IWES 2020
5th Italian Workshop on Embedded Systems

Energy-efficient PPG-based Heart-Rate Monitoring

Matteo Risso[†], Alessio Burrello[*], Daniele Jahier Pagliari[†], Simone Benatti[*],
Luca Benini[*‡], Massimo Poncino[†] and Enrico Macii[†]
[†]Politecnico di Torino, [*]University of Bologna, [‡]ETH Zürich

Key contributions

- **Temporal Convolutional Networks (TCNs)** for Heart-Rate (HR) monitoring
  - **Neural Architecture Search (NAS)** design space exploration
- **SOTA** results against other Deep Learning (DL) solutions
- **Energy-Efficient** embedded deployment

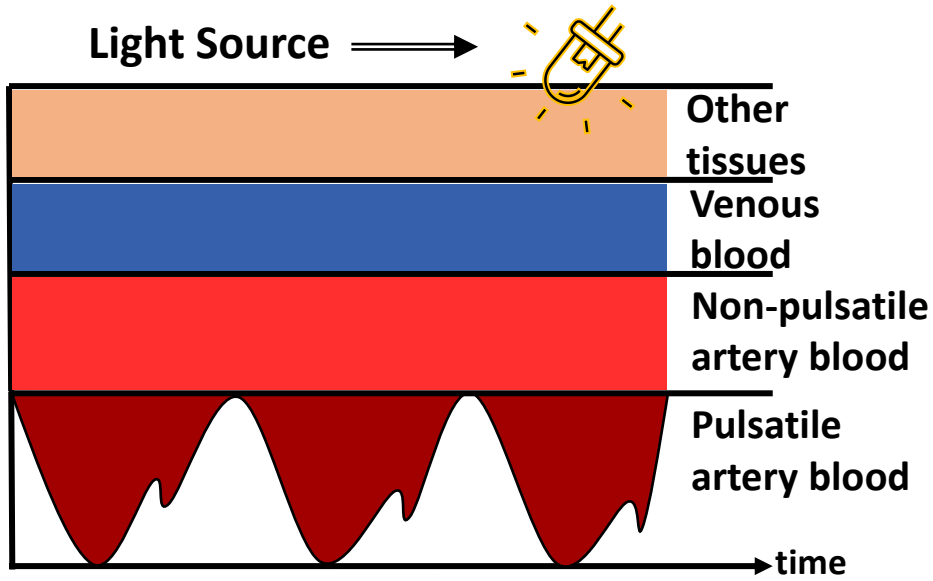
Outline

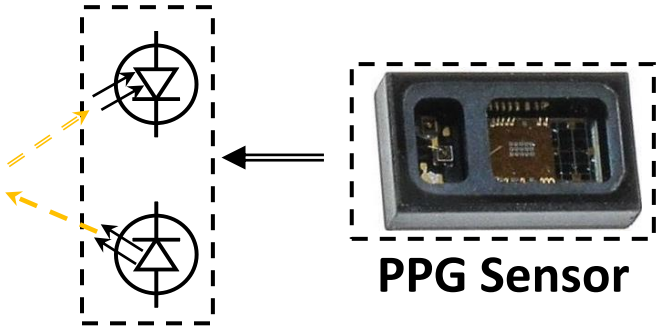
- PPG-Based Heart Rate Monitoring
- Dataset: PPG-DaLiA (PPG in *Daily Life Activities*)
- TimePPG : Optimized TCNs for PPG-based Heart Rate Monitoring
- Experimental Results

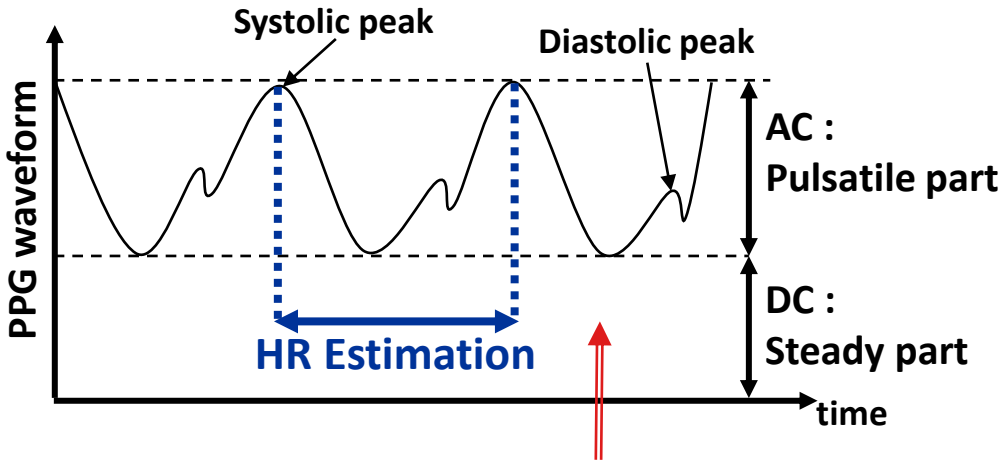
Outline

- PPG-Based Heart Rate Monitoring
- Dataset: PPG-DaLiA (PPG in *Daily Life Activities*)
- TimePPG : Optimized TCNs for PPG-based Heart Rate Monitoring
- Experimental Results

PPG-Based Heart-Rate Monitoring

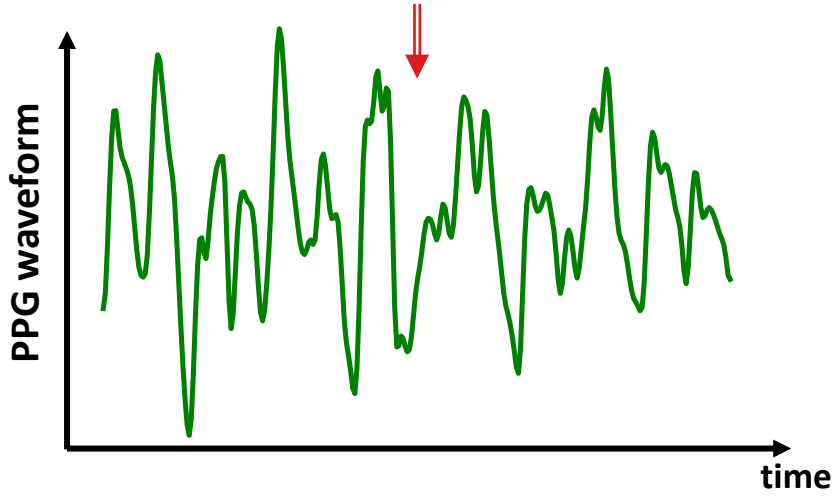






Ideal PPG waveform : easy to identify peaks

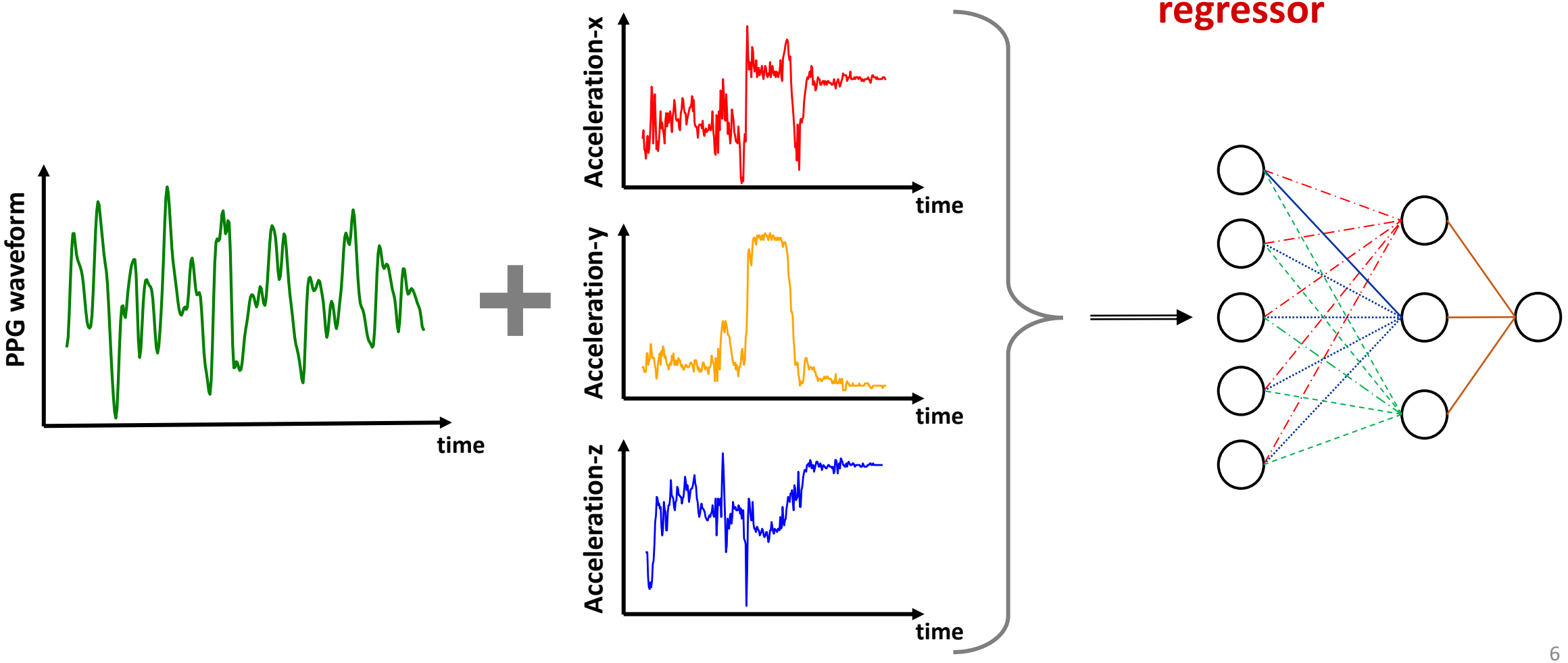
Real PPG waveform with Motion Artifacts (MAs)



Very difficult to identify peaks

How to track HR with MAs ?

Couple PPG signal with inertial sensor data → **Fed to a Neural Network regressor**



Outline

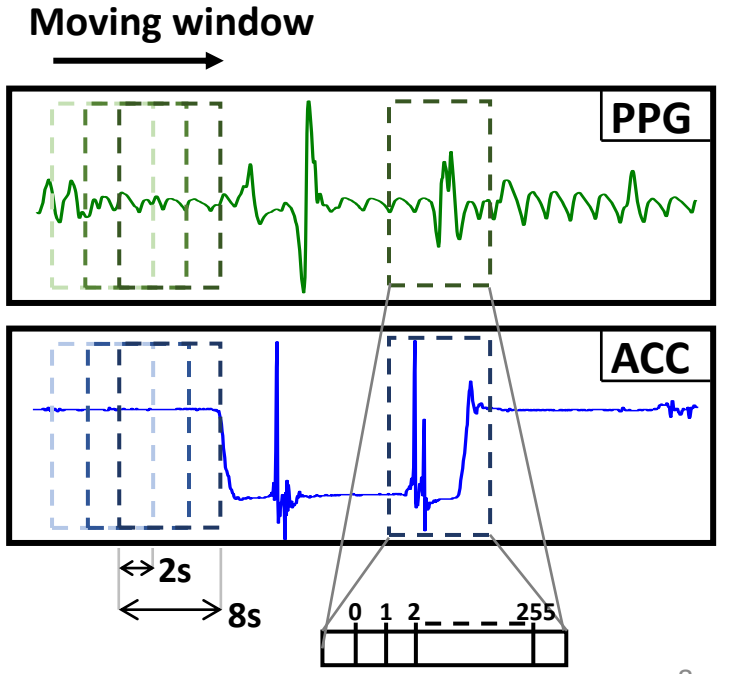
- PPG-Based Heart Rate Monitoring
- Dataset: PPG-DaLiA (PPG in *Daily Life Activities*)
- TimePPG : Optimized TCNs for PPG-based Heart Rate Monitoring
- Experimental Results

PPG-Dalia Dataset

- **15** Patients
- **37.5** hours of recordings
- **8** different daily activities, e.g., Driving, Cycling, ...
- PPG-Sensor and 3D-Accelerometer embedded in same device
- ECG data as HR ground truth

PPG-DaLiA

- PPG and Acceleration data are sliced with a **8/2 sliding window**
- Each resultant series present **256 samples**

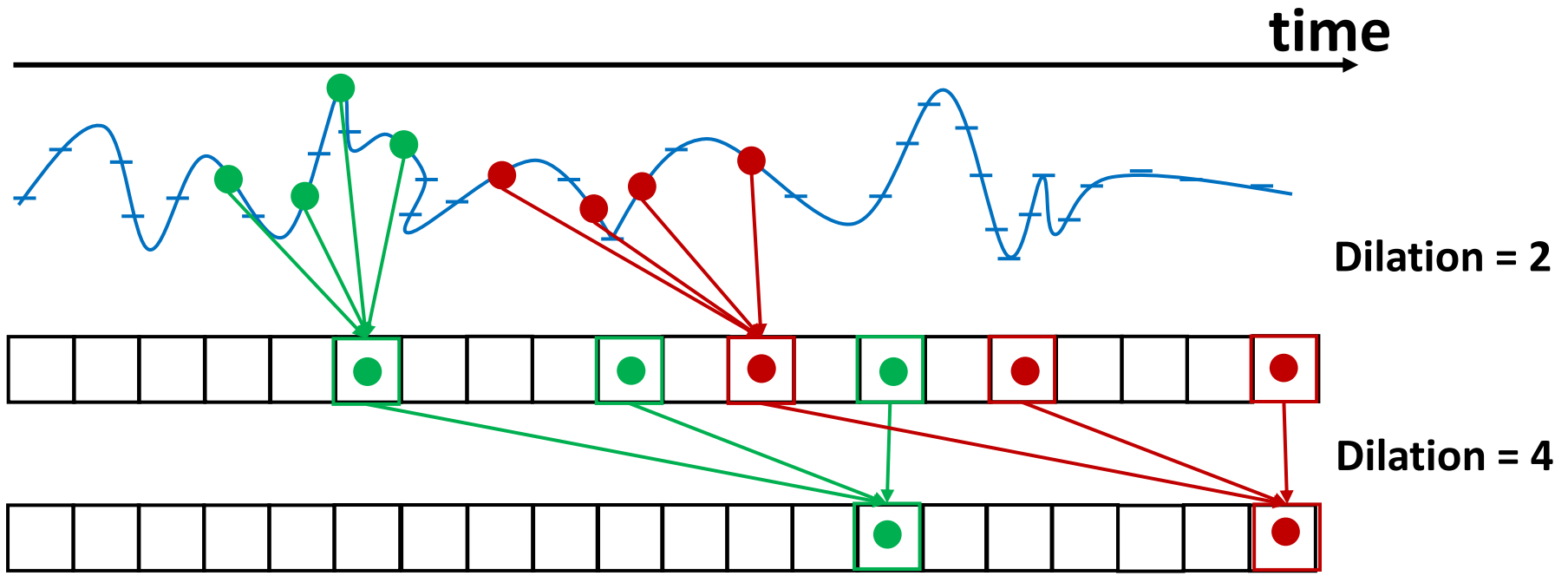


Outline

- PPG-Based Heart Rate Monitoring
- Dataset : PPG-DaLiA (PPG in *Daily Life Activities*)
- TimePPG : Optimized TCNs for PPG-based Heart Rate Monitoring
- Experimental Results

Temporal Convolutional Networks

- TCNs are a class of 1D CNN, with two main peculiarity :
  - **Causality**
  - **Dilation**



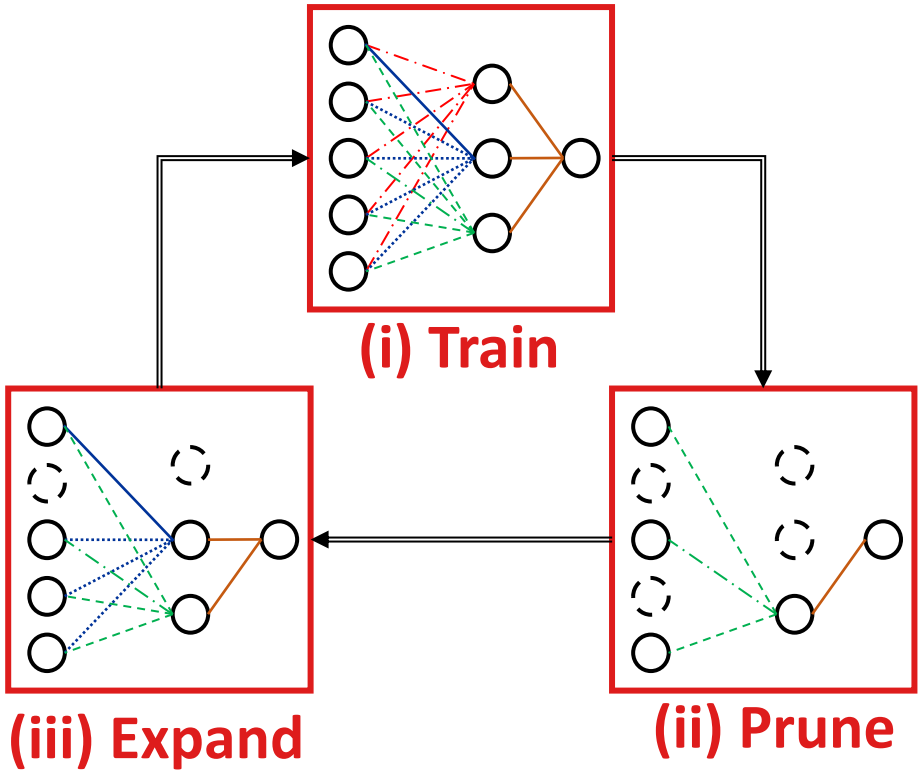
MorphNet

- MorphNet is a lightweight **NAS** algorithm, which learns the **optimal number of channel** of each convolutional layer in a seed architecture
- Possible target metrics : Network Size, MACs, Latency
- MorphNet algorithm :

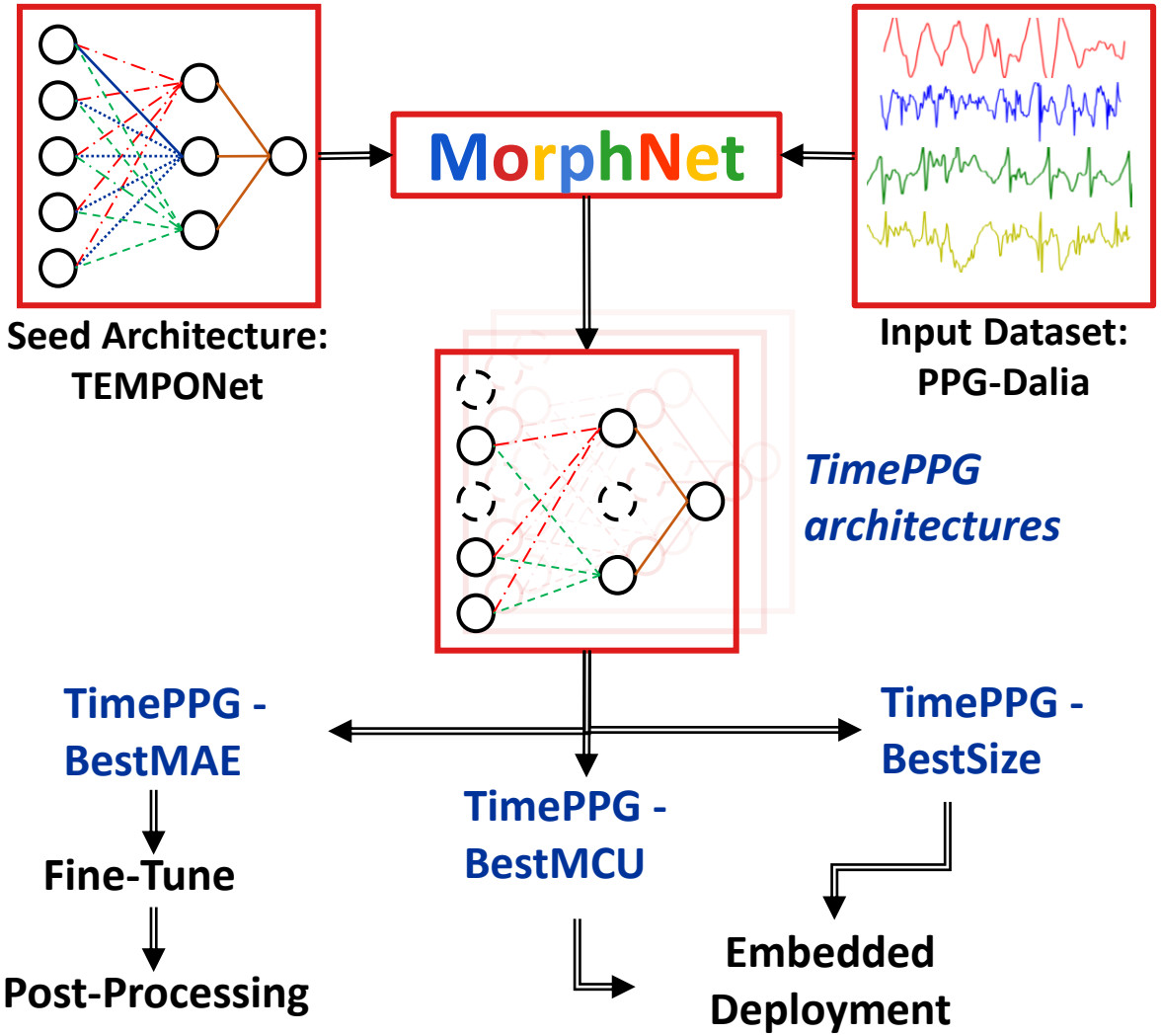
(i) Train seed network with a **sparsifying regularizer**

(ii) Prune output channels with **L1-norm ≤ threshold**

(iii) Expand output channels via **uniform width-multiplier**



TimePPG workflow



- Feed **MorphNet** NAS with TEMPONet and PPG-Dalia
- Generate a collection of **pruned** TEMPONet flavors
- Identify best networks :
  - **BestMAE**
  - **BestMCU**
  - **BestSize**

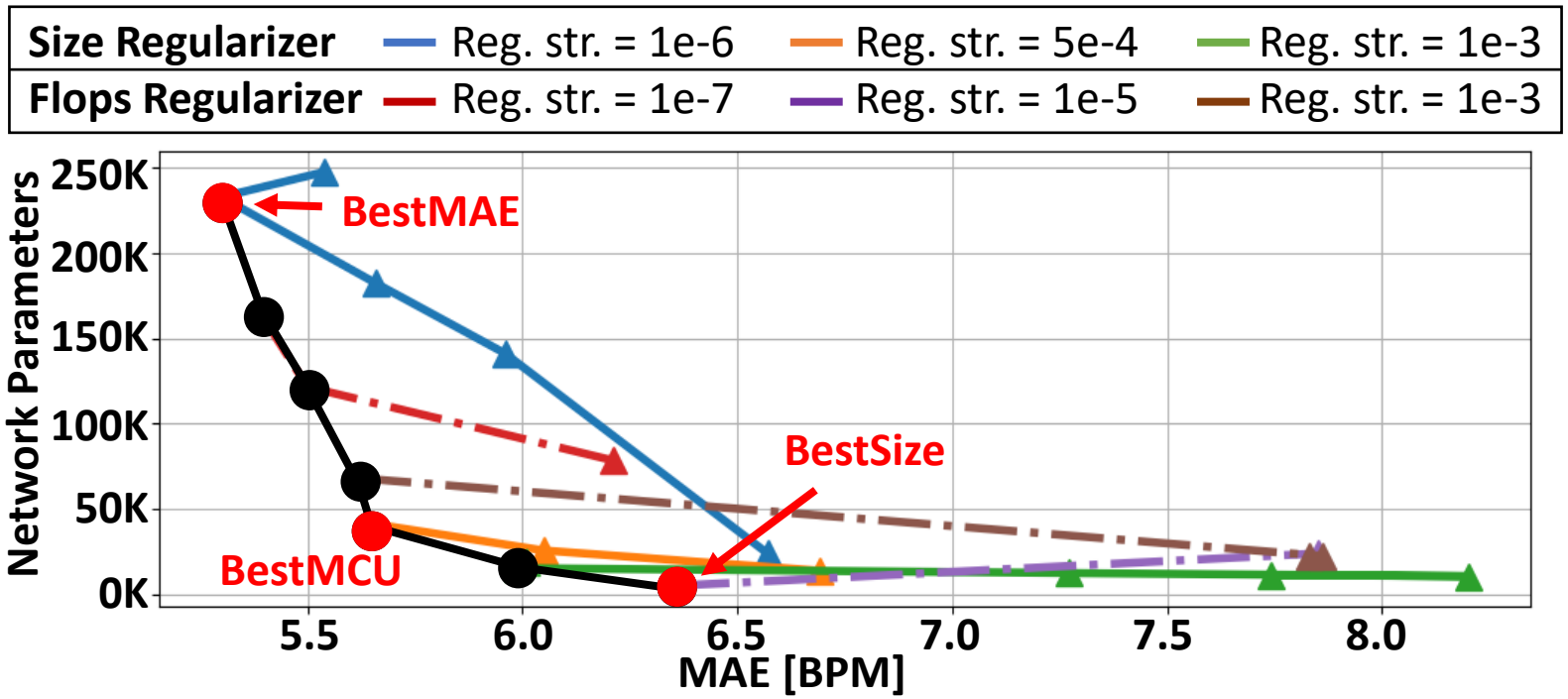
→ Deploy

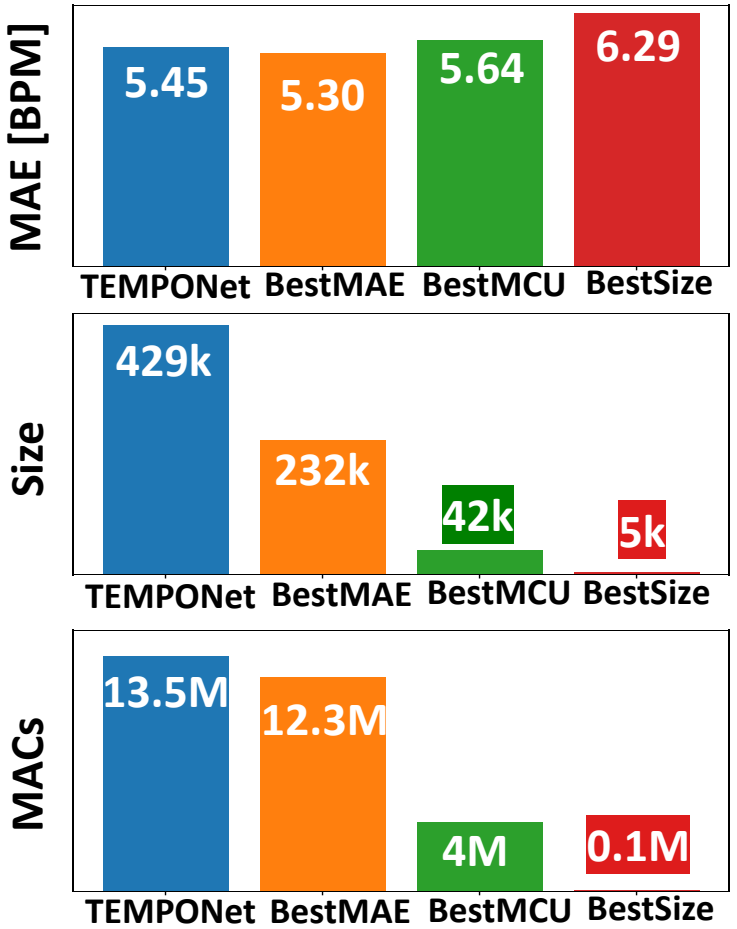
Outline

- PPG-Based Heart Rate Monitoring
- Dataset : PPG-DaLiA (PPG in *Daily Life Activities*)
- TimePPG : Optimized TCNs for PPG-based Heart Rate Monitoring
- Experimental Results

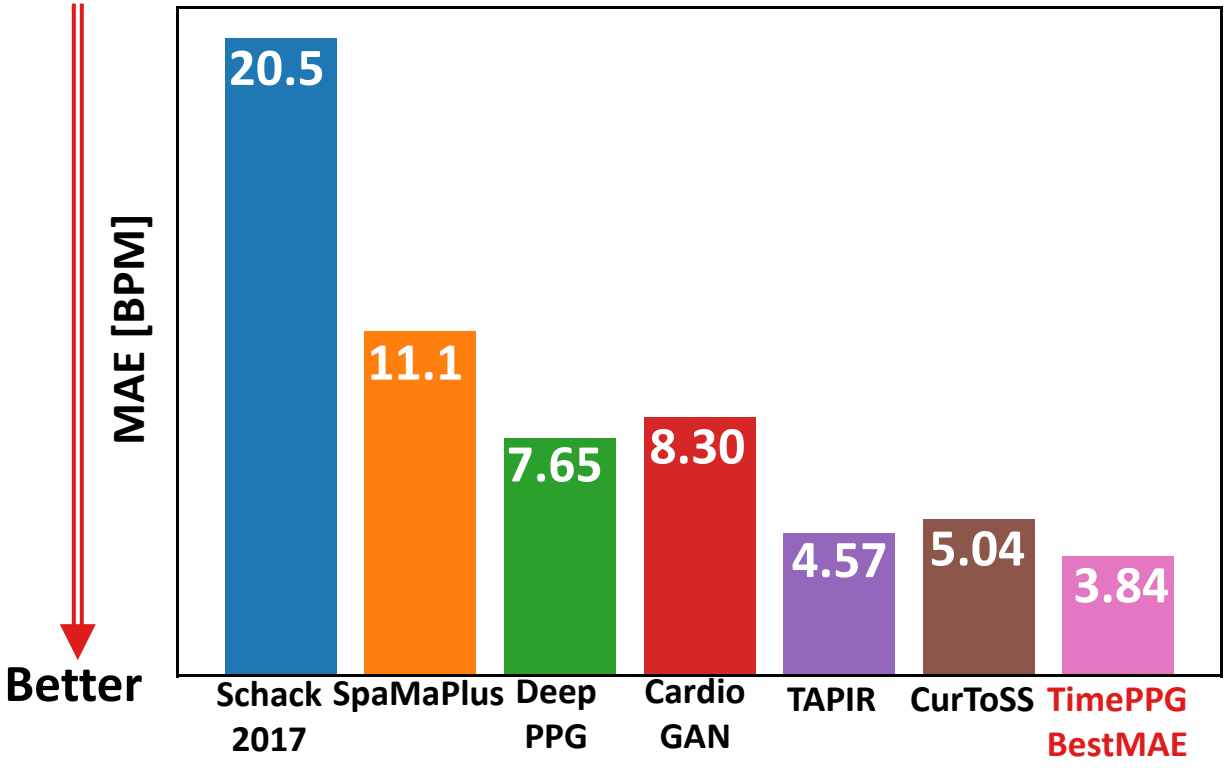
Design Space Exploration

- **Grid search** on MorphNet hyper-parameters, tweaking **regularization strength** and **pruning threshold**





State-of-the-art comparison

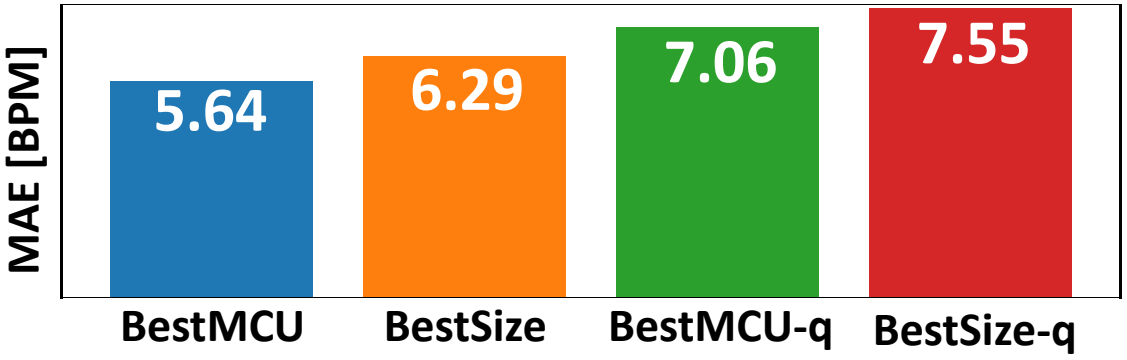


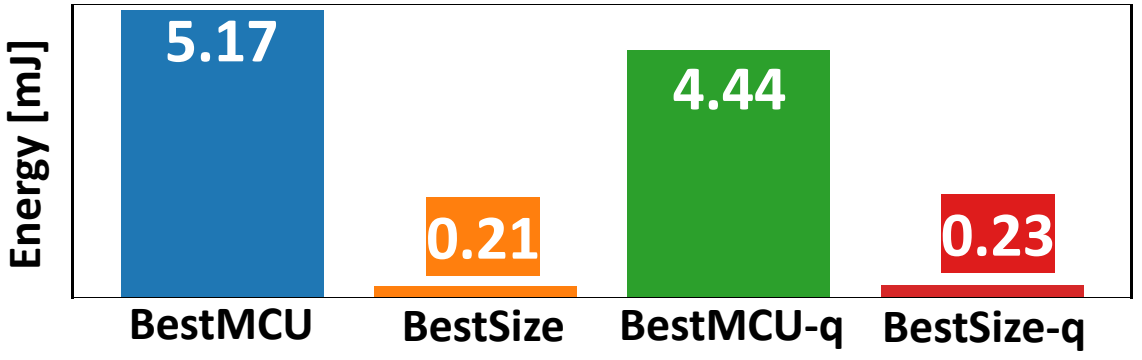
- **TimePPG-BestMAE** outperforms :
  - classical algorithms such as **Schack2017** and **SpaMaPlus** *
  - Deep Lerning methods such as **DeepPPG (CNN)** and **CardioGAN**
  - Adaptive filtering techniques such as **TAPIR** and **CurToSS**, which are optimized on PPG-Dalia

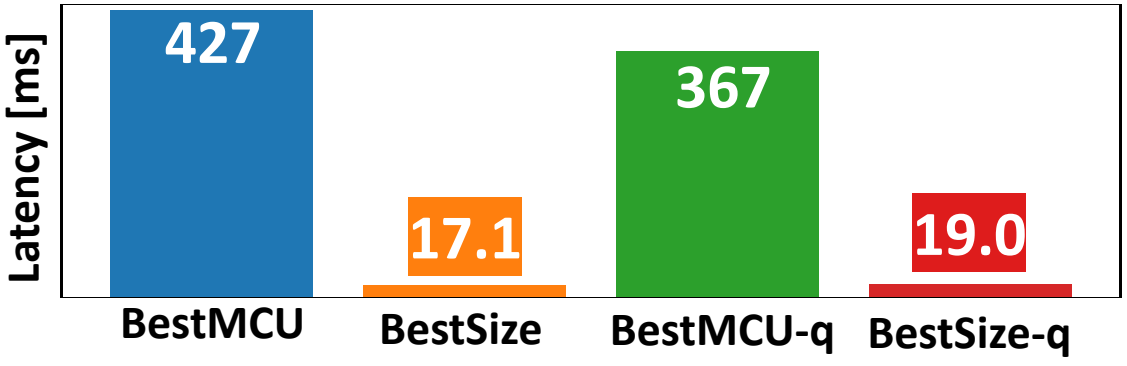
*Optimized on a different dataset

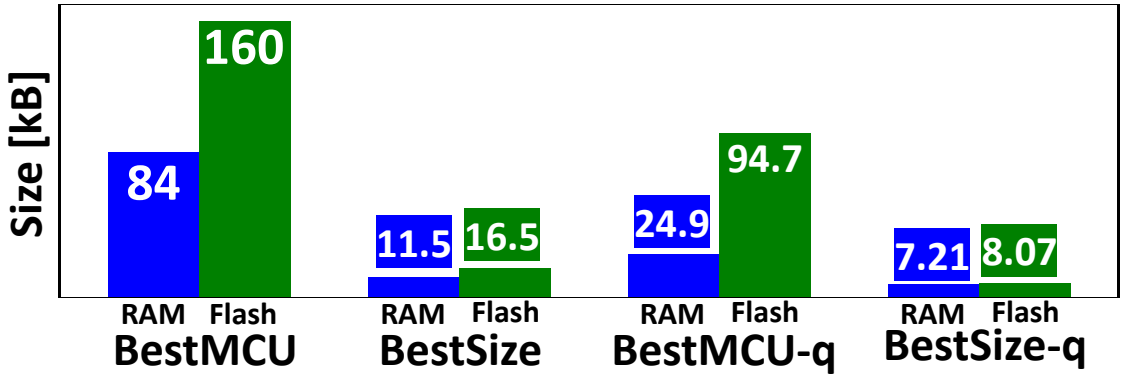
Embedded Deployment

- We deploy **BestMCU** and **BestSize** TCNs on the **STM32L476RG**
- We deploy both **float32** and **int8-quantized** variants of each TCN





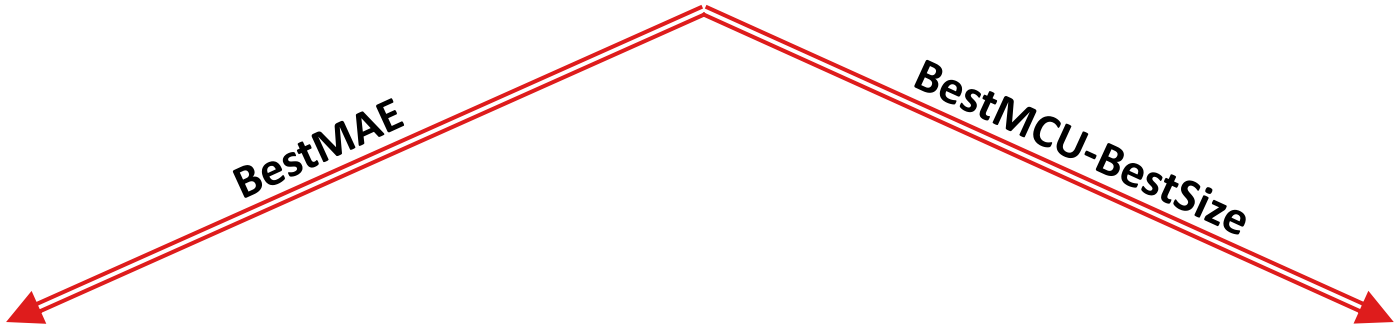




⚠ STM proprietary deployment toolchain, **X-CUBE-AI**, does not support int8 dilated convolutions ⟹ Fix dilations to 1

Conclusions

- Fast and Efficient design space exploration



- **State-of-the-art results** on the PPG-Dalia dataset with **3.84 BPM** average MAE
- Efficient deployment on a tiny commercial MCU obtaining as few as **5k** parameters with a latency of **17.1 ms** consuming just **0.21 mJ**