Tiny Spiking Neural Network for On-Edge PPG-based Blood Pressure Estimation

Rathore Shakti Singh[1], Francesco Carlucci[1], Daniele Jahier Pagliari[1], Elisa Donati[2], Benedetto Leto[1], Gianvito Urgese[1], Massimo Poncino[1], Enrico Macii[1], Vittorio Fra[1], Alessio Burrello[1]
[1] Politecnico di Torino, Turin, Italy
[2] Institute of Neuroinformatics, Zurich, Switzerland

***Abstract*—Photoplethysmography (PPG)-based blood pressure (BP) estimation is a complex biosignal processing task, particularly challenging for resource-constrained wearable devices. Deep learning methods have achieved state-of-the-art performance on large public datasets but still require hundreds of kilobytes of memory for deployment, even after strong optimization. In this work, we show the application of the spiking neural networks (SNNs). Thanks to the inherent suitability of SNNs for temporal signals, our best quantized network achieves comparable or better performance on four benchmark datasets compared to state-of-the-art (SOTA) deep neural networks (DNNs), while strongly reducing the number of parameters by up to 13.8×. We also demonstrate that our optimized SNN can be executed on wearable devices, deploying it on a low-power RISC-V-based microcontroller unit (MCU) and obtaining a memory footprint of 4.63 kB, a latency of 1.34 ms and an energy consumption of 0.068 mJ.**

***Index Terms*—PPG, Blood Pressure, Spiking Neural Network Applications, Edge Computing**

I. Introduction and Related Works

Continuous monitoring of blood pressure (BP) values can be life-changing for a large number of people, considering the high worldwide prevalence of cardiovascular diseases (CVDs), often caused by long-term hypertension conditions. The right side of Fig. 1 shows an example of arterial blood pressure (ABP) waveform and of the two key medically-relevant quantities that are extracted from it: the systolic blood pressure (SBP) corresponds to the highest peak reached during systole, when the heart contracts; diastolic pressure (diastolic blood pressure (DBP)) is instead the minimum reached when the heart relaxes and refills, drawing deoxygenated blood from the veins.

The most common way of measuring BP is the sphygmomanometer, an inflatable cuff that goes around the upper arm. However, the use of photoplethysmography (PPG) is arising as a promising alternative for non-invasive, continuous, and affordable BP measurement [1].

Although the PPG and ABP signals share a similar shape, as shown in Fig. 1, evaluating the SBP and DBP values directly from the former is non-trivial. The most common method has historically been the pulse transit time (PTT) [2], which consists of computing the speed of the wave from the heart to a distal point and uses physical models of the arteries, treated as elastic pipes, to estimate pressure inside the vessels. However, it is highly subject-dependent, given that it depends on an accurate modeling of body composition.

More recently, many machine learning (ML) algorithms have been applied to the task, such as support vector regressor (SVR) and random forest (RF) [3]. Subsequently, researchers have started exploring deep learning (DL) approaches to further improve generalization, proposing various types of deep neural networks (DNNs) that use the raw PPG signal as input. These include recurrent neural networks (RNNs), 1-dimensional convolutional neural networks (CNNs) [4] or hybrid models [5]. Some studies also investigate the usage of the attention mechanism on PPG data [6]. The current state-of-the-art (SOTA) is represented by two DNNs derived from the ResNet and UNet architecture respectively, which achieve the lowest error on different datasets of the rigorous benchmark suite proposed in [7]. While obtaining SOTA accuracy, both classic ML and DL models often have memory requirements that exceed the constraints of far-edge wearable devices (equipped with tens of kB of memory), due to an excessive number of parameters. In this work, we propose to solve this problem by investigating for the first time the usage of a spiking neural network (SNN) quantized and deployed on an ultra-low-power digital device, analysing the tradeoff between accuracy and memory footprint compared to SOTA solutions.

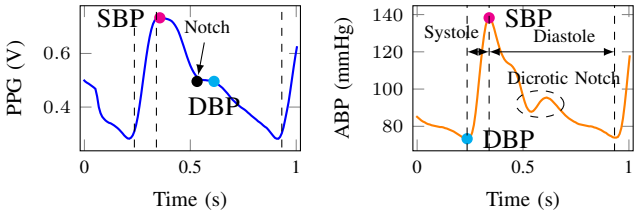

Fig. 1. SBP and DBP estimation from PPG (left) and ABP (right) signals.

A major challenge when comparing different methods for BP estimation lies in the different datasets or custom preprocessing employed by each paper. Moreover, the reported accuracy metrics can also vary drastically if training is done intra-subject (including data from the same subject within both training and test sets) or inter-subject (considering different subjects for testing compared to the ones used for training). To ensure a common preprocessing, compliant with medical device standards, and fairly evaluate all methods, this work adopts the standard benchmark of [7], testing all algorithms with inter-subject data splits on 4 different datasets.

This paper proposes, therefore, a threefold contribution:

- We report the first tiny SNN architecture for PPG-based BP estimation on the standard benchmark [7] and compare it with the SOTA in terms of both prediction error and model size.

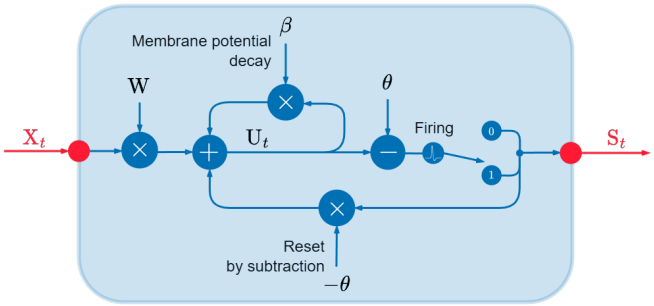

Fig. 2. Schematic of the LIF dynamic implemented in `snnTorch` by the `Leaky` model following Eq. (2) and Eq. (3).

- We optimize the SNN architecture through an application-oriented optimization of the network hyperparameters. We also compressed the model using quantization-aware training (QAT) compression, quantizing both the weights and the internal neuron states to further reduce the memory footprint and the latency of the network.
- We demonstrate that on a standardized setup our SNN achieves SOTA performance while reducing the number of parameters by up to 13.8×. When deployed on an ultra-low-power System-on-Chip (SoC), GreenWave's GAP8, our model fits in a memory of 4.63 kB, with 1.34 ms latency and consuming just 0.068 mJ per BP estimation.

II. Background

A. PPG-based Blood-Pressure Estimation

PPG is a low-cost optical technique that captures blood volume variations in tissues and has become a promising basis for cuffless BP monitoring through wearable sensors. An light-emitting diode (LED) projects light on the skin, while a photo-diode captures the reflected light, which correlates to the volume of blood and tissue absorption. From this, information can be extracted about multiple relevant medical indicators, such as heart rate (HR), respiratory rate, oxygen saturation (SpO2), and BP. Unlike techniques such as intra-arterial catheters, which are accurate (they directly measure the ABP) but invasive, or cuff-based methods such as sphygmomanometers, which are not usable for continuous monitoring, PPG enables noninvasive, continuous monitoring suitable for everyday use. PPG sensors are indeed installed on common smartwatches and can collect the signal from the wrist. The PPG signal is strongly correlated with the ABP [8], but its usage for BP estimation is not trivial, given the strong artifacts that can arise in signal collection, related to arm movement or light entering the space between the arm and the wearable. Fig. 1 shows examples of clean PPG and ABP signals with the points corresponding to SBP and DBP marked.

B. Spiking Neural Networks and LIF Neuron

With the aim of emulating event-based communication, SNNs drive inspiration from the computational functioning of the human brain, by adopting simplified models of biological neural dynamics. This brain-inspired paradigm relies on information transmission through discrete events, named action potentials and also referred to as spikes. The leaky

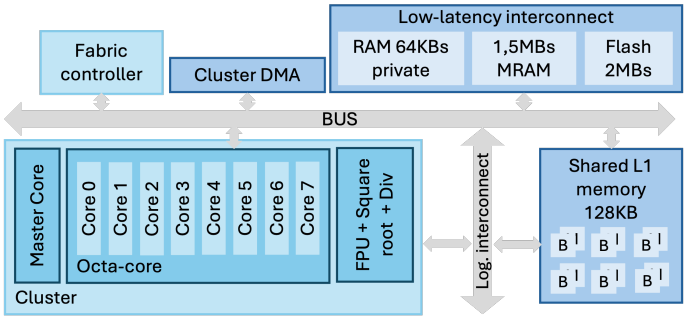

Fig. 3. GAP8 architecture used as a wearable device blueprint in this paper.

integrate-and-fire (LIF) neuron is a widely chosen model to effectively describe the mechanism leading to spike emission with a limited computational cost [9]. It describes the temporal dynamics of the membrane potential, which determines whether or not a spike is emitted, as follows:

$$\tau_m \frac{dv}{dt} = -v(t) + Ri(t) \tag{1}$$

where the membrane potential is represented by $v(t)$ and τ_m is its decay time, while $i(t)$ is a current entering the neuron and R is the electrical resistance of the neuron membrane. Spike emission occurs when the membrane potential overcomes a threshold voltage V_{th}, with the subsequent reset of $v(t)$ to a value such that $v(t+dt) < V_{th}$ [10], [11].

In most simulation frameworks, a discretized form of the solution to Eq. (1) is used. In the case of `snnTorch`, the framework adopted in this work, the LIF neuron is implemented by the so-called `Leaky` model through Eq. (2):

$$U_t = \beta U_{t-1} + WX_t - S_{t-1}\theta \tag{2}$$

with X_t being the neuron's input at time t, while W is the synaptic weights matrix, U_t models the membrane potential at time t, β is its decay factor, and $S_{t-1} \cdot \theta$ represents the reset mechanism by subtraction, with S_{t-1} being the spiking activity of the neuron at the previous time step and θ representing the threshold voltage. Coherently with the basic principles of the LIF dynamics, the spike emission of such discretized models is defined by a Heaviside function as in Eq. (3):

$$S_t = H(U_t - \theta) \quad \text{with} \quad H(0) = 0 \tag{3}$$

A graphical representation of the `Leaky` implementation of the LIF model is shown in Fig. 2.

C. Edge Platforms for Wearable Devices

The computational platform of modern energy-efficient wearables is constituted by ultra-low-power SoCs, for instance, ARM Cortex-M-class microcontroller units (MCUs) or RISC-V-based alternatives. Wearable devices operate under strict constraints of memory, compute, and energy, typically featuring a few hundred kilobytes of on-chip SRAM and often lacking hardware floating-point support. For this work, we target GAP8 [12], a parallel ultra-low-power RISC-V MCU featuring 64 kB of single-clock access memory coupled with a fully general-purpose accelerator made of 8 additional RISC-V cores. Fig. 3 shows the high-level block diagram of the platform.

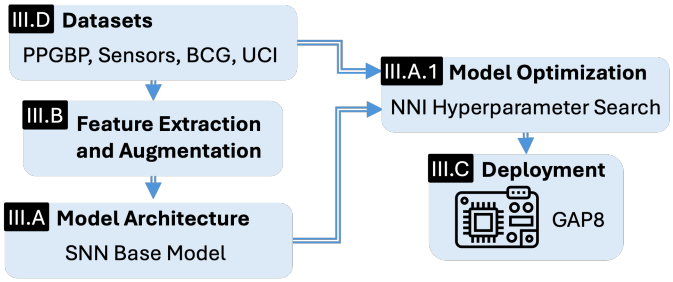

Fig. 4. Overview of the proposed pipeline for efficient BP estimation.

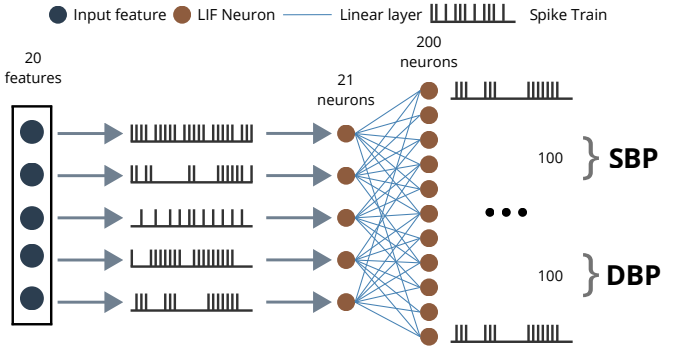

Fig. 5. SNN model for BP estimation. The network is fed with spike-encoded features and provides a twofold output for the prediction of SBP and DBP.

Efficient deployment of machine learning models on such devices requires aggressive optimization to meet both memory and energy budgets. On the other hand, most of the attention has been focused on supporting classical DNNs, with existing toolchains focusing on CNNs or Transformers. In this context, SNNs are particularly attractive, as their sparse, event-driven nature can reduce energy consumption, offering a promising path for always-on biosignal monitoring on platforms like GAP8. In our work, we introduce the first application of SNNs to PPG-based BP estimation, deployed on a commercial wearable-class SoC. While the implementation and porting are not yet optimized at the same level of the ones for DNNs, the results are promising, showing that SNNs can take over classical DL on these ultra-low-power bio-signal-based tasks.

III. Methodology

A. Spiking Neural Network for Blood Pressure Estimation

The SNN model we propose is based on the discrete-time first-order LIF neuron provided by `snnTorch` used in a two-layer fully connected architecture. A virtual input layer feeds the SNN with 20 handcrafted features computed on the raw PPG signals, min-max normalized to the range $[0, 1]$ and converted into spike trains by means of rate encoding; for each feature, the spike train is generated emitting a spike at each timestep with probability equal to the value of the feature normalised on a scale [0,1]. These spikes are transmitted to the hidden layer containing 21 neurons, which then sends its spiking activity to the output layer. The latter is made of 200 neurons, with the first 100 dedicated to the prediction of SBP and the remaining half devoted to DBP estimation. Each output neuron is assigned to a linearly distributed value within the physiological pressure range (70-190 mmHg for SBP and 45-100 mmHg for DBP). The network's predictions are computed

TABLE I
Summary of the hyperparameters, and their values, defining the search space for the HPO experiments with NNI.

Name	Description	Sampling	Values
θ_h	Threshold voltage in the hidden layer	`quniform`	[0.05, 1]
θ_o	Threshold voltage in the output layer	`quniform`	[0.05, 1]
β_h	Decay factor (hidden layer)	`quniform`	[0.3, 1]
β_o	Decay factor (output layer)	`quniform`	[0.3, 1]
α	Weight factor for the $\mathcal{L}_{MSE}$ term	`quniform`	[0.1, 0.8]
f_{max}	Target firing rate	`choice`	{0.8, 0.9, 1.0}
taper	Firing rate reduction constant	`choice`	{0.15, 0.25, 0.35}
decay	Firing rate reduction speed	`choice`	{0.25, 0.5 , 0.75 , 1.0}

as a weighted sum of the neuron's firing rate at the output layer multiplied by its associated value, as shown in Eq. 4:

$$\hat{P}_{BP} = \sum_{i=1}^{100} \hat{f}_i \cdot p_i \tag{4}$$

where $\hat{P}_{BP}$ is the predicted systolic or diastolic BP, $\hat{f}_i$ is the firing rate of each neuron, normalized so their total sum is 1, and p_i is the pressure value associated with neuron i. We train our SNN with a two-term loss function: we combine the mean squared error (MSE) between the ground truth values y and the predicted targets $\hat{y}$, with a term computing the Kullback-Leibler divergence between the output unnormalized firing rates and a target distribution, specifically designed to incentivize a smoother distribution of the output values, and avoid restricting predictions over median BP values. Such distribution assign a maximum target firing rate f_{max} to the neuron i_g associated to the pressure value closest to the ground truth, while neighbouring neurons receive a target f_i that progressively decreases according to their distance from the true value, following a power law:

$$f_i = f_{max} - taper \cdot |i_g - i|^{decay} \tag{5}$$

where f_{max}, taper and decay are tunable hyperparameters.

The final training loss is obtained through a linear combination of the two: $\mathcal{L}_{total} = \mathcal{L}_{KL} + \alpha \cdot \mathcal{L}_{MSE}$ where α is a hyperparameter that balances the trade-off between accuracy and generalization.

Hyperparameters Optimization: By relying on bio-inspired neuron models, SNNs typically require the tuning of at least two crucial hyperparameters, namely threshold voltage and decay factor [13]. Inadequately chosen values can produce models with poor performance, driving promising networks to unsuccessful results. To avoid such outcomes, hyperparameter optimization (HPO) can be a solution, especially if performed with an application-oriented approach. In our work, we perform HPO employing the Neural Network Intelligence (NNI)[1] toolkit, adapting the design presented in [14] to the target application. We defined a tailored search space and adopted

[1] github.com/microsoft/nni/

the built-in *Anneal* tuner for its exploration. The same search space, summarized in Table I, was used for all the considered datasets. The β and θ hyperparameters, namely the membrane potential decay factor and the threshold voltage, respectively, are optimized independently for the hidden and the output layer. The other hyperparameters explored are: α, which is used as a weighting factor for the $\mathcal{L}_{MSE}$ term of $\mathcal{L}_{total}$; f_{max}, decay and taper, that define the firing rate target distribution described in Eq. 5.

To evaluate the models under fair and comparable conditions, we adopted a nested cross-validation strategy. With n folds, the outer loop reserved one fold for testing, while the remaining $n-1$ folds were used within NNI to perform hyperparameter tuning through inner cross-validation. This procedure ensures that each fold had its own independent tuning process, resulting in n distinct optimized models for each dataset. As objective metric for HPO, in order to effectively account for the double prediction given as output by the model, we used the sum of the mean absolute error (MAE) for SBP and DBP on the inner validation set.

B. Feature Extraction and Dataset Augmentation

PPG signals are rich in information and numerous features can be extracted to capture temporal, morphological, and spectral characteristics (e.g., peaks, valleys, and notch coordinates of the temporal signal, as well as frequency-domain measures derived from FFT analysis). Given the abundance of potential features, a systematic selection process is essential to identify the most relevant subset for the intended application. We perform this selection by training two separate RF regressors, one for each output (SBP and DBP). Then, we estimate the relevance of each feature on a validation dataset (different from the test set used for final experiments), using the mean decrease in impurity across all 100 trees of each forest, and as impurity score the squared error. Mathematically, the importance I_f of a feature f is computed according to:

$$I_f = \sum_{t \in T_f} \frac{N_t}{N} \cdot (SE(t) - SE(t_L) - SE(t_R)) \qquad (6)$$

where T_f is the set of all nodes that split on the considered feature f, $SE(t)$ is the sum of squared errors at node t, t_L and t_R are the left and right children of node t, respectively, N is the total size of the training set, and N_t is the number of samples reaching node t. We then select the top 20 features for each dataset. The most selected features include the distance of crucial points like peaks and valleys, features extracted from the distribution of the temporal values of the signal, and frequency domain ones (for the largest datasets). A challenge specific to BP data is the narrow range of physiological values. This complicates the estimation of abnormal BP values, degrading model performance. To mitigate the issue, we implement data augmentation through stratified sampling of rare extremes. Namely, we divide each training dataset into 10 intervals between physiological values, imposing each of them to have a minimum number of samples x and replicating values where that quota is not met. On the other hand, we downsize central intervals by a maximum of $1.5\times$. We do not augment values in the highest and lowest intervals to avoid oversampling too many outliers.

C. SNN Deployment

We deployed our networks on the GAP8 platform (Sec. II-C). We deployed the original `fp32` model and an int8 quantized one, which has been trained through QAT by representing both the synaptic weights and the internal neuron states on 8 bit [15], using the same protocol described for `fp32` training.

The models are both sufficiently compact (4.63 kB and 18.91 kB) to fit entirely within the 64 kB L1 scratchpad memory of GAP8, which is directly connected to the 8-core accelerator by a single clock cycle logarithmic interconnect, ensuring rapid access to both activations and weights and avoiding costly transfers from external memory. Parallelization was performed at the neuron level across the 8 RISC-V cores, achieving a speed-up close to $8\times$ compared to the single-core execution. The sparsity of the connections is not exploited, given that GAP8 does not include specific HW features for sparsity exploitation. Notice that some SOTA work already explored SNN deployment on platforms similar to GAP8 [16]: however, they did so exploiting custom hardware modifications, not available in off-the-shelf versions of the platform, and therefore not easily embeddable in a practical wearable system. Accordingly, we did not resort to their implementation, and instead wrote our own SNN inference library for GAP8 in C.

D. Benchmarking Datasets

We consider for evaluation four publicly available datasets introduced in the benchmark study [7], which is the current SOTA reference for standardized PPG-based BP estimation. All datasets were resampled to a uniform sampling rate of 125 Hz and split according to the same subject-level protocols defined in the benchmark. **PPGBP** is the smallest dataset, comprising 619 short PPG segments of 2.1s each, collected from 218 subjects with various cardiovascular conditions. **BCG** contains about 4 hours of measurements on 40 individuals, segmented into non-overlapping 5 s windows. **Sensors** contains 11k segments of 5 s each, collected from 1195 patients. This dataset provides full continuous ABP waveforms as a reference, from which SBP and DBP ground labels can be computed. **UCI** is the largest dataset, extracted from MIMIC-II, with approximately 411k segments and an unknown number of subjects, making it significantly more heterogeneous.

We followed 5-fold cross-validation for all datasets except UCI (Hold-one-out) with subject-level splits protocol matching the reference papers [4], [7] for fair comparison. It is important to note that, under this cross-subject protocol, error values are substantially higher than medical-grade standards, which currently can only be achieved via subject-specific calibration [17], which is orthogonal to the algorithm employed.

IV. Experimental Results

A. Benchmarking & State-of-the-art Comparison

In Fig. 6, we benchmarked our best fp32 SNN models against both classical ML baselines from [7] (SVR, RF), and

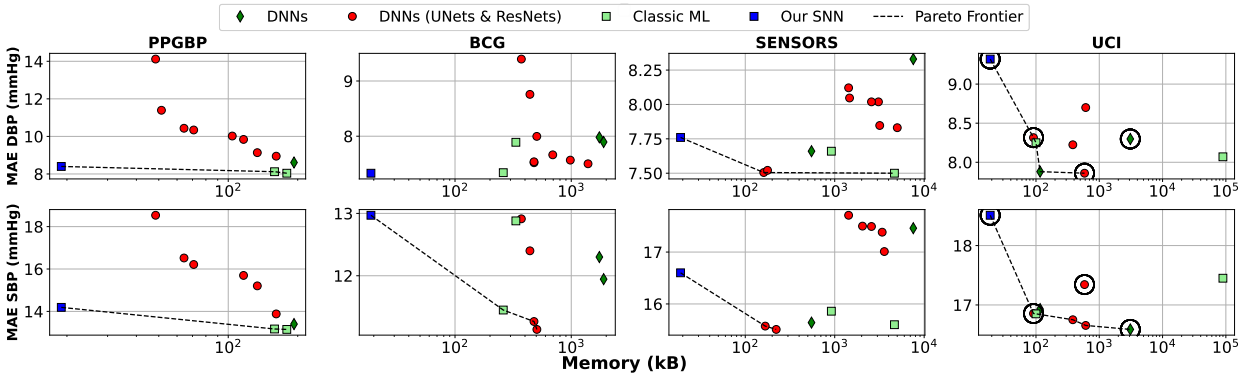

Fig. 6. Comparison with SOTA methods using `fp32` format. Classic ML and DNNs are the SOTA solutions from [7]; DNNs (UNets & ResNets) are from [4].

TABLE II
DEPLOYMENT RESULTS ON GAP8 AT 100 MHZ, WITH AN AVERAGE CONSUMED POWER OF 51 MW, CONSIDERING THE UCI DATASET.

Model	MAE (mmHg)		Size	Latency	Energy
	SBP	DBP	(kB)	(ms)	(mJ)
ResNet* [7]	18.23	**8.17**	791.8	o.o.m.	o.o.m.
Opt-ResNet* [4]	17.83	8.44	156.3	7.12	0.36
Opt-UNet* [4]	**17.2**	8.26	23.4	8.91	0.45
Ours	18.51	9.32	18.91	20.38	1.03
Ours (quantized)*	18.42	9.41	**4.63**	**1.34**	**0.068**

*Models deployed with int8 quantization

DNN baselines (ResNet and U-Net) [7], including their variants optimized for embedded deployment proposed in [4]. The results confirm that, despite its extreme compactness (∼4.8k parameters), our SNN achieves an accuracy comparable to, or in some cases even better than SOTA embedded DNNs, and most importantly, sits on the Pareto frontier in terms of error versus size, for all datasets.

On PPGBP, our SNN yields MAEs of 14.19 mmHg (SBP) and 8.40 mmHg (DBP), closely matching or surpassing optimized ResNets while using 4-9× fewer parameters. The best SOTA ML baseline is the SVR, slightly outperforming our model in terms of MAE (8.04 mmHg on DBP, 13.15 mmHg on SBP) but requiring 9.5× more parameters.

On BCG, the SNN performs on par for DBP with the best SOTA, namely an SVR, with a slight degradation in MAE for SBP of 1.52 mmHg, while reducing the parameters by 13.8×.

On Sensors, the SNN is slightly outperformed by optimized DNNs, achieving a higher MAE by 0.25 mmHg on SBP and by 1.09 mmHg on DBP, but reducing the parameters by 8.51×.

On UCI, the largest and most heterogeneous dataset, our SNN-based model achieves a MAE of 18.51 mmHg (SBP) and 9.32 mmHg (DBP), which is within 1 mmHg of degradation compared to the smallest SOTA solution (a compact U-Net), still reducing the number of parameters by 4.83×. Compared to the most accurate solutions, we achieve a MAE higher by 1.46 mmHg (DBP) and 1.93 mmHg (SBP), but with a reduction in parameters by 20.2× and 163.6×.

Notice that while our algorithm is competitive on every dataset, its architecture is not fine-tuned to each of them, demonstrating that SNNs are great candidates for these extremely low-power tasks.

B. Model Deployment

Table II compares our proposed SNN deployed on GAP8 in both `fp32` and `int8` formats with SOTA DNNs `int8` models.

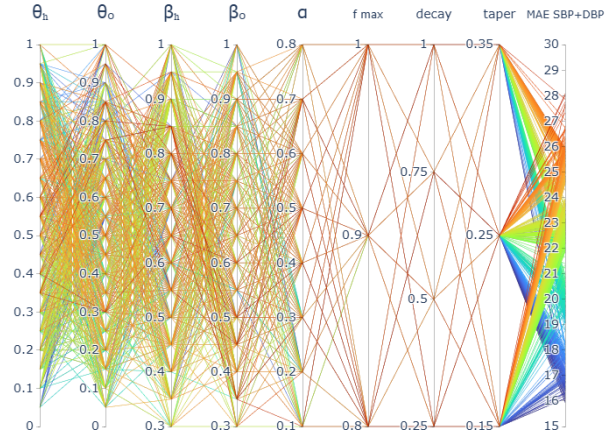

Fig. 7. Overall exploration of the search space during the HPO experiments performed with NNI. The rightmost axis represents the objective, i.e., the MAE achieved by the model on validation data.

For the sake of space, we report relevant metrics (MAE, memory footprint, latency, and energy) on the UCI dataset only. However, for our SNN, non-functional metrics are also valid for other datasets, given that the network architecture is identical. Our `fp32` network occupies 18.91 kB with a latency of 20.38 ms, being comparable in terms of performance with SOTA models, while being the smallest model. The quantized model, while maintaining the same performance in terms of MAE, reaches the smallest memory footprint of 4.63 kB.

Thanks to the parallelization on the 8-core RISC-V cluster running at 100 MHz, we obtain an inference latency of 1.34 ms per input segment, well below the real-time constraint of the application. The measured energy consumption is 0.068 mJ per inference, corresponding to an average power of 51 mW. For comparison, the smallest SOTA solution, an optimized U-Net, achieves 6.64× higher latency (8.91 ms) and 5.1× higher memory footprint (∼23.4 kB) [4].

Our implementation highlights that, even without sparsity exploitation, SNNs can be deployed efficiently on commercial low-power MCUs outperforming current edge DNNs for BP estimation.

C. Ablation Study

1) Model Optimization: The results of the HPO experiments performed on the different datasets are reported in Fig. 7, which summarizes the exploration of the search space defined in Table I. All the hyperparameter configurations evaluated

TABLE III
CHARACTERIZATION OF THE PROPOSED SNN-BASED MODEL THROUGH THE NEUROBENCH FRAMEWORK.

Metrics	Value
Activation sparsity	0.87
Effective MACs	0
Effective ACs	24201
Dense synaptic operations	115500
Membrane updates	5525
Effetive neuronal MACs	11050
Dense neuronal operations	11050

are represented across the parallel axes, showing which values have been used. Different colours emphasize how the objective metric, reported by the rightmost axis, varies depending on the specific combination of hyperparameters employed.

On validation data, adopting the cross-validation procedure previously described, the cumulative MAE on predicted DBP and SBP values varied between 15.94 mmHg and 28.07 mmHg across the different datasets, corresponding to a decrease by $\sim$15% with respect to the minimum cumulative value from [7].

2) Model Characterization: To further assess how our optimized SNN, after HPO, would perform once deployed on a more dedicated event-driven (e.g. neuromorphic) hardware platform, we employed the NeuroBench framework [18] for model characterization. Particularly, we focused on the workload metrics, i.e., activation sparsity and number of operations, as summarized in Table III. We performed the characterization using the BCG dataset, chosen because of its highest number of samples per subject [7]. All the reported values are averaged over the number of samples.

From the perspective of computational effort in an event-driven scenario, the result of 0.87 for activation sparsity is of high relevance and potential: for each sample given as input to the SNN, only 13% of the neurons are actually activated. The direct implication of such a result is that the active consumption throughout the network can be tremendously reduced, with respect to clock-based synchronous computation, when using event-driven hardware. Similar considerations can be made comparing the number of effective accumulate operations (ACs) and the number of dense synaptic operations. The former refers to synaptic operations with binary, non-zero activations, while the latter quantifies all the synaptic operations regardless of the involved activations. Consequently, as the effective ACs are representative of the number of operations required by active neurons only, a reduction by $\sim$79% with respect to the number of synaptic operations highlights the potential advantages offered by event-driven SNN-based models. Moving the attention to the neuron itself, the last three metrics reported in Table III describe the impact of the temporal evolution, ruled by Eq. (2), on the number of operations. We can further distinguish between the number of membrane updates and the corresponding operations, and, similarly to the case of synaptic operations, also at the neuron level the effective operations are those involving binary, non-zero activations.

Lastly, the number of effective multiply-accumulate operations (MACs) equal to zero is a quantitative evidence of the fully spiking operational regime of the proposed model, as this metric refers to non-binary, non-zero activations.

V. CONCLUSIONS

In this work, we presented the first systematic exploration and deployment of SNNs for PPG-based BP estimation on extreme-edge platforms. Our approach demonstrates that a compact SNN with only $\sim$4.8k parameters achieves accuracy comparable to SOTA DNNs across four public benchmarks, while providing a memory footprint below 5 kB. When deployed on the GAP8 MCU, the model achieves a latency of 1.34 ms and an energy consumption of 0.068 mJ.

These results establish SNNs as a compelling alternative to traditional DNNs for wearable BP monitoring, bridging the gap between algorithmic accuracy and embeddability. Future work will include additional architecture explorations and further optimization of the deployment code for edge digital platforms, including pruning and sparsity exploitation.

REFERENCES

[1] B. Henry *et al.*, "Cuffless Blood Pressure in clinical practice: challenges, opportunities and current limits." *Blood Pressure*, vol. 33, no. 1, p. 2304190, Dec. 2024, publisher: Taylor & Francis.

[2] R. Mukkamala *et al.*, "Toward ubiquitous blood pressure monitoring via pulse transit time: Theory and practice," *IEEE TBME*, 2015.

[3] R. He *et al.*, "Beat-to-beat ambulatory blood pressure estimation based on random forest," in *2016 IEEE 13th Conference on Wearable and Implantable BSN*, 2016, pp. 194–198.

[4] A. Burrello *et al.*, "Optimization and deployment of deep neural networks for ppg-based blood pressure estimation targeting low-power wearables," in *2024 IEEE Biomedical Circuits and Systems Conference*, 2024.

[5] N. F. Ali and M. Atef, "An efficient hybrid lstm-ann joint classification-regression model for ppg based blood pressure monitoring," *Biomedical Signal Processing and Control*, vol. 84, p. 104782, 7 2023.

[6] Z. Tian *et al.*, "A paralleled CNN and Transformer network for PPG-based cuff-less blood pressure estimation," *Biomedical Signal Processing and Control*, vol. 99, p. 106741, 2025.

[7] S. González *et al.*, "A benchmark for machine-learning based non-invasive blood pressure estimation using photoplethysmogram," *Scientific Data 2023 10:1*, vol. 10, pp. 1–16, 3 2023.

[8] G. Martínez *et al.*, "Can photoplethysmography replace arterial blood pressure in the assessment of blood pressure?" *Journal of Clinical Medicine*, vol. 7, no. 10, 2018.

[9] E. M. Izhikevich, *Dynamical Systems in Neuroscience*. The MIT Press, 2006. [Online]. Available: https://direct.mit.edu/books/book/2589/dynamical-systems-in-neurosciencethe-geometry-of

[10] W. Gerstner and W. M. Kistler, *Spiking neuron models: single neurons, populations, plasticity*. Cambridge, U.K. ; New York: Cambridge University Press, 2002.

[11] J. E. Pedersen *et al.*, "Neuromorphic intermediate representation: A unified instruction set for interoperable brain-inspired computing," *Nature Communications*, vol. 15, no. 1, p. 8122, 2024.

[12] G. Technologies, "Gap8, https://greenwaves-technologies.com/gap8_mcu_ai/," Access 16/05/2024.

[13] C. Ganguly *et al.*, "Spike frequency adaptation: bridging neural models and neuromorphic applications," *Communications Engineering*, vol. 3, no. 1, p. 22, Feb. 2024.

[14] V. Fra, "Application-oriented automatic hyperparameter optimization for spiking neural network prototyping," *arXiv preprint arXiv:2502.12172*, 2025.

[15] B. Leto *et al.*, "Variable-precision neuromorphic state space model for on-edge activity classification," *Future Generation Computer Systems*, 2026.

[16] S. Manoni *et al.*, "Spikestream: Accelerating spiking neural network inference on risc-v clusters with sparse computation extensions," in *2025 Design, Automation & Test in Europe Conference (DATE)*, 2025, pp. 1–7.

[17] T. P. Almeida *et al.*, "Aktiia cuffless blood pressure monitor yields equivalent daytime blood pressure measurements compared to a 24-h ambulatory blood pressure monitor: Preliminary results from a prospective single-center study," *Hypertension Research*, Jun. 2023.

[18] J. Yik *et al.*, "The neurobench framework for benchmarking neuromorphic computing algorithms and systems," *Nature communications*, 2025.