

# Modelling and Prediction of Cardiac Electrophysiology Signals

Tuan Truong, Tommy Peng, Mark Trew, Avinash Malik

**Abstract**—The reconstruction of Heart Surface Potential (HSP) from Body Surface Potential (BSP) has long been known as the inverse problem in electrocardiography. There have been efforts in reconstructing the HSP using numerical solution. However, there still exists limitation within this approach mainly due the ill-posedness of the inverse problem. Recently machine learning techniques have gained momentum for successfully solving many inverse problem in imaging [5][7] and show potential in tackling inverse problem in electrocardiography [8]. This project aims to take the machine learning approach to reconstruct HSP and explore the relationship between positions of sock and electrodes for a better reconstruction of HSP. Results show that our approach is able to capture well HSPs with low frequency components and gain insights into the learning of our sequence model Long Short-term Memory (LSTM) [4].

## I. INTRODUCTION

### A. Background

For the last century, the existence of electrocardiogram (ECG) has involved significantly in cardiac diagnosis and treatment by allowing practitioners to record and observe the electrical changes within the heart. In recording the heart electrical activity, the standard 12-lead ECG is a common practice that supports the view of electrical changes in 12 angles. Despite its common use, the 12-lead ECG provides limited spatial information and cannot reflect well the localization of abnormal electrical events [2]. To have a complete picture of electrical activity, additional electrodes are placed on the body to record potentials from all parts of the body. This practice is known as *Body Surface Potential Mapping* (BSPM). BSPM is useful in observing the signals from a broad surface by forming a 3D map describing the conduction of electricity. However, the signals coming from BSPM are the results of filtering and smoothing effects by body tissue, and hence do not well record the true electrical activity inside the heart.

A better understanding to cardiac electrophysiology requires an insert of catheter inside the body to measure directly electricity coming from the heart. However, this procedure is invasive and risky to the patient. A solution to this problem is to derive the mapping of HSP from BSP, which introduces no threat to a patient by invasive surgery. The mapping of HSP from BSP is collectively known as the inverse problem in electrocardiography. The inverse problem is mathematically formulated in equation 1 where  $\Phi_H$  and  $\Phi_B$  are respectively the vector of HSP and BSP.  $H$  is the transfer function that governs the transformation from HSP to BSP in the forward problem. An estimation of  $H$  would yield the numerical solution to the inverse problem. However, the inverse problem is long known to be ill-posed, meaning even a small amount

of noise in BSP can lead to huge errors in reconstructing HSP.

$$\Phi_H = H^{-1} \times \Phi_B \quad (1)$$

### B. Previous works

Besides numerical solution, there are few works using machine learning to reconstruct HSP from BSP. One of the most recent work is Malik et al. [8] who employs the *Time Delay Neural Network* (TDNN) to reconstruct HSP with an average Pearson Correlation Coefficient (PCC) of 0.7 for 3 patient. However, the architecture of TDNN allows only one signal to be predicted at a time, limiting the capability to predict many sock signals as well as observing the relationship between HSP and BSP. Another noted work by Ghimire et al. [3] also investigates the inverse solution for cardiac transmembrane potential using LSTM encoder-decoder network with constrained stochasticity and global aggregation of temporal information in the latent space. The data used in this work is simulated, thereby hardly reflecting the true state of real world data.

This leads to our contributions:

- 1) A LSTM model that learns to map between multiple BSPs (or ECGs) to HSPs (or EGMs)
- 2) An analysis on the learning of LSTM model
- 3) A proposed method in find the critical BSPs for reconstructing HSPs.

### C. Dataset

The dataset is kindly supported to us by Laura Bear at the Electrophysiology and Heart Modelling Institute, University of Bordeaux [1]. The experiment from which the data is obtained is set up with a porcine heart immersed in a body-shaped tank. There are 108 socks put over the porcine heart to measure the HSP whereas 128 electrodes are placed upon the tank to record BSP. The heart is stimulated from 3 pacings: Left Ventricular (LV), Right Ventricular (RV) and Bi-Ventricular (BiV) pacing. A special case (Left Bundle Branch Block) is also induced to record the electrical activity of a heartbeat abnormality. Together, the dataset consists of 4 sets of HSP-BSP from 4 so-called pacings. As instructed by our supervisors, the average beat signals were used in all experiments in this project. Throughout this report, we will use the name EGM instead of HSP and ECG instead of BSP for convenience.

## II. BUILDING LSTM MODEL FOR EGM RECONSTRUCTION

### A. Overview

This experiment investigates the capability of deep learning model in prediction of EGMs from ECGs. We explore this

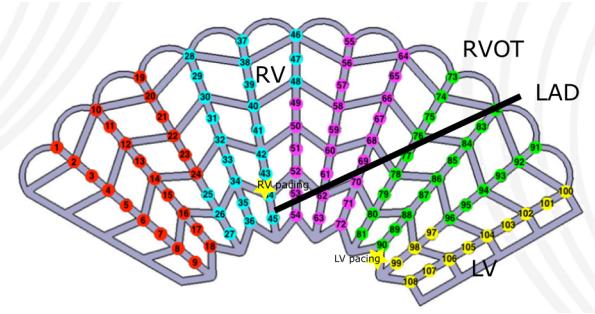


Fig. 1. The coverage of socks in the experiment set up of [1]

possibility by introducing a LSTM model for sequence prediction that takes 123 ECGs as inputs and returns 107 EGMs as outputs. The structure for this section includes the following parts:

- Data Cleaning: steps to clean and preprocess the data
- Neural Network for Sequence Modelling: a brief introduction to the model being used
- Experimental Setup: details of the architecture and parameters of the training model
- Result: result obtained from training the model

### B. Data Cleaning

The data available is the recorded sets of ECGs and EGMs from 3 pacings (LV,RV,BiV) and LBBB (the case of disease). For each set, there are 128 ECGs and 108 EGMs. However, not all 128 ECGs and 108 EGMs are usable as there are some broken leads which is defined in the data description. As a result, the first step in the data cleaning is to remove the all broken leads from the dataset that will be used as for future training and validation. The broken lead indices are shown in Table I. After removing, there will be 123 ECGs and 107 EGMs left in the dataset. Each ECG and EGM signal are then smoothed using moving average with a window of 5 samples and detrended before being fed into the model for training and validation

TABLE I  
BROKEN LEAD INDICES

EGM	1
ECG	44,61,62,124,125

### C. Neural Network for Sequence Modelling

Previous work [8] on this problem chose the Time-Delay Neural Network (TDNN) and Feed Forward Neural Network (FFNN) for prediction from ECGs to EGMs. One main disadvantage of these models is that it is too shallow to capture sufficient information from ECG for EGM prediction. In addition, both of these models are not well suited for many-to-many reconstruction problem in time series. Even though FFNN in some ways can be used for many-to-many reconstruction, the fact that each neuron in FFNN working independently from one another weakens the learning of temporal relation in time series. Hence, this experiment investigates a different neural

network known as Recurrent Neural Network (RNN) with its variant Long-Short Term Memory (LSTM). The subsections below introduce the concepts of RNN and LSTM in particular for sequence modelling.

1) *RNN*: RNN is a variant of Neural Network for sequence modelling task. What is special about RNN is its ability to remember information across every step in the sequence and can later retrieve this memory for specific tasks such as classification of the sequence into some categories or prediction of another sequence. The memory of RNN is possible by using a structure called cell. Fig.2 is a visualization of one RNN in the rolled (left) and unrolled (right) version. Like the usual Neural Network, each RNN cell has an input  $x_t$  and an output  $y_t$ . Inside each cell, there is an internal input denoted by  $h_t$ , which is referred as the *hidden state*. The hidden state  $h_t$  stores the information computed from  $x_t$  and  $h_{t-1}$  and that is why it acts as memory storage keeping information at every time step. Given  $h_t$ , the output at the last step, which is computed as a linear combination of  $h_t$  and a weight matrix, will have all information from previous steps.

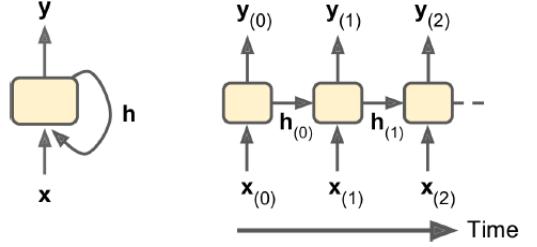


Fig. 2. The structure of an unrolled RNN cell. Source: Scency Stuff

2) *LSTM*: One disadvantage of RNN is that its memory does not last long. For long sequences, by the time RNN processes to the last step, it will have forgot most of information from the beginning. For such reason, LSTM is a variant of RNN that is capable of dealing with long-term dependencies. Besides the normal *hidden state*, LSTM has a *cell state*  $c_t$  that retains only critical information and forgets what is irrelevant. This capability is enabled by 3 gates governed by 4 weight matrices that allow what amount of information will be kept and forgot. The 3 gates are *forget gate*, *update gate*, and *output gate*. The gates are basically a sigmoid function (denoted by  $\sigma$  in Fig 3) which outputs values between 0 and 1. An output of 1 means keeping all information while an output of 0 refers to forgetting all of it. The *forget gate* specifies how much of old information in  $c_{t-1}$  should be forgot in the new cell state at  $c_t$  while the *update gate* at the same time defines how much information in the temporary cell state to be updated at  $c_t$ . In the end, *output gate* limits the amount of information to be output. Together, these three gates retain only information that is critical across a very long sequence even though the memory capability is limited. In other words, the name *long-short term memory* means that even though the memory of LSTM is short, it can still keep important information across long sequences by selecting what is worth keeping and forgetting the rest.

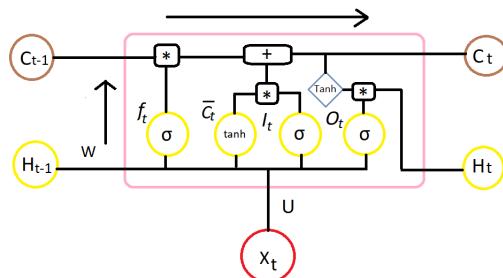


Fig. 3. The structure of a LSTM cell. Source: Medium

#### D. Experimental Setup

1) *Architecture*: Given the advantages of LSTM in sequence modelling, an LSTM model is chosen for this experiment. After a parameter search, the architecture of the LSTM is defined in Table II.

TABLE II  
ARCHITECTURE OF THE CHOSEN MODEL

Layer No.	Type of Layer	Dimension
1	Input Layer	num-examples $\times$ 123 $\times$ 700
2	LSTM Layer	num-examples $\times$ 70 $\times$ 700
3	Dense Layer	num-examples $\times$ 107 $\times$ 700
4	Regression Layer	

The input to the model is a cell array in Matlab where *num-examples* refers to how many examples (ECG-EGM pairs) to train. The dataset here contains only 4 examples from LV, RV and BiV pacings and LBBB, so three of these will be used for training and the last one will be saved for validation, which will be detailed in later section. 123 and 107 are respectively the number of ECGs and EGMs within each examples and 700 is sequence length after being limited for all examples.

Other selection of parameters is shown in Table III. These parameters have been tuned to get the best result. The Batch-Size of 3 means that the weights will be updated after all 3 examples having been passed and the updated amount is proportional to the average loss of 3 examples. Gradient threshold is set to 1 to prevent gradient exploding, which means all gradients larger than 1 will be cut off to 1. In addition, a L2 Regularization with  $\lambda = 0.0001$  is chosen to avoid overfitting.

TABLE III  
OTHER PARAMETERS

Learning Rate	0.005
MaxEpochs	500
BatchSize	3
Gradient Threshold	1
L2 Regularization	0.0001

2) *Cross Validation*: To have a proper evaluation of the performance of the model, cross-validation is designed to train on three examples and test on the rest one. For the moment, LBBB is excluded from testing but is still in the training. The detail is shown in Table IV.

TABLE IV  
CROSS-VALIDATION

Fold	Training Set	Validation Set
1	RV, BiV, LBBB	LV
2	LV, BiV, LBBB	RV
3	LV, RV, LBBB	BiV

#### E. Result

Pearson Correlation Coefficient (PCC) is used to evaluate the predicted EGMs and true EGMs. For each example, the average PCC is calculated across 107 EGMs as following:

$$\frac{1}{\text{num\_EGMs}} \times \sum_{i=1}^{\text{num\_EGMs}} \text{Corr}(\text{EGM}_{\text{pred},i}, \text{EGM}_i) \quad (2)$$

where *Corr* is the PCC function. The standard deviation (std) is also calculated across 107 coefficients. Table V shows result obtained from the training set. **Training Set Average** is the average of the average coefficients across 3 examples and **Training Set Std** is the average standard deviation across 3 examples. Table VI and Fig. 4 shows the validation result. As can be seen from Table VI, the average coefficient is much lower compared to the training set. The higher standard deviation also suggests that there are large gaps among reconstruction capability across all EGMs. As a result, further investigation is done to gain more insights into the learning process.

TABLE V  
CROSS-VALIDATION RESULT IN TRAINING SET

Fold	Training Set Average	Training Set Std
1	0.9268	0.0921
2	0.9054	0.1035
3	0.9138	0.0964

TABLE VI  
CROSS-VALIDATION RESULT IN VALIDATION SET

Fold	Validation Set Average	Validation Set Std
1 (LV)	0.5321	0.3229
2 (RV)	0.6140	0.2626
3 (BiV)	0.6561	0.2596

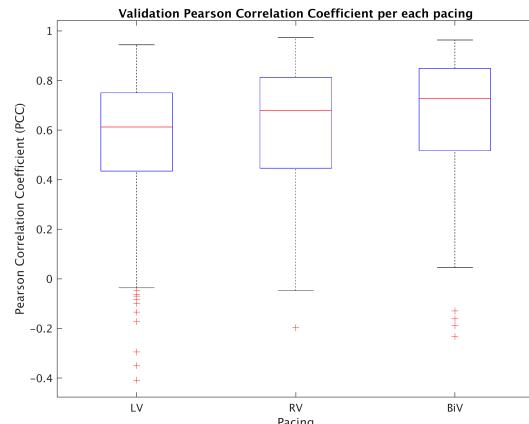


Fig. 4. Cross-Validation Pearson Correlation Coefficient per pacing

### III. UNDERSTANDING THE LEARNING OF LSTM

#### A. Overview

This section investigates the learning of the network to better understand the likely cause of performance drop in the above experiment. The first part of this section visualizes the learning weights of the network with an aim to explain the gap between training and validation evaluation. The next part investigates the success of reconstructing 107 EGMs by visualizing the coefficients in a heat map.

#### B. Weight Visualization

As this is a small size dataset, training with leave-one-out cross-validation is expected not to cause a significant change in weight distribution between folds. However, when switching to validation on a pacing that has been included in the training set of the previous trained fold, the average coefficient of reconstructed EGMs of that pacing drops by almost half. This fact suggests the weights between two different trainings might not be the same. To investigate this possibility, a small-scale experiment is conducted to map 123 ECGs to only the first 10 EGMs. After training, the weights in the LSTM layer is visualized in a scatter plot to see if there exists correlation between 2 sets of weights. In this experiment, LBBB is included during both training and validation.

Fig. 5 and Fig. 6 show the weight distribution between 2 consecutive training folds. It can be clearly seen that the weights after two trainings are not correlated, confirming the possibility of significant weight change. To ensure what actually changes, a histogram is plotted for weight distribution as shown in Fig. 7. Even though the range of values in those weight does not vary much, the number of values in the middle, especially near 0 changes a lot between 2 trainings. The change in the number of 0s can damage the learning of such network as it indicates many neurons are unusable.

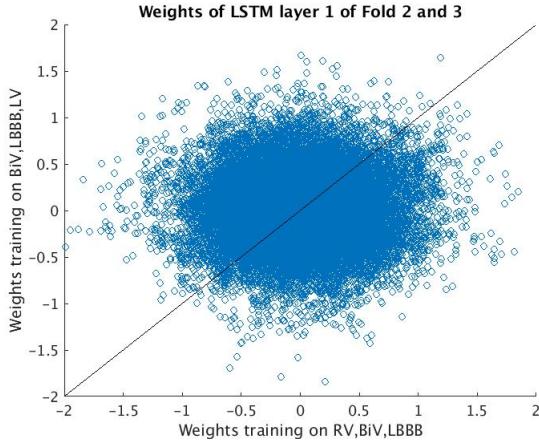


Fig. 5. Weights between training on RV-BiV-LBBB and BiV-LBBB-LV

#### C. Heat map of PCCs

The high standard deviation across all reconstructed EGMs implies that some EGMs are better reconstructed than the

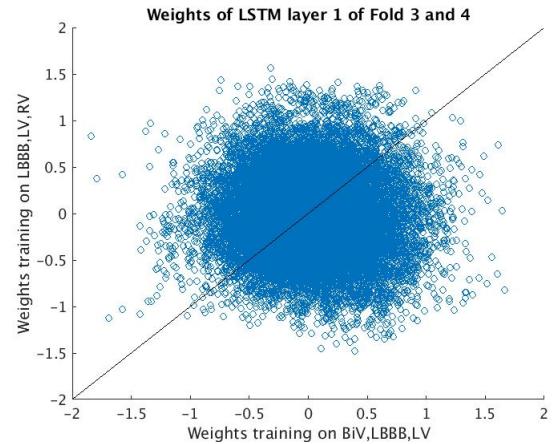


Fig. 6. Weights between training on BiV-LBBB-LV and LBBB-LV-RV

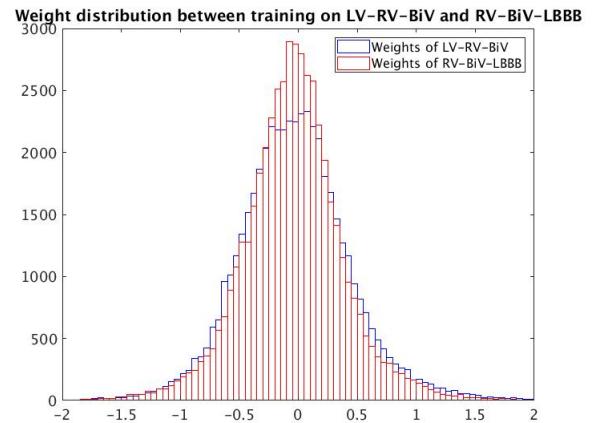


Fig. 7. Weight distribution of training on LV-RV-BiV and RV-BiV-LBBB

others. Fig. 8, 9, 10 show PCCs across 108 EGMs from each pacing. The first EGM is broken so the coefficient is 0. All reconstructed EGM coefficients are arranged based on the indices in the sock coverage (Fig. 1). These heat maps show that poor reconstructed EGMs usually localize from column 4 to 9. For each pacing, the localization of reconstructed EGM with low coefficient is not the same. One hypothesis for this could be that the activation time of those true EGMs are different. We visualize the activation time of true EGM (Appendix B and see no relation between PCC and activation time).

### IV. FINDING ECG USEFULNESS

#### A. Overview

In medical perspective it is worth knowing which ECG plays the most role in reconstructing the signals so that doctors can place the electrodes at those positions for a better prediction of EGM signals. In our neural network, the relationship between ECGs and EGMs can be expressed by the weights connecting them. This section aims to use these weights to indicate which ECG matters the most to the sock construction at a specific location in the heart.

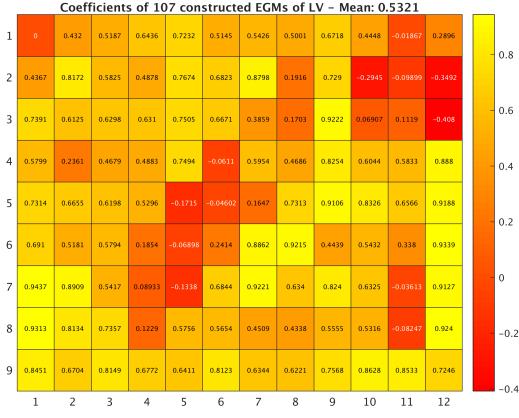


Fig. 8. Pearson Correlation Coefficients of 107 reconstructed EGMs from LV pacing

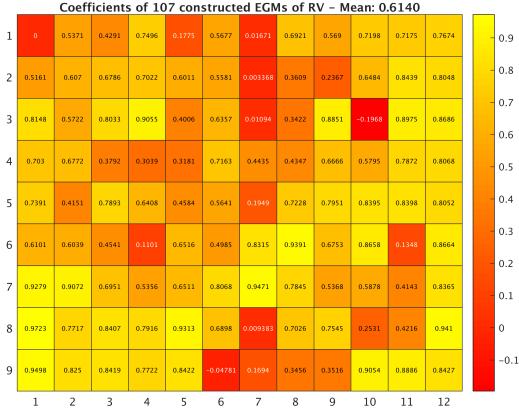


Fig. 9. Pearson Correlation Coefficients of 107 reconstructed EGMs from RV pacing

## B. Background

In machine learning, neural network belongs to the group of algorithms that are the most difficult for interpretation of their decision. Even though neural network gains significant reputation recently for their high accuracy, interpretation of its decision is still an active research with many works attempting to visualize its decision with different techniques. In image analysis domain, there are two main approaches in visualizing the saliency in neural network, especially in Convolutional Neural Network. The first approach tries to derive the gradients flowing from an activation layer to the output. This approach is taken by [14] who first introduces class activation mapping with a global average pooling layer. However [14] can only apply for the penultimate output layer while there is a great need for visualizing any activation layer. A generalization of class activation mapping is called gradient-based class activation mapping [9], which uses backpropagation to find the gradient of output w.r.t any activation layer. The second approach aims to monitor the change classification score by perturbing the input and observing the output [13]. This approach is known as *Occlusion* according to [11].

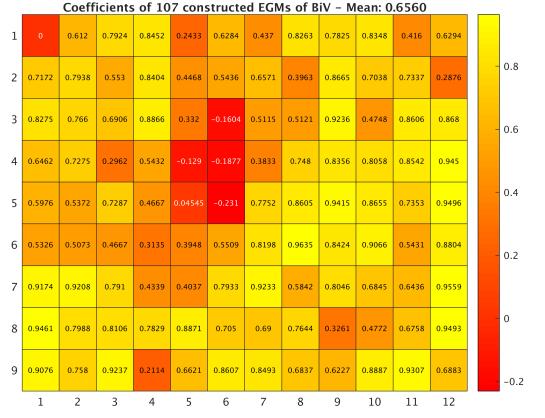


Fig. 10. Pearson Correlation Coefficients of 107 reconstructed EGMs from BiV pacing

While there are many approaches in image domain, there are currently not many techniques in visualizing decision made by sequence model. The first attempt is made by [6] in the natural language processing field. Other attempts also target at language models by introducing attention to the network during training [12]. In medical signal domain, there are limited works exploring the explanation of LSTM decision given the abundant applications of LSTM in medical signal analysis. Currently [11] and [10] give an analysis of techniques in visualizing LSTM decision in medical signals. However, all of the given techniques favor the classification rather than the regression task, thus there is a need for modification of these techniques to be compatible for regression problems.

## C. Method

Our approach is similar to the gradient-based class activation mapping method [9]. Unlike above approaches where the final class score  $S_c$  (i.e. the score before softmax or sigmoid activation) is computed for classification purpose, our problem is a regression one, which no additional class score will be calculated after the linear activation. Given this, instead of taking gradients of  $S_c$  with respect to a certain layer, our approach takes the gradient of value in each time step of a predicted EGM with respect to the input layer. Let  $X$  and  $y$  represent our input and output. The dimension of  $X$  and  $y$  are respectively  $[M \times T]$  and  $[N \times T]$  where  $M$  and  $N$  are the number of ECGs and EGMs and  $T$  is number of samples or time steps. The  $y_i$  refers to the  $i^{th}$  row of output  $y$ . Instead of computing each  $S_c$  w.r.t the input, we compute  $\frac{\partial y_i}{\partial X}$  where each time step in  $y_i$  can be treated in the same way as  $S_c$ .

$$\frac{1}{T} \sum_{i=1}^N \sum_{j=1}^T \frac{\partial y_i}{\partial X} \quad (3)$$

While all the above methods aim to visualize the saliency in time steps, in our approach we just look for the important ECGs, thus taking the average across the time steps is sensible as it is a rough estimate of how well each ECG can contribute to the reconstruction. Finding the gradient  $\frac{\partial y_i}{\partial X}$  can be made by

using backpropagation from the output layer to LSTM layer and then to the input as

$$\frac{\partial y_i}{\partial X} = \frac{\partial y_i}{\partial y_{LSTM}} \times \frac{\partial y_{LSTM}}{\partial X} \quad (4)$$

For the moment, the implementation of backpropagation in Matlab is not available, this gradient computation is done in Keras 2.2.2 and is then transferred back to Matlab for visualization purpose.

#### D. Result and Discussion

The result analysis is carried out in reconstructed EGMs which have high PCC compared with the ground truth. Given that, the sock 106 in BiV pacing set is chosen for visualization purpose. The output gradient is then rescaled to the range between -1 and 1 and visualized (Fig.11). As can be seen from the figure, the positive and negative gradients are localized, leading to our hypothesis that some electrodes have more influence than the others and gradient values near 0 should have no contribution at all. To test this hypothesis, the next step is performed similarly to the occlusion approach [11][13] such that some ECGs will be set to 0s to monitor the change in PCC.

In doing so for sock 106, the most positive gradient values show the highest impact that causes the PCC to drop quickly while values near 0 and negative have little effects on PCC. However, when extending the occlusion test to all EGMs, the behavior is different in that both negative and positive values can influence (i.e. increase or decrease) PCCs whereas values near 0 have almost no contribution to the change. We define an  $\epsilon = 0.2$  where values larger than  $\epsilon$  is defined as *large positive values* (case 1), smaller than  $-\epsilon$  as *large negative values* (case 2) and in  $[-\epsilon, \epsilon]$  as *values near 0* (case 3). These ranges of gradient values are defined on the condition that no rescaling between -1 and 1 is made. The p-value for each case is shown in Table VII. Given the resulting p-value less 0.05, the null hypothesis in case 1 and 3 is rejected, suggesting a difference in mean value during occlusion test and indicating the large influence of those ECGs on PCCs of reconstructed EGM.

TABLE VII  
P-VALUE FOR OCCLUSION TEST

Pacing	Case 1	Case 2	Case 3
LV	$4.7315e^{-17}$	0.3312	$7.3397e^{-7}$
RV	$2.8968e^{-17}$	0.1185	$4.4454e^{-10}$
BiV	$4.7726e^{-19}$	0.1441	$2.3131e^{-11}$

Apart from the general trend, some of the EGMs with fewer high frequency components (like sock 106) tend to be easier reconstructed and their PCCs are observed to increase slightly during occlusion test with removal of negative gradient values.

#### V. ACTIVATION TIME

As suggested by our supervisors, instead of evaluating the reconstruction of whole time series, comparing the activation time of predicted signals is also a measurement of how good our prediction is. In this experiment we first validate our technique using a new dataset from Laura and compare the

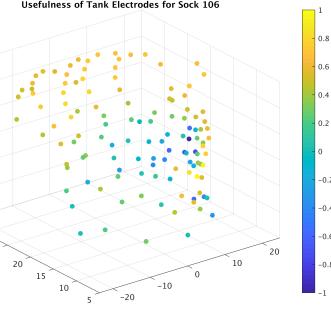


Fig. 11. Tank Electrode Usefulness for Sock 106 of BiV pacing

estimated activation time with her activation data. Then we use the trained model to predict on the new dataset and apply this technique for evaluating the activation time on the prediction.

#### A. Validation of Our Technique

We estimate the activation time of a signal by finding the point with most negative gradient ( $\frac{dV}{dt}$ ) in the signal. We apply this technique on Laura's new dataset of sock signals. In comparison with her sock activation data (Fig. 13 and Appendix C), the estimated activation time (Fig. 12 and Appendix A) is very close to, suggesting our technique is can be reliable in estimating activation time.

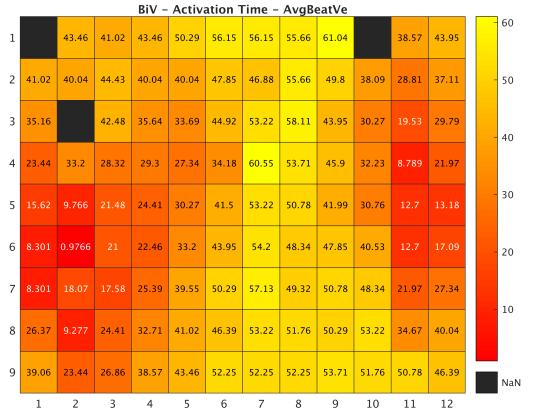


Fig. 12. Estimated activation time using our method

#### B. Activation time on predicted EGM

The new dataset is different from the old one in terms of the broken leads and shorter length of time steps. As a result, the trained network on old data cannot be applied directly on the new dataset. Given this, a new network with same architecture but with a change in input and output dimension is trained with the old dataset and retrieved later to predict the new one. The predicted signals are, however, less satisfactory in comparison to those of old data. Hence, the predicted activation time (Fig. 14 and Appendix D) has higher error compared to the ground truth (Fig. 13 and Appendix C).

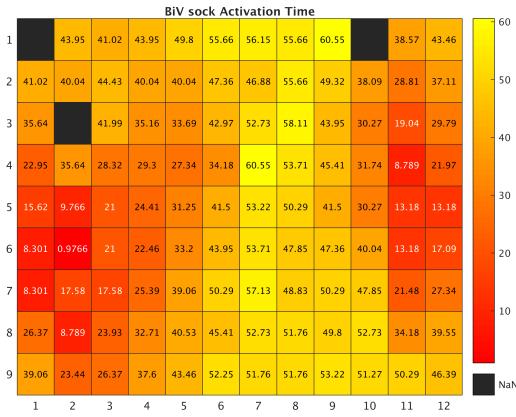


Fig. 13. Activation time taken from Laura's activation data

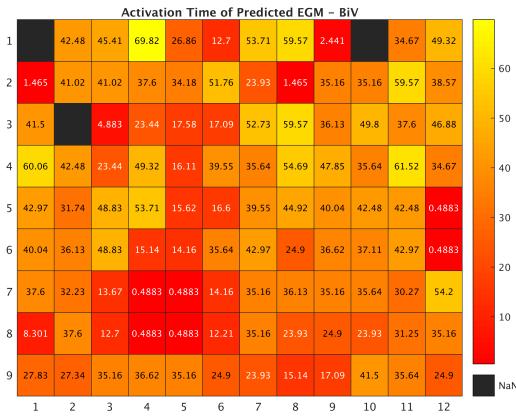


Fig. 14. Activation time from our predicted EGM

## VI. DISCUSSION

### A. On the prediction of EGM

1) *Overfitting:* It is no doubt that network suffers from overfitting with clear evidence of small loss for training and high loss for validation. As we treat each of ECG or EGM as 1 dimensional feature of input or output, this would build up roughly a hundred dimensions whereas there are only 4 examples for both training and validation. In addition, each of our signal has a length of 700 samples, which is typically longer than usual to be processed by the network. Given this and the fact our LSTM model learns and predicts step by step, the network is prone to overfitting as it only sees the current step and some steps before that for prediction but lacks the whole view of the entire series.

For such problem, an encoder-decoder model (Fig. 15) would be an option. An encoder would then learn the entire series of ECG and compress such knowledge into the hidden state of the decoder, which plays the role of predicting EGM. However training an encoder-decoder model for time series is still a major problem with few training techniques applied for medical time series. The most common training technique for encoder-decoder is teacher forcing, which uses

the forward step in the target as input for prediction of the current step during training with an aim to force the network to predict value that match with one in future time step. During inference, however, the network has to use its prediction as input for the next time step. We also experimented with this technique by training on our data but the reconstructed signals during inference soon arrive at a constant value in later time steps. This could be explained by a hypothesis that the network became lazier as it just relies on the input from the target and when left with just only its previous prediction as input, which is different from what it learns, the network simply does not know what to predict and starts losing track from the sequence.

If the encoder-decoder model is explored in further research, it is important to investigate appropriate training techniques, particularly in the decoder part. Our proposed solution is to use all output hidden states from the encoder as input for the decoder, which is more sensible since the decoder part can have access to all information at each time step of encoder instead of relying only on the last encoder output.

2) *The need of frequency analysis:* The prediction shows a strong pattern in which EGMs of low frequency components are easier to reconstruct than the others. A possible explanation for this is not yet to be found but intuitively signals of high frequency components seem generally more difficult to capture. Regarding this problem, learning simply on time series is a challenging task since most of the data is made of high frequency signals. This problem is actually what we try to avoid in the beginning of the research by introducing the Gaussian parameters to the network instead of time series. However, the Gaussian parameter approach itself still encounters the problem partly in optimizing the network to learn parameters within a tolerance. Another approach is to have a thorough analysis on the training with attention to frequency. Our proposed solution is to build a model with customized loss function that imposes a hard penalty on missing the peaks during prediction, which drives the network to pay more attention to time steps with high likelihood of seeing depolarization.

### B. On the understanding of network decision

In the above experiment we derive the usefulness of ECG using the gradient-based and occlusion approach for validation. Even though the result successfully shows which ECG the model pays more attention to, the explanation to why only those ECGs are important is still open. In addition, whether or not those important ECGs will change when tuning parameters or changing the network architecture remains as a problem. The finding of important ECGs that are invariant regardless of any changes of the network should be of higher priority as it provides a complete picture of ECG-EGM relationship that is helpful in selection of only some ECGs for EGM reconstruction. These issues require further validation of the proposed method as well as extending investigation to other techniques that help gain better insight into LSTM decision.

### C. Future work

The most important task for future work is to increase the size of dataset. Even if with the current size the network could

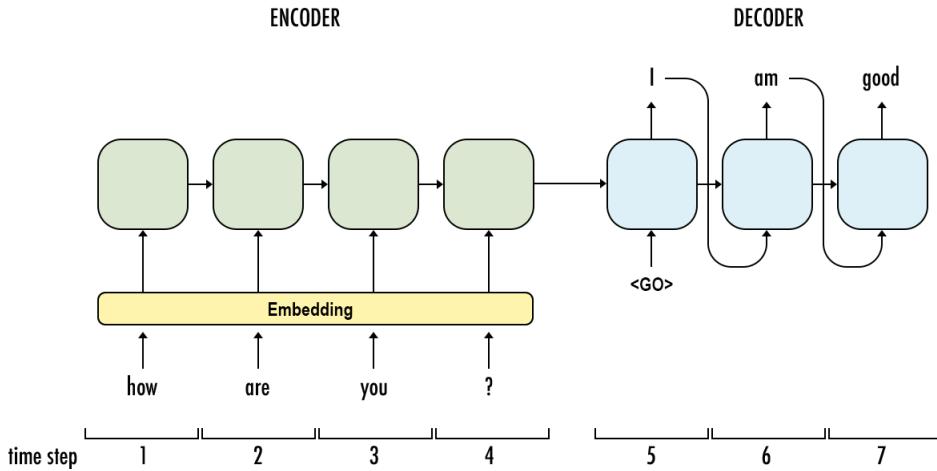


Fig. 15. A popular application of LSTM encoder-decoder model is in natural language processing. In the above figure, an encoder simply processes the question and feeds its output to the initial hidden state of the decoder. The decoder will generate the answer using all of knowledge it has from the encoder. In this decoder, the prediction at one time step is the input of the next time step. This technique is known as teacher forcing. Source: Toward Data Science

arrive at satisfactory result, it would be still less meaningful to jump at any conclusion about its generalization. A good and large enough dataset would not only enable the learning of the network but also provide a better intuition to how the network learns. Increasing the dataset size does not necessarily mean adding more data. We can increase the size by learning across the signals at different slices, thus driving the network to look at details of various sequence lengths and increasing its robustness. Conventional techniques for data augmentation like shifting and randomly perturbing the amplitude of signals are some other options to introduce variance to the network.

In addition, seeking for feature representation instead of time series approach is of high recommendation. As discussed in the above section, a time series approach poses complication to the network toward finding a solution. A feature extraction method can be implemented before a classifier to find the appropriate representation. In such case, a Convolutional Neural Network (CNN) could be an option to try on. We can use CNN as a decoder to extract feature before feeding these feature into a LSTM decoder. A disadvantage of CNN, however, is that it introduces plenty of parameters and can lead to overfitting in case of small dataset. Other engineering feature extraction methods like transforming to spectrogram are also a good way of learning representation. By transforming to a spectrogram, we can treat our problem with other image-based analysis techniques and simultaneously observe the effect of frequency as mentioned in the above section.

## VII. CONCLUSION

In this report we introduce the possibility of solving the inverse problem in electrocardiography using LSTM model. Preliminary results show that our model is capable of reconstructing the HSP from BSP but future work is needed to validate and gain more insights into network decision. Our technique as well as other machine learning techniques show enormous potential in solving the inverse problem and can be used as an alternative to numerical method. However, one

limitation of neural network in general is the interpretation of its decision, which makes it as a black box especially to practitioners. Hence our future work not only includes building a network with accurate prediction but also expands into interpreting the network for clinical applications.

## VIII. ACKNOWLEDGEMENT

This work could not have been done without the dedicated supervision of Avinash Malik, Mark Trew and Tommy Peng. Their guidance and knowledge were extremely helpful, especially in working to understand of how the network arrives at its decision. I would also like to thank the Auckland Bioengineering Institute for the use of remote server that enables training of neural network much faster. My greatest gratitude is for the University of Auckland. The awarding of Summer Research Scholarship has been a great opportunity for me to work in a new but interesting project and learn useful knowledge and skills that will be of great benefit to me in future research.

## REFERENCES

- [1] Laura R. Bear et al. "Cardiac electrical dyssynchrony is accurately detected by noninvasive electrocardiographic imaging". In: *Heart Rhythm* (2018).
- [2] Laura Bear et al. *Introduction to noninvasive cardiac mapping*. 2015. DOI: 10.1016/j.ccep.2014.11.015.
- [3] Sandesh Ghimire et al. "Improving Generalization of Sequence Encoder-Decoder Networks for Inverse Imaging of Cardiac Transmembrane Potential". In: (2018). arXiv: 1810.05713.
- [4] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* (1997).
- [5] Kyong Hwan Jin et al. "Deep Convolutional Neural Network for Inverse Problems in Imaging". In: *IEEE Transactions on Image Processing* (2017).

- [6] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. “Visualizing and Understanding Recurrent Networks.” In: *CoRR* abs/1506.02078 (2015).
- [7] Alice Lucas et al. “Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods”. In: *IEEE Signal Processing Magazine* (2018).
- [8] Avinash Malik, Tommy Peng, and Mark Trew. “A machine learning approach to reconstruction of heart surface potentials from body surface potentials”. In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2018).
- [9] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017. arXiv: 1610.02391.
- [10] Jos van der Westhuizen and Joan Lasenby. “Techniques for visualizing LSTMs applied to electrocardiograms”. In: (2017). arXiv: 1705.08153.
- [11] Jos van der Westhuizen and Joan Lasenby. “Visualizing LSTM decisions”. In: (2017). arXiv: 1705.08153. URL: <http://arxiv.org/abs/1705.08153>.
- [12] Kelvin Xu et al. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 2048–2057.
- [13] Matthew D. Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2014. arXiv: 1311.2901.
- [14] B. Zhou et al. “Learning Deep Features for Discriminative Localization.” In: *CVPR* (2016).

## APPENDIX

### A. Estimated Activation Time for new average beat data (Exp 16)

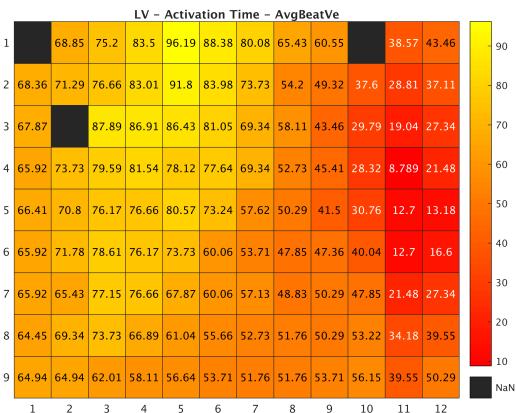


Fig. 16. Estimated activation time using our method - LV pacing

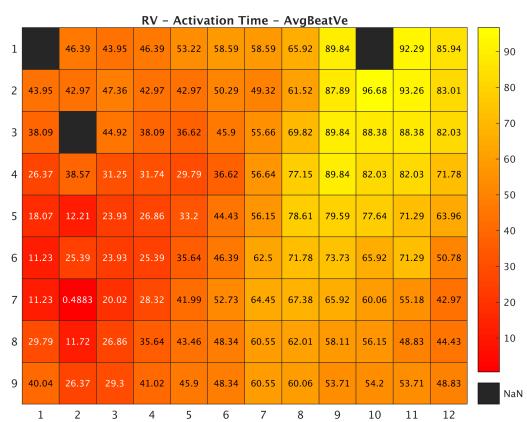


Fig. 17. Estimated activation time using our method - RV pacing

### B. Estimated Activation Time from Laura’s old average beat data (Exp 20)

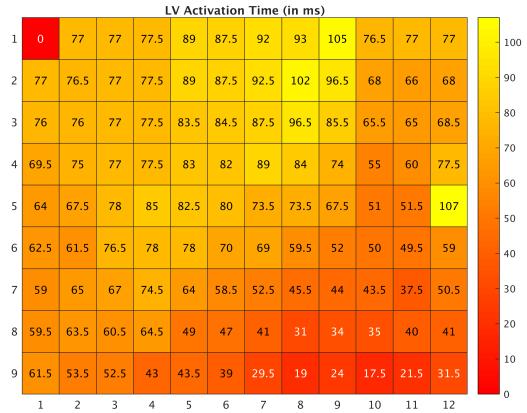


Fig. 18. Estimated activation time using our method - LV pacing

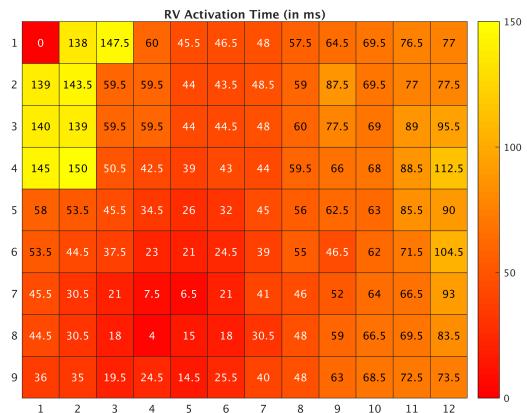


Fig. 19. Estimated activation time using our method - RV pacing

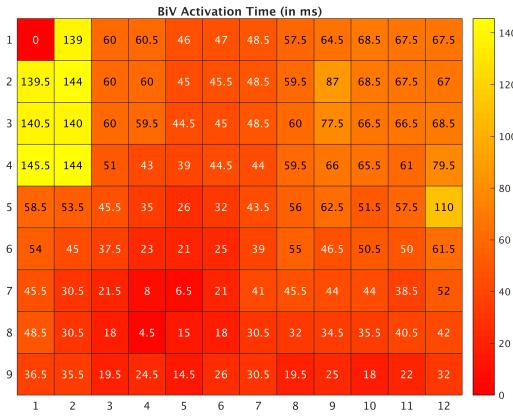


Fig. 20. Estimated activation time using our method - BiV pacing

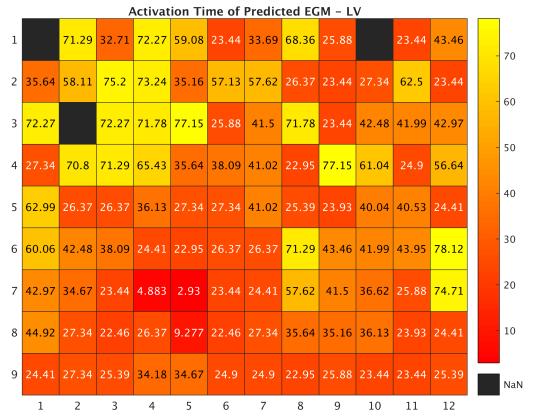


Fig. 23. Predicted Activation Time on the new data - LV pacing

### C. Activation Time from Laura's new data (Exp 16)

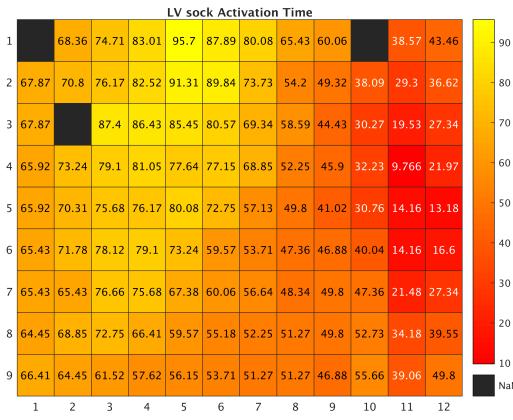


Fig. 21. Activation time from Laura's data - LV pacing

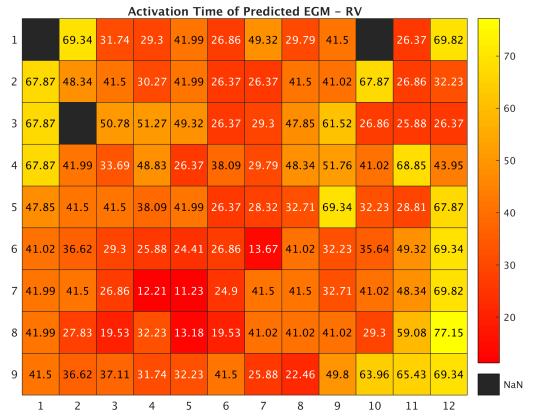


Fig. 24. Predicted Activation Time on the new data - RV pacing

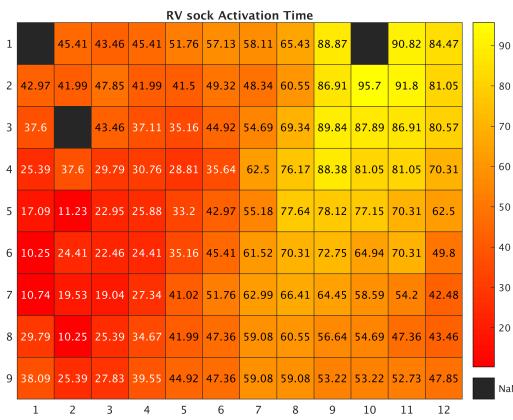


Fig. 22. Activation time from Laura's data - RV pacing

### D. Predicted Activation Time on new data (Exp 16)

This is the activation time of predicted EGM on the new data