IMPROVING GENERALIZATION IN HEART CONDITION DETECTION VIA THE USE OF TRANSFORMERS AND DATA AUGMENTATION

ANIRUDH KAILAJE [KAILAJE@SEAS], ALEXANDER RADCHENKO [XALER@SEAS], ERIK JAGNANDAN [EJAG@SEAS]

ABSTRACT. We investigate deep learning approaches for accurate diagnosis of cardiovascular diseases (CVDs) using electrocardiogram (ECG) signals. We compare two architectures: a ResNet-based model and a transformer-based model, and show that the ResNet model achieves superior test accuracy and memory efficiency. Additionally, we examine the extent to which the usage of different training sequence lengths and data augmentation techniques improves the accuracy of our ECG analysis models. We observe that manually tuned data augmentation leads to statistically significant improvements in test accuracy. However, the TaskAug method, which learns the optimal data augmentations by representing them with trainable parameters, was shown to be unable to consistently improve the test accuracy of our models.

1. INTRODUCTION

Cardiovascular diseases (CVDs) are among the leading causes of death worldwide (1). As a result, the diagnosis of CVDs is a critical task which, if done accurately, can save numerous lives. Typically, the first examination performed on patients to determine if they have a CVD is an electrocardiogram (ECG), which is a recording of the electrical signals produced by the heart (2). ECGs are usually reviewed by a cardiologist to determine a diagnosis, but this is a time-consuming and challenging task. Thus, the near-instantaneous analysis provided by an accurate diagnostic model for ECGs would greatly benefit cardiologists needing to quickly interpret ECGs in a medical emergency. Moreover, general practitioners and medical residents, for whom ECG analysis is an especially difficult task, would strongly benefit from automated assistance from a diagnostic ECG model (2).

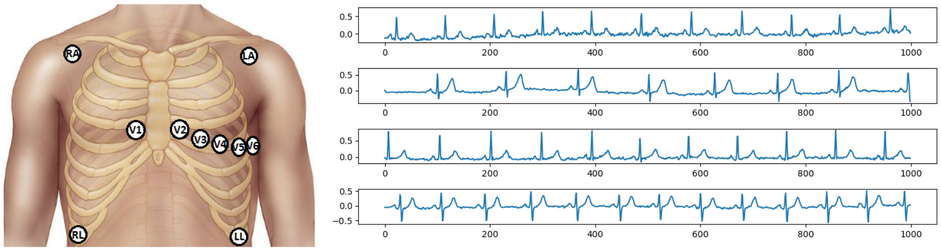
In response to this need, we have implemented a group of models that assess ECGs to detect CVDs using the PTB-XL ECG dataset (2). We use distinct models for 5 separate classification tasks, culminating in the prediction of specific diagnostic labels, as explained in the Background section. We first achieve this using ResNet-based models, which were the highest-performing model structure used in the Physionet Challenge, a deep learning competition using the PTB-XL ECG dataset. We then compare the performance of this model to that of transformer-based models to assess the applicability of transformers to the ECG analysis task. Finally, we investigate the improvements to our model performance provided by standard data augmentation techniques and the TaskAug method introduced in (3).

1.1. **Contributions.** This paper establishes the following:

(1) Implementing models for ECG analysis using a ResNet-based architecture
(2) Comparing the performance of Transformer-based models against the ResNet-based models
(3) Establishing the relative difficulty of the different modeling tasks and the reasons for their varying difficulties
(4) Demonstrating that the performance of our ResNet-based model can be improved in a statistically significant manner using data augmentations engineered to replicate the signal issues frequently observed in ECG's
(5) Examining the applicability of TaskAug data augmentation from (3) to a multi-class classification problem
(6) Understanding the limitations of our approach and outlining future work

2. BACKGROUND

An ECG measurement is taken by placing 10 metal electrodes on the chest. By measuring the voltage differences between the electrodes in a specified manner, the ECG is recorded as a 12-channel voltage waveform (4).



(A) ECG Electrode Placement

(B) ECG Measurement (first 4 channels)

There are 3 major types of Cardiac Abnormalities.

(1) **Diagnostic Abnormalities:** deviations from the normal ECG pattern that can aid in diagnosing specific cardiac conditions such as ischemia, infarction, or other cardiac disorders.
(2) **Form Abnormalities:** alterations in the shape or morphology of the ECG waveform without necessarily affecting the heart's rhythm. These abnormalities provide insights into structural or conduction abnormalities within the heart, even in the absence of significant rhythm disturbances
(3) **Rhythm Abnormalities:** disruptions in the regular sequence and timing of the heart's electrical activity, affecting the normal heartbeat. Rhythm abnormalities can have major clinical implications, limiting the heart's ability to circulate blood and potentially leading to symptoms such as palpitations, dizziness, or syncope.

The PTB-XL dataset has 71 unique labels for cardiac conditions. There are co-occurring labels in the dataset, meaning that more than one label could be true at once. Even within the categories of labels, the structure is shown in the appendix 8.1.

3. Related Work

In (2), the authors have benchmarked several architectures from the PhysioNet challenge for the classification tasks to various levels shown in 7. They benchmarked Inception-based, ResNet-based, Fully-Connected Networks, RNNs, and LSTMs, and a deep network using the signal wavelet characteristics. They found that Inception or ResNet-based networks performed the best tasks (defined in detail in Section 4.2), with AUCs ranging from 0.89 for the form to 0.95 for a diagnostic category. RNNs were slightly less performant for convolutional networks but were competitive for rhythm and sub-diagnostic categories. The deep networks vastly outperformed the network using the wavelet characteristics, although the authors point out that this may be due to the selection of their specific wavelet features.

4. Approach

4.1. Architecture.

The training dataset comprises 10 stratified folds, with Folds 9 and 10 distinguished as high-quality data where the labels are validated by a physician. We designate the 10^{th} fold as the test dataset, the 9^{th} fold as the validation dataset, and folds 1 to 8 as the training dataset. We created our own implementation of a ResNet-based model. While the PhysioNet challenge did not incorporate any transformer-based architectures, we anticipated that transformers could potentially work well since the data is sequential in nature and since RNNs were used with moderate success. To investigate this, we also developed our own transformer-based architecture. The summary of the two models is described in the appendix 1.

4.2. Classification Tasks.

We employed identical architectures for training across five distinct tasks: the diagnostic task, wherein we aimed to directly predict the 44 diagnostic labels; the sub-diagnostic class task, involving the prediction of 23 independent sub-diagnostic labels, as illustrated in 7; the super-diagnostic task, where the goal was to predict among the five super-diagnostic classes; the form task, which entailed predicting among the 19 form abnormalities; and the rhythm task, focused on predicting among the 12 rhythm abnormalities. Additionally, we trained a superclass model tasked with identifying whether the input pertained to diagnostic, form, or rhythm abnormalities.

4.3. Training Details.

As we had a dataset with high class imbalance, limited computational resources, and a complex model with numerous parameters, we decided to use training dataset folds 1-8 together for hyperparameter tuning for practicality and efficiency. This strategy ensures the representation of both majority and minority classes and strikes a balance between obtaining representative hyperparameters and managing the computational effort. Once a representative set of hyperparameters was identified, the training dataset was divided into 8 folds, and cross-validation was attempted. However, we ran into issues with our computational resources and were not able to carry it out. A sample training and test curve set from one of the trials are shown in Appendix 10 .

4.3.1. *Learning Rate.*

We trained both the ResNet-based and transformer models using a cosine annealing learning rate schedule. Using a learning rate finding algorithm, we selected a maximum learning rate of 2e-5 in both cases.

4.3.2. *Over-fitting.*

While detailed results are presented in the subsequent section, it was observed that both models exhibited a tendency to over-fit on the training dataset. To mitigate this, various techniques were explored to enhance model generalization and optimize performance with the provided dataset.

(1) **Dropout**: We introduced drop-out in the models with a $p = 0.1$

(2) **Sequence length**: Training using all 1000 samples of the measurement led to overfitting on the training set. To address this, the train-time sample size was reduced to 200-250, corresponding to 2-2.5 seconds, with random slices used during training. This improves performance for diagnostic and subdiagnostic models but negatively impacts rhythm task models. Consequently, the training sequence length for the rhythm task was increased to 600-650 samples (6-6.5 seconds), resulting in a significant improvement in validation and test performance.

(3) **Data Augmentation**: Through investigation of the dataset, we identified the nuisances that are likely to be seen at test time and designed data augmentation techniques to replicate these nuisances at training time. The introduction of these nuisances forces our model to become invariant to them. The results are discussed in 5.4.

4.4. **Data Augmentation Techniques.** Upon examining the PTB-XL ECG dataset (2), we determined that the nuisances that our model is likely to encounter at test time are signal quality issues. The dataset includes signal quality data provided by a technical expert, based on which we implemented the following augmentations:

	Description	Occurrence	Frequency	Amplitude	Probability
Baseline drift	ECG center deviates over time	7.36%	$f \sim N(0.08, 0.0005)$	0.2	0.1
Static Noise	Random noise over signal	14.94%	NA	$n \sim N(0, 0.02)$	0.15
Burst Noise	Brief high-frequency noise	2.81%	$f \sim N(100, 50)$	0.05	0.1

We also tested an automated tuning approach called TaskAug (3), in which gradient descent is used to tune the probability of applying and the magnitude of each augmentation method. The learnable parameters are the probability of each augmentation being selected, the probability of an augmentation being applied to each sample, and the magnitude of each augmentation. These parameters are selected based on the labels of the data (i.e., for each label there is a different probability and magnitude parameter). We included most of the same operations used in the TaskAug paper: Random Temporal Warp, Baseline Wander, Gaussian Noise, and Magnitude Scale. In the original paper, only binary classification was performed, in contrast our problem is both multi-class and multi-label which required modifying how probabilities and magnitudes are selected. We keep the probability of selecting each augmentation the same as in the paper. To select the probability of applying and the magnitude of each augmentation we use the mean of a point-wise multiplication of a vector of parameters with the vector of labels:

$$p_{\text{application}} = \frac{1}{N} \left(w_{\text{application}} \odot \text{labels} \right)$$

$$\text{magnitude} = \frac{1}{N} \left(w_{\text{magnitude}} \odot \text{labels} \right)$$

Where N is the number of labels. As seen below, most samples only have 1 or 2 positive labels. When there is a single positive label, since they are one-hot encoded, it is effectively used as an index for the vector of parameters, which is the same as the original paper's method.

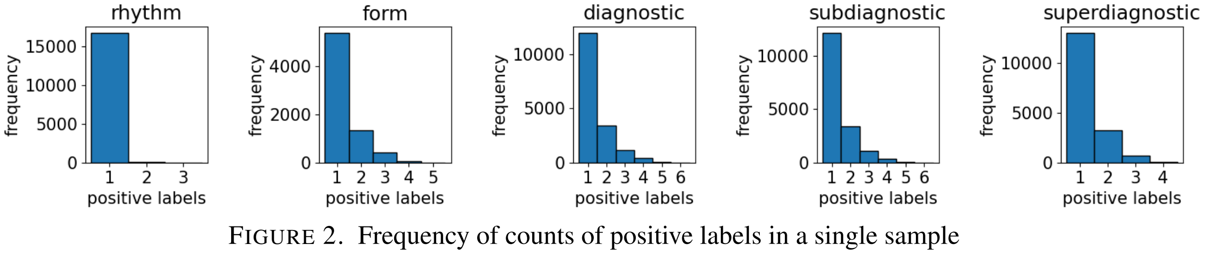


FIGURE 2. Frequency of counts of positive labels in a single sample

5. EXPERIMENTAL RESULTS

5.1. **Experiment Factors and Comparison Metrics.** We assessed our model's performance by varying architecture types, training sequence length, and employing two data-augmentation methods. We evaluated our models using the AUROC metric, with macro-weighted AUCs for each label due to high class imbalance in multilabel classification. Five models were trained for each factor setting, and performance on the test dataset was measured. The statistical significance of our results was measured using the Mann-Whitney U test, presenting p-values for each experiment trial.

5.2. **ResNet vs Transformer.** We observed that the transformer model performed similarly to the ResNet model with a shorter context length of 250 samples but quickly reached saturation, indicating overfitting on the training set and diminishing test performance. In contrast, the ResNet architecture demonstrated a lower tendency to overfit and achieved higher test performance, as seen in Figure 4. Additionally, the transformer-based architecture was harder to tune the hyperparameters for and required nearly double the amount of memory as compared to the ResNet-based model. Given these results, we opted to proceed with the ResNet architecture for all subsequent experiments. We believe that the transformer model has the capacity to learn the dataset as effectively as the ResNet model, but exhibits lower test performance because its greater model complexity renders it harder to train and more likely to overfit by "memorizing" the data.

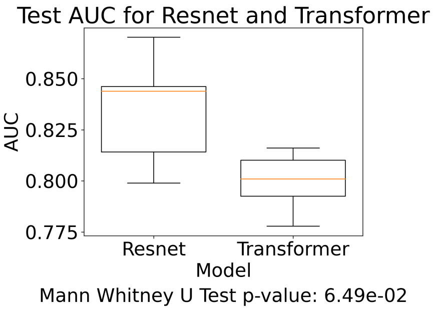


FIGURE 3. Resnet vs. Architecture SuperDiagnostic AUCs

5.3. **Training Sequence Length.** We observed that reducing the sequence length during training to 2-2.5 seconds improved performance for diagnostic and form tasks as shown in 4. The selection of random 2-2.5 second slices during training enhanced the ability of form task models to differentiate features within a heartbeat from samples outside a heartbeat. Conversely, the performance of the rhythm task model declined, as identifying abnormalities between multiple heartbeats is its primary task, and reducing the sequence length removed crucial information required for detecting these abnormalities. Various sequence lengths, including 4 seconds, 6 seconds, and the entire 10 seconds were experimented with. The results indicated that a 6-second context length provided the optimal performance for the rhythm training dataset, typically encompassing about three to six heartbeats.

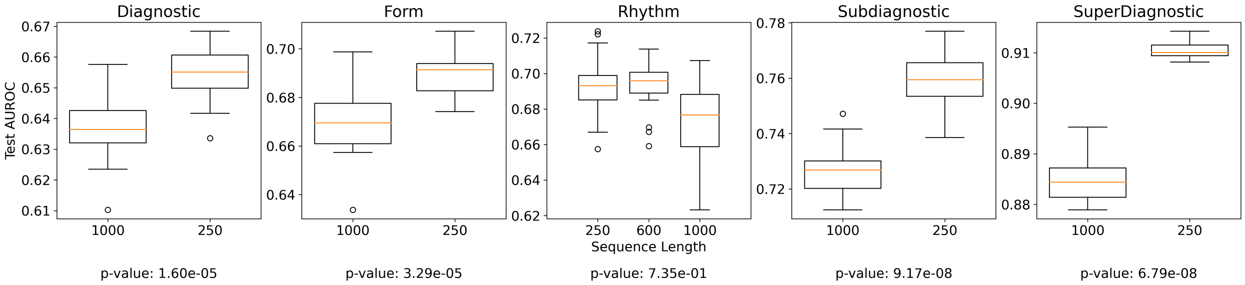


FIGURE 4. Impact on classification tasks based on Training Sequence Lengths

5.4. **Data Augmentation.** The usage of our standard data augmentation techniques consistently resulted in statistically significant improvements in performance across all models except the superdiagnostic model, as the p-values comparing each data augmentation method with the baseline model were mostly below 0.05. The superdiagnostic model likely did not see improvement because this task is already nearly optimized (as seen by its relatively high mean AUC of 0.871) prior to data augmentation, leaving little room for improvement. No one technique consistently outperforms the others across the 5 models. For the plots shown below, each model was trained $n = 5$ times without data augmentation, and then $n = 5$ times for each data augmentation technique, using a sequence length of 650 (6.5 seconds) in both cases.

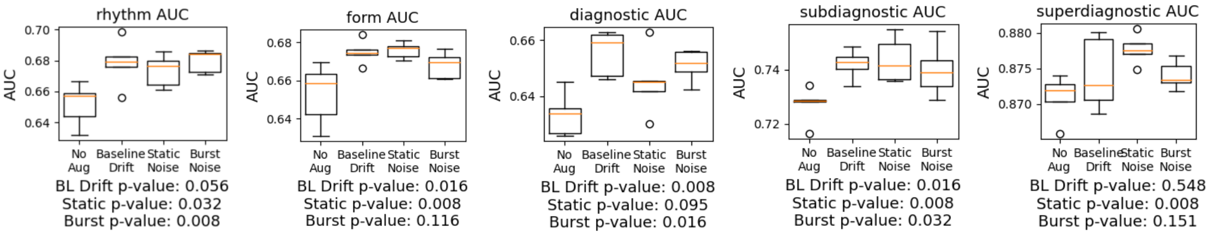


FIGURE 5. Test AUC Comparison for Each Data Augmentation Method, Across All Models

Performing data augmentation via TaskAug did not result in statistically significant improvement in AUC other than for the form model. For the plots shown below, each model was trained once without data augmentation, and then once using TaskAug. The TaskAug parameters were trained for 50 epochs with sequence length of 250 (2.5 seconds), a learning rate of 0.001, and a GD step performed every 10 epochs, starting with the $11th$ epoch. We also experimented with performing GD every 1 and 5 epochs, but this worsened performance. These models were then evaluated by bootstrapping the test set by selecting $n = 10$ random 95% samples from the test set. One possible reason for the lack of improvement caused by TaskAug is that while our method of adapting TaskAug from binary classification problems to multiclass, multilabel problems by taking a weighted average of the multiple labels for each sample appears reasonable in theory, it may not work well in practice. Alternatively, other hyper-parameters may need to be tuned or longer training may be required to see improvement from TaskAug.

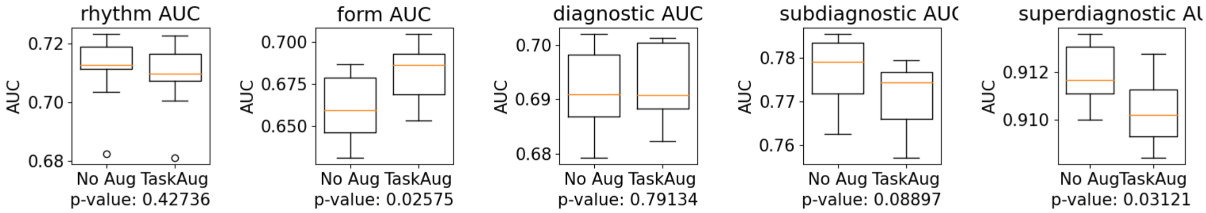


FIGURE 6. Test AUC Comparison of No Augmentation vs TaskAug, Across All Models

6. DISCUSSION

The best mean AUCs that we observed for each task are as follows:

	Diagnostic	Subdiagnostic	Superdiagnostic	Form	Rhythm
Best PTB-XL Benchmark	0.939	0.930	0.930	0.890	0.957
Our Best Performance	0.693	0.777	0.912	0.691	0.713

We can observe from the above data that we only achieve similar performance to the top performing models from PhysioNet for the superdiagnostic task. Nonetheless, this is not a major concern, as the top-performing PhysioNet models achieve very high performance which is difficult to replicate. Moreover, we have presented several notable academic findings providing insight on the ECG analysis task and the TaskAug augmentation method. First, we have demonstrated that although transformer-based models likely are capable of high performance in the ECG analysis task, they are difficult to train and prone to overfitting, and thus perform worse than ResNet-based models in practice. This informs future research efforts that transformer-based models should be avoided for ECG analysis unless further work is done to eliminate the issues they currently encounter in the training process. Additionally, we identified a tradeoff involving the sequence length used to train ResNet-based models, as the sequence length must be large enough to provide the model with sufficient information, but not so large as to enable overfitting. Critically, we have presented data augmentation techniques which produce statistically significant improvements in ECG test performance and can potentially be used to further improve high-performing ECG models. Finally, we extended TaskAug to a multiclass, multilabel problem, which the authors of (3) left as future work, and found that it did not improve performance, either due to difficulties in extending it to multilabel problems, or issues in training. Further work may be done by the academic community to use TaskAug on a different task, potentially supporting or contradicting our results, or to develop more effective modifications to TaskAug which enable it to perform well in the multiclass, multilabel setting.

7. FUTURE WORK

7.1. **Model Performance Improvement.** We aim to continue fine-tuning hyperparameters and investigating the factors contributing to the comparatively lower performance of form and rhythm task models compared to diagnostic models. One such factor limiting the performance of our models is the highly imbalanced nature of the dataset, as some labels are far less common and thus harder to learn than others. We look to address this and enable improved performance by using GANs to generate additional samples corresponding to the underrepresented classes. Additionally, we aim to amalgamate our models into a single comprehensive model capable of delivering predictions across all 71 labels.

7.2. **Model Interpretability.** The current adoption of deep networks in the medical community is limited due to interpretability demands. We would like to identify characteristic, ”tell-tale” signals corresponding to each abnormality. Our intention is to quantify model outputs for these signals, manipulate the characteristic signals by selectively removing key features, and gauge the impact on our model’s output. This approach seeks to enhance the interpretability of our model outputs, addressing concerns within the medical context. Additionally, the parameters learned by TaskAug can be extracted and studied (similar to what was done in the original paper), which provide insights as to which augmentations lead to better generalization and thus what issues may be present/absent in the data. (4)

References

[1] G. R. Dagenais, D. P. Leong, S. Rangarajan, F. Lanas, P. Lopez-Jaramillo, R. Gupta, R. Diaz, A. Avezum, G. B. Oliveira, A. Wielgosz, *et al.*, "Variations in common diseases, hospital admissions, and deaths in middle-aged adults in 21 countries from five continents (pure): a prospective cohort study," *The Lancet*, vol. 395, no. 10226, pp. 785–794, 2020.

[2] P. Wagner, N. Strodthoff, R.-D. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter, "Ptb-xl, a large publicly available electrocardiography dataset," *Scientific data*, vol. 7, no. 1, p. 154, 2020.

[3] A. Raghu, D. Shanmugam, E. Pomerantsev, J. Guttag, and C. M. Stultz, "Data augmentation for electrocardiograms," 2022.

[4] J. C. N.-C. organization and M. organization, "Ekg Machinery and Setup," feb 2 2023. [Online; accessed 2023-12-16].

8. Appendix

8.1. **Data Structure.** Below is the structure of the data

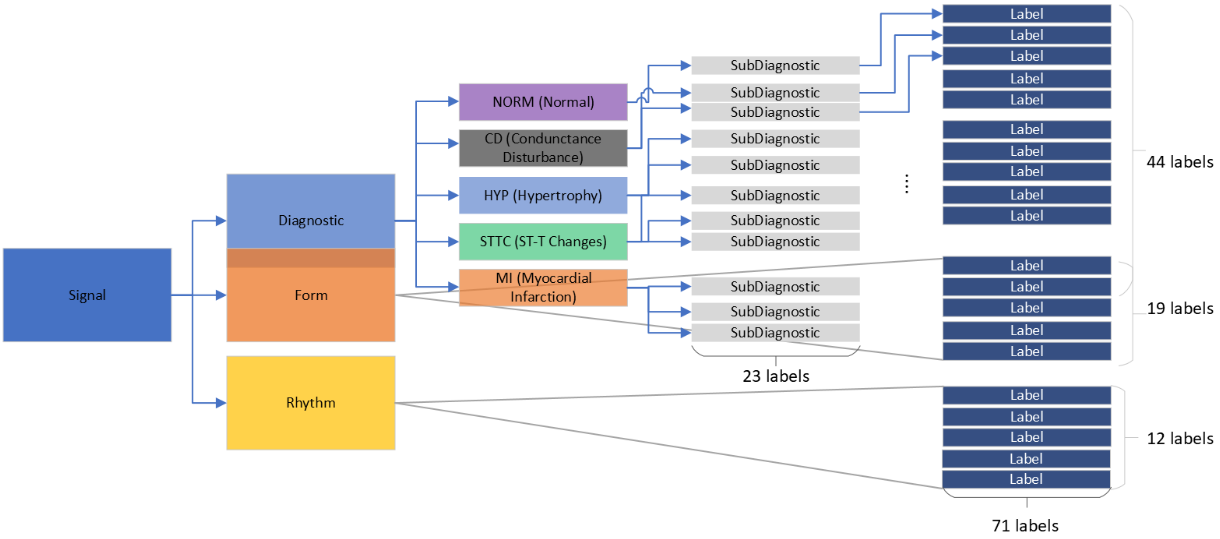


Figure 7. Label Distribution

8.2. **Model Summary.** Below are the model Summaries

Table 1. Transformer Model Architecture Summary

Layer (type:depth-idx)	Output Shape	Param #
Transformer	[1000, 3]	–
PositionalEncoding: 1-1	[1, 1000, 12]	–
TransformerEncoder: 1-2	[1, 1000, 12]	–
ModuleList: 2-1	–	–
TransformerEncoderLayer: 3-1	[1, 1000, 12]	3,884
TransformerEncoderLayer: 3-2	[1, 1000, 12]	3,884
TransformerEncoderLayer: 3-3	[1, 1000, 12]	3,884
LayerNorm: 2-2	[1, 1000, 12]	24
Linear: 1-3	[1, 1000, 64]	832
BatchNorm1d: 1-4	[1000, 64000]	128,000
Linear: 1-5	[1000, 256]	16,384,256
BatchNorm1d: 1-6	[1000, 256]	512
Dropout1d: 1-7	[1000, 256]	–
Linear: 1-8	[1000, 512]	131,584
BatchNorm1d: 1-9	[1000, 512]	1,024
Dropout1d: 1-10	[1000, 512]	–
Linear: 1-11	[1000, 3]	1,539
Total params:	16,659,423	
Trainable params:	16,659,423	
Non-trainable params:	0	
Total mult-adds (Units.GIGABYTES):	16.65	
Input size (MB):	0.05	
Forward/backward pass size (MB):	528.86	
Params size (MB):	66.63	
Estimated Total Size (MB):	595.53	

Below is the summary of the ResNet Model

TABLE 2. Resnet Model Architecture Summary

Layer (type:depth-idx)	Output Shape	Param #
ResNet	[1, 6]	–
Conv1d: 1-1	[1, 64, 500]	5,376
ReLU: 1-2	[1, 64, 500]	–
BatchNorm1d: 1-3	[1, 64, 500]	128
MaxPool1d: 1-4	[1, 64, 250]	–
Sequential: 1-5	[1, 256, 250]	–
Bottleneck: 2-1	[1, 256, 250]	–
Conv1d: 3-1	[1, 64, 250]	4,096
ReLU: 3-2	[1, 64, 250]	–
BatchNorm1d: 3-3	[1, 64, 250]	128
Conv1d: 3-4	[1, 64, 250]	12,288
ReLU: 3-5	[1, 64, 250]	–
BatchNorm1d: 3-6	[1, 64, 250]	128
Conv1d: 3-7	[1, 256, 250]	16,384
BatchNorm1d: 3-8	[1, 256, 250]	512
Sequential: 3-9	[1, 256, 250]	16,896
ReLU: 3-10	[1, 256, 250]	–
Bottleneck: x2	–	–
Sequential: 1-6	[1, 512, 125]	–
Bottleneck: x 4	–	–
Sequential: 1-7	[1, 1024, 63]	–
Bottleneck: x 23	–	–
Sequential: 1-8	[1, 2048, 32]	–
Bottleneck: x3	–	–
AdaptiveAvgPool1d: 1-9	[1, 2048, 1]	–
Linear: 1-10	[1, 6]	12,294
Sigmoid: 1-11	[1, 6]	–
Total params		28,278,918
Trainable params		28,278,918
Non-trainable params		0
Total mult-adds (Units.GIGABYTES)		1.57
Input size (MB)		0.05
Forward/backward pass size (MB)		56.49
Params size (MB)		113.12
Estimated Total Size (MB)		169.65

The same architecture's block diagram is shown below:

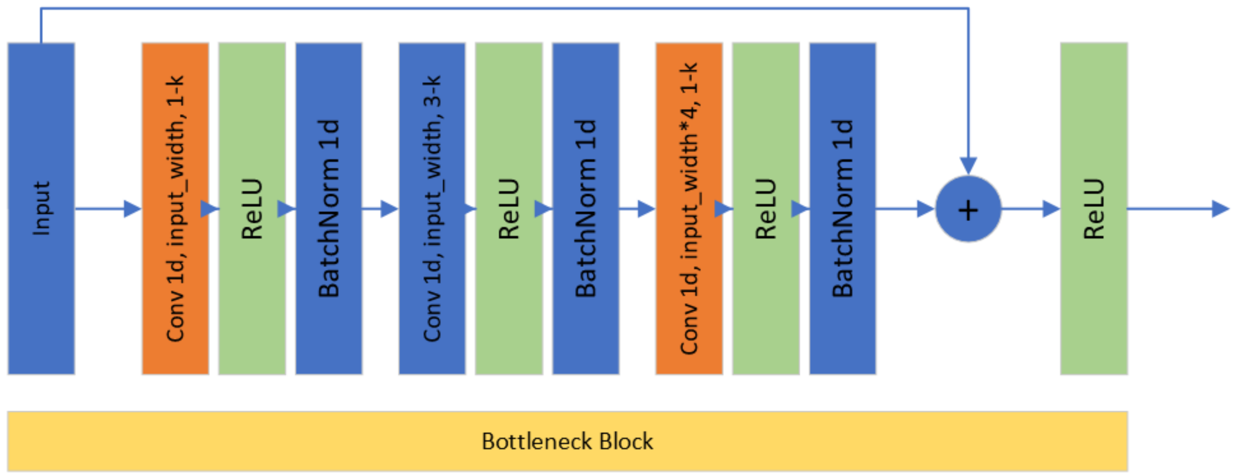

Input
Conv 1d, input_width, 1-k
ReLU
BatchNorm 1d
Conv 1d, input_width, 3-k
ReLU
BatchNorm 1d
Conv 1d, input_width*4, 1-k
ReLU
BatchNorm 1d
+
ReLU
Bottleneck Block

FIGURE 8. BottleNeck Block Used in Resnet Model

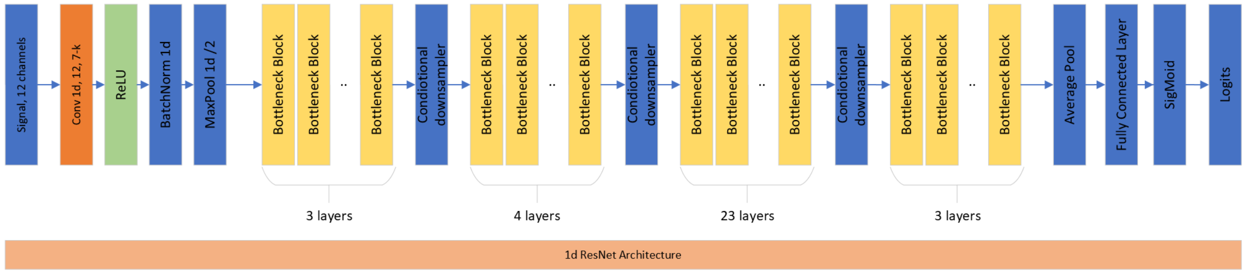


FIGURE 9. Resnet Model

8.3. **Training and Validation curves for 250 sequence length.** Below are the training and validation curves for the different tasks, with 250 sequence lengths during training.

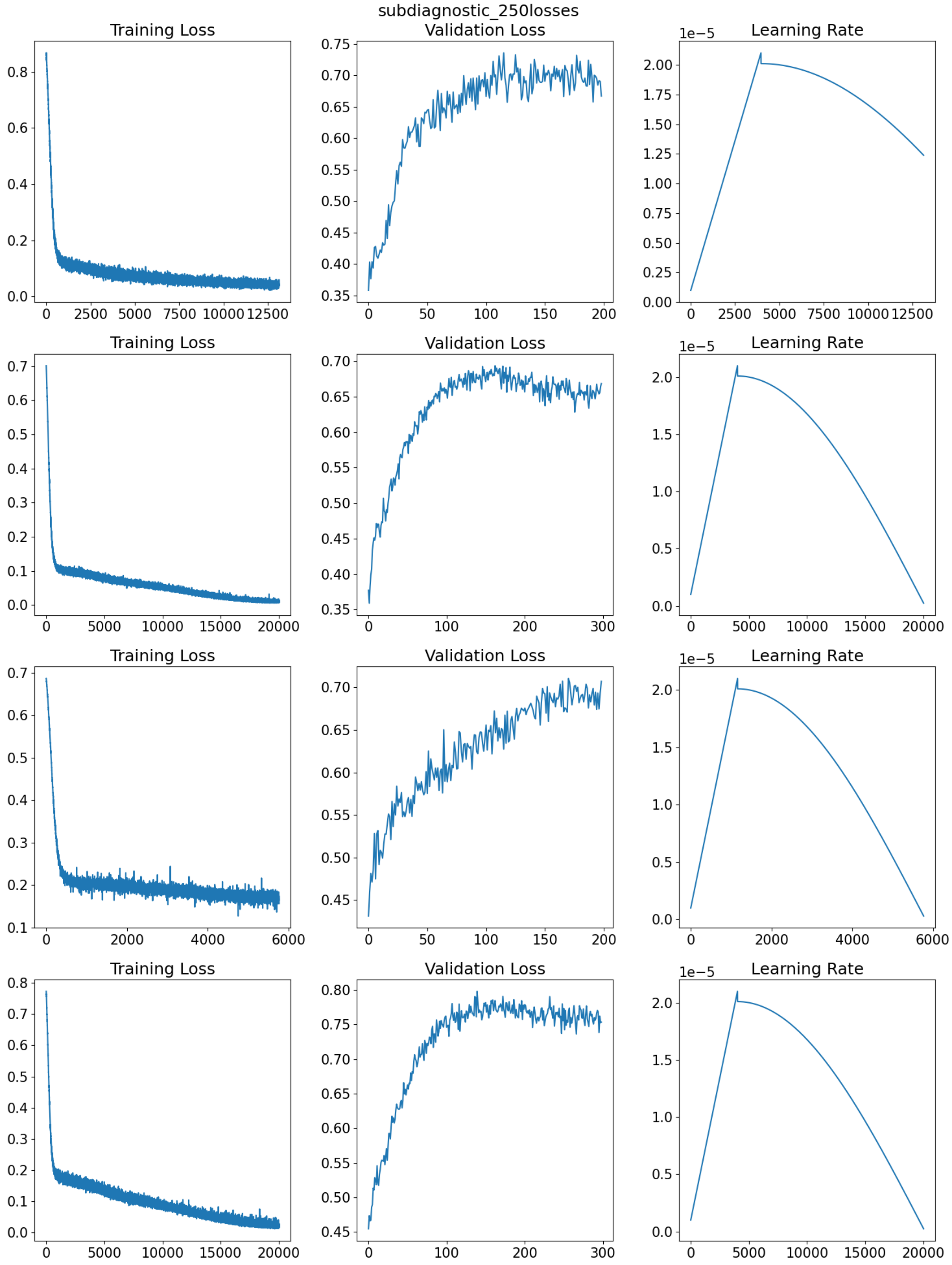


FIGURE 10. Training Curves for 250-SeqLen Models