REAP: Runtime Energy-Accuracy Optimization for Energy Harvesting IoT Devices

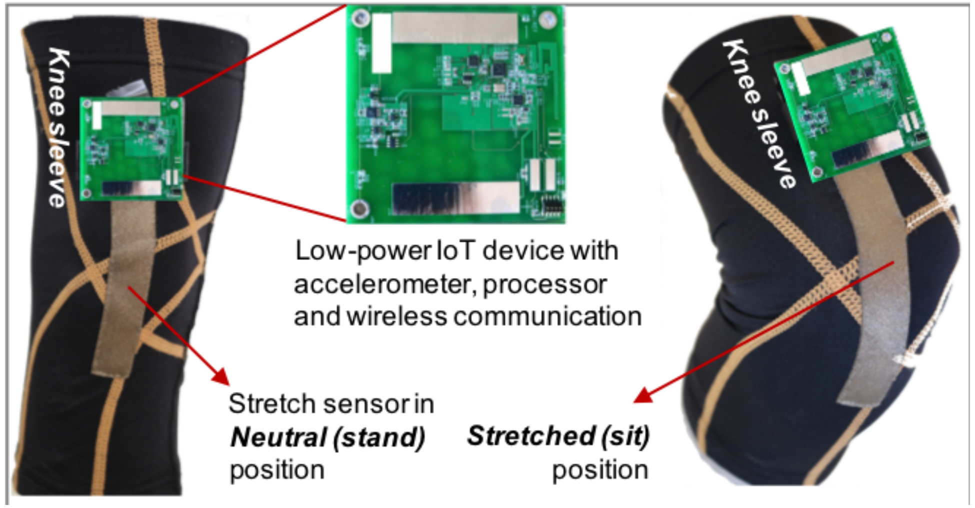
Ganapati Bhat, Kunal Bagewadi, Hyung Gyu Lee, Umit Y. Ogras

Arizona State University

56 Design Automation Conference
FROM CHIPS TO SYSTEMS – LEARN TODAY, CREATE TOMORROW

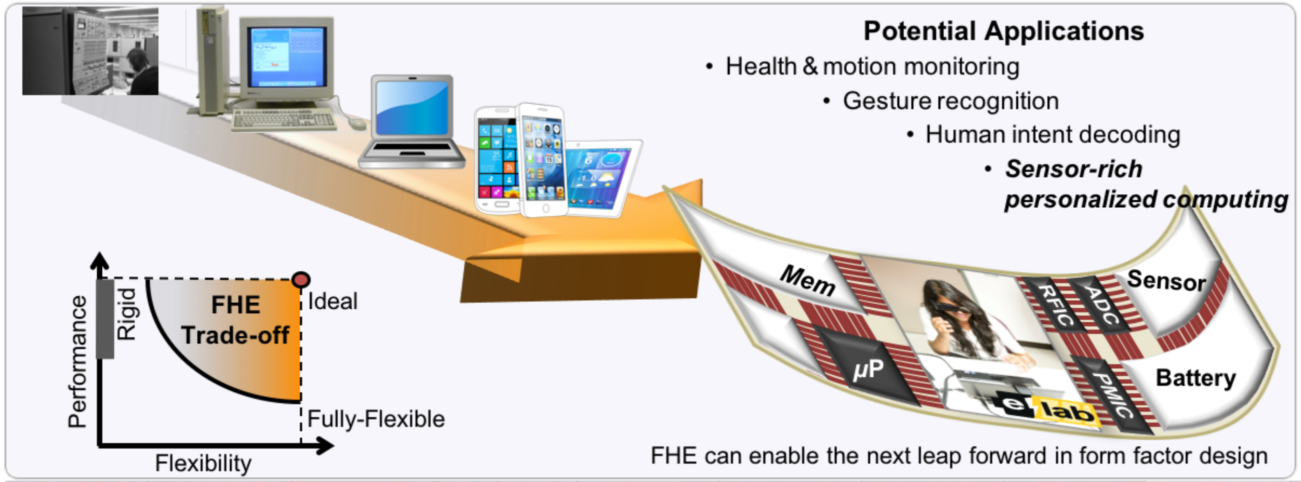
Outline

- **Motivation and Overview**
- **Runtime Energy-Accuracy Optimization**
  - Optimization Problem
  - Runtime Algorithm
- **Human Activity Recognition Case Study**
  - Baseline Implementation
  - Pareto-Optimal Design Points
- **Experimental Results**
- **Conclusions**

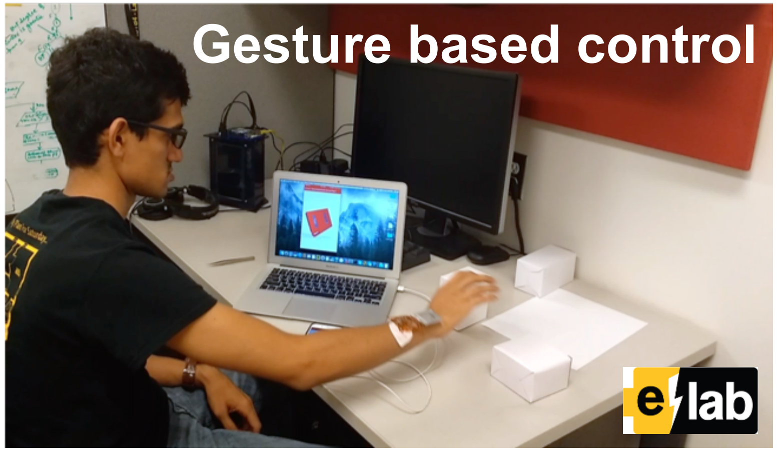


Wearable IoT Devices: Becoming Ubiquitous

- **Wearable Internet of Things devices are popular**
  - Low cost
  - Small form factor
- **Enabled by advances in low power sensors, processors, communications**

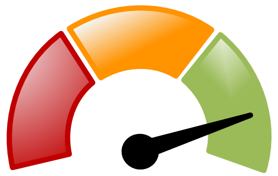


Activity Tracking



Wearable IoT Devices: Requirements

- **Conflicting objectives**
  - Maximize active time to enable continuous monitoring
  - Provide high accuracy and quality of services
- **Constraints due to wearability**
  - Small form factor limits the battery capacity
  - Bulky batteries are inflexible,
    while flexible batteries have low capacity
- **Critical need for**
  - Optimizing runtime **energy-accuracy trade-off**
  - Optimally scale operation of the device as a function of the energy budget

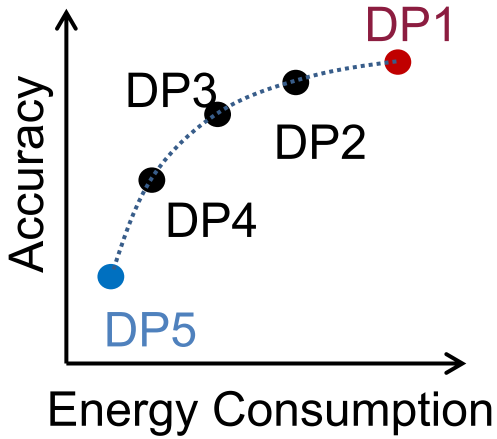
Active Time

Always On

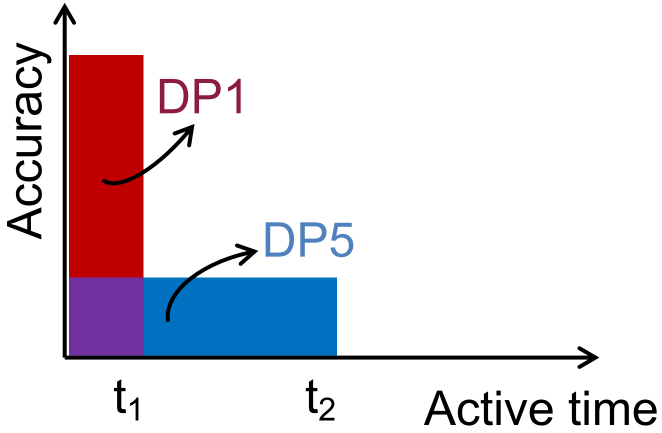
High Accuracy



Runtime Management of IoT Devices

- **Analogy: Dynamic power management techniques**
  - Switch between available power states for power-performance optimization
  - High-performance states for heavy workloads
  - Low-performance states to save power
- **Similarly, multiple design points can be used in IoT devices**
  - Multiple design points utilize the energy-accuracy tradeoff
  - Higher energy design points provide a higher accuracy
  - Sacrifice accuracy to conserve energy
- **Key Challenges**
  - Characterizing accuracy
  - Multiple design points to switch between
  - Requires use studies





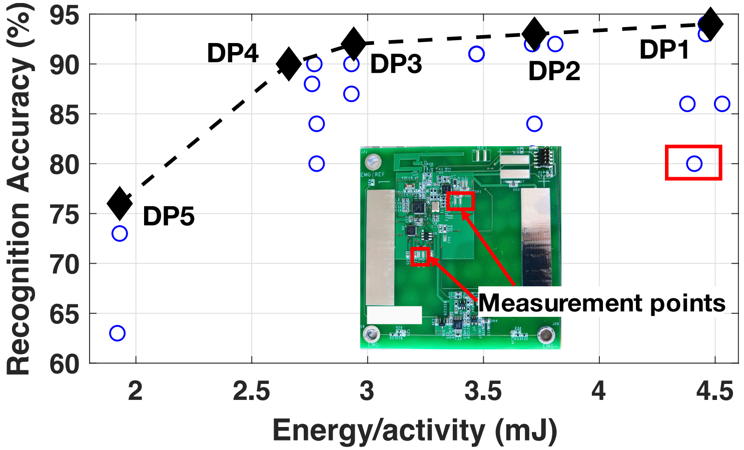
REAP Framework

- **Co-optimize accuracy and active time under tight energy budget**

User studies for HAR

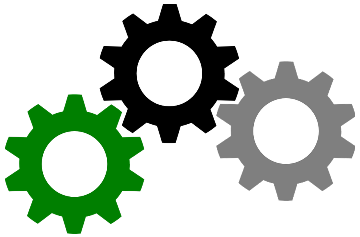
14 users, 6 activities
3553 activity windows

Pareto-optimal Design Points



24 static designs,
5 Pareto-optimal designs

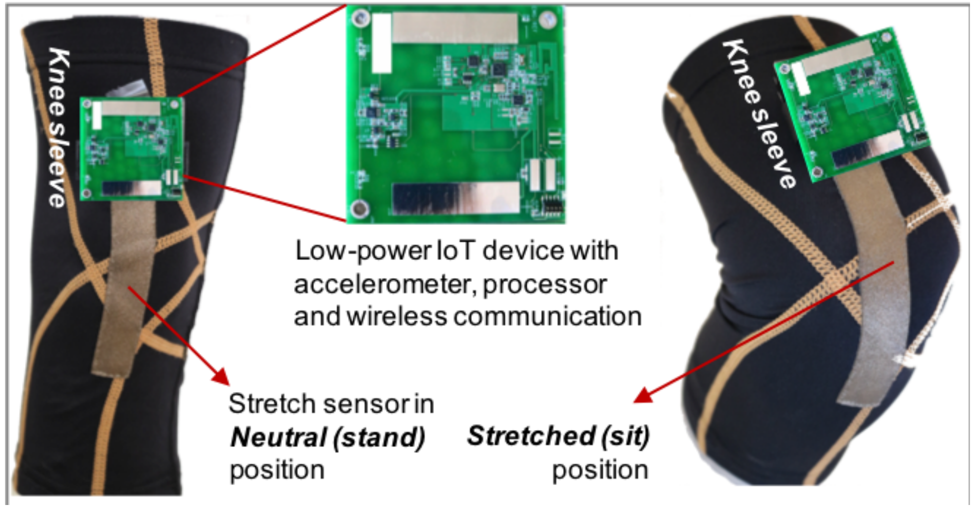
Runtime Optimization Algorithm



Simplex runtime algorithm with 1.5 ms runtime

Outline

- Motivation and Overview
- **Runtime Energy-Accuracy Optimization**
  - Optimization Problem
  - Runtime Algorithm
- **Human Activity Recognition Case Study**
  - Baseline Implementation
  - Pareto-Optimal Design Points
- Experimental Results
- Conclusions



Optimization Problem Formulation

- **Goal:**
  - Determine the optimal active time t_i of each *DP* in activity period T_P
  - Solve at runtime at beginning of each activity period
  - Operation is constrained by the energy budget E_b
- **Can be formulated as:**

$$\text{maximize} \qquad J(t) = \frac{1}{T_P} \sum_{i=1}^{N} a_i^{\alpha}\, t_i$$

$$\text{subject to} \qquad t_{off} + \sum_{i=1}^{N} t_i = T_P$$

$$P_{off} t_{off} + \sum_{i=1}^{N} P_i t_i \le E_b$$

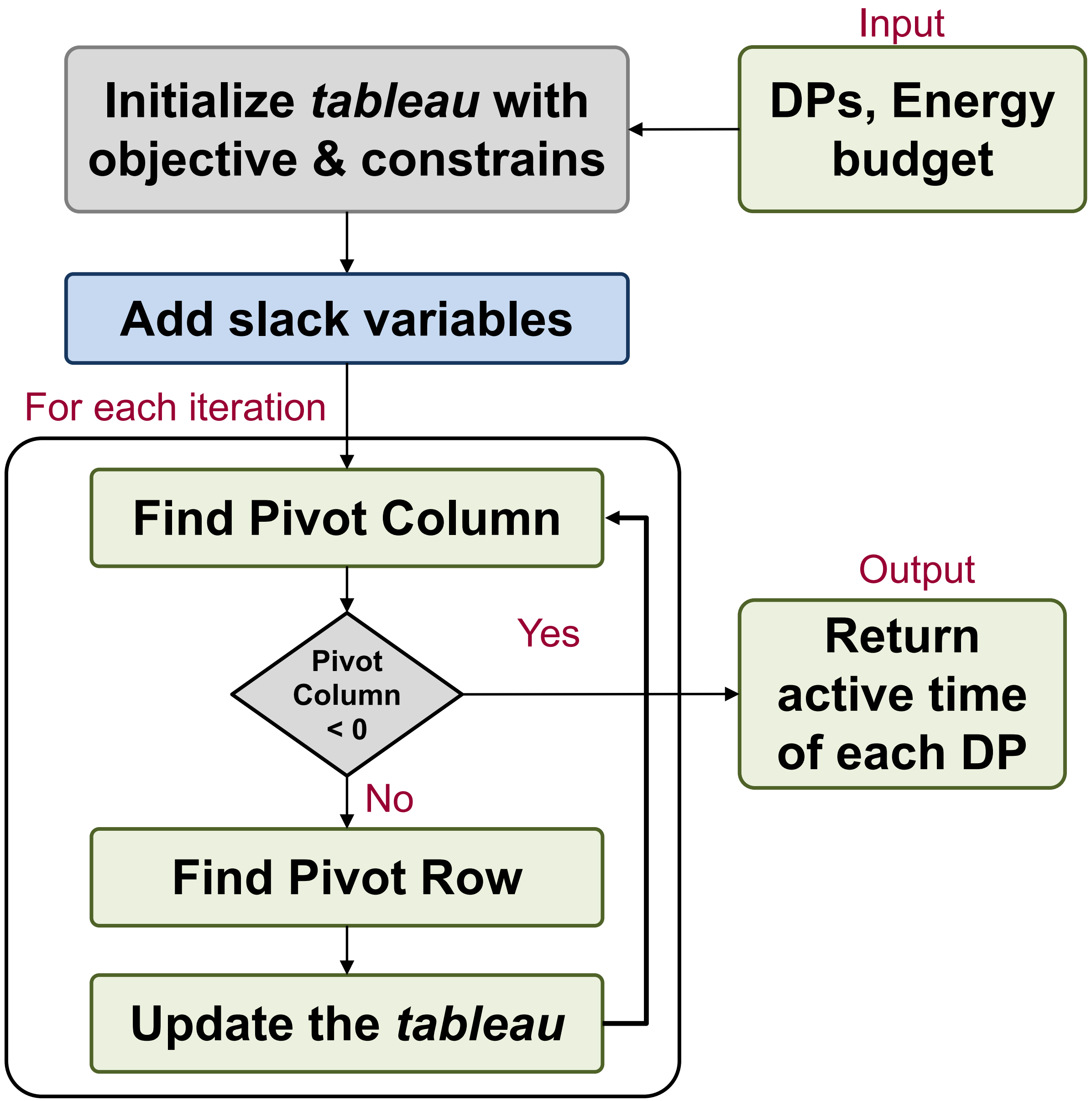
$$t_i \ge 0, 0 \le i \le N$$

- a_i is accuracy of each *DP*
- Parameter $\boldsymbol{\alpha}$ controls accuracy-active time tradeoff
- P_i is power consumption of DP_i
- P_{off} is power consumption when off

We solve this problem at runtime

Runtime Optimization Algorithm

- **Runtime algorithm solves the problem and outputs active time of each DP**
- **Executed for each activity period**
- **Pareto-optimal design points and Energy budget are inputs**
- **Complexity**
  - Polynomial for typical inputs
  - Less than five iterations for solution
  - Takes 1.5 ms for five design points



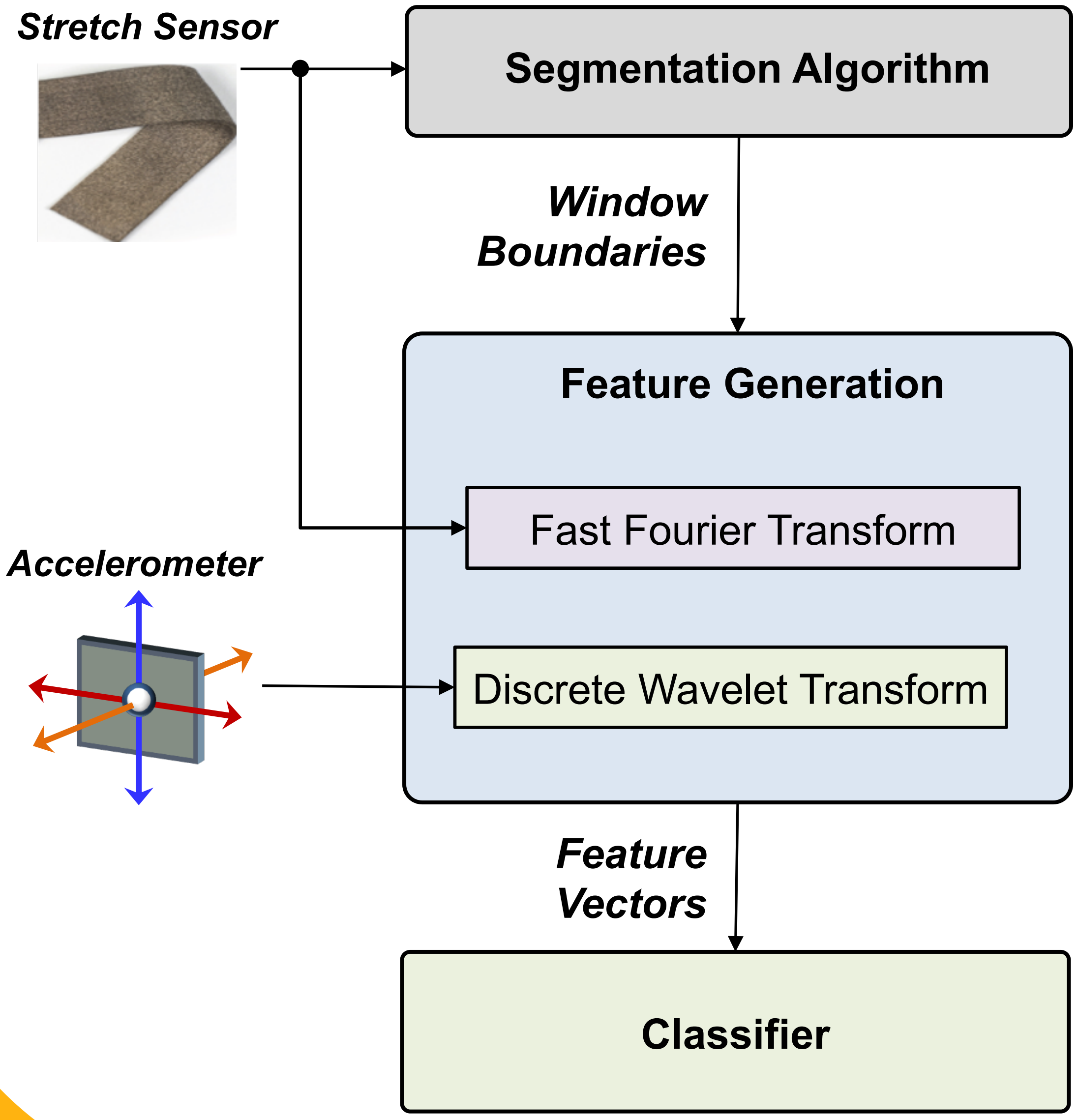
Why Human Activity Recognition (HAR)?

- HAR identifies activities, such as walking, sitting, driving, jogging
- It is the first step to solutions for movement disorders

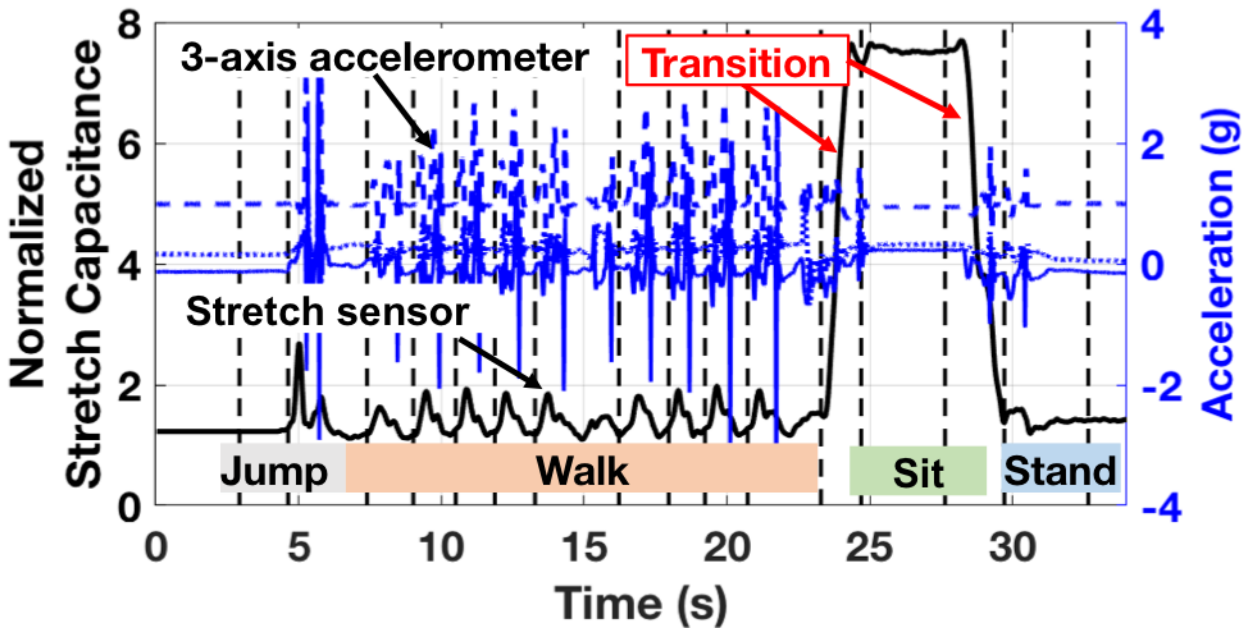
We have to know what the patient is doing to reach a conclusion

- HAR can provide valuable insight to health specialists
- Applications of HAR
  - Patient rehabilitation
  - Fall detection
  - Physical activity promotion

Baseline HAR Implementation



- **Segmentation**
  - Streaming stretch sensor data is processed to generate variable length segments



- **Classifier Design**
  - Offline training of neural network using labeled segments

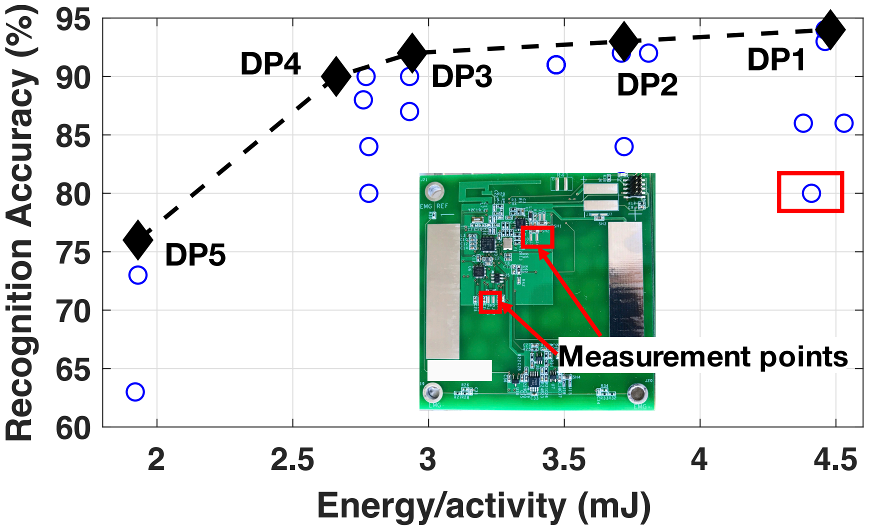
HAR Design Points

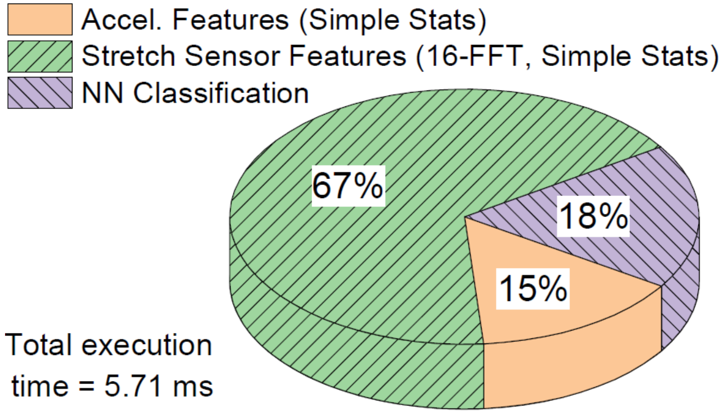
- **Energy and accuracy are functions of**
  - Sensors used
  - Active time of sensors
  - Type of features
  - Classifier complexity

Sensors		Computation		
Accel. axes	Stretch	Sensing period (%)	Signal features	NN structure
X, Y, Z		100	DWT of accel.	4×12×7
X, Y	Yes	75	16-FFT of stretch	4×8×7
X or Y	No	50	Statistics of accel.	4×7
None		40	Statistics of stretch	

Accuracy
Energy

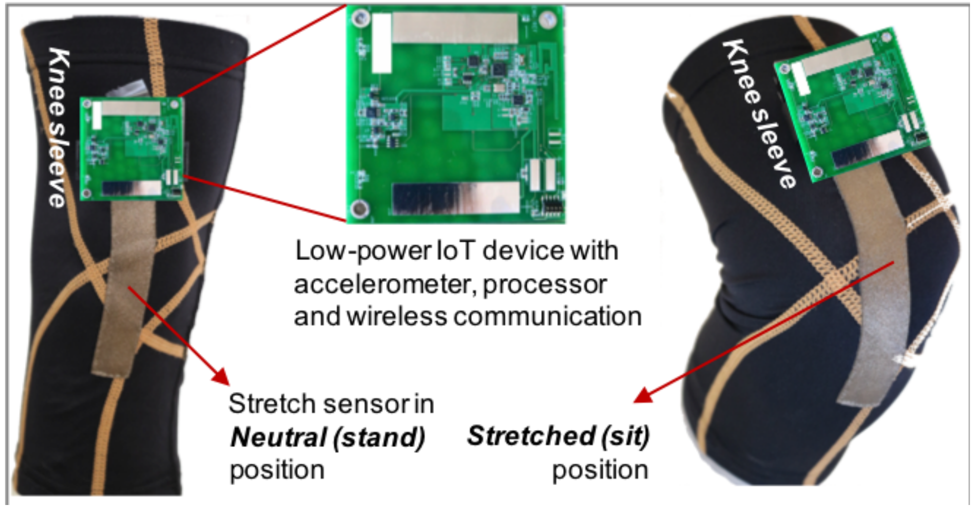
- **Use the trade-off to**
  - Design 24 design points
  - Train NN for each design
  - Characterize energy & accuracy
  - Obtain 5 Pareto-optimal designs





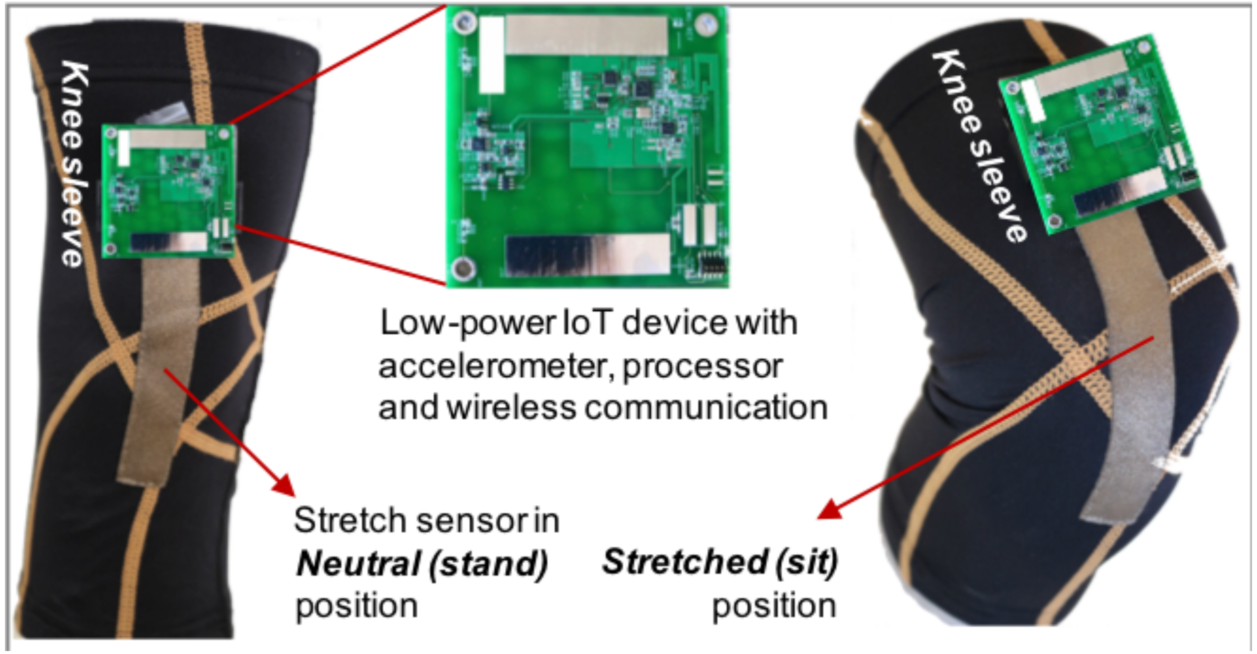
Outline

- Motivation and Overview
- Runtime Energy-Accuracy Optimization
  - Optimization Problem
  - Runtime Algorithm
- Human Activity Recognition Case Study
  - Baseline Implementation
  - Pareto-Optimal Design Points
- **Experimental Results**
- **Conclusions**



Experimental Setup

- **Wearable Device**
  - TI CC2650 MCU, InvenSense MPU
  - Stretchsense Stretch Sensor
  - MPU is sampled at 250 Hz
  - Stretch sensor at 100 Hz
- **Device Placement**
  - MPU is placed at the ankle
  - Stretch sensor is placed at the knee
- **User studies**
  - Data from 14 users
  - 3553 activity windows



Our user data is available to public at OpenHealth page

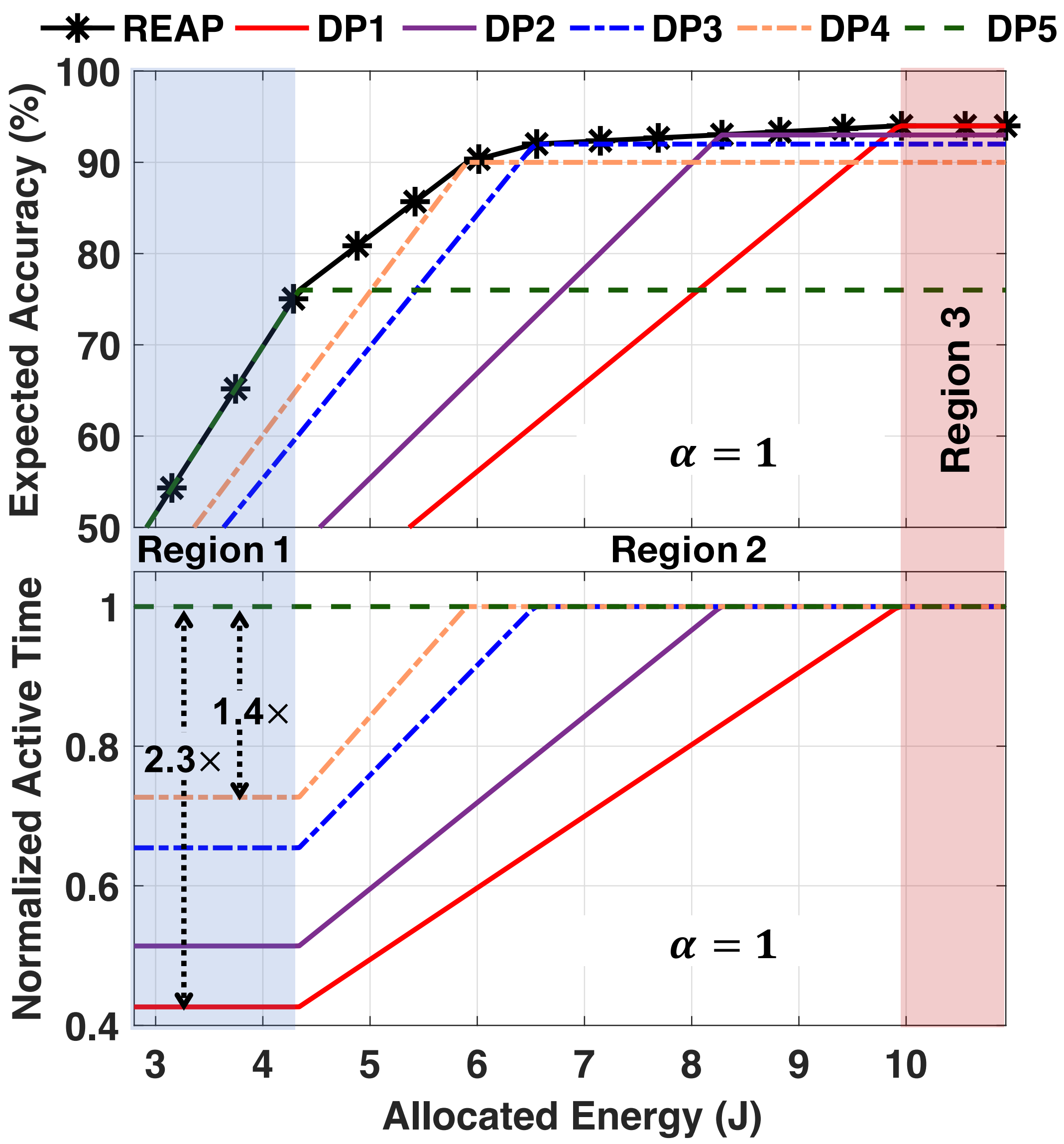
Pareto-Optimal Design Points

- **We choose five Pareto-optimal design points from 24 designs**

Design point description			MCU exec. time distribution (ms)				Per activity summary			
DP no.	Features	Accuracy (%)	Accel. Features	Stretch Features	NN Classifier	Total	MCU energy (mJ)	Sensor energy (mJ)	Energy (mJ)	Power (mW)
1	Statistical acceleration, 16-FFT stretch	94	0.83	3.83	1.05	5.71	2.38	2.10	4.48	2.76
2	Statistical y-axis accel, 16-FFT stretch	93	0.27	3.83	1.00	5.10	2.29	1.43	3.72	2.30
3	Statistical x- and y-axis accel (0.8 s), 16-FFT stretch	92	0.27	3.83	0.90	5.00	2.10	0.84	2.94	1.82
4	Statistical y-axis accel (0.6 s), 16-FFT stretch	90	0.14	3.83	1.00	4.97	2.09	0.57	2.66	1.64
5	16-FFT stretch	76	0.00	3.83	0.88	4.71	1.85	0.08	1.93	1.20

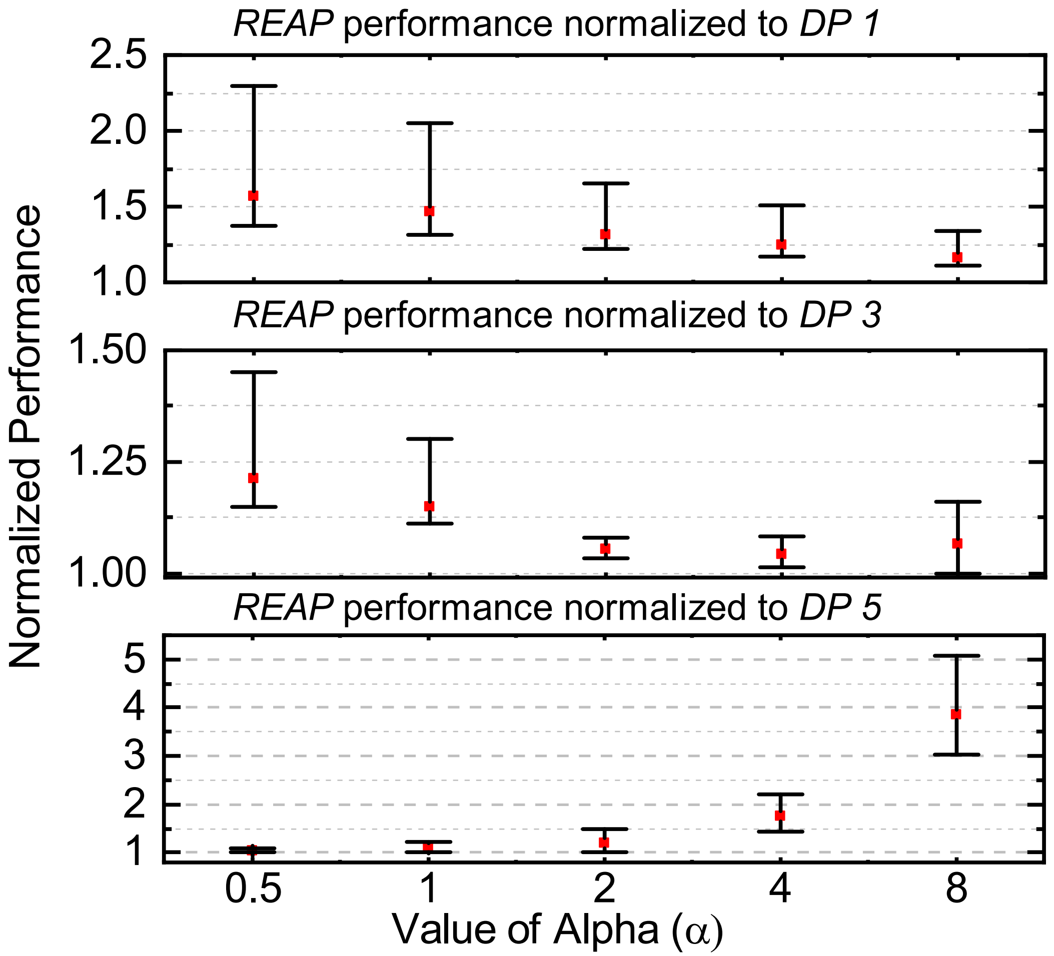
Accuracy and Active Time Analysis

- **Sweep the available energy budget**
- **Solve the problem for each budget**
- ***REAP* outperforms static DPs**
  - Utilizes multiple design points
- **2.3 times improvement in active time compared to DP1**



Case Study using Solar Energy Data

- **Solar energy data from NREL**
  - One month energy harvesting data in Golden, Colorado
- **Evaluate *REAP* with α values from 0.5 to 8**
  - Lower alpha prioritizes active time
  - Higher alpha prioritizes higher accuracy DP
- ***REAP* can adapt to changing α to choose appropriate DP**



Conclusions

- **Energy harvesting IoT devices offer great potential to enable interesting applications**
  – Health monitoring, activity tracking, gesture-based control
- **Presented a energy-accuracy optimization framework**
  – Designed Pareto-optimal design points for HAR
  – Runtime algorithm to choose active time of design points
  – Evaluations using 14 user studies
  – 46% higher expected accuracy and 66% longer active time compared to the highest performance design point
- **Data sets and source code will be made public**