





Replacing Attention with Modality-wise Convolution for Energy-Efficient PPG-based Heart Rate Estimation using Knowledge Distillation

Panagiotis Kasnesis , Lazaros Toumanidis , Daniele Jahier Pagliari  *Member, IEEE*, Alessio Burrello  *Member, IEEE*

Abstract—Continuous monitoring of Heart Rate (HR) based on photoplethysmography (PPG) sensors is an essential capability of nearly all wrist-worn devices. However, arm movements lead to the creation of Motion Artifacts (MA), affecting the accuracy of HR tracking using PPG sensors. This problem is commonly tackled by exploiting the recorded accelerometer data to correlate them with the PPG signal and eventually clean it. Thus, automatic fusion techniques based on Deep Learning (DL) algorithms have been proposed, but they are considered too large and complex to be deployed on wearable devices. The current work presents a novel and lightweight DL architecture, PULSE, improving sensor fusion by applying a multi-head cross-attention layer to the extracted temporal features. Moreover, we propose a relation-based knowledge distillation mechanism to pass PULSE’s knowledge to a student network that uses modality-wise convolutions to replace the attention module and mimic the teacher’s performance with $5\times$ fewer parameters. The teacher and student are evaluated on two datasets: a) PPG-DaLiA the most extensive available dataset, with PULSE achieving close performance to the best state-of-the-art model, and b) WESAD with PULSE reducing the mean absolute error by 22.6%. The student model is further compressed using post-training quantization and deployed on two commercial-off-the-shelf microcontrollers, demonstrating its suitability for real-time execution, having a close-to-state-of-the-art MAE of 4.81 BPM (+0.40 BPM) on the PPG-DaLiA, but a $10.9\times$ lower memory footprint of 37.9 kB, and consuming $45.9\times$ lower energy (0.577 mJ).

Index Terms—Deep Learning, Sensor Fusion, Knowledge Distillation, Attention, Heart Rate Monitoring

I. INTRODUCTION

During the last decade, the proliferation of the Internet of Things (IoT) has facilitated the development of numerous pervasive computing applications, including Human Activity Recognition (HAR) [1], [2], personalized healthcare, and medical IoT applications [3].

Preventing, diagnosing, and monitoring cardiovascular diseases is one of the most important medical IoT applications. It is commonly performed by 1-12 Electrocardiogram (ECG) leads connected through a chest band for the case of Heart

Rate (HR) monitoring or arrhythmia detection [4] and by cuff-based Blood Pressure (BP) monitors for hypertension [5]. These devices are usually bulky and uncomfortable for the users/patients. Recently, modern techniques rely on compact Photoplethysmographic (PPG) sensors, which are seamlessly embedded within the smartwatches, enabling 24/7 HR and BP monitoring [6], [5].

PPG is a non-invasive device that uses light-emitting diodes and a photodetector at the surface of the skin to measure the variations of light intensity caused by the volumetric variations of blood circulation [7]. The downside of PPG-based HR monitoring is that it suffers from Motion artifacts (MA) that may occur due to shifts in the sensor’s position on the wrist or the intrusion of ambient light between the skin and the sensor; both of which compromise signal quality [8]. Common practices utilize filtering approaches that correlate the motion data coming from IMUs and the PPG signal using classical signal processing techniques, such as adaptive filtering, peak detection, and independent component analysis, to cancel out the noise and eventually remove the MAs [9]. Then, the cleaned signal is extrapolated [10], [11], interpolated [12] or amplified [13] to obtain the HR. However, these approaches use numerous tunable hyper-parameters, which can hinder their ability to generalize across different datasets or scenarios.

Lately, Deep Learning (DL) methods have been introduced to enhance generalization, demonstrating promising outcomes on various publicly accessible datasets. DL architecture, such as ensembles of Convolutional Neural Networks (CNNs) [14], or combinations of convolutional and recurrent layers [15] were first proposed. However, such networks consist of millions of parameters, making their on-device deployment impossible. Lightweight, yet even more effective architectures are proposed in [16], [17] where the authors achieve state-of-the-art (SoA) results in accuracy and model complexity by exploiting Neural Architecture Search (NAS) [18]. On the other hand, attention-based architectures have never been explored for this task, given the large model sizes and high complexity. The upside of this neural network family stems from the so-called *Attention Module* [19], which can correlate and fuse different sensor modalities automatically [20].

In this paper, we demonstrate that attention-based networks are promising even for personalized healthcare on smartwatches and can be deployed on commercial MCUs, by employing Knowledge Distillation (KD) to reduce the network’s complexity and size without hurting its accuracy [21],

P. Kasnesis and L. Toumanidis are with ThinGenious PC, Marousi, Greece (e-mail: pkasnesis@thingenious.io, laztoum@uniwa.gr).

D. Jahier Pagliari, A. Burrello are with the Department of Control and Computer Engineering, Politecnico di Torino, Turin, Italy (e-mail: name.surname@polito.it).

This work has received financial support by Social and hUman ceNtered XR - SUN (EC, Horizon Europe No. 101092612).

This paper was produced by the IEEE Publication Technology Group. They are in Piscataway, NJ.

Manuscript received April 19, 2021; revised August 16, 2021.

[22]. In particular, we use the teacher-student KD learning paradigm, where a "teacher" model is used to train a smaller "student" model to mimic its outputs and achieve similar performance. Using this "teacher-student" KD approach, we replace the resource-intensive feature-level multi-head cross-attention layer with a modality-wise convolutional layer, developing an energy-efficient PPG-based HR estimation DL model that inherits the knowledge acquired by the attention modules. To the best of our knowledge, this is the first attempt to combine attention modules and KD to exploit the cross-modality relationships, pushing the limit of the deployment of lightweight yet state-of-the-art DNNs for HR estimation on commercial off-the-shelf low-power microcontrollers.

The main contributions of our work are the following:

- We present the concept of feature-level cross-attention-based signal fusion instead of sequence-level to discover relationships between the accelerometer and the PPG signals, introducing in the HR-estimation two novel modules, the multi-head cross-attention (MCHA) and the modality-wise convolution.
- We evaluate the precision of the developed model on the PPG-DaLiA [14] and on the WESAD [23] datasets, reaching a Mean Absolute Error (MAE) equal to 4.03 Beats Per Minute (BPM) and 3.75 BPM, respectively, outperforming the best SoA DL model by 0.85 BPM.
- We evaluate the cross-dataset generalizability of our solution by evaluating the trained teacher and student networks on the PPG-DaLiA dataset to WESAD, achieving close to SoA results, with a slight MAE increase (~ 0.40 BPM) for both networks when compared to the versions trained directly on WESAD.
- We propose a novel relation-based KD methodology to train a lightweight student model, which utilizes 1D modality-wise convolutions to replace the multi-head cross-attention layer, representing the cross-modality relationships with sparser connections. After performing a grid search over its hyperparameters, the student model achieves close to SoA MAE (4.49 BPM vs. 4.36 BPM), while requiring $4.8\times$ fewer parameters compared to the teacher and $7.1\times$ fewer parameters compared to the most accurate SoA neural network.
- We perform an ablation study on selecting different input modalities and fusion mechanisms, and on several KD schemes, performing a grid search over the proposed hyperparameters included in the proposed loss function.
- We deploy the *quantized* teacher and student models on 2 ultra-low power commercial-off-the-shelf microcontrollers (MCUs) and compare them with SoA lightweight models in terms of latency and energy consumption. The quantized teacher reduces the inference latency by $1.6\times$ compared to the SoA. The student model has a close to SoA MAE of 4.81 BPM ($+0.40$ BPM compared to the quantized deployed SoA solution), with $10.9\times$ lower memory occupation, and consuming $45.9\times$ lower energy (0.577 mJ).

The remainder of this paper is organized as follows. Section II introduces related works on HR estimation, while Section

III provides a description of the necessary background. Section IV introduces the architectures of the teacher and the student models, and the proposed KD methodology. In Section VI, we present the obtained results, and the final section concludes the article and provides ideas for future steps. It should be noted that a preliminary version of this work has been reported [24].

II. RELATED WORK

Wrist-worn PPG sensors have garnered significant attention in both industry and academia for their potential in HR monitoring and other vital signs such as blood pressure. Early algorithms for PPG-based HR estimation employed simple peak tracking methods, such as the Adaptive Threshold algorithm presented in [30], where a peak tracking algorithm was employed together with a tunable refractory period to eliminate false peaks. Recent approaches can be categorized into two main groups: classical model-driven methods, often combining adaptive filtering with peak tracking and data-driven approaches based on machine or DL. Noteworthy solutions from the literature are summarized in Table I, focusing on algorithms evaluated on the most extended PPG dataset for HR estimation publicly available, PPG-DaLiA.

In the model-driven category, TROIKA [6] paved the way to algorithm exploration, introducing a three-stage approach involving signal decomposition, spectrum estimation, and spectral peak tracking. This method achieved a MAE of 2.34 BPM on the SPC2015 dataset, introduced by the same work. The subsequent work by the same authors [10] improved TROIKA by employing spectral difference with acceleration data, reducing the MAE to 1.28 BPM. Other model-driven algorithms, like those in [31], [26], [12], incorporated methods to mitigate motion artifacts and applied Fast Fourier Transform (FFT) or Wiener filtering for HR tracking, achieving MAEs ranging from 1.03 to 1.37 BPM. Similar results, in terms of MAE, are achieved in [13], where a pipeline consisted of auto regressive spectrum estimation, MA suppression, HR amplification and tracking is proposed for HR estimation during intensive physical exercises. SpaMa, a complex algorithm introduced in [25], stands as the best performer on this dataset, with an impressive MAE of 0.89 BPM. However, model-driven algorithms tend to have numerous free parameters, making them susceptible to overfitting [14]. This issue becomes evident when they are applied to other datasets: for instance, the results reported from SpaMa (and similarly from all other model-driven algorithms developed so far) show a degradation from 0.89 BPMs to 11.06 BPM when tested on the PPG-DaLiA dataset. The latter was indeed subsequently introduced in [14], leading to the development of new model-driven algorithms such as CurToSS [27] and TAPIR [11], with impressive MAEs (5.0 BPM and 4.6 BPM, respectively), but with lower accuracy compared to the state-of-the-art on SPC2015. Additionally, classical model-driven algorithms often involve computationally intensive adaptive filters, which are unsuitable for real-time execution on low-power MCUs. Despite being characterized by a low number of operations, none of the above-mentioned approaches have been indeed deployed on wearable devices with an MCU-class computing platform.

TABLE I

STATE-OF-THE-ART COMPARISON TABLE. DIFFERENT MAE RESULTS AND COMPLEXITIES (IN TERMS OF NUMBER OF OPERATIONS) CORRESPOND TO THE BIGGEST AND SMALLEST ARCHITECTURES REPORTED IN THE ORIGINAL PAPER EVALUATED ON THE PPG-DALIA DATASET.

Work	Pre-Processing	Algorithm	Post-Processing	MCU deployment	Complexity	MAE
Classical methods						
SpaMa, 2016 [25]	0.5-3 Hz filtering, Downsampling	Spectral filtering based on Power Spectral Density	historical tracking, interpolation	No	10.28k	11.06 BPM
Schack2017 [26]	0.5-6 Hz filtering, Downsampling	Corr.-based Frequency indicating func., FFT	threshold	No	12.28k	20.5 BPM
TAPIR, 2020 [11]	0.5-4 Hz filtering	Adaptive filter, Peak detection, Linear Transform.	Notch filter	No	100.0k	4.6 BPM
CurToSS, 2020 [27]	0.5-4 Hz filtering	Sparse Spectrum Reconstruction, Curve tracking	N/A	No	15.04B	5.0 BPM
Deep Learning						
CNN, 2019 [14]	STFT, 0-4 Hz filtering	CNN	N/A	No	480M, 380k	7.65 BPM, 9.99 BPM
Q-PPG, 2021 [16]	0.5-4 Hz filtering	TCNBest	threshold, finetuning	Yes	17.5M, 63.39k	4.36 BPM, 6.07 BPM
ActPPG, 2022 [17]	0.5-4 Hz filtering	TCNBest	threshold, finetuning	Yes	12.3M, 77.6k	4.88 BPM, 5.63 BPM
BeliefPPG, 2024 [28]	0.1-18 Hz filtering, HR Augmentation	FFT, CNN + Transformer	probabilistic	No	Not defined	4.86 BPM
KID-PPG, 2024 [29]	Adaptive, HR Augmentation	Transformer	probabilistic	No	Not defined	3.79 BPM
Our Work	0.5-4 Hz filtering	Transformer + KD	threshold, finetuning	Yes	26.1M, 3.46M	4.03 BPM, 4.49 BPM

More recently, researchers have explored machine and DL approaches for HR tracking, given their success in various biosignal processing applications. While generally more complex, these algorithms are characterized by a very regular workload, which has been demonstrated to be easy to speed up on MCU-class devices [16]. The first CNN architecture coupled with an FFT was presented in [14] and outperformed classical methods [25], [26] on PPG-Dalia, achieving the best MAE of 7.65 BPM. Other works such as [15], [32], [33], [34] employed DL architectures, including CNN+Long Short-Term Memory or denoising CNNs, to remove MAs and analyze clean PPG signals, achieving results comparable to classical methods. However, deploying DL models on memory-constrained MCUs with low energy consumption and real-time latency requirements presents significant challenges due to their large memory footprint and computational demands.

The deployment of deep neural networks for HR tracking on MCUs has been first explored with BinaryCorNET [35], which proposes the implementation of a binary neural network on both application-specific integrated circuits and field programmable gate arrays, achieving an energy consumption of just 56.1 uJ per classification, but with a higher MAE of 6.78 BPM on the SPC2015 dataset. In 2021, a further step was made with Q-PPG [16], where extreme quantization and neural architecture search were exploited to minimize the energy consumption and maximize the accuracy of a TCN, achieving the best performance on PPG-DaLiA of 4.36 BPM, with a network having 17.5M operations. Subsequently, the approach was further generalized in [16], combining different TCNs by means of adaptive inference, slightly sacrificing performance (+0.48 BPM of MAE) for a reduction in operations of 29.71%.

Latest approaches [29], [28] explore using probabilistic theory to improve the HR estimation throughout PPG. Both approaches present impressively low MAE, with KID-PPG achieving the best BPM of 3.79 on the PPG-Dalia dataset (further improved when discarding windows with the highest uncertainty). On the other hand, BeliefPPG [28] converts the regression HR estimation task into a classification task using 64 bins as output and a *Prior* layer, which converts the bins back to HR to estimate the final MAE. This approach achieves a significant MAE of 3.57 when having as input 7 consecutive 8-second windows with 2 sec shift (20 sec in total), but increases to 4.86 BPM when an 8-second window is utilized. However, these methods have still not been deployed on an MCU. Furthermore, the novelty of KID-PPG is not the algorithm applied but the processing pipeline, which can be applied also to the new networks proposed in this work.

Our study marks the first attempt to embed transformer-based models for PPG-based HR tracking onto programmable, general-purpose edge MCUs. To the best of our knowledge, for the first time, we are able to embed an attention-based network for HR estimation on a low-power memory-constrained MCU, exploiting KD to transfer the knowledge of bigger networks to a simpler and more lightweight one, which can be easily fit the memory constraint. As shown in Table I and detailed in the following, our models achieve better or comparable MAE compared to all neural networks in the same

III. BACKGROUND

A. Temporal Convolutional Networks

Temporal Convolutional Networks (TCNs) are 1D-CNNs exploiting the insertion of a fixed gap r (i.e., *dilation* rate)

between the input samples [36], [37]. This leads to an increase in the temporal receptive field used by the convolutional layer, and is described by:

$$\mathbf{y}_t^m = \sum_{m=0}^{C_{out}-1} \sum_{l=0}^{C_{in}-1} \mathbf{x}_{t-r}^l \cdot \mathbf{W}_i^{l,m} \quad (1)$$

where \mathbf{x} is input feature maps, \mathbf{y} denotes the output feature maps, t stands for the output time-step, l is the current layer, and m current the output channel. Moreover, \mathbf{W} represents the filter weights, C_{in} and C_{out} the number of the input and output channels, respectively, and r denotes the dilation factor.

B. Self-Attention Module

Attention module is a layer type of that identifies and highlights the most relevant elements within the input data; multi-head self-attention (MHSA) [19] is the most common mechanism, which receives as input a tensor of sequential data and correlates it with itself. First, the sequence \mathbf{X} is projected to 3 separate tensors, called *query* \mathbf{Q} , *keys* \mathbf{K} and *values* \mathbf{V} :

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^{\text{Query}} \quad \mathbf{K} = \mathbf{X}\mathbf{W}^{\text{Key}} \quad \mathbf{V} = \mathbf{X}\mathbf{W}^{\text{Value}} \quad (2)$$

The \mathbf{Q} , \mathbf{K} and \mathbf{V} tensors are used to calculate the *scaled dot-product attention*, which is expressed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \doteq \mathbf{A} \doteq \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right) \mathbf{V} \quad (3)$$

\mathbf{A} denotes the *scaled dot-product attention*, and d represents the dimensionality of \mathbf{K} , used as a scaling factor. Finally, the h heads are produced using: $\text{head}_i = \mathbf{A}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, concatenated and transformed into \mathbf{E} using again a dense layer.

$$\mathbf{E}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^E \quad (4)$$

1) Cross-Attention Module: In contrast to MHSA, the Multi-Head Cross-Attentional (MHCA) facilitates the fusion of multiple modalities by leveraging the attention mechanism across them; it has been applied in scenarios involving text with images [38], motion sensors [2] and images with LiDAR [39]. Thus, MHCA does not correlate the input tensor \mathbf{X} with itself, but has as input two tensors \mathbf{X}_1 and \mathbf{X}_2 , with one of them forming the *key* \mathbf{K} and *value* \mathbf{V} tensors, and the other the *query* \mathbf{Q} tensor. In the current article, these tensors represent the PPG and tri-axial accelerometer-based feature maps.

C. Knowledge Distillation

Deploying large DL models is challenging, especially for edge devices with limited memory and computational capacity. To tackle this problem, model compression techniques such as KD have been proposed; KD utilizes the outputs of a heavyweight yet very accurate model, called “teacher”, as knowledge to train a more lightweight model, called “student”, so that the latter will be able to mimic the teachers’ outputs, and consequently its performance [40]. The most common type is the response-based KD, that brings the student’s output logits close to the teacher’s, by feeding them both to a loss function L_{Soft} , which usually uses Kullback-Leibler (KL) divergence [41]. KL divergence measures the difference

between the probability distributions of the teacher and student models, encouraging the student to mimic the teacher’s softened predictions, which contain richer information about class relationships than hard labels alone. The distance between the student’s outputs and the original targets is simultaneously reduced by means of another loss term L_{Hard} (e.g., the usual cross-entropy for classification tasks). The weighted sum of these two losses constitutes the final L_{KD} , with β controlling the trade-off between mimicking the teacher’s outputs and adhering to the true labels:

$$L_{KD} = \beta L_{Hard} + (1 - \beta) L_{Soft} \quad (5)$$

However, the KL divergence is not applicable to regression tasks. Therefore, in other types of KD, the teacher passes feature-based knowledge to the student: the student tries to minimize the difference between its feature activations and the teacher’s by including it in the final L_{KD} . Relation-based knowledge is a specific method of this paradigm, where the two models extract the relationships from the input feature maps, with the student trying to replicate the obtained relationships from the teacher instead of the features [42].

KD approaches can be categorized as *offline*, where the common practice is to have an already trained large teacher network and apply batch training on a static dataset, and *online*. The latter is useful in real-time learning set-ups, or where there is the need for continuous model adaptation (e.g., anomaly detection tasks). Here, the teacher and the student models are both updated continuously as they receive new data, with self-distillation (i.e., the teacher and the student are the exact same network) being a subcategory of this approach [43]. In our work, we used an *offline* teacher-student KD which could be categorized as relation-based, exploiting the correlations provided by the attention module. Therefore, we do not apply KL divergence to the last layer but instead consider pre-output feature-base correlation. To the best of our knowledge, this is the first work that exploits the relational reasoning of transformers and replaces it with 1D modality-wise convolutions to apply KD for HR estimation, which is a regression task. Other existing works on KD using wearable-based sensors, are mostly applied to classification tasks such as HAR [44], [45] and emotion recognition [46], [47], with the exception of BP waveform estimation described in [22].

D. Quantization

Neural network quantization is a commonly employed technique aimed at reducing the model’s size and latency, often with minimal or no loss of accuracy. Over the years, various quantization schemes have been developed to achieve this goal. These schemes vary from non-linear quantization algorithms, such as those discussed in [48], which offer remarkable flexibility but are not hardware-friendly, often resulting in improvements in terms of model size but not in terms of latency, to linear quantization schemes, which provide less flexibility and often a slightly higher accuracy loss, but are well-suited for implementation on general-purpose hardware.

In our work, we opted for linear quantization as described in [49] due to its suitability for MCU-based platforms. Lin-

ear quantization, while offering slightly less flexibility than non-linear approaches, has the advantage of being hardware-friendly, leading to much lower latency than floating-point operations. Specifically, linear quantization with an `int8` input format, i.e., our selected precision, usually reduces latency by up to $4\times$ with respect to `int32`, and even more with respect to `float32`, when executed on MCUs that support Single-Instruction Multiple Data (SIMD) integer arithmetic and memory operations.

More specifically, in this work, we leverage an *affine quantizer* that converts the floating point tensor \mathbf{t} (of either weights or activations), with values in the range $[\alpha_t, \beta_t]$ into a N -bit integer tensor $\hat{\mathbf{t}}$ as:

$$\hat{\mathbf{t}} = \text{round} \left(\frac{\mathbf{t} - \alpha_t}{\varepsilon_t} \right) \quad (6)$$

where $\varepsilon_t = (\beta_t - \alpha_t)/(2^N - 1)$ is the smallest value that can be represented in the quantized tensor. While the primary computation is performed in `int8`, the accumulation step of (1) is carried out using `int32` to prevent overflow [49]. This approach ensures that the quantization process remains both efficient and accurate.

Neural networks can be quantized through two primary strategies: after it has been trained, referred to as Post-Training Quantization (PTQ) [50], or during the training process using Quantization-Aware Training (QAT) [51]. QAT, though more resource-intensive as it requires retraining the model starting from its floating-point version, allows for better recovery of performance degradation, particularly when using very low bit-widths. However, this comes at the cost of retraining the model, starting from the floating-point version [51]. Conversely, in scenarios where access to training data is limited or retraining large models is impractical, PTQ is preferred. In such cases, PTQ is often used as a solution. Recent advances in PTQ algorithms have significantly reduced the performance gap, enabling near-lossless quantization in many cases [52]. In our case, we used the PTQ algorithm offered by the TensorFlow library¹, which provided a suitable balance between model accuracy and computational efficiency thanks to its fully-integer linear `int8` quantization.

E. MCUs for wearable devices

Nowadays, the computational core of most commercially available wearables is constituted by an ultra-low-power System-on-Chip (SoC), for instance, an ARM Cortex-M-class MCU or a RISC-V-based SoC. In this work, we have chosen to deploy our models on the Arm Cortex M4 of the STM32H745 evaluation board, manufactured by STMicroelectronics, and on the Greenwaves GAP8 SoC, demonstrating the feasibility of porting them directly on a wearable device.

a) ARM Cortex M4: The STM32L4R9AI board is equipped with a Cortex-M4 core, boasting 640 kB of RAM, and 2MB of Flash. The operational efficiency of this board is evident from its modest average power consumption of 13.63 mW at a clock frequency of 80 MHz [53]. The core

only supports 16-bit SIMD operations, reducing the effectiveness of executing `int8` quantized neural networks, but maintains a high efficiency thanks to the simplicity of the processing pipeline. The decision to utilize this particular platform is also guided by the presence of a software toolchain named CUBE.AI, provided by STMicroelectronics, which is specifically designed for deploying neural networks on their MCUs. Through CUBE.AI, the process of converting pre-trained neural networks from high-level frameworks such as TensorFlow Lite into optimized C code tailored for execution on the target MCU, is streamlined.

b) GAP8: On the other hand, the GWT GAP8 is a commercial Parallel-Ultra-Low-Power (PULP) system with 9 extended RISC-V cores, including one I/O core and an eight-core cluster. The configuration of the GAP8's 'cluster' involves eight 4-stage in-order single-issue pipeline RISC-V cores, designed based on the RISC-V RV32IMCxpulpV2 instruction set architecture, which incorporates the XpulpV2 extension. This extension caters to more efficient and powerful digital signal processing compared to the previously described M4 processor by incorporating features such as hardware loops, post-modified access Load/Store Units instructions, and SIMD instructions for 8-bit vector operands. These features increase the power consumption of the platform but give the possibility to execute complex workloads more efficiently.

The platform is characterized by a 3-levels memory hierarchy: the cores within the cluster share a primary memory level, specifically a 64 kB multi-banked L1 memory, accessible with single-cycle latency. A second level of memory is a 512 kB scratchpad in the SoC domain. The data movement between the L1 tightly coupled data memory and this memory is managed by the so-called 'cluster DMA', which achieves a bandwidth of up to 2 GB/s at peak frequency. Finally, a Cypress Semiconductor's HyperRAM/HyperFlash module concludes the GAPuino board, and it is managed from a GAP8 autonomous I/O subsystem named 'I/O DMA' [54], enabling an extra 64MB of storage for execution.

IV. METHODOLOGY

A. Teacher Network

For the teacher network, we propose the use of a TCN as a feature extractor combined with an MHCA module for effective sensor modality fusion. The developed model architecture is called PULSE (Ppg and imU signal fuSion for heart rate Estimation) and achieves SoA results in terms of MAE in [24], following random hyperparameter optimization.

The TCN part comprises 3 convolutional blocks to extract features out of the input modalities, where at every block the C_{out} increases. Each block contains 3 consecutive dilated 1D convolutions (similar to [16]), having same channel, kernel size equal to 5 and dilation rate equal to 2. Moreover, every 1D convolution is followed by a ReLU activation function. At the end of each convolutional block a 1D average pooling operation is performed to reduce the output values and extract statistical features from the input tensor, while dropout ratio equal to 0.5 is applied to reduce the network's overfitting.

After being processed by the three convolutional blocks, the tensors are fed to the MHCA module (Fig. 1). Here, the

¹https://www.tensorflow.org/model_optimization/guide/quantization/post_training

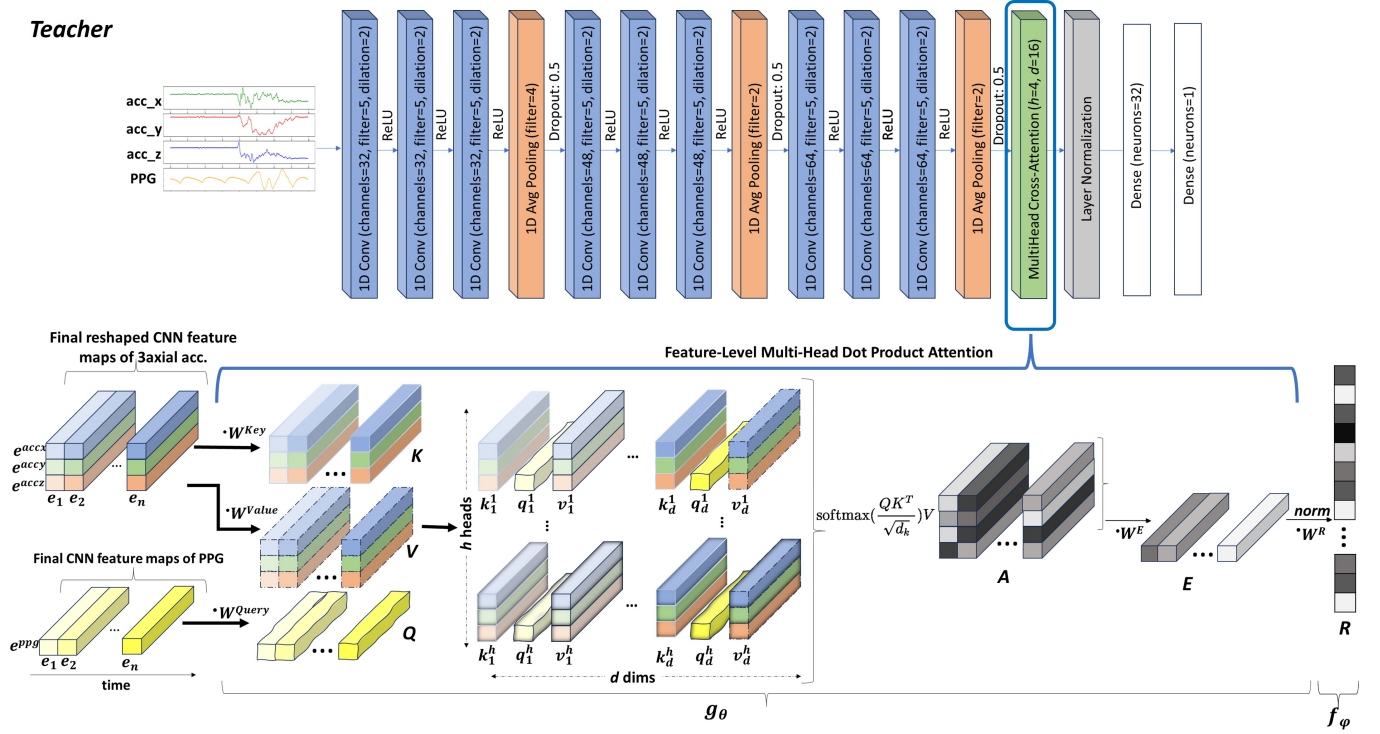


Fig. 1. Feature-level multi-head cross-attention module. The extracted set of features e^{acc} are converted to \mathbf{K} and \mathbf{V} , while the e^{ppg} to \mathbf{Q} . These are then used to compute the feature-level attention \mathbf{A} consisting of 4 heads. \mathbf{A} is converted to \mathbf{E} using a dense layer, which encapsulates all the extracted relations from PPG and accelerometer signals. \mathbf{E} is finally transformed to \mathbf{R} using f_ϕ .

PPG acts as a *query* vector \mathbf{Q} after been multiplied with \mathbf{W}^{Query} , while the tri-axial accelerometer embeddings as Key \mathbf{K} and Value \mathbf{V} vectors, multiplied with \mathbf{W}^{Key} and \mathbf{W}^{Value} , respectively. Thus, for h heads and d dimensions, the Feature-Level (FL) cross-attention is computed, and, afterwards, the h dot products (i.e., 4 for the case of PULSE) are concatenated and transformed into \mathbf{E} using a matrix multiplication with \mathbf{W}^A . After applying the MHCA the \mathbf{E} tensor is then passed to a layer normalization operation and, finally, to 2 consecutive dense layers that output the predicted HR. The architecture of the teacher is illustrated in the upper part of Fig. 3.

Compared to existing works on signal processing, this type of network has two novelties: a) FL instead of Sequence Level (SL) attention, and b) cross-attention instead of self-attention. The suggested fusion module feed to the high-level features derived from temporal convolutions applied to short-term changes in both tri-axial acceleration and arterial translucency. Furthermore, the capacity of PPG is limited when it comes to hand placement, since the blood volume oscillations are phase delayed compared to the ECG, resulting to a decreasing recording capability of the temporal dynamics of small numbers of cardiac cycles [55]. On the contrary, the MA affects immediately the PPG readings.

To address this, we used dilated convolutions along with pooling operations leading to a receptive field equal to 10 seconds after the final convolution operation, using (7) presented in [56]). In particular, the receptive field RF_0 is given by:

$$RF_0 = \sum_{l=1}^L \left((r(k_l - 1)) \prod_{i=1}^{l-1} s_i \right) + 1 \quad (7)$$

where k_l is the kernel size, s_l is the stride, and r is the dilation rate. Thus, in order to synchronize the corresponding blood volume readings with the hand movements and remove the MA we applied convolution operations. In addition to this, we propose FL attention enabling the model to discover patterns that correlate the effect of each movement axis on the blood volume variation and the MA over a specific time granularity instead of SL attention, which is an operation applied over the time domain similar to the one performed by the convolutional layers, making it somewhat redundant.

More specifically, the resulting 10 sec receptive field is larger than the input time window (i.e., 8 sec), thus, the produced set of feature maps $e = \{e_1^1, e_2^1, \dots, e_1^2, e_2^2, \dots, e_n^s\}$ considers almost all the signal values included (i.e., some values are duplicated due to padding). In particular, 16 feature maps are extracted for each modality, where e_n^m is the n -th embedding measured by the s -th sensor modality; these maps can be thought of embeddings representing short-term human actions. Such an action maybe an arm back and forth movement during walking activity or steering the wheel during driving activity.

This concept of short-term actions is similar to works relying on Relational Reasoning (RR), such as [57], [2], thus, we could describe the extracted relations from the input signals using the term \mathbf{R} (see Fig. 1) and the following equation:

$$\mathbf{R} = f_\phi \left(\sum_i (g_\theta(e_i^{ppg}, (e_i^{acc_x}, e_i^{acc_y}, e_i^{acc_z}))) \right) \quad (8)$$

where f and g are differentiable functions with parameters θ and ϕ . In particular, the g_θ function is responsible for

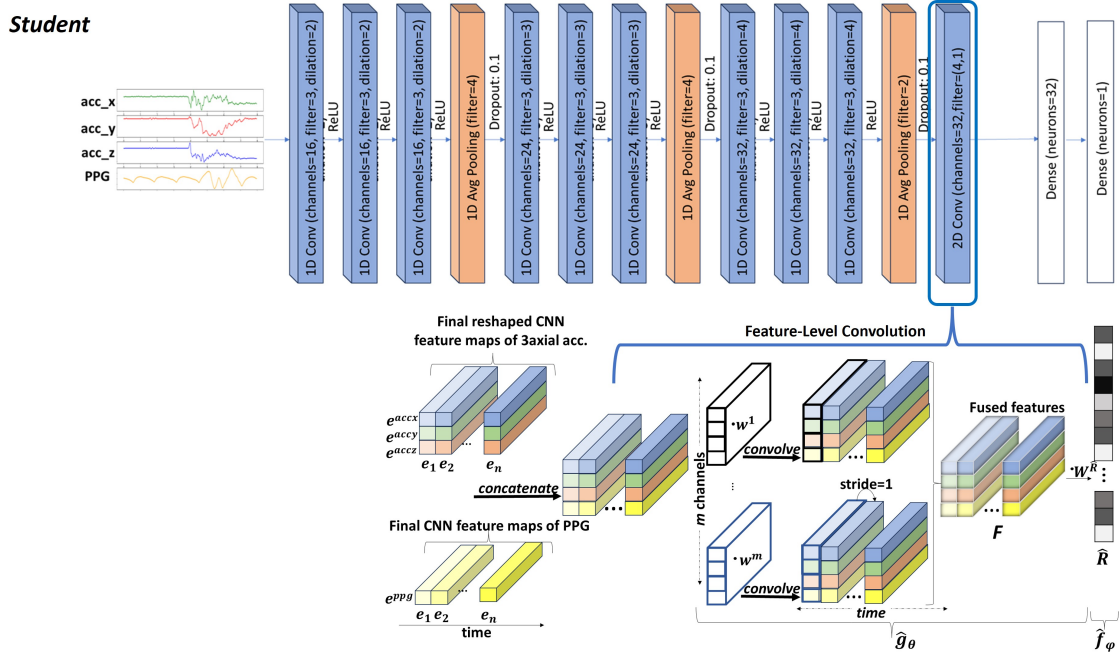


Fig. 2. Modality-wise convolutional module. The extracted set of features e^{acc} are concatenated with e^{ppg} . These then fed to a 1D convolutional modality-wise convolutions consisting of m channels. The convolved output \mathbf{F} is finally transformed to $\hat{\mathbf{R}}$ using \hat{f}_ϕ .

discovering the relationships between the actions, replacing (2),(3),(4), while f_ϕ produces the term \mathbf{E} .

One other worth mentioning aspect of this work compared to existing ones, is that fact that we selected cross-attention instead of self-attention. FL MHCA is responsible for applying relational reasoning over the PPG and the tri-axial accelerometer embeddings, while for the case of MHSA the PPG embeddings would be also correlated with themselves (leading to quadratic complexity [58]), needing more computational resources. Similar works on sensor signals use patching and extract features in the first convolutional layer, and utilize, afterwards, SL MHSA (e.g., Bioformer [59]). The work presented in [39] could be considered as the most similar using feature-based cross-attention applied to fuse images with LiDAR points with, but without taking the time domain into account. In particular, the LiDAR features representing a voxel containing a subset of points and the image features corresponding to camera pixels are already synchronized making the alignment a one-voxel-to-many-pixels problem. Finally, we developed the FL MHCA to condition the PPG over the 3 acceleration channels after being inspired by the domain expertise used in classical methods, which correlates the PPG and acceleration signals [60].

B. Student Network

As student network, we employ a late fusion approach [61], introducing the concept of modality-wise 1D convolution applied over the extracted feature maps. This is a novel sensor fusion technique that captures the intrinsic correlations among multi-source signals [62], which to the best of our knowledge has not been previously used for fusing time-series data. The intuition for designing the student model is to replace

the MHCA module with modality-wise convolution, which is lighter computationally and supported by several MCUs.

The architecture of the student network is depicted in the lower part of Fig. 3. Similarly to the teacher, it consists of 3 convolutional blocks for the feature extraction part, while using smaller kernel sizes (3 instead of 5) and half of the teacher's channels (i.e., 16, 24 and 32). In particular, each block comprises 3 consecutive dilated 1D temporal convolutional layers. The dilation rate is increasing after each block (i.e., 2, 3 and 4) in order to expand the receptive field of the student's filters without increasing the number of its trainable parameters. It should be mentioned, that each convolutional block is followed by an 1D average pooling operation to reduce the size of the output feature maps, and a dropout ratio equal to 0.1. This value is significantly lower than the teacher's since the student is less prone to overfitting.

After the feature extraction part, the modality-wise convolution is utilized to fuse and correlate the processed signals, similarly to an FL MHSA module. For this purpose, it consists of one 1D convolutional layer with kernel size 4 applied to the sensor modalities of the extracted feature maps and 32 channels. During the experimentation phase, we considered several alternative architectures exploring different hyperparameters (Section V-C), and we examined fusing the extracted e_i^{ppg} with each e_i^{acc} separately. However, as shown in V-G this type of architecture is significantly larger and less effective.

The convolved values are then passed to a relational fully connected layer followed by a linear activation function to produce the extracted relational dependencies vector $\hat{\mathbf{R}}$. The high-level equation to compute $\hat{\mathbf{R}}$ can be represented as:

$$\hat{\mathbf{R}} = \hat{f}_\phi \left(\sum_i (\hat{g}_\theta(e_i^{ppg}, e_i^{acc_x}, e_i^{acc_y}, e_i^{acc_z})) \right) \quad (9)$$

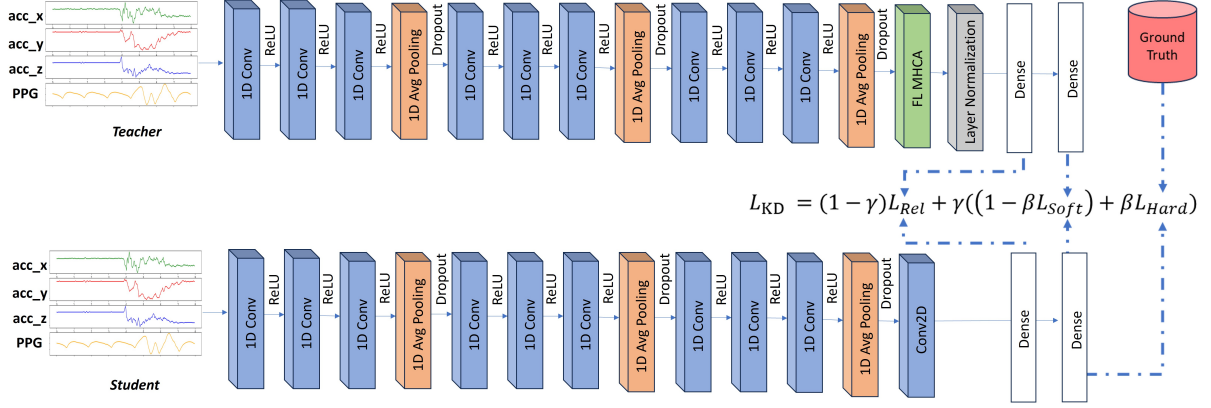


Fig. 3. Detailed illustration of the teacher-student KD. The teacher network is represented in the upper, while the student in the lower area. The distance of the outputs of prefinal dense layers of the teacher and the student model, L_{Rel} , is combined with the distances L_{Soft} and L_{Hard} to compute L_{KD} and backpropagate the error to the student's weights.

Thus, objective of the student network, as an intermediate step to precise HR estimation, is to discover the same relationships between the extracted PPG and IMU features maps as the teacher model ($\hat{\mathbf{R}} \equiv \mathbf{R}$).

Finally, $\hat{\mathbf{R}}$ values are passed to the output fully connected layer, followed by a linear activation function.

C. Teacher-Student Knowledge Distillation

When it comes to teacher-student KD, as aforementioned, there exist two losses, the L_{Hard} and the L_{Soft} [40]. L_{Hard} is similar to the loss used for training the teacher network, so for our case it is estimated using MAE:

$$L_{Hard} = \frac{1}{N} \sum_{i=1}^N |y - \hat{y}_s| \quad (10)$$

where y denotes the ground truth values and \hat{y}_s the predicted ones, by the student model. Unlike classification tasks where KL divergence is the most common equation to define L_{Soft} , since HR estimation is a regression task, we define L_{Soft} by subtracting the predicted HR of the teacher network with that of the student network, as follows:

$$L_{Soft} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_t - \hat{y}_s| \quad (11)$$

where \hat{y}_t represents the predicted values of the teacher.

Apart, from these two common losses, inspired by [4], we introduce an additional L_{Rel} loss. The latter computes the error between outputs of the pre-final dense layers of the teacher and student, which are \mathbf{R} and $\hat{\mathbf{R}}$ respectively, as follows:

$$\begin{aligned} L_{Rel} &= \frac{1}{N} \sum_{i=1}^N (\text{softmax}(\mathbf{R}) - \text{softmax}(\hat{\mathbf{R}}))^2 \\ &= \frac{1}{N} \sum_{i=1}^N (\text{softmax}(z_{t(i)}^{L-1}) - \text{softmax}(z_{s(i)}^{L-1}))^2 \end{aligned} \quad (12)$$

where z_t and z_s , depict the pre-activation outputs of the teacher and student network's penultimate layers, respectively.

In contrast to [4], using (12) we compute the distance-wise distillation losses over the extracted relationships from the feature maps, instead of the feature maps themselves. This is advantageous for two reasons: a) the fused multimodal feature maps are more semantically enriched representations than unimodal ones and b) relation-based KD offers architectural flexibility when selecting the hyperparameters of the student's convolutional blocks. In particular, in feature-based KD, the filter channels used to produce the latent representations must have the same size as those of the teacher's to be mathematically applicable, thus, leading to heavier architectures.

Moreover, instead of directly feeding the output neurons of the $L - 1$ layer to L_{Rel} , we used *softmax* for two reasons: a) numerical stability and b) smoothness. When the outputs of the pre-final dense layers differ a lot, the L_{Rel} computation can incur numerical overflows, while *softmax* normalizes the outputs and is less prone to numerical issues. Moreover, *softmax* is a smooth function, which means it is continuously differentiable. This property is crucial for gradient-based optimization algorithms. In the next section, we perform an ablation study to showcase the need of including *softmax* before the computation of L_{Rel} . The proposed complete L_{KD} is defined as follows:

$$L_{KD} = \gamma(\beta L_{Hard} + (1 - \beta)L_{Soft}) + (1 - \gamma)L_{Rel} \quad (13)$$

where β and γ are considered as hyperparameters of this loss function, having range $[0,1]$. Consequently, when γ equals 1 L_{Rel} is not used during training, and in case, also, β is 1 then the student network is trained without using KD. Different combinations of these hyperparameters are explored in the ablation study presented in the following section.

V. EXPERIMENTAL RESULTS

For our experiments we used a computer workstation equipped with a NVIDIA RTX 3090 GPU featuring 24 gigabytes RAM, 10,496 CUDA cores and a bandwidth of 484

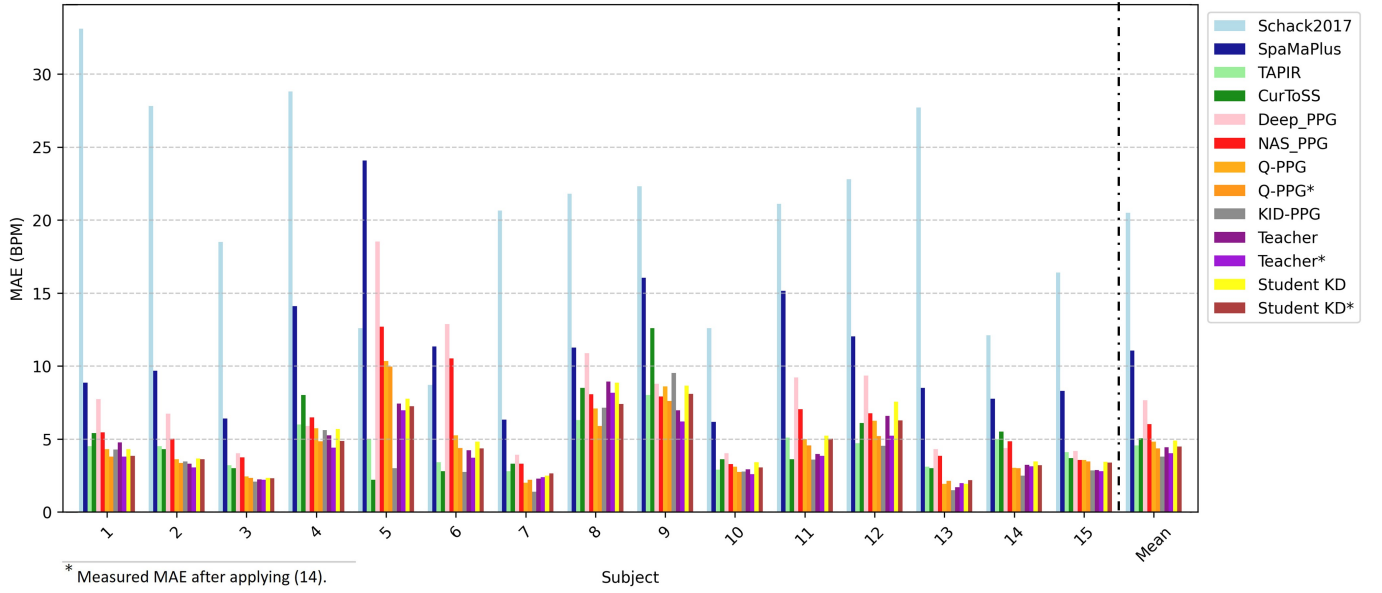


Fig. 4. Per subject MAE performance of teacher and student network on the PPG-DaLiA compared to the state-of-the-art. The * symbol denotes that the model's outputs have been post-processed using (14).

GB/s. We used Python 3.8, with the Numpy library for data preprocessing and segmentation, and the PyTorch framework to develop and train the networks.

A. Datasets

We used PPG-DaLiA (PPG dataset for motion compensation and HR estimation in Daily Life Activities) as the main dataset [14] for our experiments; it is a large and highly benchmarked public dataset including physiological and motion data for PPG-based mainly built for HR estimation. During data collection the subjects wore a E4² on their non-dominant wrist, producing one-channel PPG (sampling rate 64Hz), tri-axial accelerometer (sampling rate 32Hz), EDA and body temperature (sampling rate 4Hz) signals. For collecting the ground truth values they also wore a smart belt on their chest equipped with ECG. The dataset includes 15 subjects (aged 21–55 years), who performed 8 common everyday activities such as walking, driving, and cycling. The activities are provided as extra labels for HAR, and have been utilized in [17] to employ dynamically different models for inference.

To assess the generalizability of our approach, we also evaluated the proposed approach on the WESAD dataset [26], a multimodal dataset for wearable-based stress detection. The authors follow the same data acquisition setup with PPG-DaLiA using the same devices worn by 15 subjects. However, WESAD aims to detect and distinguish different affective states, with subjects performing mainly sedentary activities. Since the target HR values are not provided we processed the included ECG data to identify the R-peaks and calculated the mean instantaneous HR within each 8-second window to obtain the ground truth HR data as done in the SoA [14].

B. Experimental Set-up

We validated all models following the cross-validation protocol proposed in [14], [16], denoted as Leave-One-Session-Out (LOSO) Cross-Validation (CV). In particular, the 15 subjects are divided into four data folds; three are used as training set and the other one is split to create the test set (1 subject), and the validation set. Applying LOSO CV in PPG-DaLiA leads to 15 training iterations, validating this way the model's generalizability to unseen subject data. The PPG signals were downsampled to 32 Hz to match the IMU sampling rate, leading to windows of shape 4×256 . The dataset includes 64,697 samples after being segmented using an 8-second window and 80% overlap. The windows are then normalized using per-sensor modality z-score: $z_i = \frac{(x_i - \mu_i^{train})}{\sigma_i^{train}}$, where x_i denotes the samples of sensor modality i , while μ_i^{train} , σ_i^{train} depict the mean and standard deviation values, estimated on the training set.

Adam [63] is selected as network optimizer, with a learning rate of 0.0005 for training the teacher and 0.001 for the student network, following a hyperparameter search in [0.0001, 0.001] with step 0.0005. We set β_1 to 0.9, β_2 to 0.999, and ϵ_1 as done in previous works using similar networks for the same task [16], [24]. The network is trained for 500 epochs, with 256 batch size and a patience of 100. The weights of the validation model that had the lowest MAE are stored and used for evaluation on the test set. Finally, we used the post-processing method included in [16], with the estimated output values \hat{y}_n being clipped in case the prediction is more or less than a P_{th} percentage of the averaged N last estimated values:

$$\hat{y}_n^{clipped} = clip(\hat{y}_n, \sum_{i=n-N}^{n-1} \hat{y}_i \pm \frac{1}{P_{th}} \sum_{i=n-N}^{n-1} \hat{y}_i) \quad (14)$$

setting P_{th} and N equal to 10.

²<https://www.empatica.com/research/e4/>

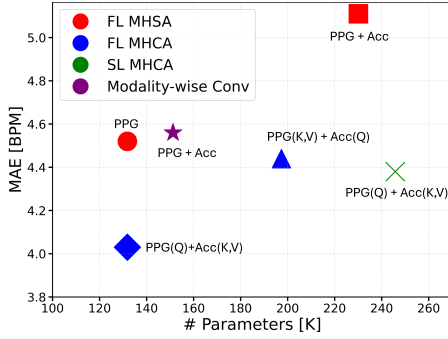


Fig. 5. Ablation study on input sensor modalities and fusion types.

C. Results on PPG-Dalia

Fig. 4 presents the MAE results produced by the teacher network (PULSE) and its distilled version compared with SoA DL-based and mathematical models. The teacher network achieved an average MAE of 4.44 BPM, which improved to 4.03 BPM with post-processing. For most subjects, PULSE surpasses or is comparable to the SoA performance of both classical and DL models; the nearest competitor is KID-PPG, which reaches the best performance of 3.79 BPM of MAE. Notice that this model strongly employs data augmentation and probabilistic modeling to improve out-of-distribution patients (patients with extremely high or low BPM). For instance, on subject 5, which has the highest average HR among all the patients of 125.84 BPM, KID-PPG improves the MAE compared to our method by an impressive 3.94 BPM. Nevertheless, KID-PPG's proposed techniques are orthogonal to our approach and can be identically applied to our pipeline. The worst MAE obtained by PULSE is equal to 8.17 BPM (subject 8), but still enables a robust evaluation of the health status. It is worth noticing that KID-PPG/Q-PPG obtains high MAE on this patient, demonstrating the difficulty of its raw signal, probably contaminated by many MAs.

Fig. 5 presents a comparison of input modality configurations, examining the use of PPG alone versus the fusion of PPG with motion data, along with various combinations applied within the attention mechanism. Noticeably, the selected FL MHCA enabled architecture (blue diamond) is not only the most accurate one, but also has the least trainable parameters. Same number of parameters are used when having solely the PPG as input and the FL MHSA module (red circle), but a higher MAE is obtained (4.52 BPM). On the other hand, when the MHCA module is modified and PPG embeddings are used as \mathbf{K} and \mathbf{V} tensors (blue triangle), both the MAE and the number of parameters are increased to 4.44 BPM and 197.36k, respectively. Furthermore, the MHSA-based network with PPG and accelerometer signals as input (red square) leads to a larger architecture than MHCA and achieves the best MAE on several subjects, but does not generalize across some others, causing a worse overall average MAE. This can be interpreted as it interpolates well on data distributions similar to the one contained in the training set, while it underperforms when it comes to extrapolation (i.e., the test subject's data distribution differs significantly from the training set's data).

TABLE II
MAE PERFORMANCE OF TEACHER AND STUDENT NETWORK ON THE WESAD COMPARED TO THE STATE-OF-THE-ART.

Work	MAE	MAE*
SpaMa, 2016 [25]	9.45	-
Schack2017 [26]	19.97	-
DeepPPG [14]	7.47	-
BeliefPPG [28], using 8-sec window	4.60	-
BeliefPPG [28], using 7x8-sec window (2s shift)	3.88	3.72
Teacher	3.97	4.24
Teacher, after applying (14)	3.75	4.16
Student	4.16	4.53
Student, after applying (14)	4.01	4.38

* Trained on PPG-DaLiA and evaluated on WESAD.

For example, it produces an 11.21 MAE for subject 5 since its records contain very high HR values, which are rarely encountered in training data, and are therefore badly predicted. Our intuition is that MHCA prevents overfitting and guides the network to exploit correlations similar to the ones of SoA classical algorithms. Additionally, the table highlights that using attention across the sequence dimension impacts the outcomes, revealing that FL attention offers better efficiency and accuracy compared to the SL MHCA approach (green "X"), given that the teacher network has a receptive field of 320 values (i.e., 10 secs, see Section IV-A).

Finally, we considered using the same hyperparameters included in the convolutional blocks of the teacher architecture and then applying modality-wise convolutions instead of attention (purple star), which results in a student-alike architecture, but almost $5.5\times$ larger. The obtained results were close to those of Q-PPG, reaching 4.56 MAE which could be improved further if a NAS method is applied, as presented in [16], [64].

D. Results on WESAD

To evaluate the generalizability of the proposed relation-based KD, we also trained the teacher model on the WESAD dataset using the same LOSO CV approach. Furthermore, we applied the same distillation approach to train our student network on this dataset, maintaining a constant memory footprint and latency of the networks trained on the PPG-Dalia dataset. As Table II presents, both the teacher and student networks obtain significantly lower MAE than the state-of-the-art, achieving a 0.63 BPM and 0.44 BPM decrease compared to the best competitor, BeliefPPG. After the application of post-processing, the improvements are increased to 0.85 and 0.59 BPM, achieving the best SoA MAE of 3.75 BPM (comparison with KID-PPG is not present, given that KID-PPG has been evaluated only on PPG-Dalia dataset). The student model achieves the second-best MAE at 4.01 BPM, with a substantially smaller memory footprint. It is also worth noting that the teacher model outperformed BeliefPPG by 0.13 BPM in MAE, even when the latter utilized seven overlapping 8-second windows with a 2 sec shift for the latter (20 sec in total). In addition to this, we evaluated the previously trained teacher and student models (V-C) on the PPG-DaLiA dataset using WESAD as a test dataset. Although the MAE increased slightly for both the student (0.37 BPM) and the teacher (0.41 BPM) when compared to the versions trained directly

on WESAD, the increase is small, proving the cross-dataset generalizability of our solution.

E. Student Hyperparameter exploration

We used a constrained random search for selecting the student model's hyperparameters. In particular, we set as a prerequisite the modality-wise convolution and experimented by changing the number of convolutional blocks, the number of convolutional layers per block, the dilation rate r , the kernel sizes, and the output channels per convolution C_{out} . Moreover, we constrained our models so that they did not exceed the size of the teacher network; otherwise, KD would not be meaningful in reducing the model size. The most significant results are presented in Fig. 6. The best results (i.e., lower MAE) of 4.47 BPM and 4.42 BPM were achieved by the 4 convolutional block architecture (red triangle) and by the 3 blocks one with $C_{out}=[32,48,64]$ (rightmost red circle). However, we did not select these architectures since they are computationally expensive, leading to 14.6 and 13.4 MOps. On the other hand, the 3-block architecture with $C_{out}=[16,24,32]$ leads to a 4.53 BPM as MAE, while only requiring 3.46 MOps. We used dilation r equal to [2,3,4] in the three blocks. The other colored points (blue, green, and purple) correspond to architectures where we reduced the dilation rate to 2 in all layers (blue circle) or even to 1 (green circle) while increasing the kernel sizes (purple circle). However, these modifications did not result in any performance improvement.

F. Window shift exploration

We also explored the non-perfect synchronization between the PPG signals and motion signals. Fig. 7 shows that there are approximately 38 samples of phase delay when comparing the PPG signal with the x-axis of the accelerometer motion artifact, which is equal to 1.2 sec (having sampling rate equal to 32Hz). This delay is a result of the combined effect of both the expected delay of the MA effect on the blood volume readings, but also of the hardware used in this dataset (Empatica E4). To understand the effect of this delay, we examined different window shifts between the PPG and the accelerometer signals by shifting the PPG signal by 16, 32, 48, and 64 samples. On the other hand, experimentally, using the same hyperparameters as those reported in Section IV.B for the student (i.e., having a receptive field equal to 10 seconds), we did not observe any statistically significant differences, with the results ranging from a decrease of -0.024 (32 shifted values) to an increase of +0.135 BPM (64 shifted values). This variation could be due to different weight initializations are used and cannot be attributed to the addition of the shift.

G. Ablation study on knowledge distillation loss

In order to select the best values for the hyperparameters β and γ included in (13) we performed a grid search presented in Fig. 8, using a fixed student network topology. The lowest BPM error was obtained using β equal to 0.5 and γ equal to 0.33, which means that all of the losses (i.e., L_{Hard} , L_{Soft} and L_{Rel}) have the same contribution to the overall L_{KD} .

The most important results from this search along with removing the *softmax* function from L_{Rel} , are also reported in Table III. First of all, without using any type of KD (i.e., the student is trained only using L_{Hard}) the obtained MAE is

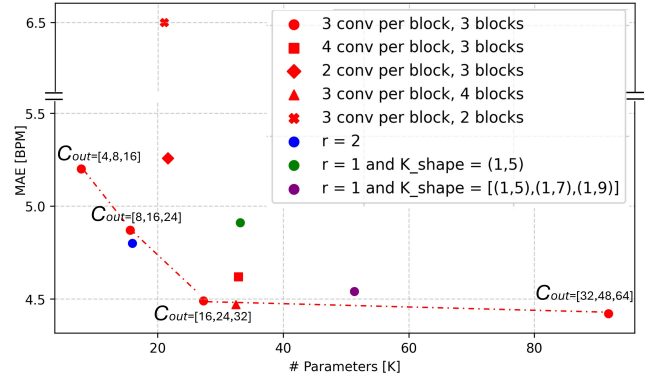


Fig. 6. Hyperparameters' effect on the student. The red circle with increasing channels per block $C_{out}=[16,24,32]$ corresponds to the selected architecture.

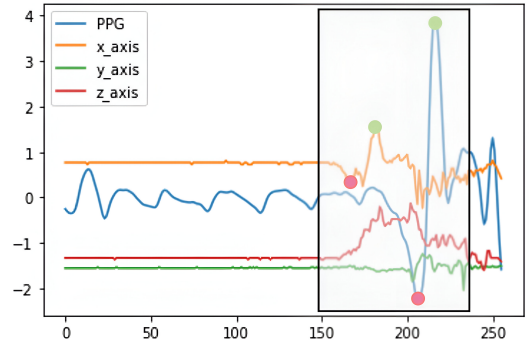


Fig. 7. The observed phase delay between the accelerometer values and the PPG signal. The red and green dots indicate significant movements on the x-axis and their corresponding effect on the PPG signal.

equal to 5.25, which is 17% higher than the proposed relation-based approach. Moreover, the common response-based KD method (i.e., not considering the L_{Rel}) obtains 4.74 MAE, while the non-*softmax*-enabled L_{Rel} approach results in a MAE equal to 4.83 BPM. This occurs because, for several subjects the student relations differ a lot from those of the teacher's, and since no activation function is applied to $\hat{\mathbf{R}}$ and \mathbf{R} , this causes numerical instability during gradients' computation, and the network struggles to converge [65]. On the other hand, using the *softmax* function compresses the L_{Rel} values into a range between 0 and 2 (i.e., it is normalized) which is beneficial for the optimizer, since the algorithm converges faster as shown in Fig. 9.

TABLE III
ABLATION STUDY ON DISTILLATION TYPES AND STUDENT NETWORKS.

KD Approach	Network	MAE	#KParams
None ($\beta = 1, \gamma = 1$)	Student	5.25	27.41
Response-based ($\beta = 0.5, \gamma = 1$)	Student	4.74	27.41
Relation-based ($\beta = 0.5, \gamma = 0.66$)	Student	4.83	27.41
Relation-based ⁺ ($\beta = 0.5, \gamma = 0.66$)	Student	4.49	27.41
Relation-based ⁺ ($\beta = 0.5, \gamma = 0.66$)	Student ^φ	4.66	41.58
Relation-based ($\beta = 0.5, \gamma = 0.66$)	Student ^θ	4.69	47.69

⁺ *softmax* function included in L_{Rel} .

^θ No modality-wise convolutions included (i.e., replaced by a dense layer).

^φ Modality-wise convolutions are applied to the extracted e_i^{ppg} with each e_i^{acc} separately.

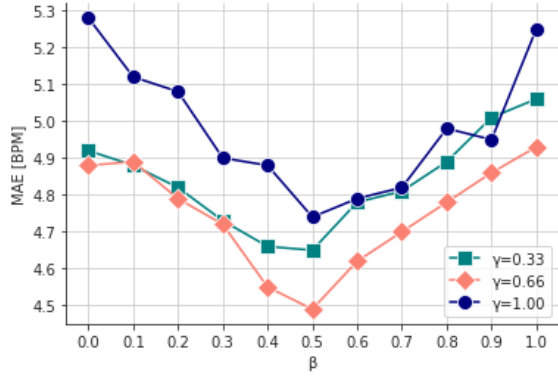


Fig. 8. Hyperparameters' β and γ effect on the obtained student KD MAE.

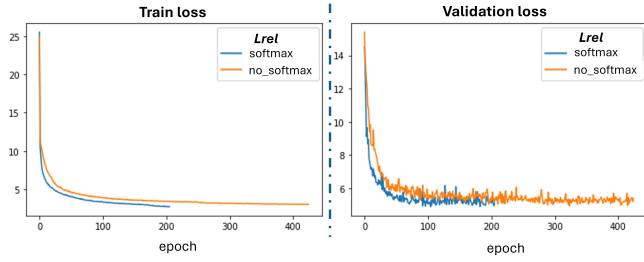


Fig. 9. Per-epoch decrease of the training (Subject 5, left) and validation loss (Subjects 6,7,8, right) when including *softmax* or not (orange line).

Furthermore, as mentioned in Section IV-B we considered to separately fuse the produced PPG feature maps with each accelerometer axis, which was the most intuitive thing to do to mimic the teacher's MHCA module. However, this architectural choice results in higher BMP error and increases the network's size by almost 50%, since this fusion outputs three tensors instead of one. In addition to this, we examined substituting completely the modality-wise convolutional layer with a fully connected layer. This architectural decision led to a 0.2 error increase, while having around 75% more parameters than the proposed fusion layer.

Finally, Fig. 10 illustrates the MAE versus parameters trade-off achieved by the developed teacher and student models, against the most lightweight SoA DL-based models. As shown, the teacher achieves the lowest MAE, and also a lower size with respect to SoA models having MAE < 5.00 BPM. The lightest model is TimePPG-Small [17], having 5 \times less parameters than the developed student, which however surpasses it significantly in accuracy (1.14 BPM lower MAE).

H. Deployment on wearable devices

1) *Arm Cortex M4*: In this section, we present the deployment results of both the teacher network and the knowledge-distilled network on an M4 MCU. The experimental setup involved using a STM32L4R9I-EVAL development board with a frequency of 80 MHz. Noteworthy, this platform is a proxy to simulate the deployment on a typical wearable device that exhibits MCUs of the same class. To deploy the neural networks, we used the STM32CubeMX tool with the X-CUBE-AI package, version 8.0.0. All networks underwent

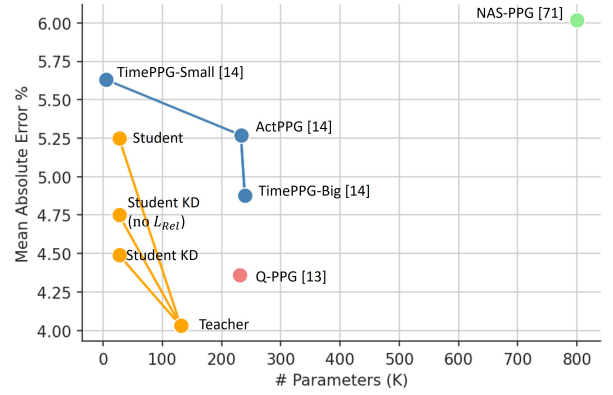


Fig. 10. Presentation of the tradeoff between number of parameters and achieved MAE of the student (with and without KD) and teacher models, compared to lightweight SoA DL-based models.

post-training quantization (using Tensorflow Lite quantization) to *int8*. Furthermore, dilated filters are filled with 0 to address constraints imposed by the toolchain, preventing the use of quantization and dilation higher than 1. Table IV shows the results of our model, together with the SoA models of [17] in terms of memory, power, latency, energy consumption, and MAE. The memory occupation is extracted from the CUBE-AI tool, while the average power consumption has been extracted from the STM32L4R9AI datasheet for the running mode at 80 MHz. The energy is computed as the product of latency (measured on the actual hardware) and average power consumption. The reported energy exclusively accounts for the network execution without considering additional energy contributions from the sensors or during idle periods. This is done to isolate and compare the efficiency of different network architectures without external confounding factors.

The distilled model achieves an impressive low latency of 169.6 ms, 11.2 \times lower compared to the most accurate SoA network (Q-PPG-L), with an energy consumption as low as 2.311 mJ, while achieving a near-SoA performance of 4.81 BPM of MAE. Furthermore, our model achieves 10.9 \times lower memory footprint (37.9 kB). TimePPG-Small is the fastest model from the SoA, requiring only 19.02 ms to run inference and consuming only 0.259 mJ on an M4 core. However, it achieves a critically higher MAE of 5.60 BPM.

The most accurate models from the SoA are TimePPG-Big and Q-PPG-L, with 4.87 BPM and 4.41 BPM, respectively. However, both networks achieve substantially higher latency and energy consumption. The first one runs on an M4 core, achieving a latency of 1301.85 ms (7.68 \times more) for processing and consuming 17.74 mJ. The second one runs on the STM32WB55, featuring the same M4 MCU, with a latency of 1.9 seconds and an energy consumption of 47.65 mJ. Our most accurate quantized network, the teacher attention-based model, outperforms them on all metrics: it achieves the best MAE (4.09 BPM), a lower latency of 1202 ms, and an energy consumption of 16.385 mJ per inference. Note that the *fp32* version of the teacher required more RAM and Flash memory than the one provided by the MCU, and was not deployed.

TABLE IV
DEPLOYMENT OF THE DEVELOPED MODELS ON STM32L4R9AI (CORTEX-M4 CORE) AND GAP8

Model	Ram [kB]	Flash [kB]	params	OPs	MCU	P. [mW]	Cycles	Fr.[Hz]	lat. [ms]	E. [mJ]	MAE
TimePPG-Small ^{†‡} [17]	13.31	8.55	8.76k	224.8k	STM32L4R9AI	13.63	1.52M	80M	19.02	0.259	5.60
TimePPG-Big ^{†‡} [17]	34.06	884.26	902.2k	33.3M	STM32L4R9AI	13.63	104.1M	80M	1,301.85	17.74	4.87
ActPPG ^{†‡} [17]	129.64	922.25	239.2k	6.17M	STM32L4R9AI	13.63	52.26M	80M	653.28	8.903	5.27
Q-PPG-L ^{†‡} [16]	411.9	-	-	24.4M	STM32WB55	13.63	-	-	1,900.0	47.65	4.41
Teacher* [‡]	85.87	308.31	211.9k	46.6M	STM32L4R9AI	13.63	96.17M	80M	1,202.2	16.385	4.09
Student KD [‡]	106.37	170.23	37.84k	5.82M	STM32L4R9AI	13.63	52.03M	80M	650.43	8.865	4.49
Student KD* [‡]	47.95	98.92	37.84k	5.64M	STM32L4R9AI	13.63	13.59M	80M	169.6	2.311	4.81
Student KD *	37.90	78.22	27.2k	3.46M	GAP8	262.2	572.5K	260M	2.20	0.577	4.81

[†] With `int8` quantization-aware training.

* With `int8` post-training quantization.

[‡] Dilation reduced to one, with 0-padded filters to maintain the receptive field, to cope with toolchain limitations.

2) *Greenwaves GAP8*: Finally, we deployed the `int8` quantized version of the student to a GAP8 processor, which supports a dilation rate larger than 1. For deployment, we used the DORY deployment toolchain [66] together with the PULP-TCN library [67], which is optimized for the deployment of `int8` quantized neural networks. The memory occupation is extracted from the compiled code, while the average power is computed throughout the execution of the network on board. The energy consumption is computed as reported for the M4.

We do not deploy the teacher network, since the attention modules are not supported by the deployment toolchain. The achieved latency was remarkable (equal to 2.2 ms) with only 0.577 mJ energy consumption. This result is achieved thanks to the low-power consumption of the platform and the DSP-oriented instruction set architecture, which includes SIMD operations for `int8` multiply-accumulate operations. Furthermore, compared to the M4 case, the network has been deployed without any 0-padding, given that dilated filters are supported by the toolchain, allowing GAP8 to extract the maximum performance from this specific topology. Note that the deployed network is similar to the ones of the SoA in terms of operations used and size, while the KD allowed us to improve the accuracy of this miniaturized model, which otherwise would be more difficult to train, given the lower capability of extracting features in the hidden layers.

VI. CONCLUSION

In this paper, we introduced a lightweight deep learning model, PULSE, for heart rate estimation utilizing both temporal convolutional and attention layers. The proposed architecture achieves close to state-of-the-art results on the most benchmarked dataset, PPG-DaLiA, and state-of-the-art results on the WESAD dataset, by effectively fusing PPG signals with tri-axial accelerometer data. Moreover, we reduce significantly its size 11.17 \times by employing a relation-based teacher-student knowledge distillation strategy, firstly, and a post quantization training process afterwards, and manage to deploy it on two commercial-off-the-shelf ultra-low power microcontrollers.

We advocate that the produced quantized student model could be deployed to existing commercial smartwatch devices. However, the effectiveness and the efficiency of PULSE and its student could be further improved by utilizing neural architecture search methods. One other step towards empowering

the model's performance and explainability is to employ a multi-task learning setup where PULSE will also predict the performed activity and visualize the produced attention maps displaying the correlations between the PPG and tri-axial signals. Finally, supervised learning for discarding motion artifacts [68] could be useful to increase the model's robustness.

REFERENCES

- [1] S. Deng *et al.*, "Lhar: Lightweight human activity recognition on knowledge distillation." *IEEE journal of biomedical and health informatics*, vol. PP, 2023.
- [2] P. Kasnesis *et al.*, "Modality-wise relational reasoning for one-shot sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 146, pp. 90–99, 2021.
- [3] D. V. Dimitrov, "Medical internet of things and big data in healthcare," *Healthcare Informatics Research*, vol. 22, pp. 156 – 163, 2016.
- [4] M. Sepahvand *et al.*, "A novel method for reducing arrhythmia classification from 12-lead ecg signals to single-lead ecg with minimal loss of accuracy through teacher-student knowledge distillation," *Inf. Sci.*, vol. 593, pp. 64–77, 2022.
- [5] Z. Liu *et al.*, "Cuffless blood pressure measurement using smartwatches: A large-scale validation study," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, pp. 4216–4227, 2023.
- [6] Z. Zhang *et al.*, "Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise," *IEEE Transactions on biomedical engineering*, vol. 62, no. 2, pp. 522–531, 2014.
- [7] D. Castaneda *et al.*, "A review on wearable photoplethysmography sensors and their potential future applications in health care," *International journal of biosensors & bioelectronics*, vol. 4, pp. 195 – 202, 2018.
- [8] S. M. A. Salehizadeh *et al.*, "A novel time-varying spectral filtering algorithm for reconstruction of motion artifact corrupted heart rate signals during intense physical activities using a wearable photoplethysmogram sensor," *Sensors*, vol. 16, 2016.
- [9] M. B. Mashhadi *et al.*, "An improved algorithm for heart rate tracking during physical exercise using simultaneous wrist-type photoplethysmographic (ppg) and acceleration signals," *2016 23rd Iranian Conference on Biomedical Engineering and 2016 1st International Iranian Conference on Biomedical Engineering (ICBME)*, pp. 146–149, 2016.
- [10] Z. Zhang, "Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction," *IEEE transactions on biomedical engineering*, vol. 62, no. 8, pp. 1902–1910, 2015.
- [11] N. Huang *et al.*, "Robust ppg-based ambulatory heart rate tracking algorithm," in *in IEEE EMBC*. IEEE, 2020, pp. 5929–5934.
- [12] A. Temko, "Accurate heart rate monitoring during physical exercises using ppg," *IEEE Transactions on Biomedical Engineering*, vol. 64, pp. 2016–2024, 2017.
- [13] M. B. Mashhadi *et al.*, "Low complexity heart rate measurement from wearable wrist-type photoplethysmographic sensors robust to motion artifacts," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 921–924, 2018.
- [14] A. Reiss *et al.*, "Deep ppg: large-scale heart rate estimation with convolutional neural networks," *Sensors*, vol. 19, no. 14, p. 3079, 2019.
- [15] D. Biswas *et al.*, "Cornet: Deep learning framework for ppg-based heart rate estimation and biometric identification in ambulant environment,"

- IEEE transactions on biomedical circuits and systems*, vol. 13, no. 2, pp. 282–291, 2019.
- [16] A. Burrello *et al.*, “Q-ppg: Energy-efficient ppg-based heart rate monitoring on wearable devices,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 15, pp. 1196–1209, 2021.
 - [17] A. Burrello *et al.*, “Embedding temporal convolutional networks for energy-efficient ppg-based heart rate monitoring,” *ACM Transactions on Computing for Healthcare*, vol. 3, no. 2, pp. 1–25, 2022.
 - [18] D. J. Pagliari *et al.*, “Plinio: A user-friendly library of gradient-based methods for complexity-aware dnn optimization,” *arXiv preprint arXiv:2307.09488*, 2023.
 - [19] A. Vaswani *et al.*, “Attention is all you need,” *arXiv preprint arXiv:1706.03762*, 2017.
 - [20] R. Girdhar *et al.*, “Imagebind one embedding space to bind them all,” *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 180–15 190, 2023.
 - [21] V. Sanh *et al.*, “Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter,” *ArXiv*, vol. abs/1910.01108, 2019.
 - [22] C. Ma *et al.*, “Kd-informer: A cuff-less continuous blood pressure waveform estimation approach based on single photoplethysmography,” *IEEE Journal of Biomedical and Health Informatics*, vol. 27, pp. 2219–2230, 2022.
 - [23] P. Schmidt *et al.*, “Introducing wesad, a multimodal dataset for wearable stress and affect detection,” *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, 2018.
 - [24] P. Kasnesis *et al.*, “Feature-level cross-attentional ppg and motion signal fusion for heart rate estimation,” *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2023.
 - [25] S. Salehizadeh *et al.*, “A novel time-varying spectral filtering algorithm for reconstruction of motion artifact corrupted heart rate signals during intense physical activities using a wearable photoplethysmogram sensor,” *Sensors*, vol. 16, no. 1, p. 10, 2016.
 - [26] T. Schäck *et al.*, “Computationally efficient heart rate estimation during physical exercise using photoplethysmographic signals,” in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 2478–2481.
 - [27] M. Zhou *et al.*, “Heart rate monitoring using sparse spectral curve tracing,” in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 5347–5352.
 - [28] V. Bieri *et al.*, “Belieppg: Uncertainty-aware heart rate estimation from ppg signals via belief propagation,” in *Uncertainty in Artificial Intelligence*. PMLR, 2023, pp. 173–183.
 - [29] C. Kechris *et al.*, “Kid-ppg: Knowledge informed deep learning for extracting heart rate from a smartwatch,” *arXiv preprint arXiv:2405.09559*, 2024.
 - [30] H. S. Shin *et al.*, “Adaptive threshold method for the peak detection of photoplethysmographic waveform,” *Computers in biology and medicine*, vol. 39, no. 12, pp. 1145–1152, 2009.
 - [31] K. Arunkumar *et al.*, “Robust de-noising technique for accurate heart rate estimation using wrist-type ppg signals,” *IEEE Sensors Journal*, vol. 20, no. 14, pp. 7980–7987, 2020.
 - [32] A. Shyam *et al.*, “Ppgnet: Deep network for device independent heart rate estimation from photoplethysmogram,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 1899–1902.
 - [33] H. Chung *et al.*, “Deep learning for heart rate estimation from reflectance photoplethysmography with acceleration power spectrum and acceleration intensity,” *IEEE Access*, vol. 8, pp. 63 390–63 402, 2020.
 - [34] X. Chang *et al.*, “Deepheart: A deep learning approach for accurate heart rate estimation from ppg signals,” *ACM Trans. Sen. Netw.*, vol. 17, no. 2, Jan. 2021. [Online]. Available: <https://doi.org/10.1145/3441626>
 - [35] L. G. Rocha *et al.*, “Binary cornet: accelerator for hr estimation from wrist-ppg,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 4, pp. 715–726, 2020.
 - [36] S. Bai *et al.*, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” *arXiv:1803.01271*, 2018.
 - [37] C. Lea *et al.*, “Temporal convolutional networks: A unified approach to action segmentation,” in *European Conference on Computer Vision*. Springer, 2016, pp. 47–54.
 - [38] H. Chang *et al.*, “Muse: Text-to-image generation via masked generative transformers,” *ArXiv*, vol. abs/2301.00704, 2023.
 - [39] Y. Li *et al.*, “Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection,” *ArXiv*, vol. abs/2203.08195, 2022.
 - [40] G. E. Hinton *et al.*, “Distilling the knowledge in a neural network,” *ArXiv*, vol. abs/1503.02531, 2015.
 - [41] T. Kim *et al.*, “Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation,” in *International Joint Conference on Artificial Intelligence*, 2021.
 - [42] W. Park *et al.*, “Relational knowledge distillation,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3962–3971, 2019.
 - [43] L. Zhang *et al.*, “Be your own teacher: Improve the performance of convolutional neural networks via self distillation,” *2019 IEEE/CVF Int. Conference on Computer Vision (ICCV)*, pp. 3712–3721, 2019.
 - [44] J. Ni *et al.*, “Progressive cross-modal knowledge distillation for human action recognition,” *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.
 - [45] M. Mardanpour *et al.*, “Human activity recognition based on multiple inertial sensors through feature-based knowledge distillation paradigm,” *Inf. Sci.*, vol. 640, p. 119073, 2023.
 - [46] J. Choi *et al.*, “Weighted knowledge distillation of attention-lrcn for recognizing affective states from ppg signals,” *Expert Systems with Applications*, 2023.
 - [47] Z. Wang *et al.*, “Fldnet: Frame-level distilling neural network for eeg emotion recognition,” *IEEE Journal of Biomedical and Health Informatics*, vol. 25, pp. 2533–2544, 2021.
 - [48] T. Liang *et al.*, “Pruning and quantization for deep neural network acceleration: A survey,” *Neurocomputing*, vol. 461, pp. 370–403, 2021.
 - [49] B. Jacob *et al.*, “Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.
 - [50] A. Capotondi *et al.*, “Cmix-nn: Mixed low-precision cnn library for memory-constrained edge devices,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020.
 - [51] Z. Cai *et al.*, “Rethinking differentiable search for mixed-precision neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2349–2358.
 - [52] G. Xiao *et al.*, “SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models,” Jun. 2023, arXiv:2211.10438 [cs]. [Online]. Available: <http://arxiv.org/abs/2211.10438>
 - [53] S. Microelectornics. Stm32h7. [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32l4r9ai.pdf>
 - [54] A. Pullini *et al.*, “ μ dma: An autonomous i/o subsystem for iot end-nodes,” in *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation*. IEEE, 2017, pp. 1–8.
 - [55] G. Lu *et al.*, “A comparison of photoplethysmography and ecg recording to analyse heart rate variability in healthy subjects,” *Journal of Medical Engineering & Technology*, vol. 33, pp. 634 – 641, 2009.
 - [56] A. Araujo *et al.*, “Computing receptive fields of convolutional neural networks,” *Distill*, 2019, <https://distill.pub/2019/computing-receptive-fields>.
 - [57] A. Santoro *et al.*, “A simple neural network module for relational reasoning,” in *NIPS*, 2017.
 - [58] C.-F. Chen *et al.*, “Crossvit: Cross-attention multi-scale vision transformer for image classification,” *2021 IEEE/CVF ICCV*, pp. 347–356, 2021.
 - [59] A. Burrello *et al.*, “Bioformers: Embedding transformers for ultra-low power semg-based gesture recognition,” *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1443–1448, 2022.
 - [60] Z. Zhang, “Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction,” *IEEE Transactions on Biomedical Engineering*, vol. 62, pp. 1902–1910, 2015.
 - [61] P. Kasnesis *et al.*, “Perceptionnet: A deep convolutional neural network for late sensor fusion,” in *Intelligent Systems with Applications*, 2018.
 - [62] Y. Xu *et al.*, “Cross-modal fusion convolutional neural networks with online soft-label training strategy for mechanical fault diagnosis,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, pp. 73–84, 2024.
 - [63] D. P. Kingma *et al.*, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.
 - [64] S. b. Song *et al.*, “Nas-ppg: Ppg-based heart rate estimation using neural architecture search,” *IEEE Sensors Journal*, vol. 21, pp. 14 941–14 949, 2021.
 - [65] C. M. Bishop, *Neural networks for pattern recognition*. CLARENDON Press, 1995.
 - [66] A. Burrello *et al.*, “Dory: Automatic end-to-end deployment of real-world dnns on low-cost iot mcus,” 2020.
 - [67] A. Burrello *et al.*, “TCN Mapping Optimization for Ultra-Low Power Time-Series Edge Inference,” in *2021 IEEE/ACM Int. Symposium on Low Power Electronics and Design (ISLPED)*, Jul. 2021, pp. 1–6.
 - [68] Z. Guo *et al.*, “A supervised machine learning semantic segmentation approach for detecting artifacts in plethysmography signals from wearables,” *Physiological Measurement*, vol. 42, no. 12, p. 125003, dec 2021.