Can Generative Pre-trained Transformers for PPG Signals Run on Microcontrollers?

Marek Żyliński[1], Bartosz Tomasz Śmigielski[1], Gerard Cybulski[1]

[1] IInstitute of Metrology and Biomedical Engineering, Faculty of Mechatronics, Warsaw University of Technology, Warsaw, Poland

Abstract

The PPG-PT, small generative model for PPG, utilizes a 101-token vocabulary to represent signal amplitudes and a 500-token context window (equivalent to 10 seconds of data at 50Hz) processed through 8 transformer layers. Consequently, the model comprises 443,622 parameters, stored as 4-byte floats. The memory requirements exceed the capacity of common wearable microcontrollers, such as the nRF52840, which possesses only 1024KB of Flash memory. In this study, we explored the optimization and deployment of this generative pre-trained transformer onto resource-constrained microcontrollers.

To refine the model for the constraints of an embedded system, we conducted a layer redundancy analysis using the Mean Block Influence metric. This approach quantifies the contribution of each transformer block and allows to rank layer and find ones that performing nearly identity-like transformations and may be redundant. We systematically evaluated the model by removing layers with the lowest influence scores one by one. Our pruning experiments revealed that while the architecture remains functional with as few as three transformer layers, there is a noticeable degradation in the morphological reliability. At this reduced depth, the model successfully generates a periodic, pulse-like waveform. However, critical physiological details, such as the inflection point and the dicrotic notch, are lost.

1. Introduction

The Generative Pre-trained Transformers (GPT) architectures can be directly adapted from natural language processing to complex physiological time series. By tokenizing physiological waveforms into discrete representations and training compact decoder-only models to predict the next token, networks can learn a general, efficient representation of underlying cardiac dynamics [1]. Foundational understanding acquired during pre-training can be utilized during fine-tuning. The model can be fine-tune on small dataset for clinical tasks, such as arrhythmia detection and heart condition classification.

Davies *et al.* [2] introduced specialized pre-trained models, including the Photoplethysmography Pre-Trained Transformer (PPG-PT) and its electrocardiography counterpart (ECG-PT). Training PPG-PT over 500,000 iterations with a batch size of 64 required just over 5 days on an NVIDIA RTX A2000 12GB GPU. The model was trained on a dataset combining over 128 million PPG tokens from Capnobase, BIDMC, and MIMIC II subsets, as well as 42 million ECG tokens from Lead-I PhysioNet Challenge data. In this study, we evaluate the feasibility of deploying PPG-PT on wearable devices.

PPG-PT is a compact architecture. The model has 8 transformer blocks with a total of 443,622 parameters stored in 64-bit precision (`https://github.com/harryjdavies/HeartGPT`). Its memory footprint remains a challenge. Modern microcontrollers can perform advanced computations and even execute machine learning models as Flash and RAM capacities continue to increase with each generation, making them capable to run machine learning models [3]. Nevertheless, this model still exceeds the hardware constraints of microcontrollers commonly used in edge wearables. For instance, the Bioboard employs an nRF52840 with only 1 MB of Flash and 256 KB of RAM [4], while the OpenEarable uses an nRF5340 with 512 KB of RAM [5].

In large language models, a single transformer layer barely contributes to the final output [6] and can therefore be removed without degrading model performance. In this study, we evaluate whether PPG-GT can be pruned to fit within microcontroller memory constraints. To this end, we assess the impact of both context size and the number of transformer layers on model performance.

2. Methods & Results

In the study we used Pulse Transit Time PPG Dataset [7] (`https://physionet.org/content/pulse-transit-time-ppg/1.1.0/`). The dataset has the recordings are from 22 healthy subjects perform-

ing 3 physical activities. In the analysis we used first 15 seconds of $pleth_2signal$ (infrared wavelength PPG from the distal phalanx of the left index finger, recorded using MAX30101 sensor). The signals were inverted, baseline drift were removed using moving average of 1 second, the signals were filter (bandpass filter 1-40Hz), resampled to 50 Hz, normalized and tokenized before analysis.

Context size is defined as the number of previous samples used to generate a new sample. We evaluated the model using reduced context sizes ranging from 500 down to 1 sample, in steps of 50 samples. For each context size, we predicted the next 500 samples across all recordings and calculated the median absolute error (MAE). As a proxy to evaluate relative computation times on microcontrollers, all benchmarks were performed on an Intel Core i5-13500 CPU.

Figure 1 shows the impact of context size on both the computation time and prediction accuracy of the model. The generation time for a new sample is non-linear relative to the context size, with larger contexts significantly increasing runtime. For a context size of 250, the model generated 500 samples in approximately 10 seconds; increasing the context size by 40% (to 350) increased the computation time by 60% (to 16 seconds). Conversely, increasing the context size above 200 did not yield meaningful improvements in overall accuracy. Based on these trade-offs, we selected a fixed context size of 250 samples for subsequent experiments.

Men *et al.* [8] showed that same layers of LLMs contribute minimally to the overall network functionality. Redundant layers increase computation overhead and do not benefit model accuracy. Men *et al.* [8] proposed block influence metrics to evaluate contribution of single transformer:

$$\mathbf{BI_i} = \mathbf{1} - \mathbb{E}_{(\mathbb{X},\approx)} \left[\frac{\mathbf{X_{i,t}^\top X_{i+1,t}}}{\|\mathbf{X_{i,t}}\|_2 \cdot \|\mathbf{X_{i+1,t}}\|_2} \right]$$

Where, $X_(i,t)$ means the t-th row of hidden states of the i-th layer. Lower BI score implies layer makes lower transformations to the hidden states and is therefore less important.

Table 1. Mean Block Influence metric across PPG-PT transformer layers.

Transformer Block	Mean Block Influence
1	0.0555
2	0.0187
3	0.0113
4	0.0087
5	0.0089
6	0.0120
7	0.0189
8	0.3079

Table 1 presents influence metrics for each transformer

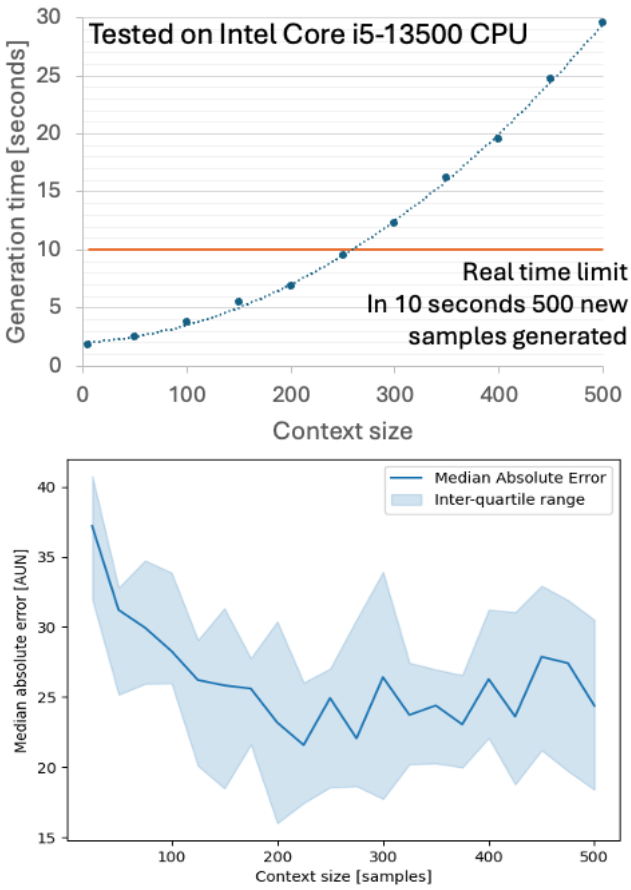

Figure 1. On the left: Relation between context size, number of previous samples used to prediction of new sample, and time of computation on CPU. On the right: relationship between context size and median absolute error.

block in network. Based on them we removed transformer blocks from less influential to the most. We set sequence of blocks removal as: 4, 5, 3, 6, 2, 7, 8, and 1. Removing of transformer block reduce computation time and memory footprint. Each transformer has around 50,000 parameters that must be stored in RAM. Computing time for each block on Intel Core i5-13500 CPU take on average 1.1721 second.

However, removing of transformers reduce prediction accuracy of the model. Figure 2 shows absolute error for predicted tokens for networks with different number of transformers. In each case prediction error accumulate, absolute error increase with increased number of predicted tokens. Next prediction is based on predictions. First token is predicted on signal, next one replaces one sample with own prediction etc. Model with 8 transformers can predict around 0.5 secund of signal with acceptable error. Removing of the transformer block increase speed of increase of absolute error for next prediction.

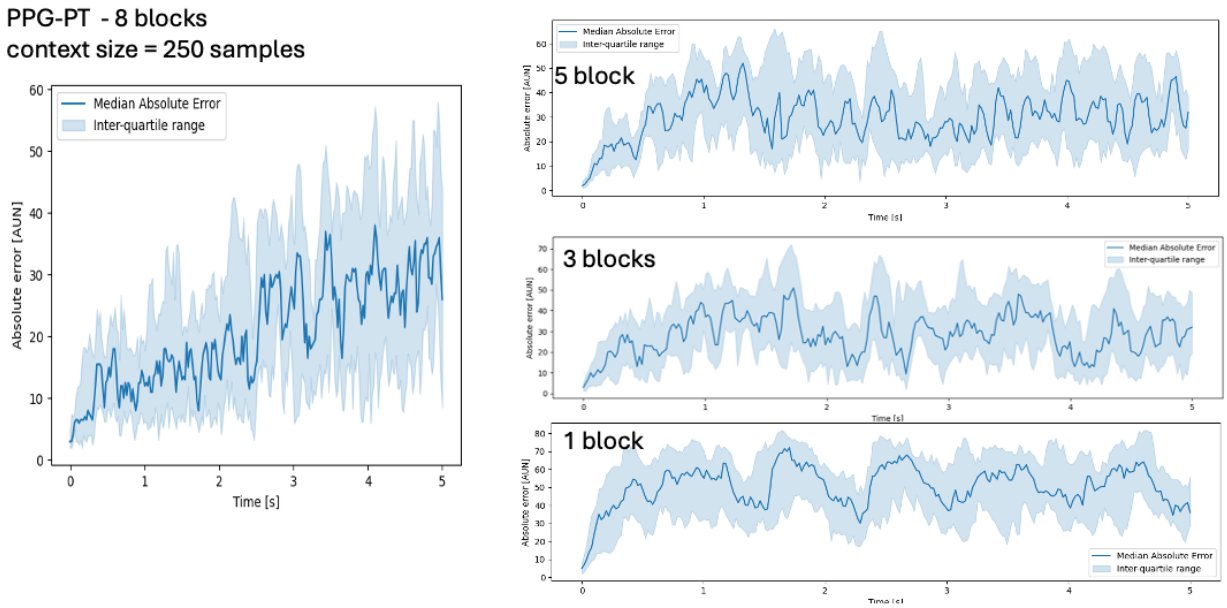

Figure 2. Reduction of the number of transformer blocks reduces predictive accuracy. With fewer transformer blocks, the absolute error of forward predictions increases faster, limiting the model's ability to predict further samples accurately.

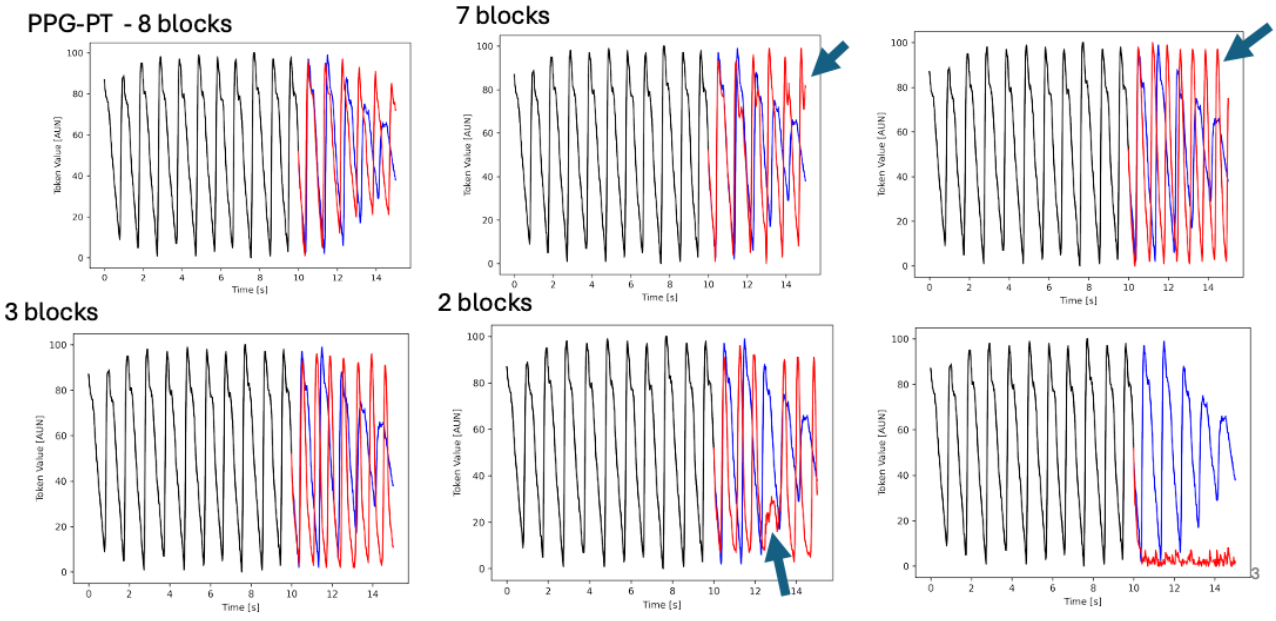

Figure 3. Reduction of number of transformers block reduce accuracy of the model's predictions. Removing of transformers reduce model's ability to track changes in details and further changes of PPG signal. The 7-blocks model lost track of changes in PPG amplitude. For 5-3 block models details of signal like inflection point or fluctuation of amplitude are missing. For less than 3-blocks model predictions tend to collapse.

Figure 3 compare predicted signal with true PPG. Removing of transformer blocks reduce model capability to track details of the PPG signal. Network with 7 transformers lost track of changes in amplitude of PPG. Network with 5 blocks cannot reconstructed inflection point. 3 transformers are enough to generate PPG like oscillations, for less blocks prediction totally collapse.

3. Discussion

The PPG-GT is a Generative Pre-trained Transformers aiming to predict next sample based on previous one, learning representation of underlying cardiac dynamics. The model can be fine-tuned for atrial fibrillation and beat detection [2]. Besides its small size the PPG-GT cannot be deploy into edge device in current form. Microcontrollers used in wearables[4, 5] don't have enough computation power and memory to handle the model.

Main strategies to reduce memory footprint of neural models are quantization and pruning [9]. In this work we focused on pruning, removing of redundant transformers. Operation reduces both computation time and memory footprint. Unfortunately for small model like the PPG-GT removing of the transformers reduced reduce accuracy of the model's predictions. The model lost information of underlying cardiac and PPG dynamics. Removing of transformers reduced details in predicted PPG (Fig. 3) and in consequence accuracy of the predicted signal (Fig. 2).

Computational latency further prohibits the deployment of this model on edge wearables. Even on a desktop CPU (Fig. 1), inference takes several seconds; transferring this workload to resource-constrained microcontrollers would substantially prolong computation time. A standard solution to this bottleneck is the use of dedicated hardware accelerators [10]. However, in edge wearable systems, this approach introduces significant drawbacks, including increased power consumption, additional physical footprint, and greater overall system complexity.

On the other hand, smaller network can be deployed into advanced microcontrollers. Busia *et al.* [11] deployed a tiny transformer model for the analysis of the ECG signal, requiring only 6k parameters and reaching 98.97% accuracy in the recognition of the 5 most common arrhythmia classes from the MIT-BIH Arrhythmia database. Only 8-bit integer inference was required for efficient execution on low-power microcontroller-based devices. Busia *et al.* run the model on the parallel ultra-low-power GAP9 processor, where inference execution requires 4.28ms and 0.09mJ.

In conclusion, the adaptation of generative, decoder-only transformers to physiological time series offers a promising framework for unified cardiac signal analysis, yet significant hardware bottlenecks hinder their direct deployment on wearable edge devices. As demonstrated by our benchmarking, PPG-PT's memory footprint and compute demands far surpass the Flash, RAM, and power budgets of standard wearable microcontrollers such as the nRF52840 and nRF5340. Future progress in continuous, wearable AI will require co-designing compact model architectures alongside energy-efficient microcontroller platforms, ensuring that the predictive power of physiological foundation models can be employed within strict edge hardware boundaries.

References

[1] Gaudilliere PL, Sigurthorsdottir H, Aguet C, Van Zaen J, Lemay M, Delgado-Gonzalo R. Generative pre-trained transformer for cardiac abnormality detection. In 2021 Computing in Cardiology (CinC), volume 48. IEEE, 2021; 1–4.

[2] Davies HJ, Monsen J, Mandic DP. Interpretable pre-trained transformers for heart time-series data. arXiv preprint arXiv240720775 2024;.

[3] Żyliński M, Nassibi A, Mandic DP. Design and implementation of an atrial fibrillation detection algorithm on the arm Cortex-M4 microcontroller. Sensors 2023;23(17). ISSN 1424-8220. URL https://www.mdpi.com/1424-8220/23/17/7521.

[4] Nassibi A, Zylinski M, Malik A, Davies HJ, Williams I, Papavassiliou C, Mandic D. Bioboard: A multimodal physiological sensing platform for wearables. TechRxiv 2025;URL https://doi.org/10.36227/techrxiv.174285696.68126720/v.

[5] Röddiger T, Küttner M, Lepold P, King T, Moschina D, Bagge O, Paradiso JA, Clarke C, Beigl M. Openearable 2.0: Open-source earphone platform for physiological ear sensing. Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies 2025;9(1):1–33.

[6] Liu S, Zeng C, Li L, Yan C, Fu L, Mei X, Chen F. FoldGPT: Simple and effective large language model compression scheme. arXiv preprint arXiv240700928 2024;.

[7] Mehrgardt P, Khushi M, Poon S, Withana A. Pulse Transit Time PPG Dataset. PhysioNet March 2022;URL https://doi.org/10.13026/jpan-6n92. Version 1.1.0.

[8] Men X, Xu M, Zhang Q, Yuan Q, Wang B, Lin H, Lu Y, Han X, Chen W. ShortGPT: Layers in large language models are more redundant than you expect. In Findings of the Association for Computational Linguistics: ACL 2025. 2025; 20192–20204.

[9] Żyliński M, Nassibi A, Rakhmatulin I, Malik A, Papavassiliou CM, Mandic DP. Deployment of artificial intelligence models on edge devices: A tutorial brief. IEEE Transactions on Circuits and Systems II Express Briefs 2023;71(3):1738–1743. URL https://doi.org/10.1109/TCSII.2023.3336831.

[10] Samanta A, Hatai I, Mal AK. A survey on hardware accelerator design of deep learning for edge devices. Wireless Personal Communications 2024;137(3):1715–1760.

[11] Busia P, Scrugli MA, Jung VJB, Benini L, Meloni P. A tiny transformer for low-power arrhythmia classification on microcontrollers. IEEE Transactions on Biomedical Circuits and Systems 2025;19(1):142–152.

Address for correspondence:

Marek Żyliński
Ul. Św. A. Boboli 8
02-525 Warsaw
Poland marek.zylinski@pw.edu.pl