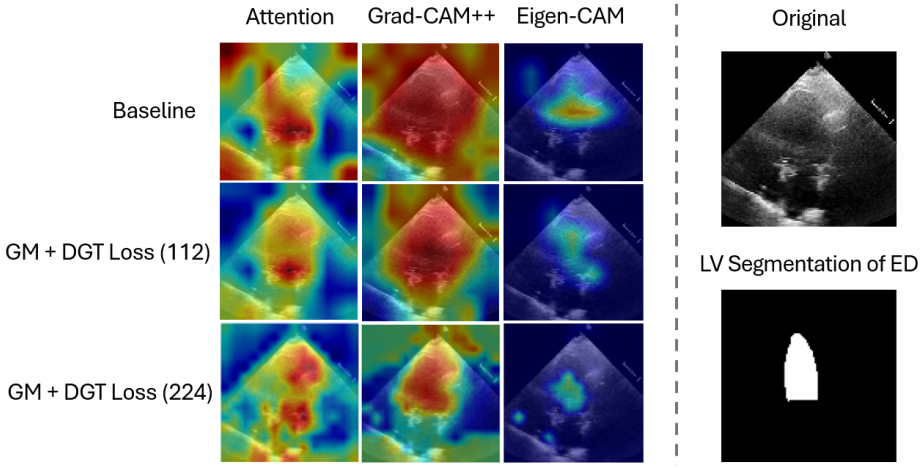


Figure 6.1: Example of the activation maps produced for an input that all proposed methods achieved a high RMSE. The RMSE scores are as follows: 10.25, 7.734, and 8.57 for the Baseline, DGT Loss + GM (112) and DGT Loss + GM (224) models respectively. The sample frame from the video input upon which the activation map is overlayed is shown on the right. The LV segmentation of the ED frame of the input is also shown on the right. Attention refers to attention rollout.

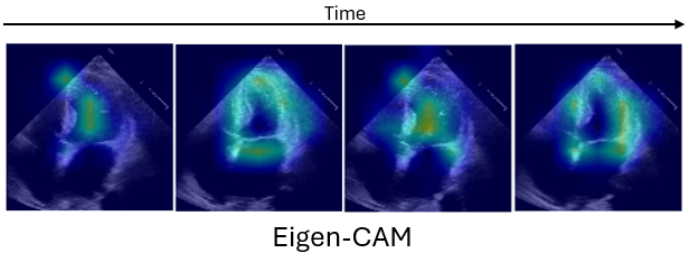


Figure 6.2: Figure showing the Eigen-CAM activation map on examples of 4 frames of a single input video passed to the DGT Loss + GM (112) model.

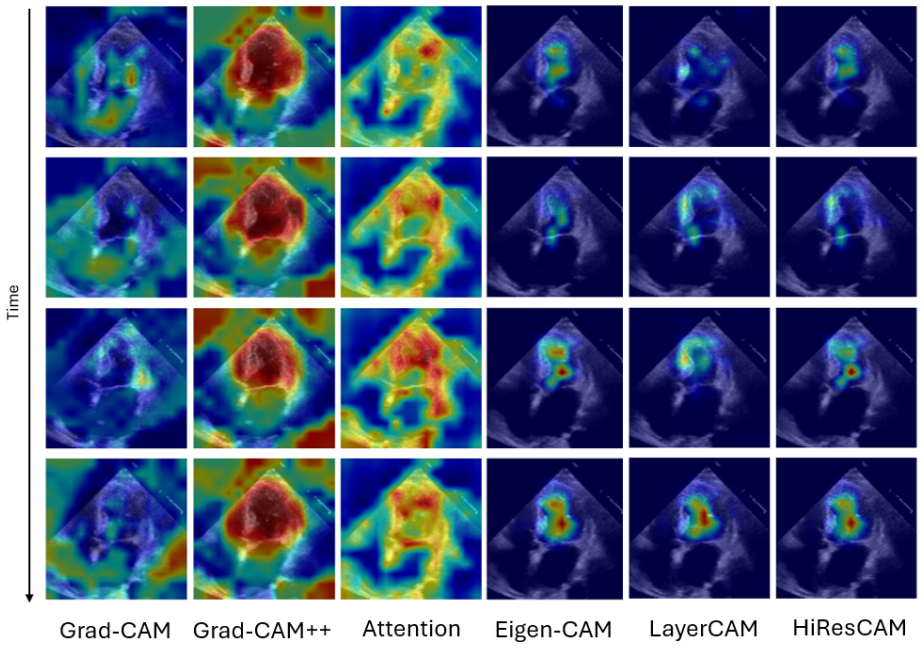


Figure 6.3: Figure showing examples of 4 frames of a single input video passed to the DGT Loss + GM (224) model where the activation map of the corresponding technique has been overlayed. Attention refers to attention rollout.

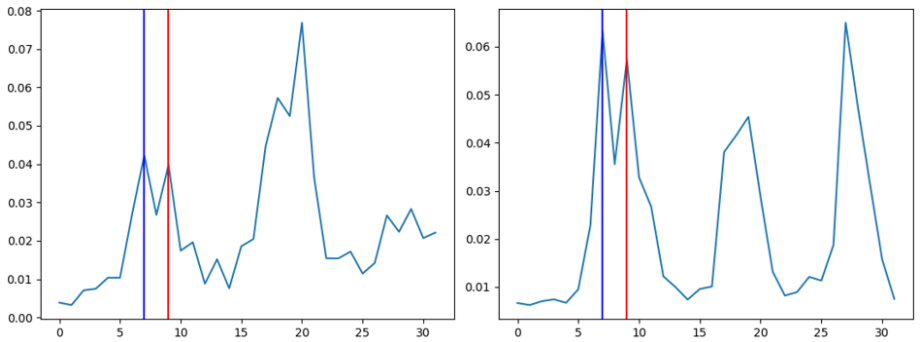


Figure 6.4: Figure showing the attention scores (y-axis) given to the 32 frames (x-axis) by the CLS token in the temporal module for the baseline model (left) and the DGT Loss + GM (112) model (right) for the same input. The blue and red lines indicate the ES and ED frame indices.

Reynaud et al. [29], although guidelines advise operators to select the heart beat cycle with the largest change in ES and ED areas, in practice they will usually pick a large cycle, but not necessarily the largest. If this was the case for the data used, not only would training be negatively impacted, but the evaluation carried out against the human rater labels may falsely penalise the model for accurately detecting ES and ED locations that may have been mislabelled by the human labellers, thus leading to lower metrics.

The training of the model was also negatively impacted by the absence of ED and ES frames after sampling, which meant that it was impossible for the model to accurately calculate the LVEF. It also meant that our DGT loss function could not be applied to facilitate training for such instances. As mentioned in the Other Experimentation section, although attempts were made to retain both ES and ED frames after sampling, the experiments resulted in the model achieving worse results, possibly due to the mismatch in the format of the training and validation data.

However, the biggest limitation was the lack of GPU compute available. This meant that our best model was not optimised, leading to non-optimal performance. It also meant that observing the changes in the original GEMTrans model when trained with the proposed DGT loss was also not possible.

# CHAPTER 7

# Conclusion

To conclude, we have shown that our proposed novel DGT Loss improved the model performance significantly, and that GridMask data augmentation can be a suitable method in this domain to reduce overfitting and improve model performance on validation and test sets. We also quantitatively evaluated different SOTA CAM-based techniques and attention rollout, concluding that Eigen-CAM provided the best results both quantitatively and by visual inspection of individual results which showed that the model focused on the LV region before predicting the LVEF value. We also showed that as well as visualising the spatial module, attention visualisation could be applied to the temporal module of our model, thus facilitating the visual explanation of all key sub-modules on each input.

Additionally, the CAM-based techniques can be applied to any CNN architecture and any vision transformer network, provided the outputs in one of the later layers can be reshaped to represent the features maps of a CNN, thus making our research beneficial for LVEF studies as these are the most common architectures used.

Overall, we demonstrated that the use of data augmentation, attention supervision, and explainability can improve our spatio-temporal video transformer for predicting LVEF in Echocardiography.

## 7.1 Future Work

A modified version of the GEMTrans model was used to reduce the model size and to enable experimentation using the proposed methods with the limited GPU compute available. Future work should explore applying the proposed changes to the original GEMTrans model (assuming the model weights and hyperparameters will be provided) as the DGT Loss, GridMask data augmentation and all explainability methods used are transferable to the original model.

Additionally, if more GPU compute was available, our best model could have been hyperparameter tuned to achieve higher results, possibly competing with

or outperforming methods like GEMTrans but with a much smaller model size and consequently fewer training iterations.

Future work should also explore improving the DGT loss. A possible limitation of our loss is that it penalises potential ES and ED frames that are not labelled in the same way as those furthest away from the labelled ES and ED frames. Instead, an ideal loss function should penalise the model less for paying attention to such frames, and potentially even encourage it to give higher attention scores, as these could contribute to a more accurate calculation of the LVEF. Although we tried to address this using the mirror sampling technique as discussed previously, our approach resulted in poor performance.

# CHAPTER 8

# Appendix

## 8.1 GitLab

Code available at: *https://git.cs.bham.ac.uk/projects-2023-24/jxs1740*. As stated in the README.md file:

- Download the dataset from *https://echonet.github.io/dynamic/index.html*
- Download the pre-trained ViT from *https://github.com/lukemelas/PyTorch-Pretrained-ViT/releases/tag/0.0.2*
- Then update the paths in default.yml as per README.md

The src folder contains the code needed to train and test the model. The configs folder contains the default.yml file which defines the configurations including the key hyperparameters. The sweep folder contains the configuration file to run a wandb sweep. The code to evaluate the explainability techniques is provided in evaluate_gradcam.py (with the corresponding output in cam_results.csv). The CAM outputs of the best model can be seen in the activation_map_outputs folder. A more detailed overview of the structure is provided in the README.md.

Our best model path is available at:
*https://drive.google.com/file/d/1HWaEpXPIzx_vqpXYBRFFnGhACnGR9Sxd/view?usp=sharing*

Run the following using Python 3.9.0 to install the dependencies:

```
>> pip install torch==1.12.1+cu113 torchvision==0.13.1+cu113
torchaudio==0.12.1 --extra-index-url https://download.pytorch.org/whl/cu113
>> pip install -r requirements.txt
```

Then run the following to evaluate the model (or train without the –test flag):

```
>> python run.py --config_path <path_to_training_config> --save_dir
<dir_to_save_ouput_to> --test
```

# Bibliography


[1] S. Abnar and W. Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[3] A. Baevski and M. Auli. Adaptive input representations for neural language modeling. *arXiv preprint arXiv:1809.10853*, 2018.

[4] L. Biewald et al. Experiment tracking with weights and biases. 2020.

[5] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[6] P. Chen, S. Liu, H. Zhao, and J. Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[7] D. Chicco, M. J. Warrens, and G. Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623, 2021.

[8] S. Desai and H. G. Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 972–980, 2020.

[9] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

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

[11] R. L. Draelos and L. Carin. Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks. *arXiv preprint arXiv:2011.08891*, 2020.

[12] L. Fazry, A. Haryono, N. K. Nissa, N. M. Hirzi, M. F. Rachmadi, W. Jatmiko, et al. Hierarchical vision transformers for cardiac ejection fraction estimation. In *2022 7th International Workshop on Big Data and Information Security (IWBIS)*, pages 39–44. IEEE, 2022.

[13] J. Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

[14] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021.

[15] A. Kosaraju, A. Goyal, Y. Grigorova, and A. N. Makaryus. *Left Ventricular Ejection Fraction*. StatPearls Publishing, Treasure Island (FL), 2023.

[16] S. Leclerc, E. Smistad, J. Pedrosa, A. Østvik, F. Cervenansky, F. Espinosa, T. Espeland, E. A. R. Berg, P.-M. Jodoin, T. Grenier, et al. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. *IEEE transactions on medical imaging*, 38(9):2198–2210, 2019.

[17] K. Li, Y. Wang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:2201.04676*, 2022.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[19] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[20] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[21] F. A. Maani, N. Saeed, A. Matsun, and M. Yaqub. Coreecho: Continuous representation learning for 2d+ time echocardiography analysis. *arXiv preprint arXiv:2403.10164*, 2024.

[22] M. Mokhtari, N. Ahmadi, T. S. Tsang, P. Abolmaesumi, and R. Liao. Gemtrans: A general, echocardiography-based, multi-level transformer framework for cardiovascular diagnosis. In *International Workshop on Machine Learning in Medical Imaging*, pages 1–10. Springer, 2023.

[23] M. Mokhtari, T. Tsang, P. Abolmaesumi, and R. Liao. Echognn: explainable ejection fraction estimation with graph neural networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 360–369. Springer, 2022.

[24] M. B. Muhammad and M. Yeasin. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2020.

[25] R. Muhtaseb and M. Yaqub. Echocotr: Estimation of the left ventricular ejection fraction from spatiotemporal echocardiography. In L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, pages 370–379, Cham, 2022. Springer Nature Switzerland.

[26] S. Nazir, D. M. Dickson, and M. U. Akram. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. *Computers in Biology and Medicine*, 156:106668, 2023.

[27] D. Ouyang, B. He, A. Ghorbani, N. Yuan, J. Ebinger, C. P. Langlotz, P. A. Heidenreich, R. A. Harrington, D. H. Liang, E. A. Ashley, et al. Video-based ai for beat-to-beat assessment of cardiac function. *Nature*, 580(7802):252–256, 2020.

[28] 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. *Journal of machine learning research*, 21(140):1–67, 2020.

[29] H. Reynaud, A. Vlontzos, B. Hou, A. Beqiri, P. Leeson, and B. Kainz. Ultrasound video transformers for cardiac ejection fraction estimation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VI 24*, pages 495–505. Springer, 2021.

[30] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[31] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[32] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[33] K. K. Singh, H. Yu, A. Sarmasi, G. Pradeep, and Y. J. Lee. Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond. *arXiv preprint arXiv:1811.02545*, 2018.

[34] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms, 2012.

[35] S. Thomas, A. Gilbert, and G. Ben-Yosef. Light-weight spatio-temporal graphs for segmentation and ejection fraction prediction in cardiac ultrasound. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 380–390. Springer, 2022.

[36] A. Vaswani. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[37] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.

[38] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

[39] S. Wright. Correlation and causation. *Journal of agricultural research*, 20(7):557, 1921.

[40] J. Yang, X. Ding, Z. Zheng, X. Xu, and X. Li. Graphecho: Graph-driven unsupervised domain adaptation for echocardiogram video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11878–11887, 2023.

[41] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.