Fast-tracking the Deep Residual Network Training for Arrhythmia Classification by Leveraging the Power of Dynamical Systems

Tuan Aqeel Bohoran
School of Science and Technology
Nottingham Trent University
Nottingham, UK
tuan.bohoran@ntu.ac.uk

Polydoros N. Kampaktsis
Division of Cardiology
Columbia University Irving Medical Center
New York City, NY, USA

Gerry P. McCann
Department of Cardiovascular Sciences
University of Leicester
NIHR Leicester Biomedical Research Centre
Glenfield Hospital
Leicester, UK

Archontis Giannakidis
School of Science and Technology
Nottingham Trent University
Nottingham, UK

***Abstract*—Arrhythmia, characterised by irregular heartbeats, poses significant health risks. Residual networks, a subset of deep learning architectures, have emerged as a potent tool in detecting electrocardiogram (ECG) signal anomalies. However, the enhanced accuracy and capabilities afforded by increasing network depth in these models come at the cost of heightened computational demands, posing a considerable challenge to their practical applicability. Addressing this critical bottleneck, this paper presents a methodology for the resource-economical development of machine learning-enabled systems for arrhythmia detection. The proposed methodology, grounded in the dynamical systems perspective of residual networks, initiates the training process with a shallow network, and then progressively augments its depth. We validate the method rigorously on the PhysioNet's MIT-BIH arrhythmia dataset using heartbeat spectrograms as training inputs. The results show that the proposed training necessitates a minimum of 40% fewer parameters per epoch, when compared with the conventional vanilla training, a feat achieved without sacrificing, and in fact potentially enhancing, the overall performance. Our findings suggest the methodology not only drastically reduces training time but also promises significant savings in energy consumption and environmental costs, offering a glimpse into a future of more sustainable and resource-efficient machine learning developments in arrhythmia detection.**

***Index Terms*—Computer vision, Machine learning, Neural Nets, Ordinary differential equations, Physiological signals, Training**

I. Introduction

Many people sustain irregular heartbeats which can be fatal [1] in some cases. As a result, the topic of arrhythmia classification by analysing the widely-used electrocardiogram (ECG) signals has been receiving great attention [2] in recent years. The manual analysis of heartbeats is both labour-intensive and subject to human errors, thus necessitating the advent of automated approaches. Within this context, deep learning architectures, particularly residual networks [3], have been successfully applied [4] lately towards detecting ECG signal anomalies.

Evidence in existing literature [5] underscores the pivotal role of network depth in enhancing the accuracy and capabilities of residual models. However, the computational demands during training accompanying increased depth pose significant hurdles, thereby limiting practical applicability of such models.

In this paper, in search of resource-economical development of systems for the machine learning-enabled arrhythmia detection, we propose to dynamically adjust the computation workload during residual network training. Essentially, the methodology entails initiating the training process with a shallow network, progressively augmenting its depth as training advances. To this end, we leverage the dynamical systems perspective [6] of residual networks that conceptualises a network constituted of N residual blocks as an ordinary differential equation with N temporal intervals. Unlike other strategies [7] for residual network training acceleration which are mostly heuristic, the proposed approach is theoretically grounded.

Our pipeline is evaluated on the large-scale freely-available PhysioNet's MIT-BIH arrhythmia dataset [8], [9]. Drawing inspiration by some recent deep learning sound classification studies [10] that provided more accurate results in noisy conditions, we employ the heartbeat spectrograms to train the deep residual networks. The spectrogram being an image itself means that it aligns seamlessly with the input requisites of deep residual networks. Apart from training time reductions, we also gauge savings on energy consumption and environmental cost by using the proposed pipeline.

Tuan Aqeel Bohoran is funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 801604.

This research not only underscores the reductions in training time afforded by our methodology but also foregrounds the associated savings on energy consumption and environmental impact. By fostering more resource-efficient machine learning development frameworks for arrhythmia detection, this investigation contributes a salient step forward in the ongoing efforts to enhance automated ECG signal analysis.

II. Materials And Methods

A. Dataset

This paper utilises the PhysioNet MIT-BIH Arrhythmia ECG dataset [8], [9]. In our experiments, we have used only the ECG lead II. The MIT-BIH dataset consists of 47 patients (109446 data points per example at sampling frequency 125 Hz). The ECG signals are annotated into five classes by at least two cardiologists according to the Association for the Advancement of Medical Instrumentation (AAMI) EC57 standard [11]. Table I lists the AAMI EC57 annotation standards for the five categories. The predefined five class dataset consists of 87554 ECG signals in the training set and 21892 ECG signals in test set. We randomly chose 10% of the ECG signals from the training set as the validation set. A representative heartbeat and the generated spectrogram are illustrated in Figure 1.

TABLE I
Summary of mappings between beat annotations and AAMI EC57 [11] categories.

Category	Annotations
N	• Normal • Left/Right bundle branch block • Atrial escape • Nodal escape
S	• Atrial premature • Aberrant atrial premature • Nodal premature • Supra-ventricular premature
V	• Premature ventricular contraction • Ventricular escape
F	• Fusion of ventricular and normal
Q	• Paced • Fusion of paced and normal • Unclassifiable

B. Dynamical Systems Viewpoint

The forward propagation in a residual network block can be expressed as:

$$y_{j+1} = y_j + h\mathcal{F}(y_j, W_j), \quad j = 0, 1, \ldots, D-1, \qquad (1)$$

where $\mathcal{F}$ is the residual module, and D is the number of layers. Here, $h > 0$ is a sufficiently small parameter that has been included without loss of generality. $\mathcal{F}$ encompasses batch normalisation, ReLU activation and convolutional layers. Eq. 1 can be rewritten as

$$\frac{y_{j+1} - y_j}{h} = \mathcal{F}(y_j, W_j). \qquad (2)$$

Eq. 2 can be seen as the forward Euler discretisation for the following initial value ordinary differential equation (ODE)

$$\dot{y}(t) = \mathcal{F}(y(t), W(t)), \quad y(0) = y_0, \quad \text{for} \quad 0 \le t \le T, \quad (3)$$

where features $y(t)$ and parameters $W(t)$ are viewed in their continuous limit as functions of time $t \in [0, T]$, the evolution time T corresponds to the network depth D, $y(0)$ is the input feature map after the initial convolution, and $y(T)$ is the output feature map before the softmax classifier. Consequently, the problem of learning the model parameters, W, is equivalent to solving an optimal control problem involving the ODE in Eq. 3.

The theoretical analyses for the network growing dynamics and the feasibility of effective residual network growing during training can be found in [12].

C. Automated Adaptive Training Algorithm

We briefly describe the adaptive training algorithm [12] that we employed. In this study, we go a step further and test the algorithm's validity in a different target research area, in particular in a challenging cardiovascular healthcare task.

In each training epoch, the model parameters are updated as usual. Then, a growing scheduler determines if it is needed to increase the network depth. Given that the upper bound of the temporal error is monotonously correlated with the maximum Lipschitz constant of $\mathcal{F}(D)$, then, the growing scheduler is designed to make sure that the Lipschitz constant will not become too large. Specifically, if it exceeds a pre-determined risk tolerance r_{tol}, then, the network grow is triggered.

Given that a residual block comprises convolutional layers, ReLU activation layers, and batch normalisation, then, the Lipschitz constant of the aggregate function is simply the product of the individual Lipschitz constants of each component. The latter are amenable to efficient calculation [13]. In fact, it has been shown [14] that this Lipschitz constant calculation imposes a negligible overhead on the total training time.

To incorporate adaptive growing, the learning rate scheduler is designed such that, after each growth, the cycle in a standard cosine learning rate scheduler is reset as

$$\eta = \eta_{\text{min}} + \frac{1}{2}(\eta_{\text{max}} - \eta_{\text{min}}) \left(1 + \cos\left(\frac{T_{\text{cur}} - T_{\text{grow}}}{T_{\text{tot}} - T_{\text{grow}}}\pi\right)\right), \quad (4)$$

where η_{min} and η_{max} represent the minimum and maximum learning rates, respectively. T_{cur} denotes the current epoch, T_{tot} is the total number of epochs, and T_{grow} refers to the epoch at the last growth occurrence. Initially, $T_{\text{grow}} = 0$.

To ensure effective training, cloning initialisation is applied as a method of growing. This method simply clones the residual blocks from the nearest time points of the previous network to populate the new network. Such an approach also ensures the efficient continual optimisation after the grow. An implicit step size scaling is also implemented after growth to maintain a roughly constant sum of residuals. The number of layers is doubled in each growth, whereas a certain number of epochs is reserved exclusively for the training of the final model.

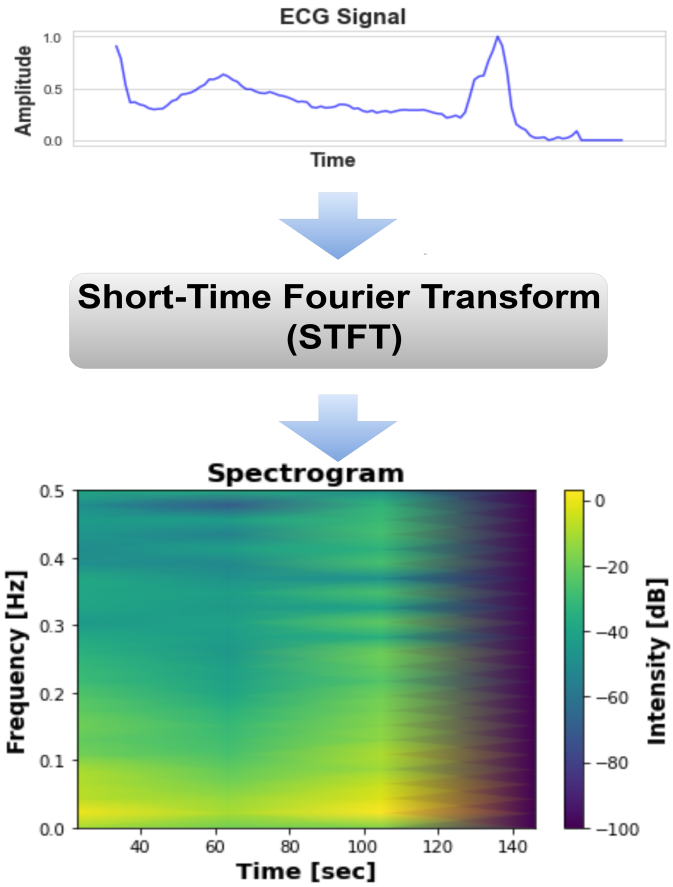

Fig. 1. Instance of an ECG signal and the generated spectrogram. The brighter the colour and higher the energy of the singal.

D. Experiments

We employed ResNet-50 and ResNet-74 to train using the proposed adaptive method. For a fair comparison, vanilla ResNet-50 and vanilla ResNet-74 (fixed models) were also trained. The training and testing mini-batch sizes were 128 and 100, respectively. The learning rate was fixed at 0.1. For the optimiser, we chose weight decay and momentum constants to be equal to 0.0002 and 0.9, respectively. The growth risk tolerance for the Lipschitz constant was (r_{tol} =) 1.4. Each network was grown two times, doubling the depth each time. All networks were trained for 64 epochs (three runs), making sure to reserve for the proposed method at least 20 epochs to train the final model (after the second growth). All the experiments were carried out on a workstation with NVIDIA RTX A6000 48GB GPU and Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz CPU.

E. Evaluation Metrics

To represent the actual training time, we depict learning curves with respect to wall clock time (rather than epoch number). For quantitative evaluation purposes, on top of validation and test accuracy, we also employ the parameters per epoch (PPE) metric [12]. This measure portrays the computational load (memory and processor that were utilised for training); yet it is detached from hardware settings and its usage. Lastly, we present the total carbon dioxide equivalent (CO_2eq) (in g) and energy consumption (in kWh) for each model using the *Carbontracker* method [15], which is reliant on the type of hardware used. For each quantitative measure, the mean and standard deviation over the three runs of training are provided.

III. Results and Discussion

Figures 2 and 3 illustrate representative training and validation error rates (error against wall clock time in minutes) for ResNet-74. By inspection, it is apparent that the proposed method exhibits substantial training acceleration while achieving similar accuracy at the end.

Table II lists the validation and test accuracies achieved by the proposed (adaptive) and the vanilla (fixed) training methods. The proposed method has slightly higher values for both ResNet-50 and ResNet-74.

Table III outlines the CO_2eq (in g), energy spent (in kWh) and PPE for the two training methods. In regard to ResNet-74, the proposed method produced $\sim$ 45% less environmental pollution, consumed $\sim$ 39% less energy, and required $\sim$ 46% less parameters to train, when compared to the vanilla method. Similarly, with respect to ResNet-50, the proposed method emits $\sim$ 33% less carbon emissions, is $\sim$ 25% more energy

efficient and requires $\sim$ 40% times less parameters to train, when compared with the vanilla method.

TABLE II
COMPARISON OF PROPOSED AND VANILLA TRAINING METHODS FOR RESNET-50 AND RESNET-74. VALIDATION (VAL. ACC.) AND TEST (TEST ACC.) ACCURACIES ARE PRESENTED AS MEAN $\pm$ STANDARD DEVIATION OVER THE THREE TRAINING RUNS.

Method	ResNet	Val. Acc.(%)	Test Acc.(%)
Proposed	50	96.33 ± 0.46	97.14 ± 0.08
	74	96.41 ± 0.42	97.10 ± 0.05
Vanilla	50	96.19 ± 0.44	97.09 ± 0.03
	74	96.38 ± 0.51	96.93 ± 0.16

TABLE III
COMPARISON OF PROPOSED AND VANILLA TRAINING METHODS FOR RESNET-50 AND RESNET-74 IN TERMS OF CO_2EQ, ENERGY CONSUMPTION, AND PARAMETERS PER EPOCH (PPE). RESULTS ARE PRESENTED AS MEAN $\pm$ STANDARD DEVIATION OVER THE THREE TRAINING RUNS.

Method	ResNet	CO_2eq (g)	Energy (kWh)	PPE ($\times 10^6$)
Proposed	50	411.34	1.32	0.46 ± 0.01
	74	410.09	1.31	0.63 ± 0.06
Vanilla	50	611.49	1.75	0.76
	74	738.23	2.14	1.15

IV. CONCLUSIONS

In this paper, a method was proposed to reduce the training time of deep residual networks for the challenging arrhythmia classification task, without compromising the model performance. To achieve our goal, we exploited the dynamical systems perspective of deep residual networks.

For deciding the network growth, the Lipschitz constant is calculated at every epoch. The Lipschitz constant has recently received considerable attention in the deep learning community, mostly to improve stability [16] and robustness against adversarial attacks [17].

Extensive experiments on the MIT-BIH arrhythmia dataset demonstrated that the proposed method required at least 40% less parameters per epoch to train when compared with conventional vanilla training, while retaining or improving performance. It also reduced carbon emissions by at least 1/3 and improved energy efficiency by at least 1/4.

This research heralds a new chapter in the pursuit of sophisticated, efficient, and sustainable solutions in the cardiovascular healthcare technology landscape.

For future work, we will explore further algorithmic refinements and try to extend this framework to encapsulate a broader spectrum of deep learning architectures. We will also apply the technique to a diverse array of cardiovascular tasks, thus broadening its impact.

REFERENCES

[1] Srinivasan, N. & Schilling, R. Sudden Cardiac Death and Arrhythmias. *Arrhythm Electrophysiol Rev*. **7**, 111-117 (2018,6)

[2] Alarsan, F. & Younes, M. Analysis and classification of heart diseases using heartbeat features and machine learning algorithms. *Journal Of Big Data*. **6** (2019,8), https://doi.org/10.1186/s40537-019-0244-x

[3] He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference On Computer Vision And Pattern Recognition (CVPR)*. pp. 770-778 (2016)

[4] Kachuee, M., Fazeli, S. & Sarrafzadeh, M. ECG Heartbeat Classification: A Deep Transferable Representation. *2018 IEEE International Conference On Healthcare Informatics (ICHI)*. pp. 443-444 (2018)

[5] Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference On Learning Representations*. (2015)

[6] E, W. A Proposal on Machine Learning via Dynamical Systems. *Communications In Mathematics And Statistics*. **5**, 1-11 (2017,3,1), https://doi.org/10.1007/s40304-017-0103-z

[7] Huang, G., Sun, Y., Liu, Z., Sedra, D. & Weinberger, K. Deep Networks with Stochastic Depth. *Computer Vision – ECCV 2016*. pp. 646-661 (2016)

[8] Moody, G. & Mark, R. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering In Medicine And Biology Magazine*. **20**, 45-50 (2001)

[9] Moody, G. & Mark, R. MIT−BIH Arrhythmia Database. (physionet.org,1992), https://physionet.org/content/mitdb/

[10] Khamparia, A., Gupta, D., Nguyen, N., Khanna, A., Pandey, B. & Tiwari, P. Sound Classification Using Convolutional Neural Network and Tensor Deep Stacking Network. *IEEE Access*. **7** pp. 7717-7727 (2019)

[11] Association for the Advancement of Medical Instrumentation, *Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms*, ANSI/AAMI EC38, vol. 1998, 1998.

[12] Dong, C., Liu, L., Li, Z. & Shang, J. Towards Adaptive Residual Network Training: A Neural-ODE Perspective. *Proceedings Of The 37th International Conference On Machine Learning*. **119** pp. 2616-2626 (2020,7,13), https://proceedings.mlr.press/v119/dong20c.html

[13] Y. Tsuzuku, I. Sato, and M. Sugiyama, "Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 31, 2018. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/485843481a7edacbfce101ecb1e4d2a8-Paper.pdf

[14] Gouk, H., Frank, E., Pfahringer, B. & Cree, M. Regularisation of Neural Networks by Enforcing Lipschitz Continuity. (2020)

[15] Anthony, L., Kanding, B. & Selvan, R. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models. (ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems,2020,7), arXiv:2007.03051

[16] Qi, G. Loss-sensitive generative adversarial networks on lipschitz densities. *International Journal Of Computer Vision*. **128**, 1118-1140 (2020)

[17] Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y. & Usunier, N. Parseval networks: Improving robustness to adversarial examples. *International Conference On Machine Learning*. pp. 854-863 (2017)

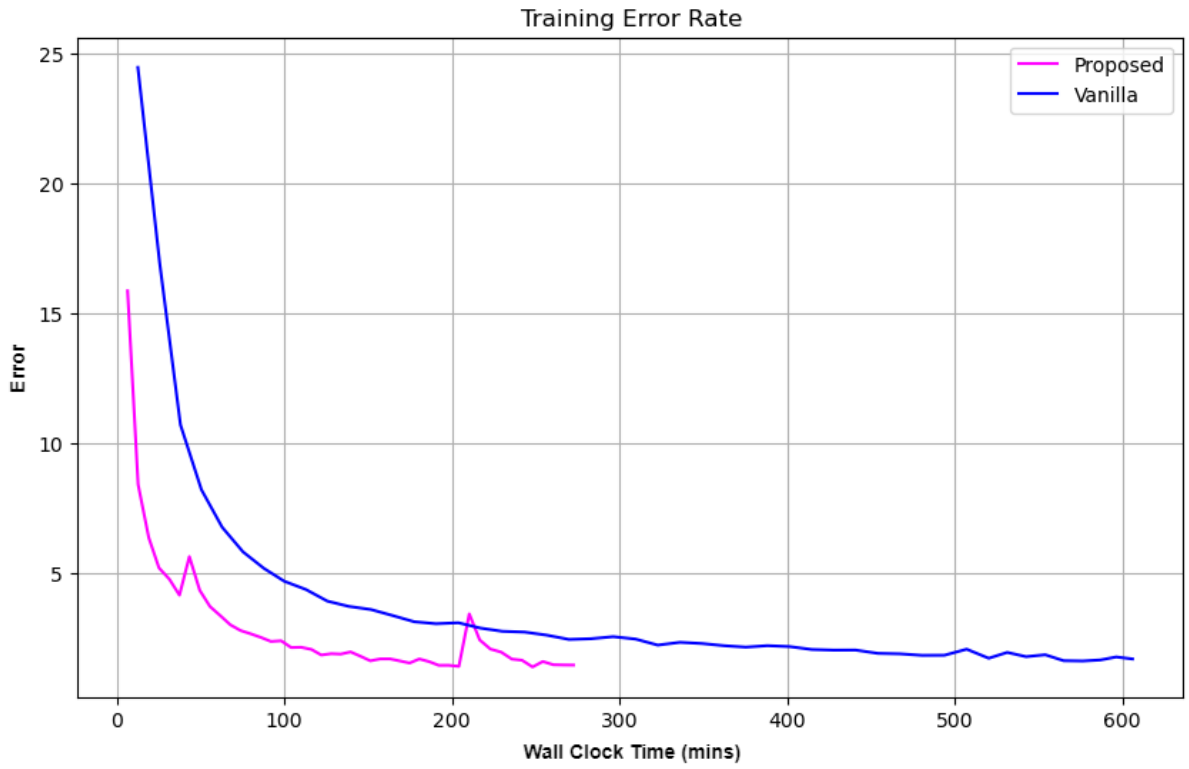

Fig. 2. The training error over wall clock time comparison of the proposed and vanilla training methods. Time is measured in minutes.

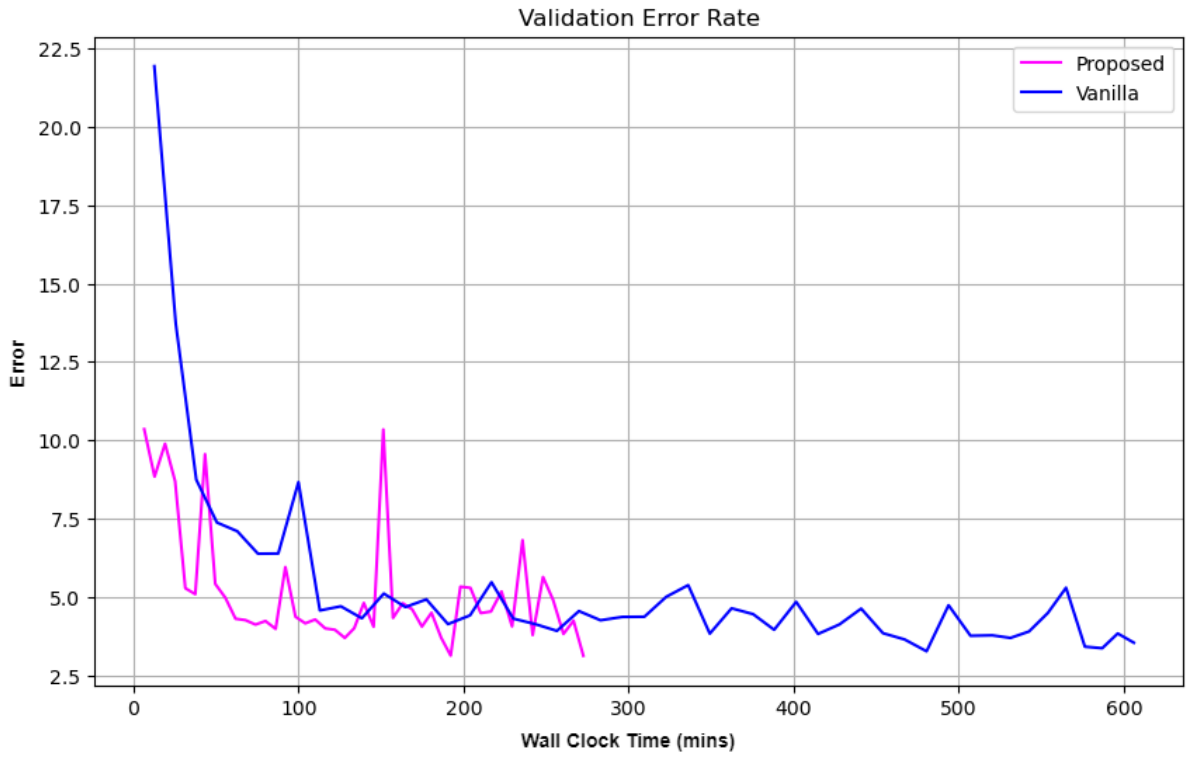

Fig. 3. The validation error over wall clock time comparison of the proposed and vanilla training methods. Time is measured in minutes.