# Low-Complexity Compressed Analysis in Eigenspace with Limited Labeled Data for Real-Time Electrocardiography Telemonitoring

***Kai-Chieh Hsu***, *Bo-Hong Cho,*
*Ching-Yao Chou and An-Yeu (Andy) Wu*

National Taiwan University

# ECG Telemonitoring with Edge Computing

- Mobile Telemedicine with Wireless Body Area Network (WBAN) [1]
  - Patient-centered health-care
  - Ubiquitous health-care
- ECG Telemonitoring [2], [3]
  - Record the electrical activity of the heart
  - Standard practice in hospitals for diagnoses
- Edge Computing [4]
  - bandwidth cost saving
  - battery life constraint
  - latency requirement

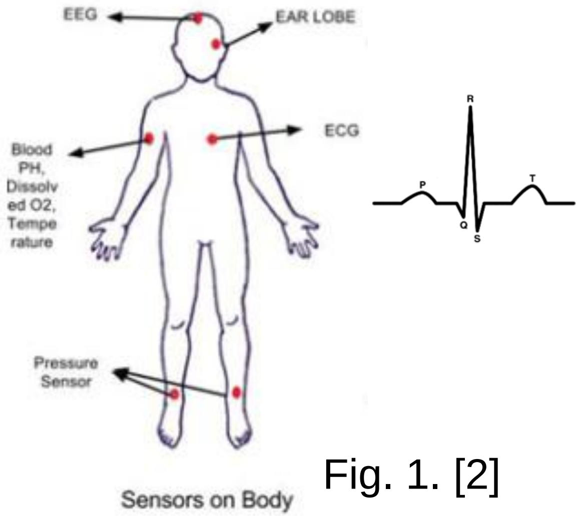



Fig. 1. [2]

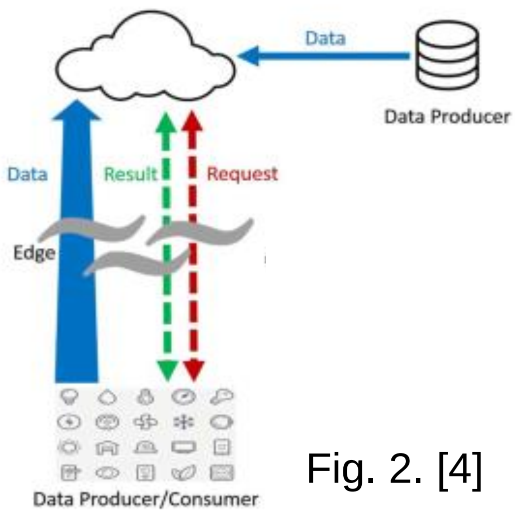
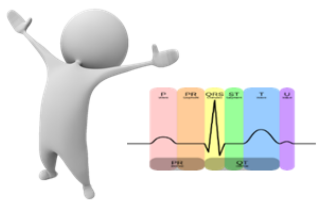



Fig. 2. [4]

# Edge Computing under Existing IoT Systems

Wireless Body Area Network



**Massive Sampled Data**

Transmit

Existing Module


Receive

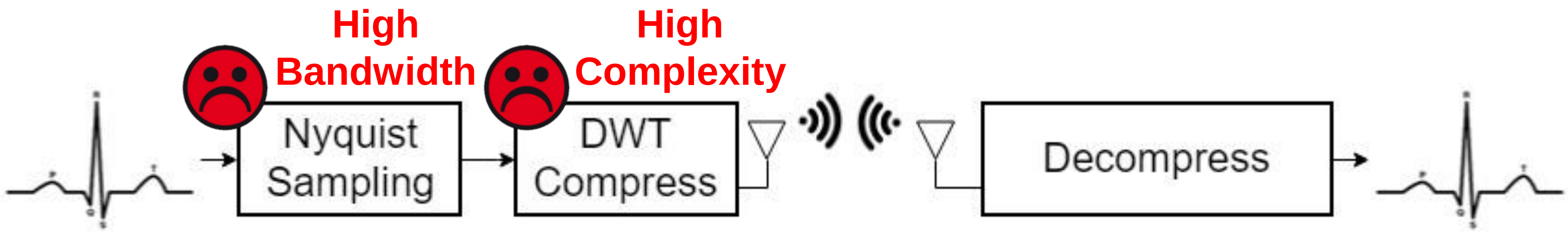
Machine Learning

**Massive Sampled Data**

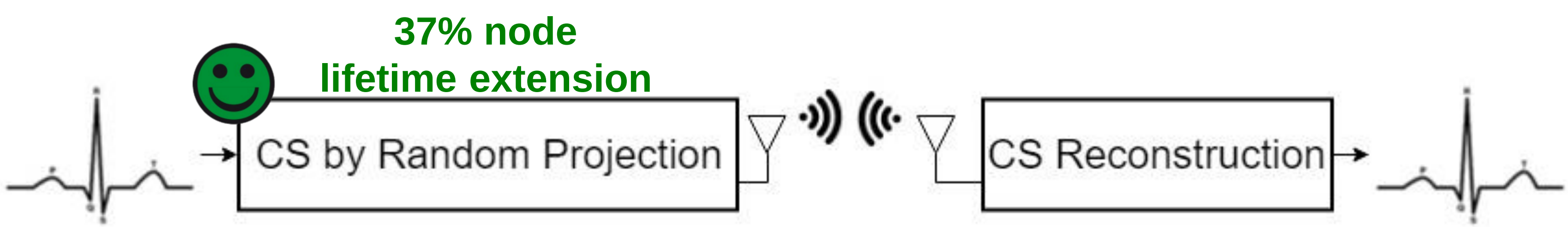
Edge-Devices

# Compressed Sensing for ECG Telemonitoring

- Problems of Digital Wavelet Transform (DWT)
  - High bandwidth incompatible to ADC (Nyquist sample rate)
  - High Computational Complexity (Compression)



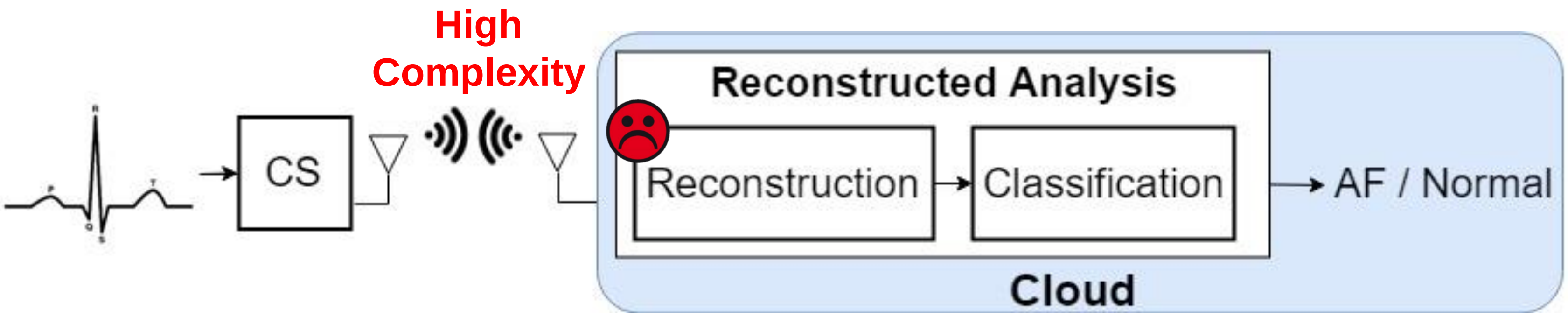

- Compressed sensing (CS) combines sampling and compressing
  - Reduce cost and latency in sampling
  - CS-based sensors achieves a **37%** node lifetime extension [2]



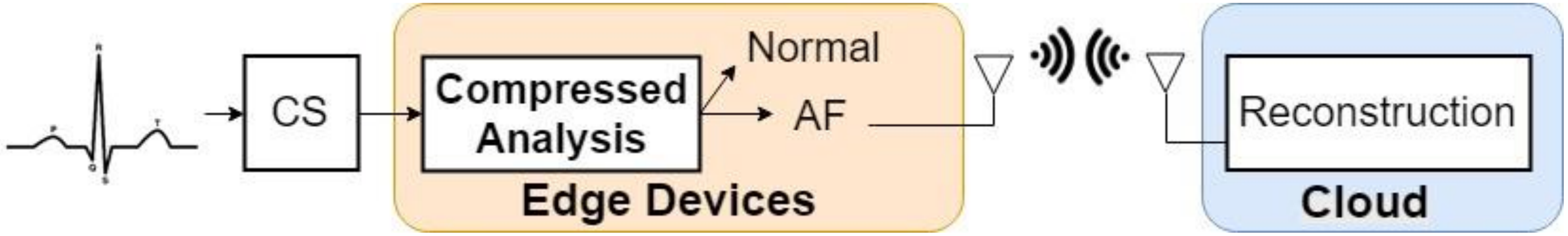
# Compressed Analysis for ECG Telemonitoring

- Reconstructed Analysis (RA)
  - High computational complexity because of CS reconstruction algorithms
  - Inappropriate at edge devices.

AF: Atrial Fibrillation

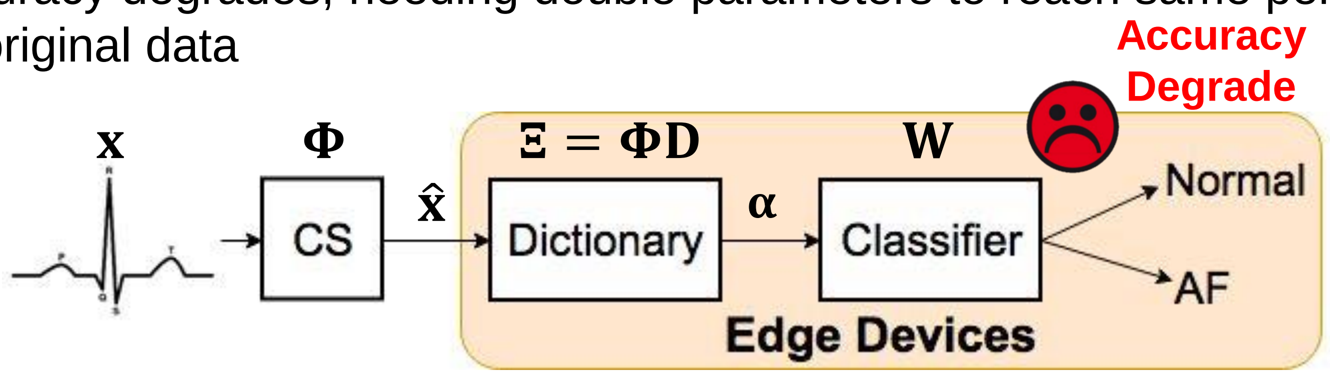



- Compressed Analysis (CA)
  - Reduce power (classification on compressed signals), suitable at edge devices
  - Reduce the bandwidth requirement (only transmitting AF signals)



# Naïve CA (CA-N)

- Combining CS with Task-Driven Dictionary Learning (TDDL)
  - What is TDDL [5]
    - Learning a dictionary ($\mathbf{D}$) to provide predictive sparse coding ($\boldsymbol{\alpha}$) at given data set
    - Learning a classifier ($\mathbf{W}$) to classify by the sparse coding $\boldsymbol{\alpha}$
  - Why we choose TDDL?
    - Low Complexity → Overcome battery constraint and bandwidth scarcity
    - High Generalization → Limited label of ECG dataset
- The on-line inference mode of CA-N
  - $\mathbf{D}$ and $\mathbf{W}$ learned on original data ($\mathbf{X}$)
  - Accuracy degrades, needing double parameters to reach same performance on original data



# Contribution of Proposed Scheme (1/2)

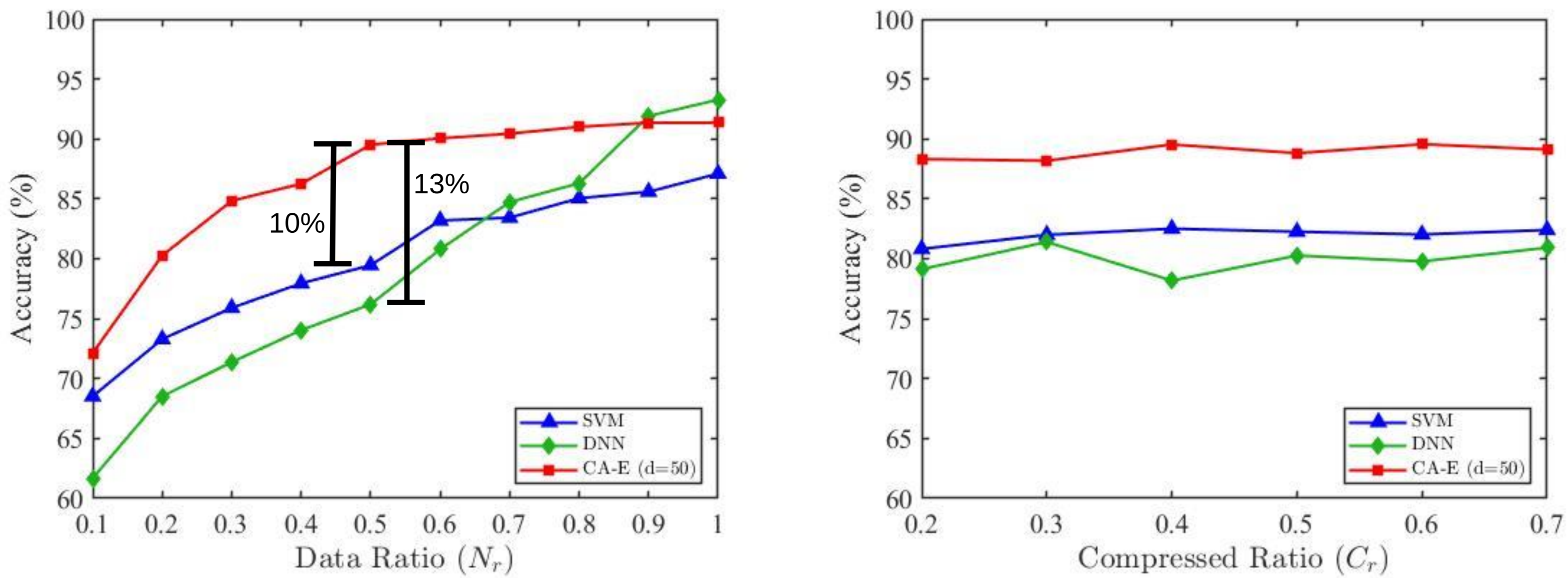
- Low-Complexity (overcame battery and bandwidth requirement)
  - Our proposed Eigenspace-aided Compressed Analysis (CA-E) vs Naïve Compressed Analysis (CA-N)

| Model | # Parameters | Training Time (s) | Inference Time (ms) | Accuracy (%) |
|---|---|---|---|---|
| **CA-N** | 13k | 452.56 | 26.94 | $89.24 \pm 0.520$ |
| **CA-E (Our proposed)** | 4.25k | 107.15 | 3.50 | $90.05 \pm 0.256$ |

  - Reduce about **67%** parameters (Memory↓)
  - Reduce about **87%** inference time (Power↓)
  - Reduce about **76%** training time (Power↓)

# Contribution of Proposed Scheme (2/2)

- High-Stability
  - CA-E outperforms DNN and SVM by over 10% when the amount of data is halved. (Overcame limited label of ECG dataset)
  - CA-E reaches about 90% under all compressed ratio (Stable under all compressed ratio)



# Eigenspace-Aided CA (Training)

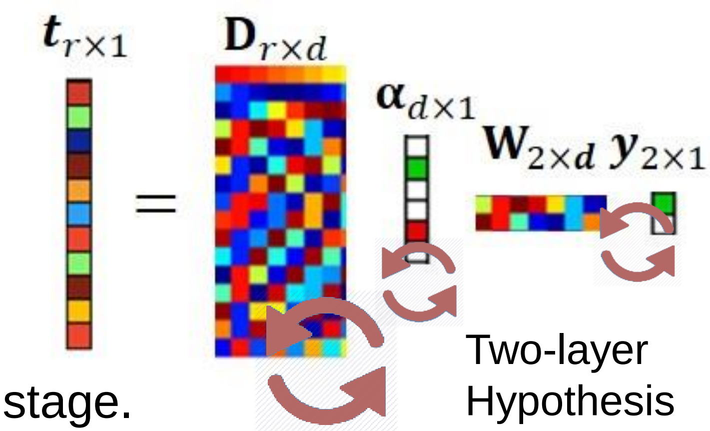
- Principal Component Analysis (PCA)
  - Record mean vector ($\boldsymbol{\mu}$) of dataset ($\mathbf{X}$)
  - Learn eigenspace ($\boldsymbol{\Psi} \in \mathbb{R}^{N\times r}$) of $\mathbf{X}$
  - Transpose to eigenspace by $\mathbf{T} = \boldsymbol{\Psi}^{\mathbf{T}}(\mathbf{X} - \boldsymbol{\mu})$
- TDDL to learn $\mathbf{D}$ and $\mathbf{W}$ on $\mathbf{T}$
  - Stage I. Initialize
    - Dictionary: online dictionary learning (ODL) [6]
    - Weight: square / logistic loss
  - Stage II. **Co-optimize $\mathbf{D}$ and $\mathbf{W}$ with labels**
    - Alternates between $\mathbf{A}$ and $\mathbf{D}, \mathbf{W}$
    - Update dictionary with back propagation rule
  - Sparse coding plays an important role in both stage.

$$\boldsymbol{\alpha}_{\mathbf{D}} \triangleq \operatorname*{argmin}_{\alpha\in\mathbb{R}^d} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

$$\min_{\alpha,\mathrm{D}} \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}\|\mathbf{x}_\mathbf{i} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

$$\min_{\mathbf{W}} \frac{1}{m}\sum_{i=1}^{m} l_s(\mathbf{y}_i, \mathbf{W})$$

$$\min_{\mathbf{W},\mathbf{D}} \frac{1}{m}\sum_{i=1}^{m} l_s(\mathbf{y}_i, \mathbf{W}, \boldsymbol{\alpha}_{\mathbf{D}}) + \frac{\gamma}{2}\|\mathbf{W}\|_F^2$$




Two-layer Hypothesis

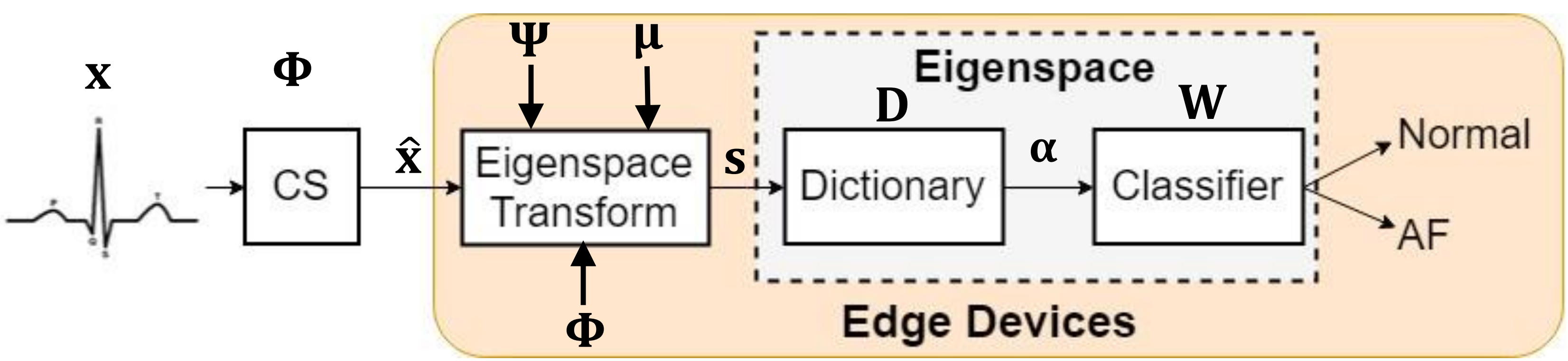
# Eigenspace-Aided CA (Inference)

- Eigenspace Transform
  - Compressed sensing signal is transmitted with known sensing matrix ($\mathbf{\Phi}$), the decoding data is obtained by

$$\mathbf{s} = (\mathbf{\Phi\Psi})^{+}(\hat{\mathbf{x}} - \mathbf{\Phi\mu}) = \mathbf{\Theta}^{+}(\hat{\mathbf{x}} - \mathbf{\Phi\mu})$$

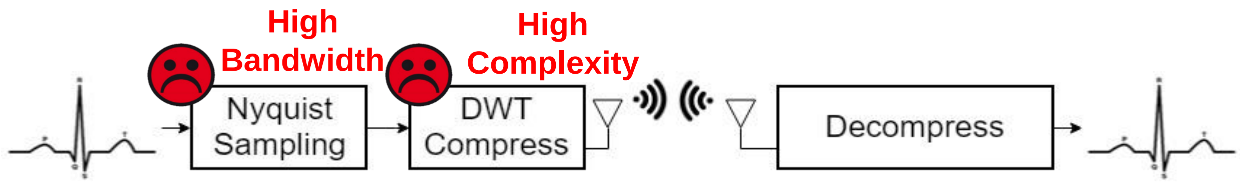
  - $(\quad)^{+}$: pseudo-inverse
- The decoding vector ($\mathbf{s}$) then pass through TDDL-based classifier
  - Get sparse coding $\boldsymbol{\alpha}(\mathbf{s}, \mathbf{D})$
  - Simple linear classifier $\mathbf{W}$

$$\boldsymbol{\alpha}_{\mathbf{D}} \triangleq \underset{\alpha \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$$

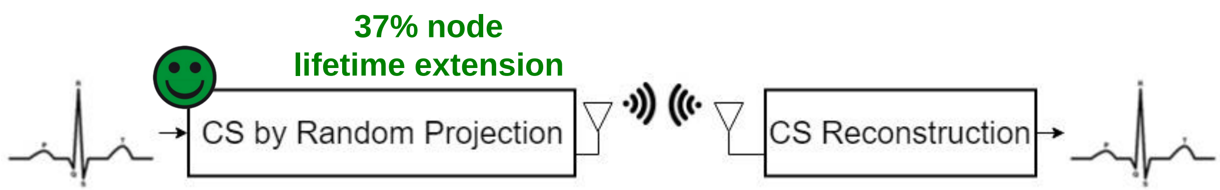


# Scheme Development (1/2)

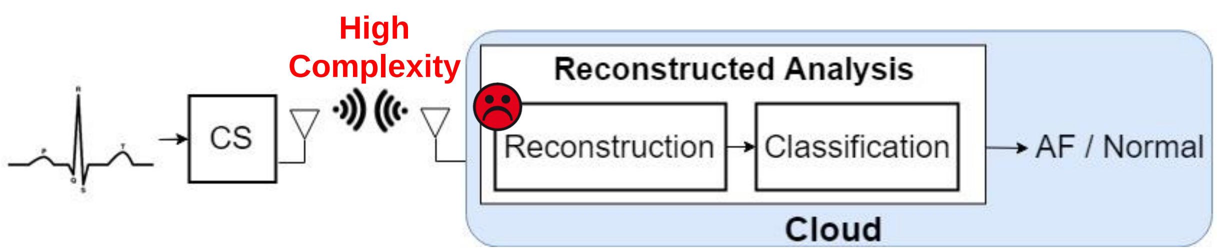
- Sensor
  - DWT-Based



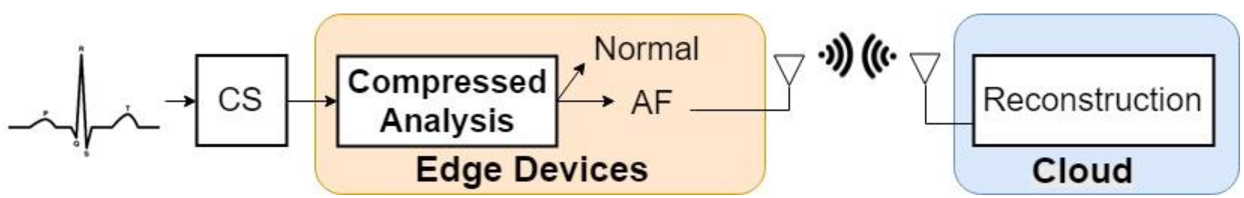

  - CS-Based




- Analysis
  - Reconstrcted (RA)

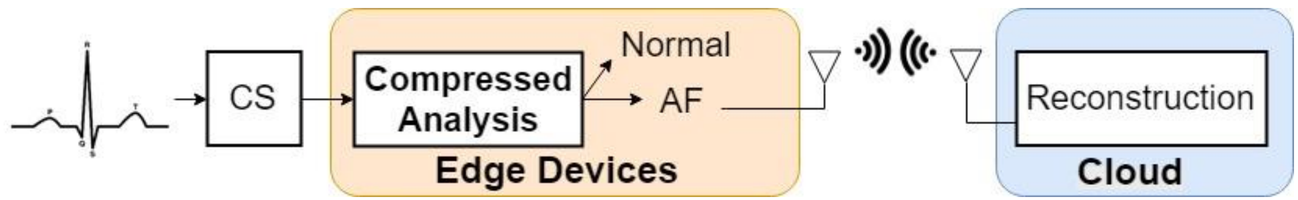



  - Compressed (CA)

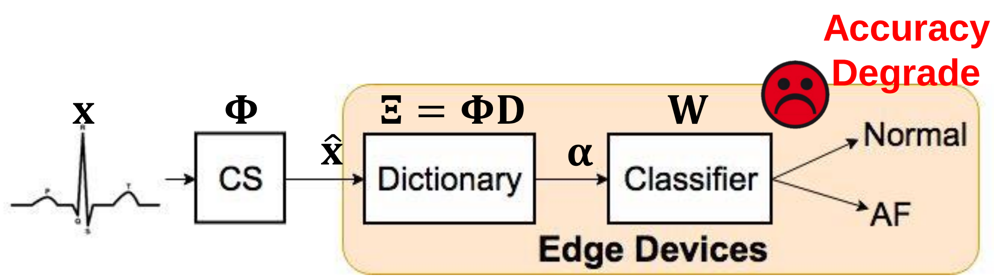


# Scheme Development (2/2)

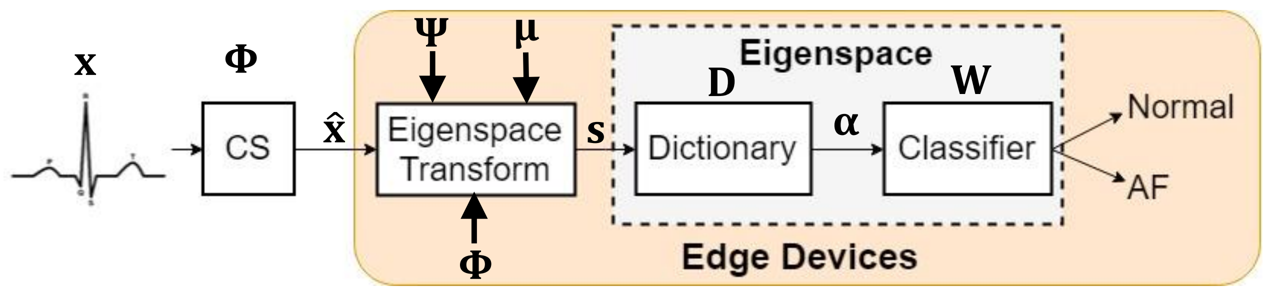
- Compressed Analysis
  - Prototype




  - Naïve

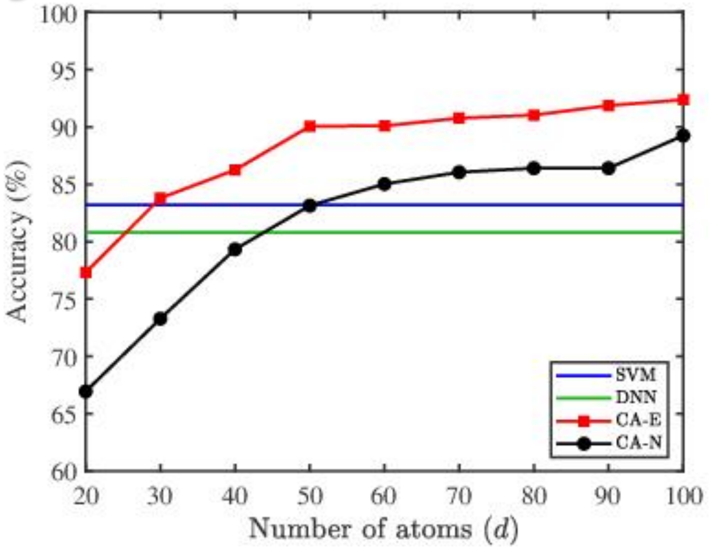



  - Eigenspace-aided



# Simulation Results (1/3)
# Different Dictionary Size

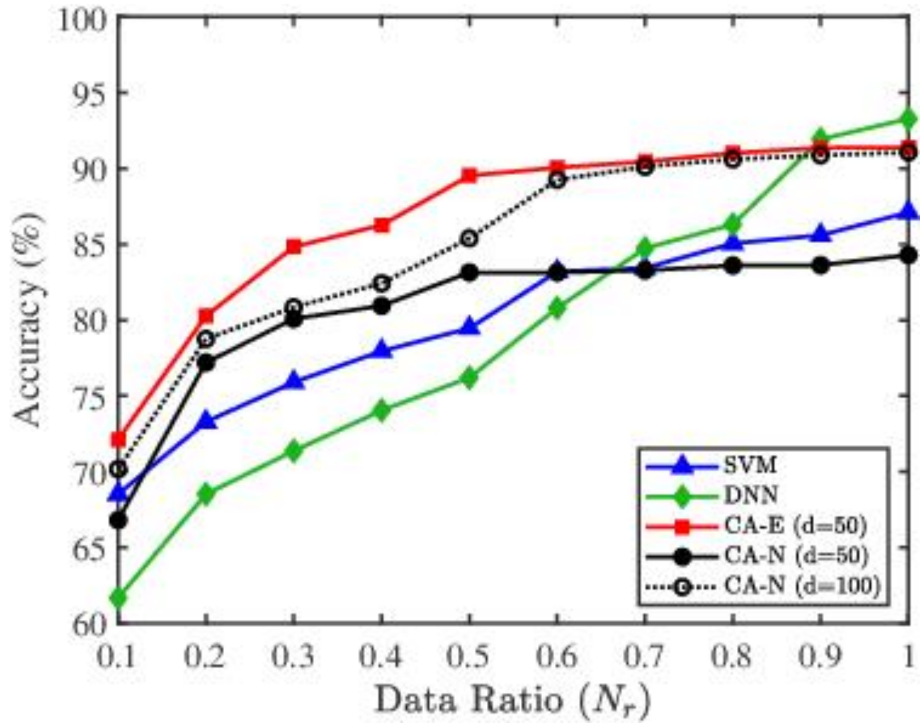
- Accuracy vs Dictionary Size
  - To surpass DNN & SVM (~85%), CA-E needs **30 atoms**, but CA-N needs 60 atoms.
  - Under same number of atoms, CA-E outperforms CA-N by about 7%.
- CA-E-50 *vs.* CA-N-100
  - Reduce about **67%** parameters (Memory↓)
  - Reduce about **87%** inference time (Power↓)
  - Reduce about **76%** training time (Power↓)




| Model | # Parameters | Training Time (s) | Inference Time (ms) | Accuracy (%) |
|---|---|---|---|---|
| **CA-N (100)** | 13k | 452.56 | 26.94 | 89.24 $\pm$ 0.520 |
| **CA-E (50)** | 4.25k | 107.15 | 3.50 | 90.05 $\pm$ 0.256 |

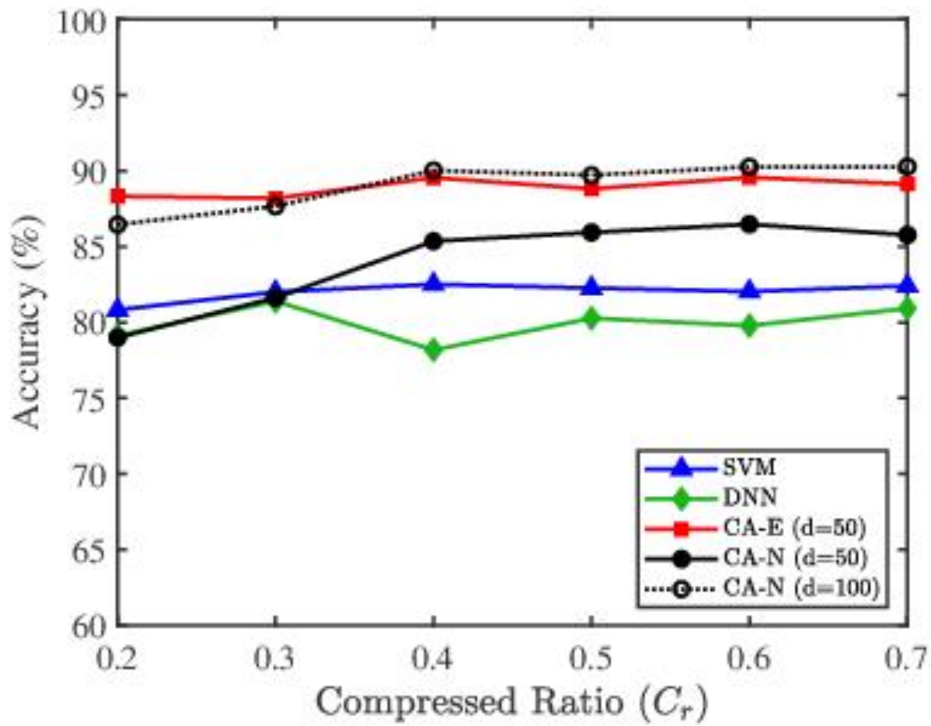
# Simulation Results (2/3)
# Different Data Set Size

- CA-E is **more immune** to limited data challenge (ex. $N_r \leq 0.5$)
  - SVM and DNN dramatically drops below 80%
  - CA still maintain the performance
  - CA-E outperforms CA-N in **7% margin** when the number of atoms is the same.



# Simulation Results (3/3)
# Different Compressed Ratio

- CA-E can achieve about 90% accuracy under all compressed ratios
  - CA-N requires 100 atoms to achieve same level of performance
  - SVM and DNN have only about 80%
- CA-E is robust and address the entailed problems of variation of compress ratio



# Conclusion

- We propose an eigenspace-aided compressed analysis for ECG telemonitoring, using
  - PCA to mitigate the influence of sensing matrix and reduce the dimension
  - TDDL to learn predictive sparse coding at eigenspace.
- The proposed eigenspace-aided compressed analysis achieves
  - Low complexity
  - High generalization
  - High stability of different compressed ratios

# Reference

[1] F. Touati and R. Tabish, “U-healthcare system: state-of-the-art review and challenges,” in *Journal of Medical Systems*, pp.120, May 2013.

[2] H. Mamaghanian, N. Khaled, D. Atienza and P. Vandergheynst, “Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes,” in *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2456-2466, Sep. 2011.

[3] M. Y. Tsai, C. Y. Chou and A. Y. Wu, “Robust compressed analysis using subspace-based dictionary for ECG telemonitoring systems,” in *Proc. IEEE Int. Workshop on Signal Process. Syst.*, Oct. 2017

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” in *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637-646, Oct. 2016.

[5] J. Mairal, F. Bach and J. Ponce, "Task-driven dictionary learning," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791-804, April 2012.

[6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. “Online dictionary learning for sparse coding,“ ICML, 2009.

# Thanks for your attention

# Q&A

# Backup

# Experimental Setting

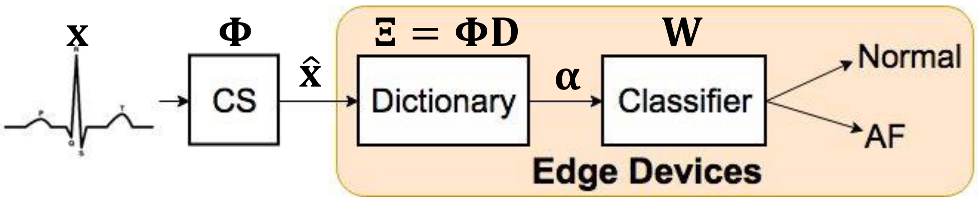
- ECG signals were recorded from the intensive care unit (ICU) of stroke in National Taiwan University Hospital (NTUH)
  - 231 normal records and 58 AF records (labeled by doctors)
  - Sample Frequency: 512 Hz
  - Each record randomly sample 2250 seconds
    - 1250 for training
    - 1000 for testing
- CS setting
  - Entries of sensing matrix: *Bernoulli (0.5)*
- Simulation Environment
  - Measured on Intel i5-4200M CPU @ 2.5 GHz
  - Using Python3

TABLE I: Parameters setting for learning models

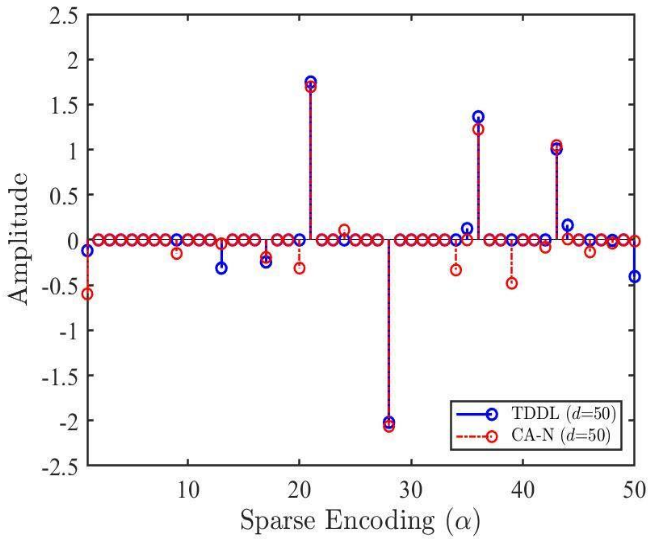
| | |
|---|---|
| CA-E and CA-N | |
| $\ell_1$-Constraint ($\lambda_1$) | [0.2, 0.5, 0.8] |
| Regularization ($\nu$) | [$10^{-5}$, $10^{-4}$] |
| SVM | |
| Kernel | Radial Basis Function |
| Gamma ($\gamma$) | [0.08, 0.10, 0.12, 0.15, 0.2] |
| Cost ($C$) | [500, 800, 1000] |
| DNN | |
| Hidden Layer Dimension | [ (16,32), (32,64), (64,128), (128,256), (8,16,32), (16,32,64), (32,64,128) ] |

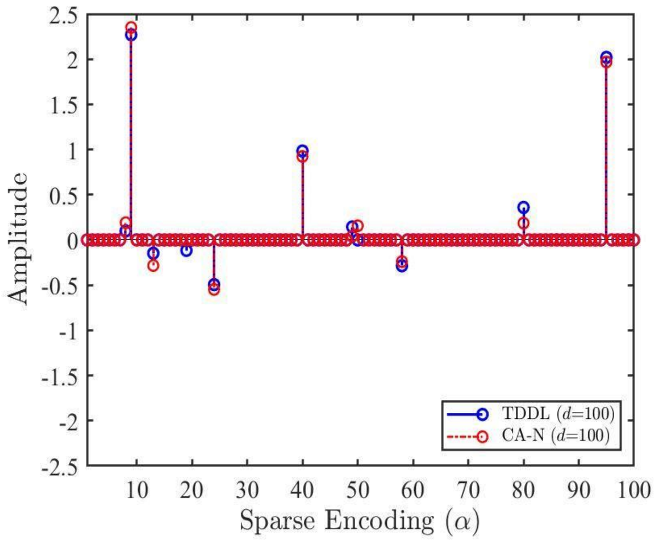
# Analysis of CA-N

- We need to increase the number of atoms in dictionary to compensate the performance degrade




- Figures below also present the sparse codings in original domain as comparison group.



# Comparison of CA-E and CA-N

- CA-E-50 *vs.* CA-N-100
  - Reduce about **67%** parameters
  - Reduce about **76%** training time,
  - Reduce about **87%** inference time
  - Smaller performance variance

**Far smaller classifier with faster training and inference time**

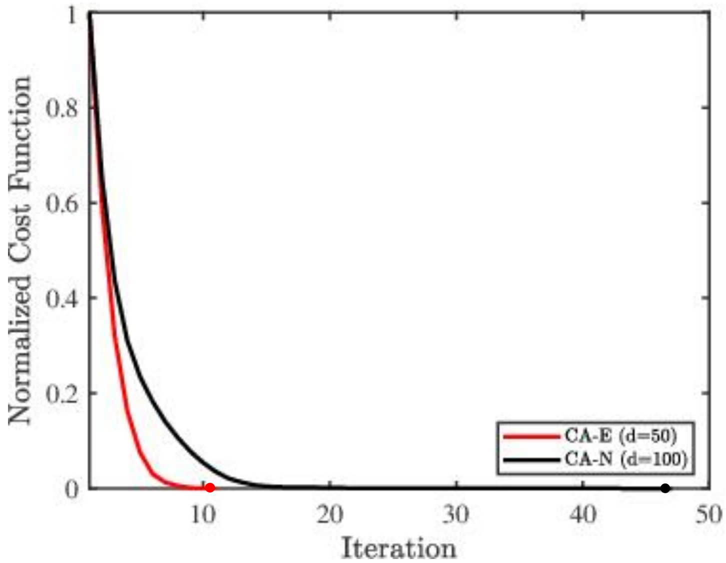
- The bottleneck of training and inference time lies in **FISTA**

| Model | # Parameters | Training Time (s) | | Inference Time (ms) | | | | Accuracy (%) |
|---|---|---|---|---|---|---|---|---|
| | | Total | FISTA | Total | FISTA | # Iter | 1 Iter | |
| **CA-N (d=100)** | $M \times d + d \times N_c$ (13k) | 452.56 | 306.33 | 26.94 | 26.94 | 35.5 | 0.759 | 89.24 $\pm$ 0.520 |
| **CA-E (d=50)** | $r \times d + d \times N_c$ (4.25k) | 107.15 | 61.59 | 3.50 | 3.49 | 15.2 | 0.229 | 90.05 $\pm$ 0.256 |

$M$ = 128, $r$ = 83 and $N_{\mathrm{r}}$ = 0.6

# Detailed Timing Analysis

- The bottleneck in FISTA is $\nabla_\alpha f$ operation
  - $\nabla_\alpha f = \nabla_\alpha \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 = \mathbf{D}^\mathrm{T}\mathbf{D}\boldsymbol{\alpha} - \mathbf{D}^\mathrm{T}\mathbf{x} \rightarrow O(d^2)$
  - Above order matches the following table
- CA-E accelerates FISTA by
  - **Cut off the complexity of each iteration**
  - **Reducing the number of iteration**




| Model | Training Time (s) | | Inference Time (ms) | | | |
|---|---|---|---|---|---|---|
| | Total | FISTA | Total | FISTA | # Iter | 1 Iter |
| CA-N (d=100) | 452.56 | 306.33 | 26.94 | 26.94 | 35.5 | 0.759 |
| CA-E (d=50) | 107.15 | 61.59 | 3.50 | 3.49 | 15.2 | 0.229 |

# Reference

[1] I. Daubechies, M. Defrise, and C. D. Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Commun. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.

[2] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”, *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, 2009.