sEMG-driven Hand Dynamics Estimation with Incremental Online Learning on a Parallel Ultra-Low-Power Microcontroller

sEMG-driven Hand Dynamics Estimation with Incremental Online Learning on a Parallel Ultra-Low-Power Microcontroller

Marcello Zanghieri,* *Graduate Student Member, IEEE*, Pierangelo Maria Rapa, *Graduate Student Member, IEEE*, Mattia Orlandi, *Graduate Student Member, IEEE*, Elisa Donati, *Member, IEEE*, Luca Benini, *Fellow, IEEE*, Simone Benatti, *Member, IEEE*

***Abstract*— Surface electromyography (sEMG) is a State-of-the-Art (SoA) sensing modality for non-invasive human-machine interfaces for consumer, industrial, and rehabilitation use cases. The main limitation of the current sEMG-driven control policies is the sEMG's inherent variability, especially cross-session due to sensor repositioning; this limits the generalization of the Machine/Deep Learning (ML/DL) in charge of the signal-to-command mapping. The other hot front on the ML/DL side of sEMG-driven control is the shift from the classification of fixed hand positions to the regression of hand kinematics and dynamics, promising a more versatile and fluid control. We present an incremental online-training strategy for sEMG-based estimation of simultaneous multi-finger forces, using a small Temporal Convolutional Network suitable for embedded learning-on-device. We validate our method on the HYSER dataset, cross-day. Our incremental online training reaches a cross-day Mean Absolute Error (MAE) of (9.58 ± 3.89)% of the Maximum Voluntary Contraction on HYSER's RANDOM dataset of improvised, non-predefined force sequences, which is the most challenging and closest to real scenarios. This MAE is on par with an accuracy-oriented, non-embeddable offline training exploiting more epochs. Further, we demonstrate that our online training approach can be deployed on the GAP9 ultra-low power microcontroller, obtaining a latency of 1.49 ms and an energy draw of just 40.4 uJ per forward-backward-update step. These results show that our solution fits the requirements for accurate and real-time incremental training-on-device.**

***Index Terms*— Continual Learning, Deep Learning, Electromyography, Embedded, Human-Machine Interaction, Human-Machine Interfaces, Incremental Learning, Low-Power, Machine Learning, Microcontroller, On-Device Learning, Online Learning, Parallel Computing, Prosthetics, Ultra-Low Power, PULP, Real Time, Regression, Temporal Convolutional Networks, TinyML, Training-on-Device.**

Submitted on 22 February 2024.

This manuscript is submitted as an invited extension of the conference paper M. Zanghieri *et al.*, "Online unsupervised arm posture adaptation for sEMG-based gesture recognition on a parallel ultra-low-power microcontroller," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2023, pp. 1-5. DOI: 10.1109/BioCAS58349.2023.10388902. IEEE Xplore URL: https://ieeexplore.ieee.org/document/10388902.

*Corresponding author.

M. Zanghieri, P. M. Rapa, M. Orlandi, L. Benini, and S. Benatti are with the Department of Electrical, Electronic, and Information Engineering, University of Bologna, 40136 Bologna, IT (e-mail: {marcello.zanghieri2, pierangelomaria.rapa, mattia.orlandi, luca.benini, simone.benatti}@unibo.it).

E. Donati is with the Institute of Neuroinformatics, University of Zürich and ETH Zürich, 8057 Zürich, CH (e-mail: elisa@ini.uzh.ch).

L. Benini is also with the Integrated Systems Laboratory, Department of Information Technology and Electrical Engineering, ETH Zürich, 8092 Zürich, CH (e-mail: lbenini@iis.ee.ethz.ch).

P. M. Rapa and S. Benatti are also with the Department of Engineering, University of Modena and Reggio Emilia, 41125 Modena, IT (e-mail: pierangelomaria.rapa@unimore.it, simone.benatti@unimore.it).

This research was supported in part by the EU Horizon Europe project IntelliMan (g.a. 101070136) and by the ETH Zürich's Future Computing Laboratory funded by a donation from Huawei Technologies.

I. Introduction

Decoding surface electromyographic (sEMG) [1], [2] signals is nowadays a widespread approach for driving Human-Machine Interfaces (HMIs) in an intuitive and non-invasive way in scenarios involving robotics, prosthetics, and other industry and consumer use cases [3]–[8]. The current dominant approach exploits classical Machine Learning (ML) [9]–[15] or Deep Learning (DL) [16]–[20] to automatize and optimize the signal-to-command strategy, mapping the sEMG to hand positions, movements, or forces.

The main obstacle to the ML/DL ability to generalize accurately and robustly is represented by the numerous variability factors inherent in the sEMG signal, such as anatomical variability, fatigue, skin perspiration, and electrode repositioning. The current SoA ML/DL methodology has tackled the variability issue by using statistical methods to align data across days [21] or by producing larger datasets and performing multi-session training of the models [6], [22]–[24], i.e., by training deep models on sEMG data acquired in diverse conditions, postures, electrode placements, or users [10], [25]. This approach is mainly applied to DL since multi-session training benefits Deep Neural Networks (DNNs) more than non-deep ML [26], [27]. However, multi-session training has high computational and memory costs since it requires storing training datasets representative of several conditions and involves iteratively training deep models for several epochs. In addition, collecting sEMG datasets sufficiently large for multi-session training remains challenging since curating and releasing high-quality datasets requires several participants and sessions and is labor-intensive [6].

Another major limitation of the ML/DL approach to sEMG-driven HMIs is the focus on classification, i.e., the recognition of hand postures or gestures. Even if accurate, sEMG recognition limits the control to a discrete set of fixed predefined movements. In contrast, regression of hand kinematics or dynamics promises a more fluid and versatile control. Recent advances in sEMG regression have been made thanks to public sEMG datasets [28]–[30] and accurate models [31]–[33]. However, very few sEMG regression works target the deployment and profiling of the regression algorithms onto embedded computational devices [34], [35].

To advance wearable sEMG-driven HMIs, it is necessary to put effort into the hardware-software co-design required to port the accurate SoA algorithms into the domain of Tiny Machine Learning (TinyML) [36]–[38], i.e., the approach to ML that targets resource-constrained platforms and seeks to optimize the tradeoff between accuracy and execution requirements such as memory footprint, computational burden, and latency [39].

In this work, we propose a strategy of incremental online learning for simultaneous multi-finger force estimation on an embedded platform in real time. The method is based on retraining a small Temporal Convolutional Network (TCN) [40], [41], designed for combining accuracy and a reduced computational and memory budget, for time-domain processing of raw sEMG [24], [34], [42] and other raw biosignals [7], [43], [44]. Our contribution is three-fold:

- we propose an incremental online learning setup to learn new multi-finger force patterns processing the retraining data in streaming mode, consuming them one-by-one, and seeing them just once as in an online adaptation session on an embedded device that does not save the whole retraining set into volatile or non-volatile memory;
- we validate our method on the HYSER sEMG regression dataset [29], obtaining a cross-day Mean Absolute Error (MAE) of $(9.58 \pm 3.89)\%$ of the Maximum Voluntary Contraction (MVC) on HYSER `RANDOM`, the dataset's most challenging section containing improvised, non-predefined force sequences, closest to real scenarios; this MAE is on par with a baseline offline training exploiting more computational resources, and compatible with the literature addressing HYSER in easier settings;
- we deploy and profile the retraining on the GAP9 parallel ultra-low power Microcontroller Unit (MCU), obtaining a power draw of $27.1\,\mathrm{mW}$; for each forward-backward-update step, we measure an energy consumption of $40.4\,\mu\mathrm{J}$ and a latency of $1.49\,\mathrm{ms}$, shorter than the input data frequency used for training and thus consistent with the requirements of real-time training-on-device.

The present work is an extension of our previous paper [25] and targets the same challenge of sEMG's pattern variability across different conditions and sessions. As schematized in Table I, the present work contributes an incremental re-learning strategy which is more powerful than [25] since it advances from sparse, low-count to high-density, high-count electrodes, from classification of fixed gestures to regression, and from cross-arm-posture adaptation to incremental learning of new simultaneous multi-finger force patterns. This progress is promising for the future of intuitive and reliable non-invasive wearable HMIs.

To allow the research community to reproduce our work, we make the code developed for this paper available open-source.[1]

II. Related Work

This section outlines the SoA of the ML/DL-based HMI control policies that address the two main current challenges of sEMG-driven HMIs: the inherent variability of the sEMG signal patterns (II-A) and the advance from recognition to regression (II-B).

A. sEMG Inherent Variability

In the ML/DL framework for sEMG-driven control, the SoA methods for addressing the sEMG signal's inherent variability are mainly multi-session training and model adaptation. In multi-session training, a ML/DL model is refit on data collected in different conditions, e.g., diverse users, sessions (involving a different positioning of the sEMG sensors), or arm postures [10]. In contrast, model adaptation only involves partial retraining on data from a novel sEMG session. Multi-session training and model adaptation are mainly applied to DL models since multi-session training benefits deep models more than non-deep ML [26], and the deep nets' modularity allows to only retune a subset of layers or modules, e.g., batch-normalizations in AdaBN [45], [46], or a model's final classifier block in continual learning [27].

Multi-session training and deep net adaptation improve recognition accuracy but imply a large computational burden and a high memory footprint. On the one hand, multi-session training requires storing multiple sEMG sessions as training sets to be merged and training the model on a server. On the other hand, adaptation based on fine-tuning can lower the memory footprint. For instance, latent replay stratagems [47] only save a reduced subset of intermediate activations [27]. Other solutions only retune the final or BN layers, avoiding the computational load of back-propagation [46]. However, these methods were not deployed on resource-constrained computing devices.

To the best of our knowledge, we are the first to propose an on-device learning setup for sEMG regression, which fits the resource constraints of embedded devices suitable for wearable HMIs.

B. sEMG Regression

The estimation of the hand kinematics and dynamics, framed as a ML/DL regression task, promises a more fluid and natural control compared to the classification that maps the sEMG to a limited, fixed set of static positions.

The literature addressing sEMG regression can be classified based on the variable chosen as the estimation target: kinematics variables (e.g., joint angle and joint velocity) or dynamic variables (finger forces). Joint angles constitute the most representative variable for hand kinematics and are the quantities

[1] https://github.com/pulp-bio/incremental_hyser/

TABLE I: Outline of the present work as an extension of our previous publication [25].

WORK	sEMG sensing	ML task	Addressed variability	Validation
Zanghieri *et al.* [25] (extended here)	4 channels: low-density, low-count	classification: fixed gestures	new arm postures	within-session
This work	64 channels: high-density, high-count	regression: simultaneous multi-finger forces	new force patterns	cross-day

most used as a target variable. For instance, [28] targets the estimation of the joint angles of a dataglove, using a Wiener filter and attaining a median R^2 of 0.63. However, this work does not investigate other algorithms to improve the regression score since it focuses on the control quality perceived by users. The work [31] employs a Long Short-Term Memory deep neural network on the same dataset as [28], reaching a Mean Absolute Error (MAE) of $7.04°$. The limitation of [31] is that it does not explore convolutional networks, which are more amenable to parallelization and hence more suitable for embedded control. An alternative kinematic approach targets joint velocity to model finger motion in addition to finger position. For example, the work [15] estimates velocity using a hybrid classification-regression setup, which is based on thresholding the speed's dynamic range into 3 intervals; though novel, this method still limits the estimation to the recognition of discrete classes.

A different and independent research direction in sEMG regression aims to model hand dynamics, i.e., finger forces. The work [48] targeted a multiple-Degrees-of-Freedom (multi-DoF) force estimation but is limited to grasp movements. In contrast, in the present work, we address the richer set of force patterns present in the public sEMG HYSER dataset (III-A) [29].

The limitation of all the sEMG regression works mentioned above is that they do not focus on the embedded deployment onto resource-limited computational platforms. The interest for accuracy alone is common to other sEMG regression works that implement convolutive deep learning [49], [50]. The deployment of a SoA TCN for sEMG hand kinematics regression is addressed in [34], matching the SoA accuracy on the NinaPro DB8 [28] dataset and deploying the TCN onto the parallel ultra-low power microcontroller GAP8 (predecessor of GAP9, which we use in this work). However, [34] does not address cross-session validation and online learning, which we target in the present work.

III. Materials & Methods

A. HYSER Dataset

This work targets the High-densitY Surface Electromyogram Recordings (HYSER)[2] [29], a rich sEMG dataset released for research on sEMG recognition and force regression. HYSER features 20 healthy participants undergoing 2 acquisitions in separate days at a distance of 3 to 25 days (8.5 ± 6.7 days on average). sEMG data are collected with four 8×8 HD-sEMG arrays (256 electrodes in total) on the forearm covering the extensor and flexor muscles, employing an OT Bioelettronica Quattrocento system sampling at $2048\,\mathrm{samples/s}$. Force ground-truth signals are collected in isometric contractions with a sensor-amplifier pair for each finger, employing Huatran SAS sensors and Huatran HSGA amplifiers sampling at $100\,\mathrm{samples/s}$. In all the experiments of this work, we comply with the recommended filtering of the HYSER data [29]: (*i*) on the HD-sEMG data, an 8^{th}-order band-pass Butterworth filter with pass-band between $10\,\mathrm{Hz}$ and $500\,\mathrm{Hz}$, and a notch filter to attenuate the power line interference at $50\,\mathrm{Hz}$ and harmonics up to $400\,\mathrm{Hz}$; (*ii*) on the force data, an 8^{th}-order lowpass Butterworth filter with cut frequency $10\,\mathrm{Hz}$.

The HYSER dataset is composed of 5 sub-datasets:

1) `PR`: pattern recognition on 34 hand gestures (not used in this work);
2) `MVC`: trials for determining the MVC of every finger's flexion and extension;
3) `1-DoF`: single-finger contractions for 1-DoF force estimation, subdivided into 5 trials $\times$ 5 fingers;
4) `5-DoF`: multi-finger contraction following prescribed combinations and trajectories for 5-DoF force estimation in controlled conditions, subdivided into 5 trials $\times$ 5 fixed finger combinations;
5) `RANDOM`: multi-finger contractions performed in a fashion defined *random task*, i.e. with no prescribed protocol of combinations or trajectories, subdivided into 5 trials (each lasting 25 s, performed with a 5 s inter-trial rest to prevent muscle fatigue);

This work uses datasets 2 to 5, oriented to simultaneous multi-finger force estimation. In particular, HYSER `RANDOM` is the closest to real conditions in that the force patterns and dynamic ranges are not fixed and are allowed to differ between Day 1 and Day 2, making HYSER `RANDOM` the most challenging part of HYSER.

Most works that have addressed the HYSER dataset have targeted discrete gesture recognition on the HYSER's `PR` sub-dataset; only little research has been conducted on continuous force estimation on HYSER's `1-DoF`, `5-DoF`, and `RANDOM` sub-datasets. We report the SoA literature on HYSER regression in Table II, also comparing it against this work. It is worth remarking that this work addresses the two HYSER's most challenging settings: (*i*) cross-day validation, and (*ii*) the `RANDOM` dataset, with force patterns and dynamic ranges that are not fixed by protocol and allowed to differ between Day 1 and Day 2. Targeting these two scenarios means using the HYSER dataset in the way closest to real, out-of-the-lab conditions. We provide an in-depth discussion in Section V.

[2] https://www.physionet.org/content/hd-semg/2.0.0/

TABLE II: Literature summary of force estimation on the HYSER HD-sEMG dataset.

WORK	Purpose	Algorithm	HYSER sub-dataset	Validation	Results	HW?	incremental/online learning?
Jiang *et al.* [29] (2021)	HYSER release	FIR kernel	**RANDOM**	within-day, leave-1-trial-out	RMSE $= (8.57 \pm 5.27)\%$ MVC	✗	✗✗
Jiang *et al.* [51] (2022)	channel selection	FIR kernel, random masks	5-DoF	**cross-day**, leave-1-subject-out	RMSE $= (8.66 \pm 0.96)\%$ MVC	✗	✗✗
Jiang *et al.* [52] (2023)	robustness, physiological explainability	deep forests	1-DoF	**cross-day: train on Day 1, test on Day 2**	RMSE $= (8.0 \pm 2.3)\%$ MVC $r_{\text{Pearson}} = 0.900 \pm 0.101$ $R^2 = 0.631 \pm 0.172$	✗	✗✗
Wu *et al.* [53] (2023)	motor units extraction	gCKC BSS, cumulative spike train	1-DoF	no ML-style validation	$r_{\text{Pearson}} = 0.908 \pm$ n.a.	✗	✗✗
This work	incremental online learning embedded on a parallel ultra-low-power MCU	TCN	**RANDOM**	**cross-day: train on Day 1, test on Day 2**	MAE $= (9.58 \pm 3.89)\%$ MVC	✓	✓✓

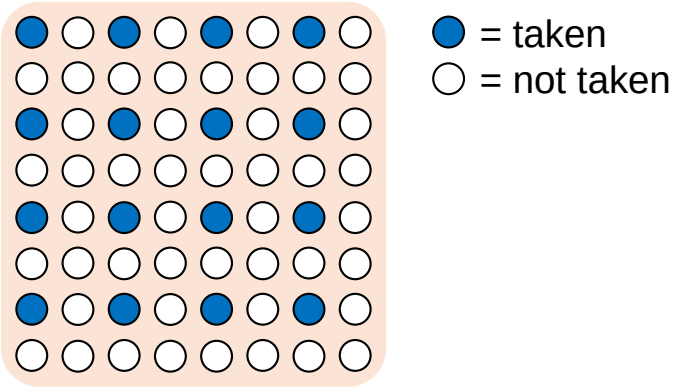


Fig. 1: Uniform downsampling applied to the HYSER HD-sEMG channels. The same $2\times$ downsampling in both directions is applied to all 4 HYSER's 8×8 sensor patches, lowering the channel count from 256 to 64.

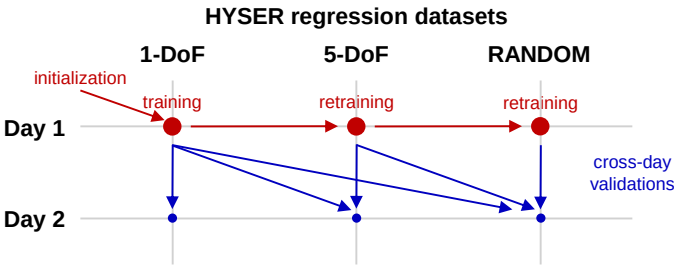


Fig. 2: Scheme of the incremental protocol implemented on the HYSER dataset.

B. Tiny Temporal Convolutional Network

We address force estimation by designing an extremely compact Temporal Convolutional Network (TCN) [40]–[42] that fits the resource constraints of embedded computational platforms in the spirit of TinyML. The TCN's complete structure is detailed in Table III. The net is composed of 5 convolutional blocks followed by a final 3-layer fully connected block; all convolutions and poolings are 1-dimensional, acting only in the time dimension, and have kernel 2, stride 2, and no padding (also termed *valid* padding).

The model's input is constituted by sEMG time-windows of size $64\,\text{channels} \times 63\,\text{samples}$, corresponding to a duration of $\Delta = 30.8\,\text{ms}$; these windows are taken with slide $64\,\text{samples}$, i.e., $31.2\,\text{ms}$, in both training and validation. The 64 channels are obtained by undersampling HYSER's original 256 channels uniformly over the 4 sensor patches; each of the 4 HYSER's 8×8 patches (III-A) is downsampled spatially by $2\times$ along both dimensions, as illustrated in Fig. 1, yielding a total downsampling factor of 4.

The model's output is the 5-dimensional output $\mathbf{y} \in \mathbb{R}^5$ of the last fully connected layer (without any non-linearity), constituting the 5-finger forces estimates. In compliance with the real-time TCN paradigm [40], [41] (which has been yielding advances in the efficient processing of sEMG [24], [34] and other raw biosignals [43], [44]), we adopt a *causal* setup by pairing each ground-truth force $\mathbf{y}$ at time $t_\mathbf{y}$ with the sEMG window $t \in [t_\mathbf{y} - \Delta, t_\mathbf{y}]$, so that each TCN pass only takes into account data from the past, without any leak of information from the future. In contrast, future data leak would happen if pairing the ground truth with the sEMG signal in $t \in \left[t_\mathbf{y} - \frac{1}{2}\Delta, t_\mathbf{y} + \frac{1}{2}\Delta\right]$.

This TCN is designed to be as compact as possible while exploiting the numerous sEMG sensors available on the dataset. We do so by reducing the channels from 64 to 16 in the first convolutional layer. As shown in Table III, the overall size of this TCN is 3317 parameters and 150 kMAC, which is very hardware-friendly for embedded devices [36]–[38].

C. Incremental & Online Learning Protocol

On top of the HYSER dataset, we define a training protocol that is both incremental and online.

1) Incremental Learning: We employ the HYSER 1-Dof, 5-DoF, and RANDOM datasets (III-A) as a sequence of increasingly complex patterns to be learned. The incremental protocol (shwon in Fig. 2) follows three stages, performed separately for each of HYSER's 20 subjects:

TABLE III: Structure of the proposed tiny TCN.

BLOCK	Layer	Size			# MAC (forward pass)	
		input	parameters (weights + biases)	output		
Convolutional 0	conv-ReLU	64×63	2064	16×62	127968	(86.01%)
	max-pooling	16×62		16×31		
Convolutional 1	conv-ReLU	16×31	528	16×30	15840	(10.65%)
	max-pooling	16×30		16×15		
Convolutional 2	conv-ReLU	16×15	264	8×14	3696	(2.48%)
	max-pooling	8×14		8×7		
Convolutional 3	conv-ReLU	8×7	136	8×6	816	(0.55%)
	max-pooling	8×6		8×3		
Convolutional 4	conv-ReLU	8×3	136	8×2	272	(0.18%)
	max-pooling	8×2		8×1		
(flatten)		8×1		8		
Dense	dense-ReLU	8	72	8	72	(0.05%)
	dense-ReLU	8	72	8	72	(0.05%)
	dense	8	45	5	45	(0.03%)
TOTAL		Total parameters: 3317			Total MAC: $149 \cdot 10^3$	

- **Stage 0:** we initialize the TCN (III-B) and train it on HYSER `1-DoF`, Day 1, to learn single-finger force estimation.
- **Stage 1:** we retrain the net on HYSER `5-DoF`, Day 1, to extend the learning to fixed combinations of multi-finger forces;
- **Stage 2:** we retrain on HYSER `RANDOM`, Day 1, to teach the model more diverse, improvised multi-fingers forces not belonging to predefined finger combinations.

All (re)trainings only see data from the Day 1 of the HYSER sub-dataset that is the stage's target. After every (re)training, we validate on Day 2 of `1-Dof`, `5-DoF`, and `RANDOM` to compare the regression quality attained at each stage; in particular, we validate on Day 1 of `5-DoF` and `RANDOM` also before the training on their Day 1, to assess how good the model is before seeing on the new force patterns. It is worth stressing that Day 1 data are used for (re)training, and validations are always performed on Day 2 data, assessing the cross-day generalization capabilities of the trained model. The described procedure is strictly incremental in that the model never sees data from different sessions during a training stage: we never perform multi-session or multi-day training.

2) Online Learning: In this work, we implement and evaluate an online learning mode that mirrors the amount of data and computational budget available in an embedded system working in real time to process the stream of data produced by a new sEMG acquisition session. For our real-time embedded perspective, we define *online learning* as a (re)training of the models that is performed with the following constraints:

- the training examples (i.e., $64\,\text{channels} \times 63\,\text{samples}$ windows) are consumed for training one-by-one: a full forward-backward-update is computed exploiting a single input window;
- this one-by-one procedure strictly happens in the order of acquisition, feeding the input windows to the training in the exact order they are recorded;
- this pass on the session is only done once, so that each input window only contributes once to the training, i.e., with just one forward-backward-update pass (performed individually for each window, due to the requirement of the first point).

In our application, each example element is a $64\,\text{channels} \times 63\,\text{samples}$ sEMG time-window, taken with slide $64\,\text{samples}$, i.e., $31.2\,\text{ms}$, as detailed in III-B. Therefore, in online learning, each of these input windows is processed to perform one iteration of gradient descent of the TCN, consisting of a forward pass, a backward pass, and a model update. We compare the online learning against a baseline represented by a longer, computation-intensive offline training, more favorable to accuracy but not feasible on-device due to the memory footprint of complete training sets and the computational burden of several epochs. In detail, we implement and compare the following two training configurations:

- **Offline baseline:** randomized mini-batching, mini-batch size 32, Adam optimizer with initial learning rate $1 \cdot 10^{-4}$, 32 epochs (i.e., the whole training set is seen 32 times);
- **Online:** sequential mini-batching, mini-batch size 1, gradient descent with fixed learning rate $2 \cdot 10^{-4}$, 1 epoch (i.e., just 1 pass over the training set).

The online settings implement the constraints of real-time training on a computational platform with limited memory: each example is exploited for one forward-backward-update as soon as it is acquired, then discarded; no second traversal on the training session's data is performed since an embedded device is typically not able to store an sEMG session on board.

It is worth remarking that the incremental setup and the online setup are orthogonal experimental settings. In particular, the offline training baseline follows the incremental protocol like the online training but does not implement the latter's streaming behavior. All (re)trainings have the Mean Squared

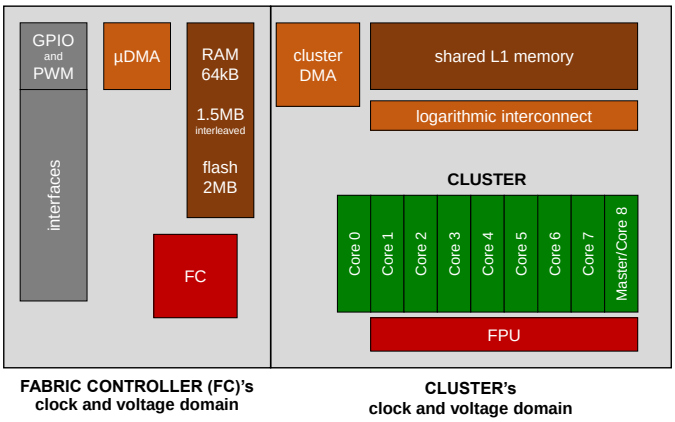


Fig. 3: High-level block diagram of the GAP9 MCU.

Error as a loss function. We always perform training on data from HYSER's Day 1 and validation on Day 2 (III-A), thus obtaining a cross-day assessment that accounts for the sEMG's inherent cross-session variability mostly due to electrode repositioning [23], [24], [54]. We conduct all experiments within-subject, without any multi-subject training or cross-subject validation. In this way, we target the adaptation to new force patterns and the cross-day generalization in a subject-specific scenario. The sEMG's inter-subject variability is outside the scope of this work.

D. Deployment on a Parallel ULP MCU

We execute our application on the commercial microcontroller GWT GAP9.[3] It is equipped with a Parallel Ultra-Low Power (PULP)[4] [56], [57] 9-core cluster accelerator exploiting a RISC-V Instruction Set Architecture extended with instructions specialized for digital signal processing and ML (linear algebra). Fig. 3 shows a high-level block diagram of the platform. GAP9 represents the SoA of low-power MCUs since it has ranked first in latency and energy consumption in the MLPerf Tiny v1.0[5] benchmarking.

We have deployed the TCN model's (III-B) learning via online back-propagation (III-C.2) by using the `TrainLib_Deployer` tool of the PULP-TrainLib[6] [58], [59] library, the first open-source DNN training library for RISC-V-based multi-core MCUs (such as GAP9, mounting PULP), developed to enable the On-Device Learning on this category of parallel ultra-low-power platforms. The library offers matrix multiplication primitives for training and utilities for back-propagation in `float32` and `float16` on multiple cores. The library's `TrainLib_Deployer` is the automated code generator for validating and training user-specified DNNs on a PULP-based platform. The deployer's output is a C source code project to run and profile the training update stages of the model, performing back-propagation with the mini-batch size and a number of epochs set by the user.

[3]Product brief at https://greenwaves-technologies.com/gap9_processor/; for a thorough exposition, refer to the paper about GWT GAP8 [55], the predecessor of GAP9.

[4]https://pulp-platform.org/

[5]https://mlcommons.org/en/inference-tiny-10/

[6]https://github.com/pulp-platform/pulp-trainlib

We performed the online training experiments by streaming the sEMG windows of the HYSER trials from a PC to the GAP9 MCU via a serial interface; GAP9 processes each window executing one training step, i.e., forward pass, backward pass, and weights update. GAP9 repeats the reception-forward-backward-update loop online for each sEMG window individually.

The profiling of the net's training updates was conducted by executing with GAP9's most energy-efficient settings, namely, $V_{\text{dd core}} = 0.65\,\text{V}$ and $f_{\text{CLK}} = 240\,\text{MHz}$. Latency was measured in cycles exploiting the performance counter of the API of PMSIS,[7] the open-source system layer for GAP9's operating system. Latency in physical time was determined as num_cycles$/f_{\text{CLK}}$. The power consumption was measured experimentally with GAP9's Evaluation Kit[8][9] and a Nordic Semiconductor Power Profiler Kit II (PPK2).[10] We used the PPK2 to measure the GAP9 cluster's current draw (excluding the peripherals and the off-chip memories), we used a GPIO to synchronize the current measurement with the code execution, and we determined the energy draw as power$\times$latency.

IV. Experimental Results

A. Regression Error

Fig. 4 shows the MAE obtained with the incremental learning protocol (III-C.1) on all trials of all 20 subjects of the HYSER dataset, using the baseline offline training and the online training (III-C.2). The MAEs of Day 1 of (`1-DoF`, Stage 0), of (`5-DoF`, Stage 1), and of (`1-RANDOM`, Stage 2) are training MAEs, reported to check the cross-day training-to-validation accuracy drop; all other distributions are validation MAEs.

The most relevant difference between the offline (Fig. 4a) and the online training (Fig. 4b) is that the latter yields higher regression errors, both in training and validation. This difference is expected since the online training works in streaming by doing a single pass on the training set (i.e., only 1 epoch) and has thus less opportunity to exploit the training data. However, only the online setup is deployable on resource-constrained embedded devices.

A second general trend, observed in both the offline and the online training, is that in all validations (i.e., on Day 2), the models retrained more recently (based on the protocol stages illustrated in III-C.1) are more accurate. In detail:

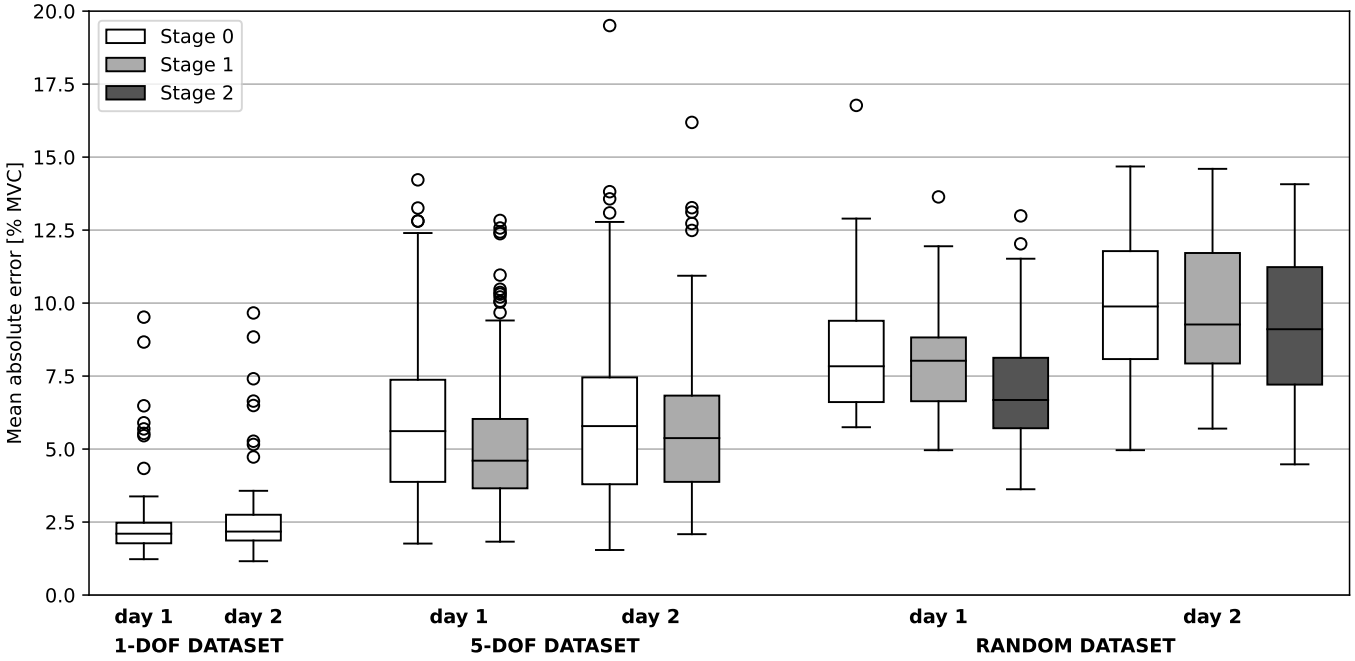
- on Day 2 of the `5-DoF` dataset, the Stage-1 TCN has a lower error distribution than the Stage-0 TCN since the latter is specialized on the single-finger force patterns of the `1-DoF` dataset whereas the former has seen Day 1 of the `5-DoF` dataset;
- analogously, on Day 2 of the `RANDOM` dataset, the Stage-2 TCN has a lower error distribution than the Stage-1

[7]https://greenwaves-technologies.com/manuals/BUILD/HOME/html/index.html

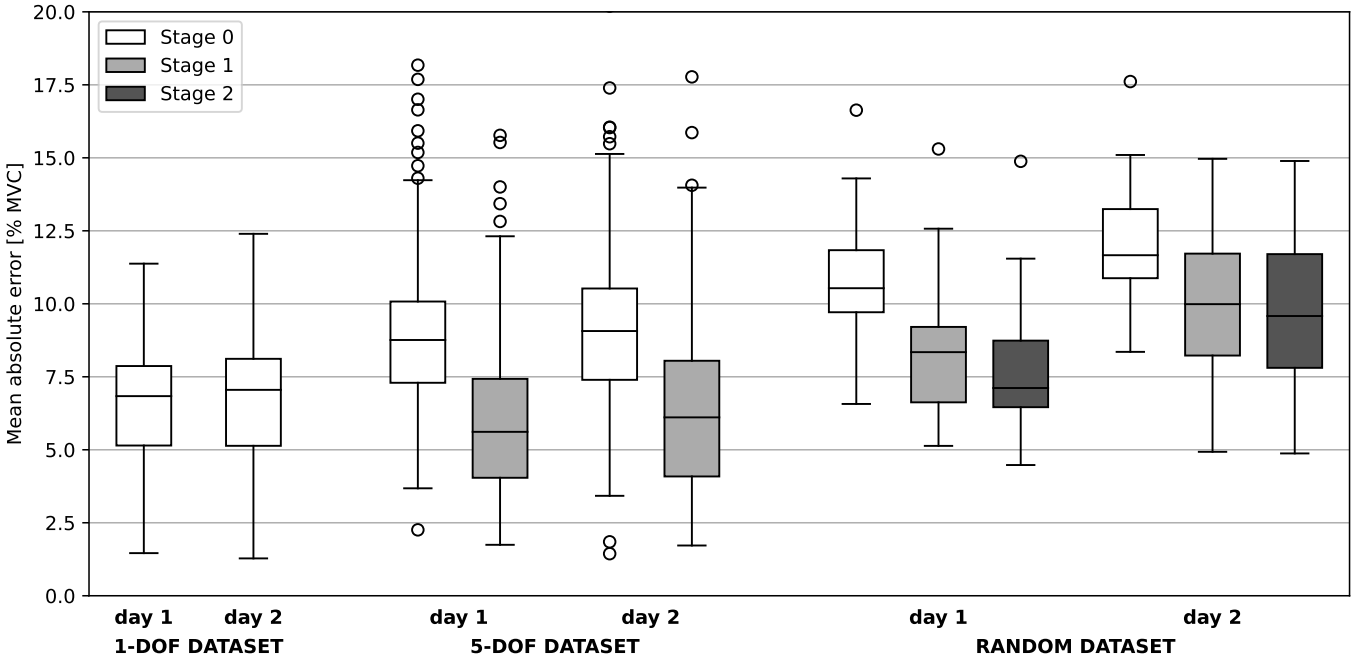
[8]https://greenwaves-technologies.com/product/gap9_evk-gap9-evaluation-kit-efused/

[9]https://greenwaves-technologies.com/product/gap9-resources/

[10]https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2



(a)



(b)

Fig. 4: Experimental distributions of the MAE obtained for the incremental learning stages for baseline offline training and online training. **4a:** Baseline offline training. **4b:** Online training. The MAEs of $(\text{Day } 1, \texttt{1-DoF}, \text{Stage } 0)$, of $(\text{Day } 1, \texttt{5-DoF}, \text{Stage } 1)$, and of $(\text{Day } 1, \texttt{RANDOM}, \text{Stage } 2)$, are *training* MAEs, since Day 1 is the training set of these dataset-stage pairs, as explained in III-C.1; all other distributions are validation MAEs; the training metrics are reported to check the cross-day training-to-validation accuracy drop. The lower (resp., upper) whisker is set at the lowest datum above $Q_1 - 1.5 \cdot \text{IQR}$ (resp., $Q_3 + 1.5 \cdot \text{IQR}$), with Q_1 and Q_3 the first and third quartiles respectively, and $\text{IQR} \triangleq Q_3 - Q_1$ the interquartile range.

TABLE IV: Results of the ablation experiments that reduce the input time window, the HD-sEMG data rate, or the number of HD-sEMG channels (one at a time). The baseline is the setup using 63-sample input time-windows of 64 HD-sEMG channels sampled at 2048 Hz (detailed in III-B; dedicated results in IV-A).

Ablation experiment		MAE, reported as (median ± IQR) % MVC
input time window	47 samples	10.15 ± 3.93
	31 samples	10.54 ± 4.02
data rate 1024 Hz		10.33 ± 3.98
48 HD-sEMG channels		9.98 ± 3.91
Reference setup (III-B; IV-A)		$\mathbf{9.58 \pm 3.89}$

TCN (which in turn performs better than the Stage-0 TCN) since the Stage-2 TCN has seen multi-fingers force patterns out of the fixed combinations of 5-DoF during retraining on RANDOM's Day 1.

This result confirms the motivation for the computational investment in retraining, showing that it is required to improve the regression accuracy. It is worth observing that this improvement is present for both the baseline offline training and the online training, showing that the online training can learn even with the online protocol's limitations (4b).

The most significant validation for summarizing our method's accuracy and generalization ability is the cross-day validation on HYSER RANDOM, which contains diverse non-predefined force patterns like a real-world scenario. In this cross-day validation, the baseline non-deployable offline training and the online training reach a median regression error of $(9.10 \pm 4.02)\%$ MVC and $(9.58 \pm 3.89)\%$ MVC, respectively, expressed as median ± Inter-Quartile Range (IQR). This result indicates that the online training protocol produces a negligible accuracy loss compared to the physiological variability range across subjects and trials. Moreover, this regression error is comparable to that attained by the SoA literature targeting the HYSER dataset, which is in the range $(8.5 \pm 5.0)\%$ MVC in simpler settings, such as within-day validation [29], validation on 5-DoF [51], or validation on 1-DoF [53]. This comparison demonstrates that our setup matches the SoA regression accuracy on the dataset. Moreover, the cited regression works on HYSER only target the regression error without addressing the deployment on embedded hardware and the real-time operation.

B. Ablation Study

The previous subsection (IV-A) exposes the results of the setup that uses 63-sample input time-windows of 64 HD-sEMG channels sampled at 2048 Hz (detailed in III-B). We identified these settings as a sweet spot providing an effective trade-off between accuracy and a low resource budget. We motivate this choice by reporting in Table IV the results of ablation experiments that decrease the computation and memory requirement by reducing the following parameters (one at a time):

- input time window: reduced from 63 samples to 47 or 31 samples (amounting to 22.9 ms or 15.1 ms, respectively);
- data rate: subsampled from 2048 Hz to 1024 Hz;
- channels: reduced from 64 to 48; the 48 channels were obtained by downsampling the 4 HYSER's 8×8 sensor patches by $2\times$ and $3\times$ longitudinally and transversely with respect to the forearm, respectively (the inverse choice, i.e., $3\times$ longitudinally and $2\times$ transversely, yielded worse accuracy).

These ablations were applied one at a time, i.e., one for each ablation experiment. As can be seen from the results in Table IV, all these more compact settings yield sub-optimal regression errors that are not competitive with the reference setup, motivating our choices. The following subsection exposes the deployment and execution results that show that the proposed reference setup is efficient and suitable for an embedded implementation.

C. Profiling on a Parallel ULP MCU

The profiling of the memory occupation of each TCN's forward-backward-update step with mini-batch size 1 yielded 110.9 KiB, which amounts to 86.6% of the 128 KiB available in the L1 memory of GAP9's cluster (III-D). This compact memory footprint shows that our model combines high accuracy and a hardware-friendly size suitable for TinyML applications on resource-constrained devices. As to execution, the profiling yielded a latency of 356 k cycles for each forward-backward-update step of the whole net; this corresponds to 1.49 ms at $f_{\text{clk}} = 240$ MHz, which is the clock frequency of GAP9's most energy-efficient configuration (III-D). This latency is consistent with the training setting of consuming one data window every 31.2 ms (III-B), indicating that our application fulfills the requirements of real-time training. The consensus for the latency requirement for sEMG-based HMIs to make users perceive the control as real-time is 300 ms [60]. Taking into account our 30.8 ms input window plus the 1.5 ms of computation latency, our application matches the sEMG HMI real-time requirements, proving capable of providing a fluid control without a significant perceived delay. The measured power draw was 27.1 mW, which yields an energy consumption of 40.4 µJ for every forward-backward-update step. This value shows that our solution fits the requirements of low-power operation on embedded platforms.

V. Discussion

This section discusses the contribution of this work in comparison with the related SoA literature. In particular, our contribution can be better identified based on the scheme we provide in Table II, reporting the SoA works that use the HYSER dataset targeting the regression task of HD-sEMG-based force estimation. These works represent a minority since most research on the HYSER dataset addresses its PR subdataset for discrete recognition of fixed hand positions; this limitation is common to the domain of sEMG-based hand modeling in general [6].

On the HYSER dataset, the most relevant assessment to determine a model's accuracy and generalization robustness is on the RANDOM dataset (featuring diverse forces not fixed by protocol) in a cross-day validation to be close to a realistic

utilization. In these conditions, the proposed online training achieves a regression error $(9.58 \pm 3.89)\%$ MVC (IV-A). This regression error falls in the same range as the error achieved in literature, which lies in the range of $(8.5 \pm 5.0)\%$ MVC for validations conducted in settings that are easier from the Machine Learning viewpoint, namely, validation performed within-day [29], on `5-DoF` [51], or on `1-DoF` [53]. A remarkable result is that the constraints introduced by the online learning protocol only cause a negligible regression error increase, compared to the physiological variability range across subjects and trials (this physiological variability is an established trend also observed in the literature, as can be seen in the numerical results of Table II). Globally, the comparison highlights that our proposed solution is competitive with the SoA accuracy of sEMG-based force estimation.

As far as deployment and execution are concerned, we are the first to report several progresses. The SoA literature does not address the implementation on resource-constrained hardware, which is a requirement of enabling wearable HMIs. In contrast with this general limitation of all the cited works, we present a TinyML solution that is deployed and profiled an embedded computation device [36]–[38] suitable for a low-power real-time wearable sEMG-based controller. Moreover, our solution is more advanced compared to other TinyML sEMG applications that, though successfully exploiting resource-constrained embedded hardware, only deploy inference [20], [24], [34]. In contrast, we also implement real-time incremental online learning as a key contribution.

VI. Conclusion

In this work, we have presented an incremental online-training solution for sEMG-based estimation of simultaneous multi-finger forces. We propose a tiny TCN devised so that its training computational budget fits the strict resource requirements of embedded platforms. We evaluate our strategy on the HYSER dataset in a cross-day validation, accounting for the sEMG's inherent inter-session variability. With an incremental training protocol, our online training reaches a cross-day MAE of $(9.58 \pm 3.89)\%$ MVC on HYSER's `RANDOM` subset, which has no predefined force pattern sequences and is thus the most challenging and closest to a real scenario. This error is in the same range reached by an offline training with more epochs and resources and is compatible with the SoA literature tackling HYSER with easier training and validation settings. We profile our TCN's online training on the SoA parallel ultra-low power microcontroller GAP9, obtaining a latency of $1.49\,\mathrm{ms}$ and an energy consumption of $40.4\,\mu\mathrm{J}$ for each forward-backward-update step. This proves that our solution fits the requirements of real-time training-on-device on an embedded platform. This work expands our previous paper [25] by contributing a more advanced retraining strategy: we progress from sparse low-count to high-count, high-density electrodes, from classification of fixed gestures to regression, and from cross-arm-posture adaptation to the learning of new simultaneous multi-finger force patterns. This advance is promising for the future intuitive and realiable non-invasive wearable HMIs.

In future work, we will exploit this work's insights about the computational budget of incremental online learning for HD-sEMG-based force regression to implement a high-count sEMG setup and to realize a novel research dataset for hand force modeling [29], [30].

Acknowledgment

We thank Davide Nadalini (University of Bologna, Polytechnic of Turin) and Alberto Dequino (University of Bologna, Polytechnic of Turin) for the assistance in using their PULP-TrainLib library. We thank GreenWaves Technologies for the preview access to the GAP9 SDK.

References

[1] C. J. De Luca, "The use of surface electromyography in biomechanics," *Journal of Applied Biomechanics*, vol. 13, no. 2, pp. 135–163, May 1997. DOI: 10.1123/jab.13.2.135.

[2] R. M. Rangayyan, *Biomedical Signal Analysis*. Wiley, Apr. 2015. DOI: 10.1002/9781119068129.

[3] S. Benatti *et al.*, "Multiple biopotentials acquisition system for wearable applications," in *Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 1*, ser. BIOSTEC 2015, SCITEPRESS - Science and Technology Publications, Lda, 2015, pp. 260–268. DOI: 10.5220/0005320302600268.

[4] C. Castellini and P. van der Smagt, "Surface EMG in advanced hand prosthetics," *Biological Cybernetics*, vol. 100, no. 1, pp. 35–47, Nov. 2008. DOI: 10.1007/s00422-008-0278-1.

[5] L. Guo *et al.*, "Human-machine interaction sensing technology based on hand gesture recognition: A review," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 300–309, 2021. DOI: 10.1109/THMS.2021.3086003.

[6] A. Phinyomark and E. Scheme, "EMG pattern recognition in the era of big data and deep learning," *Big Data and Cognitive Computing*, vol. 2, no. 3, 2018. DOI: 10.3390/bdcc2030021.

[7] Y. Wei *et al.*, "A review of algorithm & hardware design for AI-based biomedical applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 2, pp. 145–163, 2020. DOI: 10.1109/TBCAS.2020.2974154.

[8] R. Meattini *et al.*, "Experimental evaluation of a sEMG-based human-robot interface for human-like grasping tasks," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 1030–1035. DOI: 10.1109/ROBIO.2015.7418907.

[9] S. Benatti *et al.*, "Analysis of robust implementation of an EMG pattern recognition based control," in *Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4*, ser. BIOSTEC 2014, SCITEPRESS - Science and Technology Publications, Lda, 2014, pp. 45–54. DOI: 10.5220/0004800300450054.

[10] B. Milosevic *et al.*, "Exploring arm posture and temporal variability in myoelectric hand gesture recognition," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2018, pp. 1032–1037. DOI: 10.1109/BIOROB.2018.8487838.

[11] V. H. Cene *et al.*, "Open database for accurate upper-limb intent detection using electromyography and reliable extreme learning machines," *Sensors*, vol. 19, no. 8, 2019. DOI: 10.3390/s19081864.

[12] S. Benatti *et al.*, "Online learning and classification of EMG-based gestures on a parallel ultra-low power platform using hyperdimensional computing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 3, pp. 516–528, 2019. DOI: 10.1109/TBCAS.2019.2914476.

[13] E. Donati *et al.*, "Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 5, pp. 795–803, 2019. DOI: 10.1109/TBCAS.2019.2925454.

[14] A. Vitale *et al.*, "Neuromorphic edge computing for biomedical applications: Gesture classification using EMG signals," *IEEE Sensors Journal*, vol. 22, no. 20, pp. 19 490–19 499, 2022. DOI: 10.1109/JSEN.2022.3194678.

[15] A. Krasoulis and K. Nazarpour, "Myoelectric digit action decoding with multi-output, multi-class classification: An offline analysis," *Scientific Reports*, vol. 10, no. 1, Oct. 2020. DOI: 10.1038/s41598-020-72574-7.

[16] Y. Hu *et al.*, "A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition," *PLOS ONE*, vol. 13, pp. 1–18, Oct. 2018. DOI: 10.1371/journal.pone.0206049.

[17] E. Ceolini *et al.*, "Hand-gesture recognition based on EMG and event-based camera sensor fusion: A benchmark in neuromorphic computing," *Frontiers in Neuroscience*, vol. 14, 2020. DOI: 10.3389/fnins.2020.00637.

[18] S. Tam *et al.*, "A fully embedded adaptive real-time hand gesture classifier leveraging HD-sEMG and deep learning," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 2, pp. 232–243, 2020. DOI: 10.1109/TBCAS.2019.2955641.

[19] F. Chamberland *et al.*, "Novel wearable HD-EMG sensor with shift-robust gesture recognition using deep learning," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 17, no. 5, pp. 968–984, 2023. DOI: 10.1109/TBCAS.2023.3314053.

[20] N. Leroux *et al.*, "Online transformers with spiking neurons for fast prosthetic hand control," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2023, pp. 1–6. DOI: 10.1109/BioCAS58349.2023.10388996.

[21] E. Donati *et al.*, "Long-term stable electromyography classification using canonical correlation analysis," in *2023 11th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2023, pp. 1–4. DOI: 10.1109/NER52421.2023.10123768.

[22] P. Kaufmann *et al.*, "Fluctuating EMG signals: Investigating long-term effects of pattern matching algorithms," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2010, pp. 6357–6360. DOI: 10.1109/IEMBS.2010.5627288.

[23] S. Amsüss *et al.*, "Long term stability of surface EMG pattern classification for prosthetic control," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2013. DOI: 10.1109/embc.2013.6610327.

[24] M. Zanghieri *et al.*, "Robust real-time embedded EMG recognition framework using temporal convolutional networks on a multicore IoT processor," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 2, pp. 244–256, 2020. DOI: 10.1109/TBCAS.2019.2959160.

[25] M. Zanghieri *et al.*, "Online unsupervised arm posture adaptation for sEMG-based gesture recognition on a parallel ultra-low-power microcontroller," in *2023 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2023, pp. 1–5. DOI: 10.1109/BioCAS58349.2023.10388902.

[26] M. Zanghieri *et al.*, "Temporal variability analysis in sEMG hand grasp recognition using temporal convolutional networks," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2020, pp. 228–232. DOI: 10.1109/AICAS48895.2020.9073888.

[27] A. Burrello *et al.*, "Tackling time-variability in sEMG-based gesture recognition with on-device incremental learning and temporal convolutional networks," in *2021 IEEE Sensors Applications Symposium (SAS)*, 2021, pp. 1–6. DOI: 10.1109/SAS51076.2021.9530007.

[28] A. Krasoulis *et al.*, "Effect of user practice on prosthetic finger control with an intuitive myoelectric decoder," *Frontiers in Neuroscience*, vol. 13, 2019. DOI: 10.3389/fnins.2019.00891.

[29] X. Jiang *et al.*, "Open access dataset, toolbox and benchmark processing results of high-density surface electromyogram recordings," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1035–1046, 2021. DOI: 10.1109/TNSRE.2021.3082551.

[30] A. Matran-Fernandez *et al.*, "SEEDS, simultaneous recordings of high-density emg and finger joint angles during multiple hand movements," *Scientific Data*, vol. 6, no. 1, Sep. 2019. DOI: 10.1038/s41597-019-0200-9.

[31] P. Koch *et al.*, "Regression of hand movements from sEMG data with recurrent neural networks," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 3783–3787. DOI: 10.1109/EMBC44109.2020.9176278.

[32] T. Bao *et al.*, "A deep Kalman filter network for hand kinematics estimation using sEMG," *Pattern Recognition Letters*, vol. 143, pp. 88–94, 2021. DOI: 10.1016/j.patrec.2021.01.001.

[33] L. Meng *et al.*, "Evaluation of decomposition parameters for high-density surface electromyogram using fast independent component analysis algorithm," *Biomedical Signal Processing and Control*, vol. 75, p. 103 615, 2022. DOI: 10.1016/j.bspc.2022.103615.

[34] M. Zanghieri *et al.*, "sEMG-based regression of hand kinematics with temporal convolutional networks on a low-power edge microcontroller," in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, 2021, pp. 1–6. DOI: 10.1109/COINS51742.2021.9524188.

[35] M. Zanghieri *et al.*, "Event-based low-power and low-latency regression method for hand kinematics from surface EMG," in *2023 9th International Workshop on Advances in Sensors and Interfaces (IWASI)*, 2023, pp. 293–298. DOI: 10.1109/IWASI58316.2023.10164372.

[36] L. Dutta and S. Bharali, "TinyML meets IoT: A comprehensive survey," *Internet of Things*, vol. 16, p. 100 461, 2021. DOI: 10.1016/j.iot.2021.100461.

[37] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, 2022. DOI: 10.1016/j.jksuci.2021.11.019.

[38] Y. Abadade *et al.*, "A comprehensive survey on TinyML," *IEEE Access*, pp. 1–1, 2023. DOI: 10.1109/ACCESS.2023.3294111.

[39] L. Ravaglia *et al.*, "Memory-latency-accuracy trade-offs for continual learning on a RISC-V extreme-edge node," in *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, 2020, pp. 1–6. DOI: 10.1109/SiPS50750.2020.9195220.

[40] C. Lea *et al.*, "Temporal convolutional networks for action segmentation and detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1003–1012. DOI: 10.1109/CVPR.2017.113.

[41] S. Bai *et al.*, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *CoRR*, vol. abs/1803.01271, 2018. DOI: 10.48550/arXiv.1803.01271.

[42] A. Burrello *et al.*, "TCN mapping optimization for ultra-low power time-series edge inference," in *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2021, pp. 1–6. DOI: 10.1109/ISLPED52811.2021.9502494.

[43] M. Zanghieri *et al.*, "Low-latency detection of epileptic seizures from iEEG with temporal convolutional networks on a low-power parallel MCU," in *2021 IEEE Sensors Applications Symposium (SAS)*, 2021, pp. 1–6. DOI: 10.1109/SAS51076.2021.9530181.

[44] A. Burrello *et al.*, "Embedding temporal convolutional networks for energy-efficient PPG-based heart rate monitoring," *ACM Trans. Comput. Healthcare*, vol. 3, no. 2, Mar. 2022. DOI: 10.1145/3487910.

[45] Y. Li *et al.*, *Revisiting batch normalization for practical domain adaptation*, 2016. DOI: 10.48550/arXiv.1603.04779.

[46] Y. Du *et al.*, "Surface EMG-based inter-session gesture recognition enhanced by deep domain adaptation," *Sensors*, vol. 17, no. 3, 2017. DOI: 10.3390/s17030458.

[47] L. Pellegrini *et al.*, "Latent replay for real-time continual learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 203–10 209. DOI: 10.1109/IROS45743.2020.9341460.

[48] C. Castellini *et al.*, "Fine detection of grasp force and posture by amputees via surface electromyography," *Journal of Physiology - Paris*, vol. 103, no. 3, pp. 255–262, 2009. DOI: 10.1016/j.jphysparis.2009.08.008.

[49] R. C. Sîmpetru *et al.*, "Accurate continuous prediction of 14 degrees of freedom of the hand from myoelectrical signals through convolutive deep learning," in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2022, pp. 702–706. DOI: 10.1109/EMBC48229.2022.9870937.

[50] R. C. Sîmpetru *et al.*, "Sensing the full dynamics of the human hand with a neural interface and deep learning," *bioRxiv*, 2022. DOI: 10.1101/2022.07.29.502064.

[51] X. Jiang *et al.*, "Random channel masks for regularization of least squares-based finger EMG-force modeling to improve cross-day performance," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2157–2167, 2022. DOI: 10.1109/TNSRE.2022.3194246.

[52] X. Jiang *et al.*, "Explainable and robust deep forests for EMG-force modeling," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 6, pp. 2841–2852, 2023. DOI: 10.1109/JBHI.2023.3262316.

[53] W. Wu *et al.*, "A new EMG decomposition framework for upper limb prosthetic systems," *Journal of Bionic Engineering*, Jul. 2023. DOI: 10.1007/s42235-023-00407-0.

[54] F. Palermo *et al.*, "Repeatability of grasp recognition for robotic hand prosthesis control based on sEMG data," in *2017 International Conference on Rehabilitation Robotics (ICORR)*, 2017, pp. 1154–1159. DOI: 10.1109/ICORR.2017.8009405.

[55] E. Flamand *et al.*, "GAP-8: A RISC-V SoC for AI at the edge of the IoT," in *2018 IEEE 29th International Conference on Application-*

specific Systems, Architectures and Processors (ASAP), 2018, pp. 1–4. DOI: 10.1109/ASAP.2018.8445101.

[56] F. Conti *et al.*, "PULP: A ultra-low power parallel accelerator for energy-efficient and flexible embedded vision," *J. Signal Process. Syst.*, vol. 84, no. 3, pp. 339–354, Sep. 2016. DOI: 10.1007/s11265-015-1070-9.

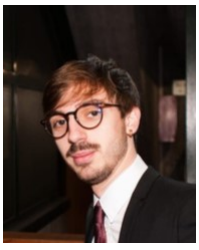
[57] A. Garofalo *et al.*, "PULP-NN: Accelerating quantized neural networks on parallel ultra-low-power RISC-V processors," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 378, no. 2164, p. 20 190 155, Dec. 2019. DOI: 10.1098/rsta.2019.0155.

[58] D. Nadalini *et al.*, "PULP-TrainLib: Enabling on-device training for RISC-V multi-core MCUs through performance-driven autotuning," in *Embedded Computer Systems: Architectures, Modeling, and Simulation*, Cham: Springer International Publishing, 2022, pp. 200–216. DOI: 10.1007/978-3-031-15074-6_13.

[59] D. Nadalini *et al.*, "Reduced precision floating-point optimization for deep neural network on-device learning on microcontrollers," *Future Generation Computer Systems*, vol. 149, pp. 212–226, 2023. DOI: https://doi.org/10.1016/j.future.2023.07.020.

[60] B. Hudgins *et al.*, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993. DOI: 10.1109/10.204774.

Marcello Zanghieri (Graduate Student Member, IEEE) received his M.Sc. in Physics (cum laude) from the University of Bologna, Italy, in 2019. He is currently working toward his Ph.D. in Data Science and Computation under the supervision of Prof. Luca Benini at the Energy-Efficient Embedded Systems Laboratory (EEES Lab) of the University of Bologna. His research interests focus on time series analysis with machine learning and deep learning, focusing on sEMG, ultrasounds, and EEG to advance real-time human-machine interaction based on embedded computing platforms, including parallel (ultra-)low-power microcontrollers.



Pierangelo Maria Rapa (Graduate Student Member, IEEE) received his M.Sc. degree in Electronic Engineering from the University of Bologna, Italy, in 2022. He is currently working on his Ph.D. in Automotive Engineering for Intelligent Mobility under the supervision of Prof. S. Benatti at the DEI department, University of Bologna. His research interests include biosignals and ultra-low-power Human-Machine Interfaces with specific emphasis on their application in the automotive industry.

Mattia Orlandi (Graduate Student Member, IEEE) received his M.Sc. degree in Artificial Intelligence from the University of Bologna, Italy, in 2022. He is currently working toward his Ph.D. Data Science and Computation under the supervision of Prof. S. Benatti at the Energy-Efficient Embedded Systems Laboratory (EEES Lab), DEI Department, University of Bologna. His research activities involve bio-signal processing with machine learning on low-power computing platforms. He is investigating how to decode EMG signals into spike trains to develop advanced human-machine interfaces.

Elisa Donati (Member, IEEE) received the B.Sc. and M.Sc. degrees (cum laude) in biomedical engineering from the University of Pisa, Pisa, Italy, and the Ph.D. degree in biorobotics from the Sant'Anna School of Advanced Studies, Pisa. She is currently a Research Fellow with the Institute of Neuroinformatics, University of Zürich and ETH Zürich. Her research activities include the interface of the neuroscience and neuromorphic engineering. She is interested in understanding how the biological neural circuits carry out the computations and apply them in biomedical application and neurorobotics. She was the Co-Coordinator of the H2020 EU CSA Project NEUROTECH.

Luca Benini (Fellow, IEEE) holds the chair of digital Circuits and systems at ETH Zurich, Switzerland, and is Full Professor at the Università di Bologna, Italy. He received his Ph.D. from Stanford University. Dr. Benini's research interests are energy-efficient parallel computing systems, smart sensing micro-systems, and machine learning hardware. He is a Fellow of the ACM and a member of the Academia Europaea. He is the recipient of the 2016 IEEE CAS Mac Van Valkenburg Award, the 2020 EDAA Achievement Award, the 2020 ACM/IEEE A. Richard Newton Award, and the 2023 IEEE CS E.J. McCluskey Award.

Simone Benatti (Member, IEEE) received his Ph.D. degree in Electronics, Telecommunications, and Information Technologies from the University of Bologna under the supervision of Prof. Luca Benini. During the Ph.D., he was a visiting fellow at BWRC - University of California, Berkeley (supervisor Prof. Jan Rabaey). Currently, he serves as Assistant Professor at the University of Modena and Reggio Emilia while pursuing his collaboration with the EEES Lab of the University of Bologna. In 2023, he was appointed Visiting Professor at EFCL-ETHZ. His research interests focus on energy-efficient embedded systems for IoT and biomedical applications. This includes hardware/software codesign to efficiently address performance, as well as advanced algorithms. Dr. Benatti works on designing and optimizing energy-efficient embedded systems for biopotential (EMG and EEG) acquisition and processing and on Brain-Machine Interfaces for Human-Machine Interaction. In this field, he has published more than 90 papers in international peer-reviewed conferences and journals. He has ongoing collaborations with several international research institutes, such as ETHZ-EFCL, EPFL, TU Graz, FBK, and Politecnico di Torino. Dr. Benatti is recipient of the GHAIA grant (H2020-MSCA-RISE-2017, g.a. 777822).