UCIrvine

Novel data processing techniques for high-performance brain-computer interface

Ph.D. Defense

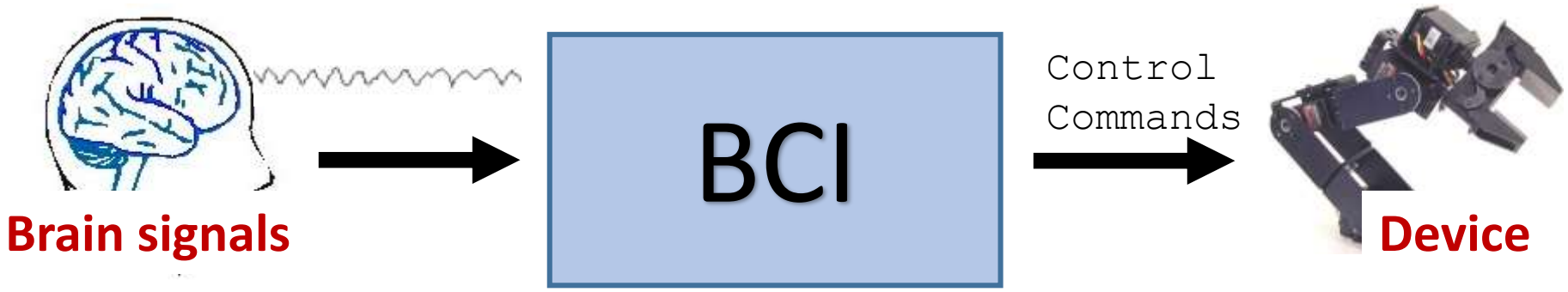
Po T. Wang

Biomedical Engineering

June 6th, 2014

What is a **brain-computer interface (BCI)**?

- BCI is a direct communication pathway between human brain and a computer device.
  - Reads signals from the brain
  - Decodes brain signals into intentions
  - Commands an output device



Overview of this presentation

1. Introduction and background
2. High-performance EEG-based BCI applications
    1. BCI Spelling device
    2. BCI Gait project
3. Conclusions and future works

Introduction & background

What BCI can do

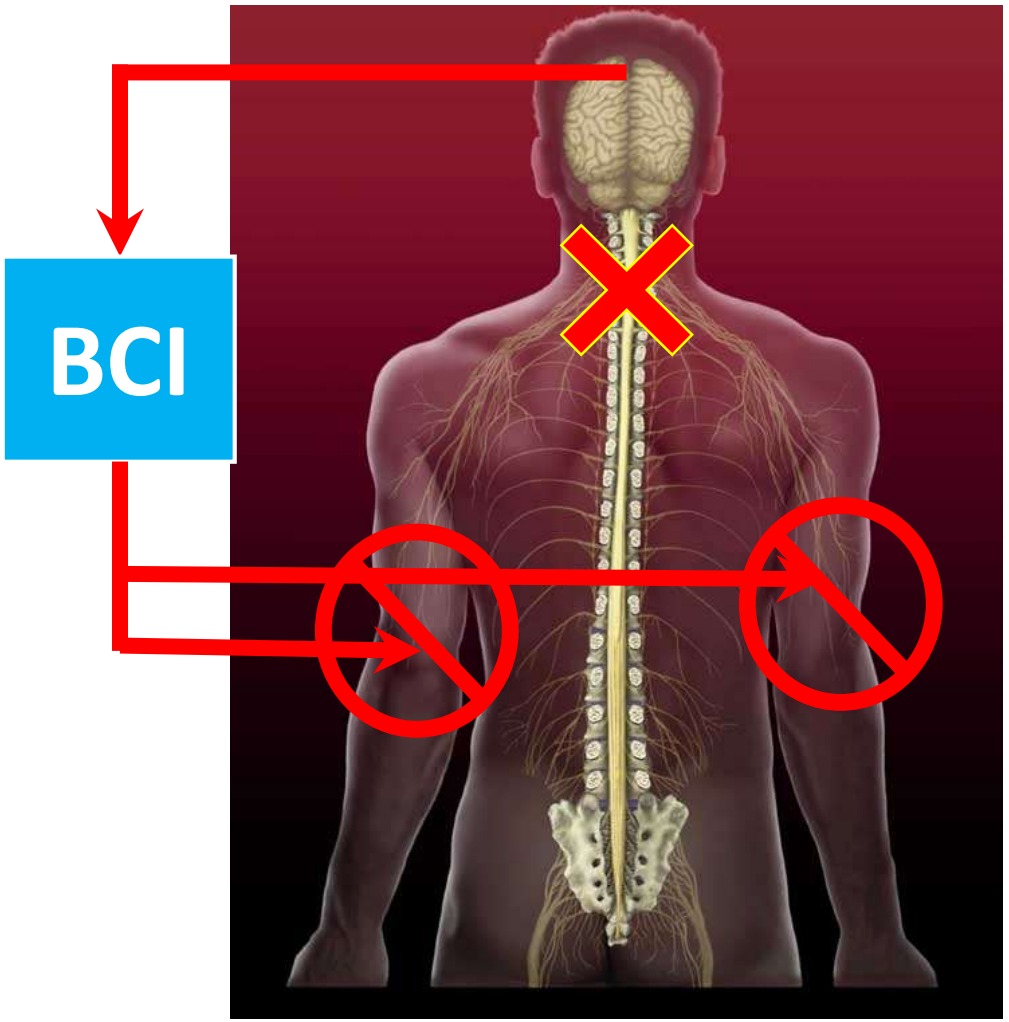


Image from: Cedars-Sanai

Normal people can control their muscles by sending control signals down the spinal cord.

People with injuries to their brain or spinal cord may lose control of their own limbs and body.

A BCI can bypass the damaged nerves and directly control the limbs via a prosthesis or exoskeleton.

Communicate

Typing words

Graz University of Technology, Austria (2004)

Filmed by Muller-Putz. [Pfurtscheller et al (2003). Thought control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia]

60 Minutes, CBS, Oct 31, 2008.

http://www.cbsnews.com/news/harnessing-the-power-of-the-brain/

University of Pittsburgh (2011)

[Wang et al (2013). An Electrocorticographic Brain Interface in an Individual with Tetraplegia]

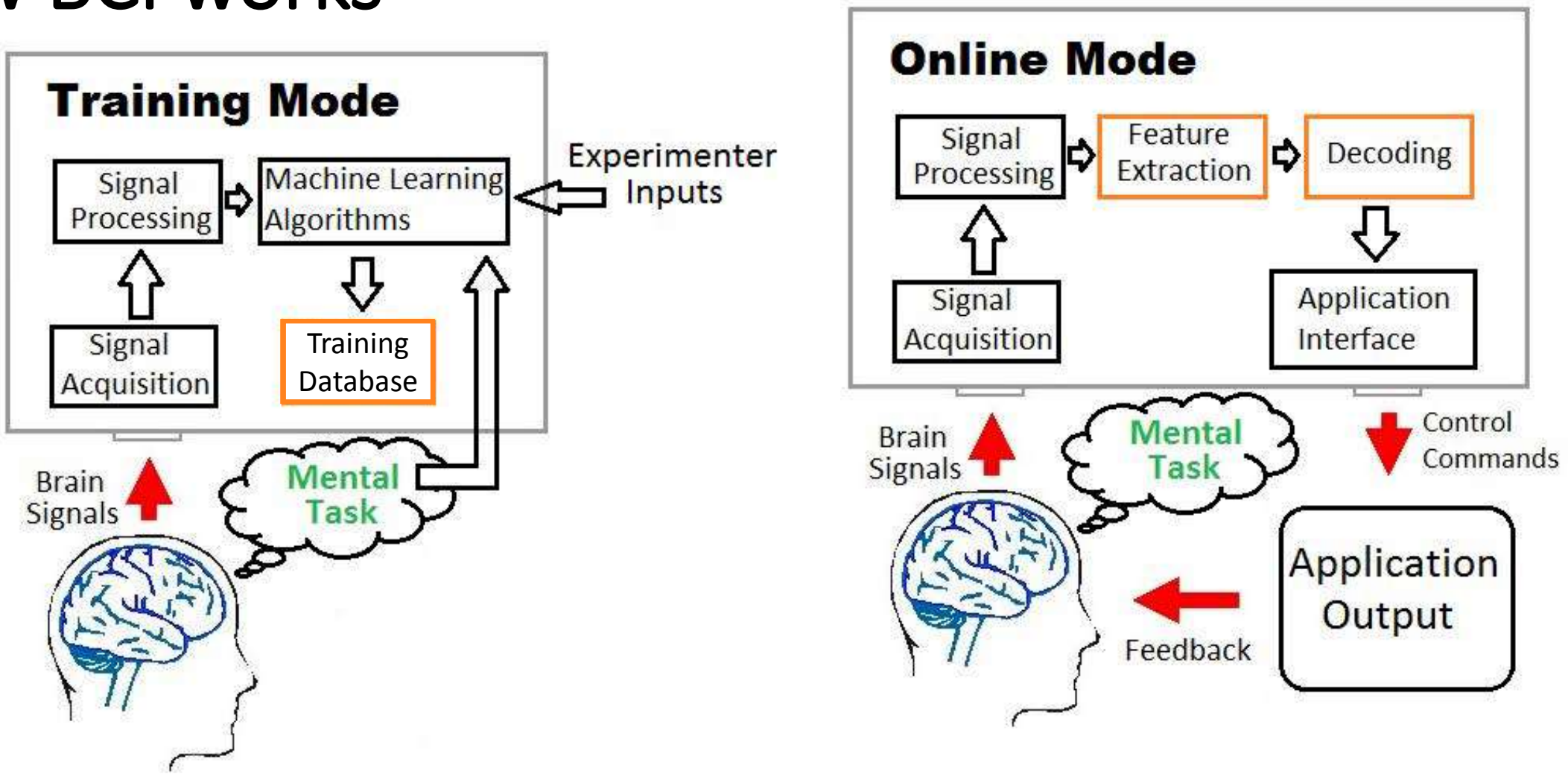
Motivation

- Spinal cord injury 350,000 (US), 25,000 new cases / year.*
  - Healthcare: $40.5 billion p.a.
  - Society: $306 billion p.a.
- Wheelchair dependence leads to:
  - Cardiovascular diseases
  - Diabetes
  - Osteoporosis
  - Muscular atrophy
  - Pressure ulcers
- Stroke >7 million (US), 800,000 new cases / year.**
  - Healthcare: $30 billion p.a.
- BCIs can potentially help them reduce reliance on caregivers, reduce medical costs

*National Spinal Cord Injury Statistical Center (NSCIS). Spinal cord injury facts and figures at a glance, 2013.

**Rogers et al, American Heart Association.

How BCI works

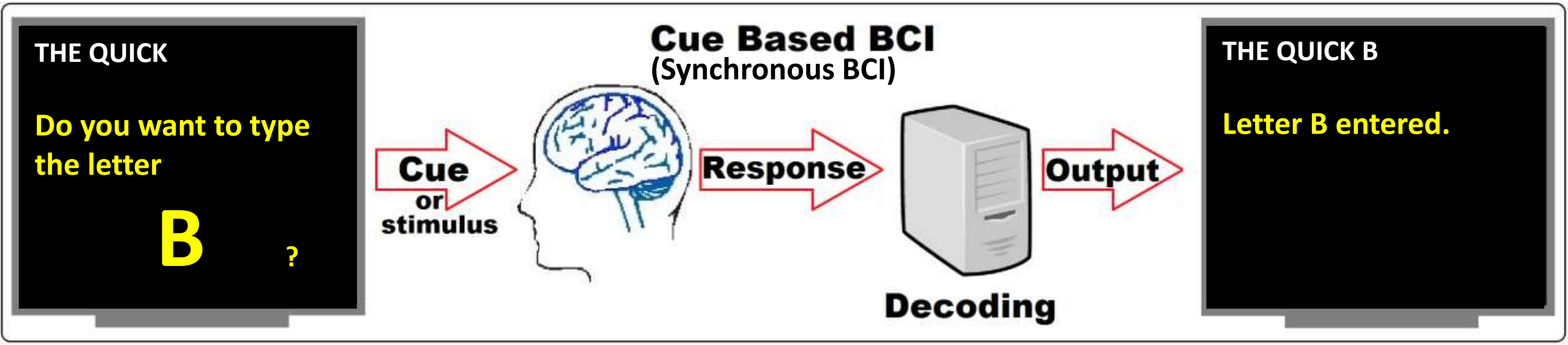


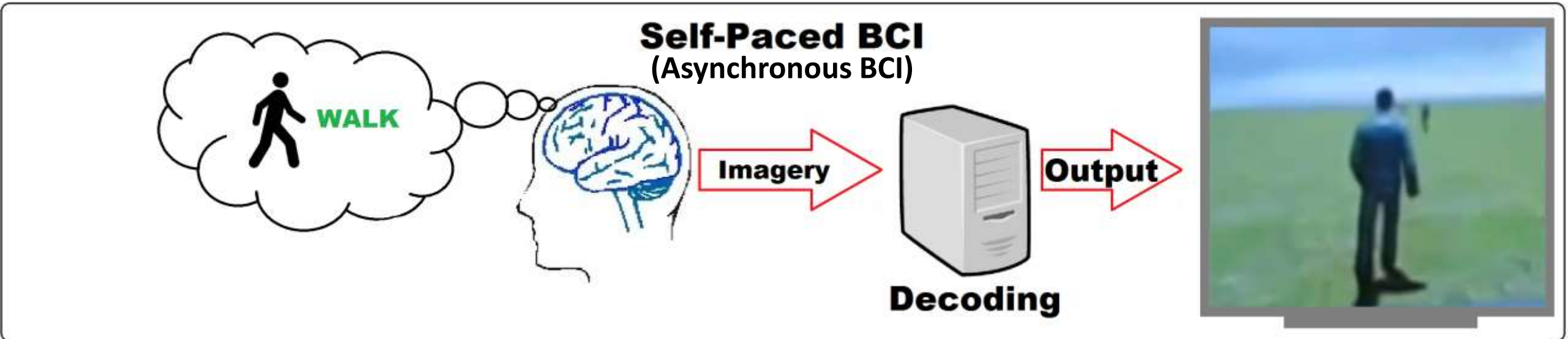
Training paradigms

- User-training:
  - Users are asked to learn how to change the amplitudes of their brain wave rhythms
  - 2-3 weeks, if not months training required
  - Usually not intuitive

- Computer-training:
  - Users are asked to perform intuitive **mental imagery** or **attempted movement**
  - Short training time (10-20 minutes) required

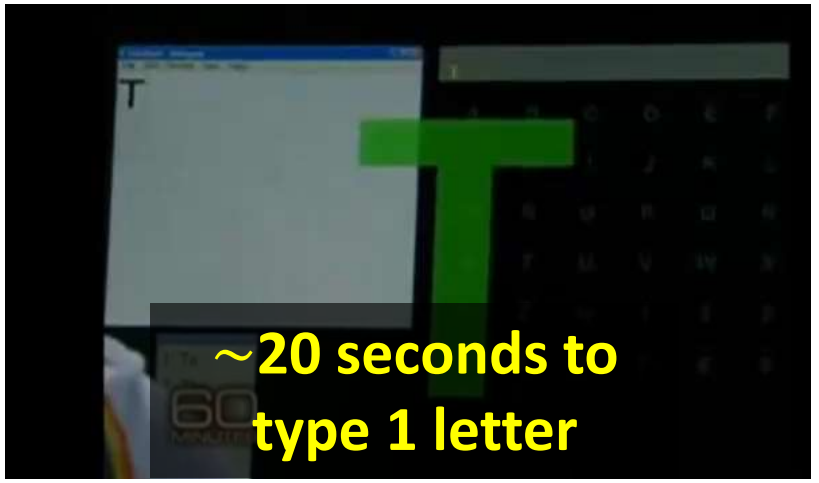
Control paradigms





Limitations of non-invasive EEG-based BCIs

1. Slow speed (< 1 bit/s)
2. Unintuitive operation (e.g. imagine foot and tongue movements to operate hand prosthesis)
3. Extensive training procedure (e.g. months)
4. Sensitive to artifacts



Potential solutions: Novel signal processing and pattern recognition techniques can

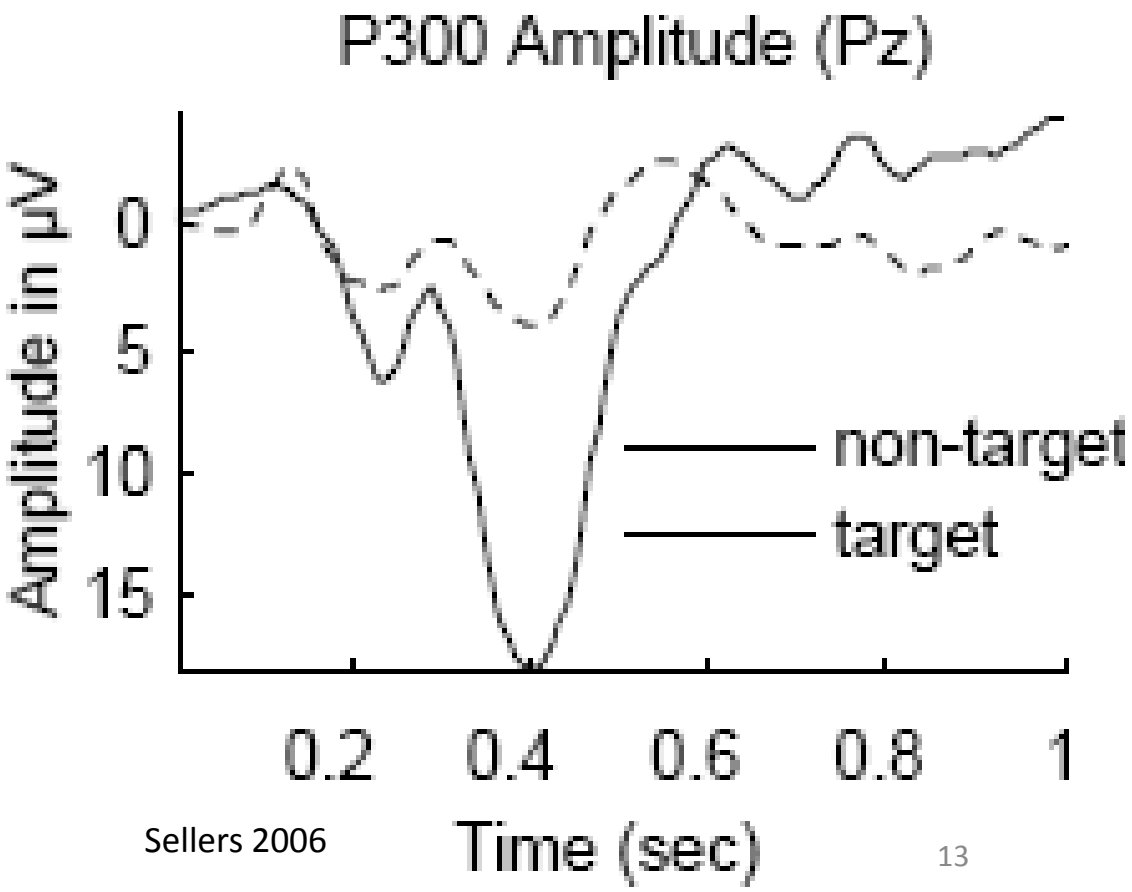
1. Improve BCI speed (> 3 bits/s)
2. Provide intuitive operation
3. Shorten the training time (15 minutes)

BCI Speller

P300 Speller

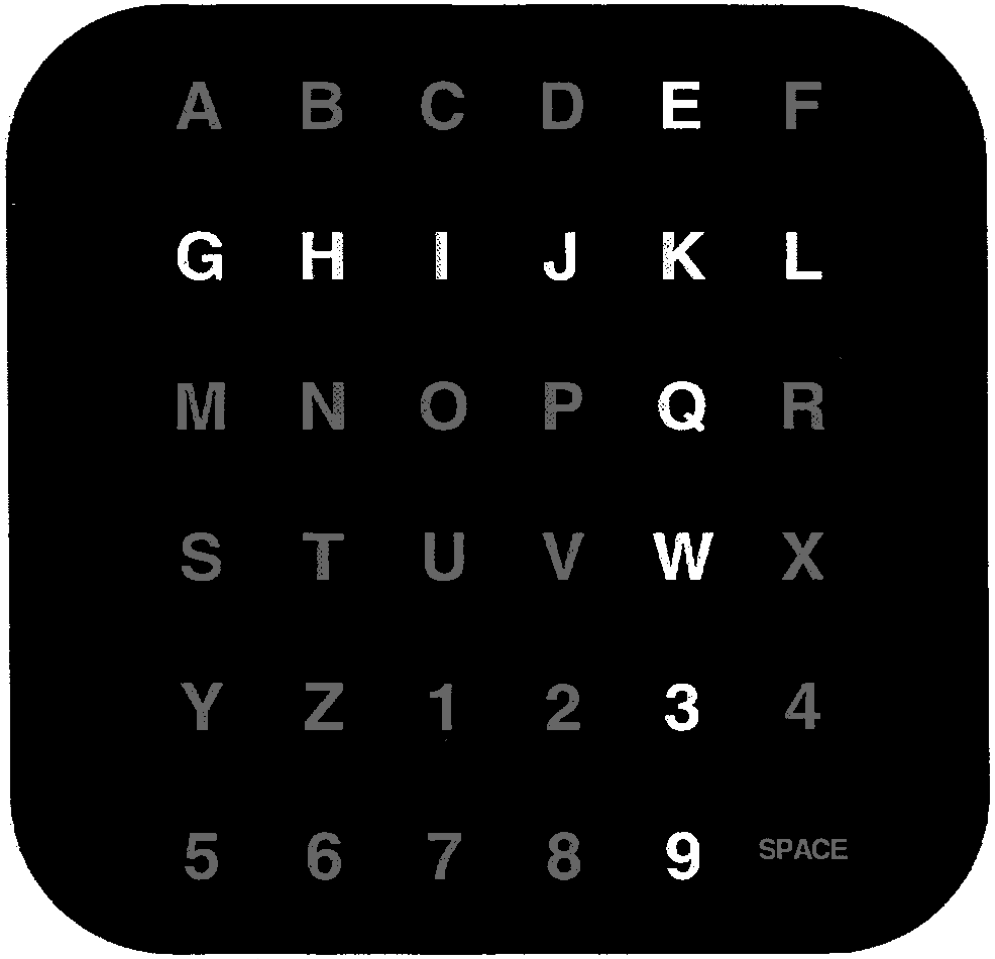
Electroencephalography (EEG)

- Farwell and Donchin (1988)
- Based on the visual **oddball** event-related potential (ERP) paradigm.
- Oddball = Rare and wanted object.
- ERP is an EEG response to a stimulus.
  - Weak and difficult to discern on a single-trial basis
  - Obtained by averaging many identical trials
- P300 wave



Sellers 2006

P300 Speller

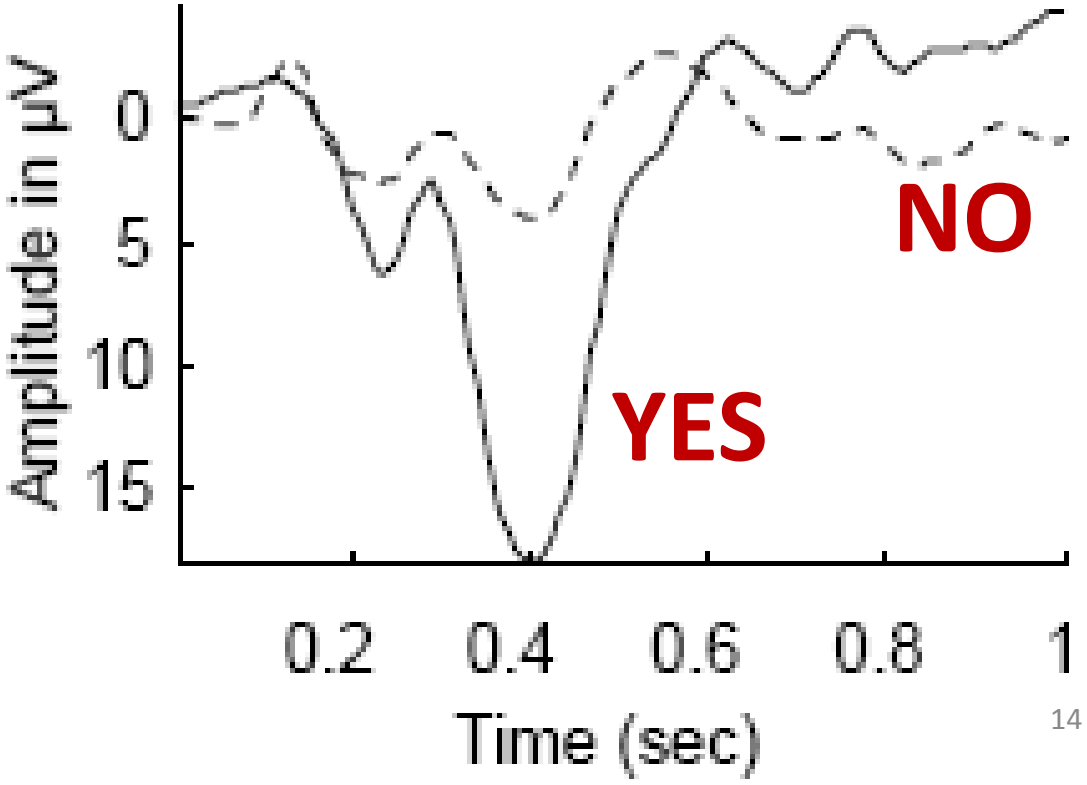


Rows and Columns paradigm in
Farwell & Donchin (1988), Sellers (2000), etc.

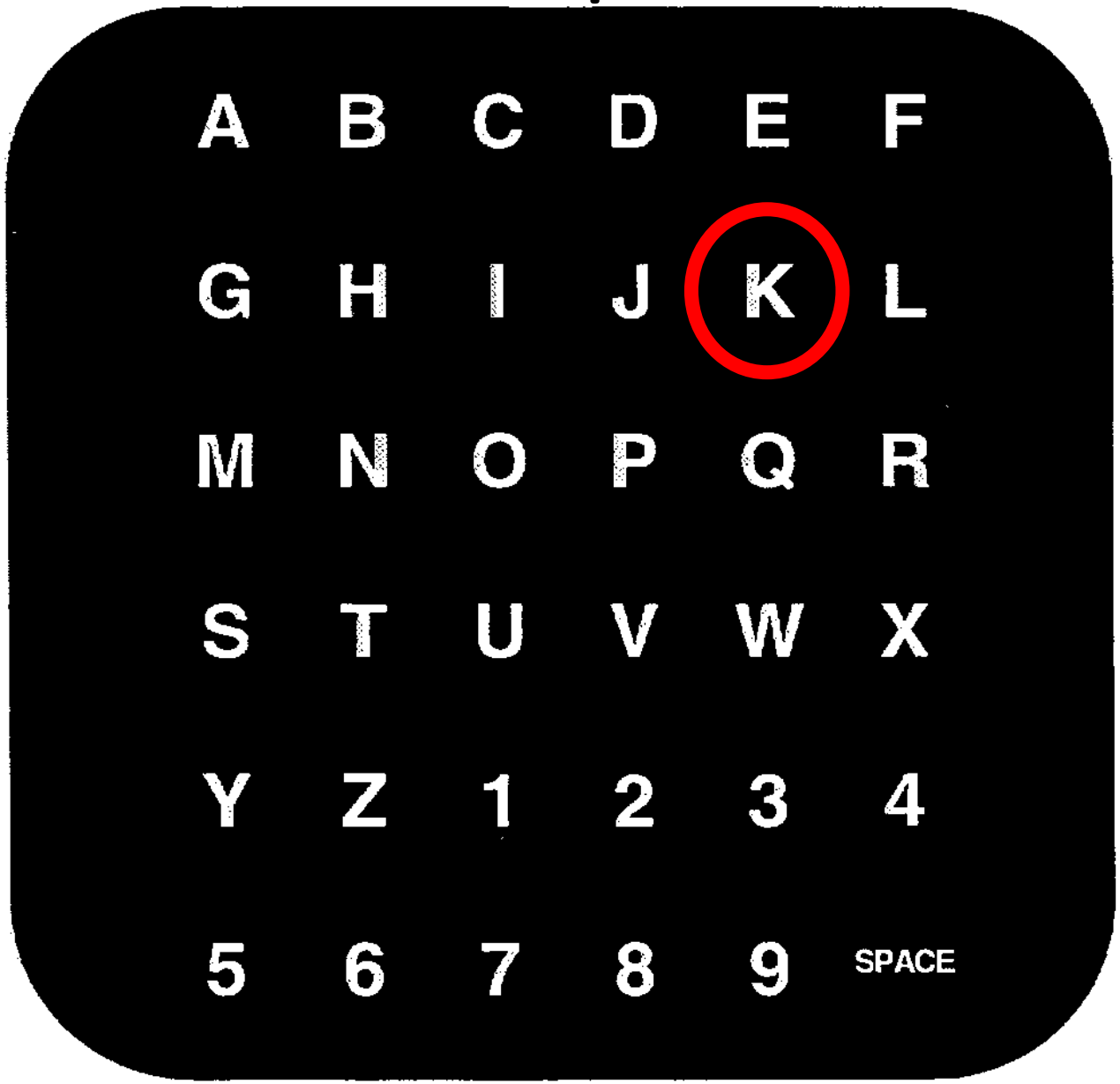
Essentially, the computer "probes" the human brain with this question:

Is the letter you want in this group?

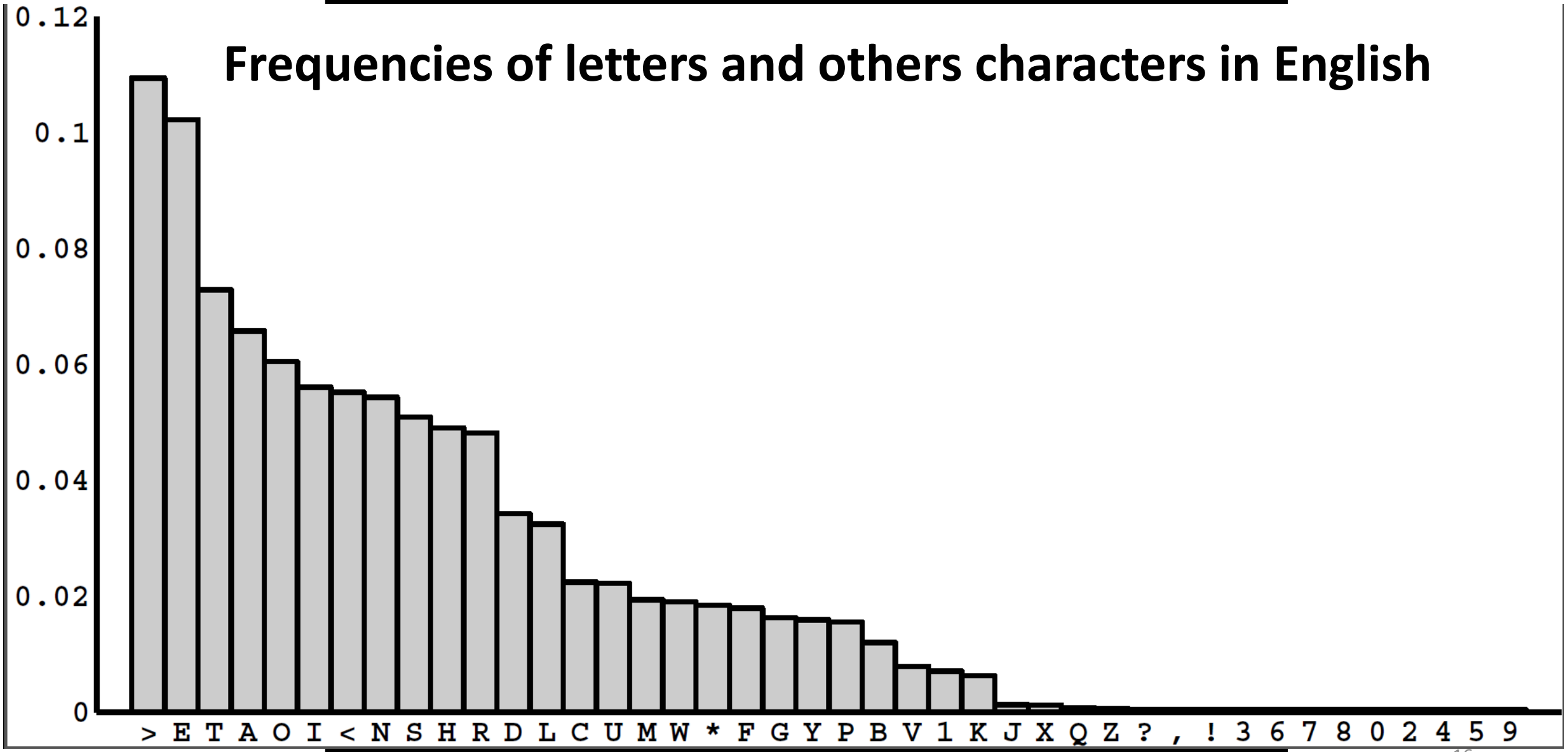
The only signals the brain can respond are:



P300 Speller

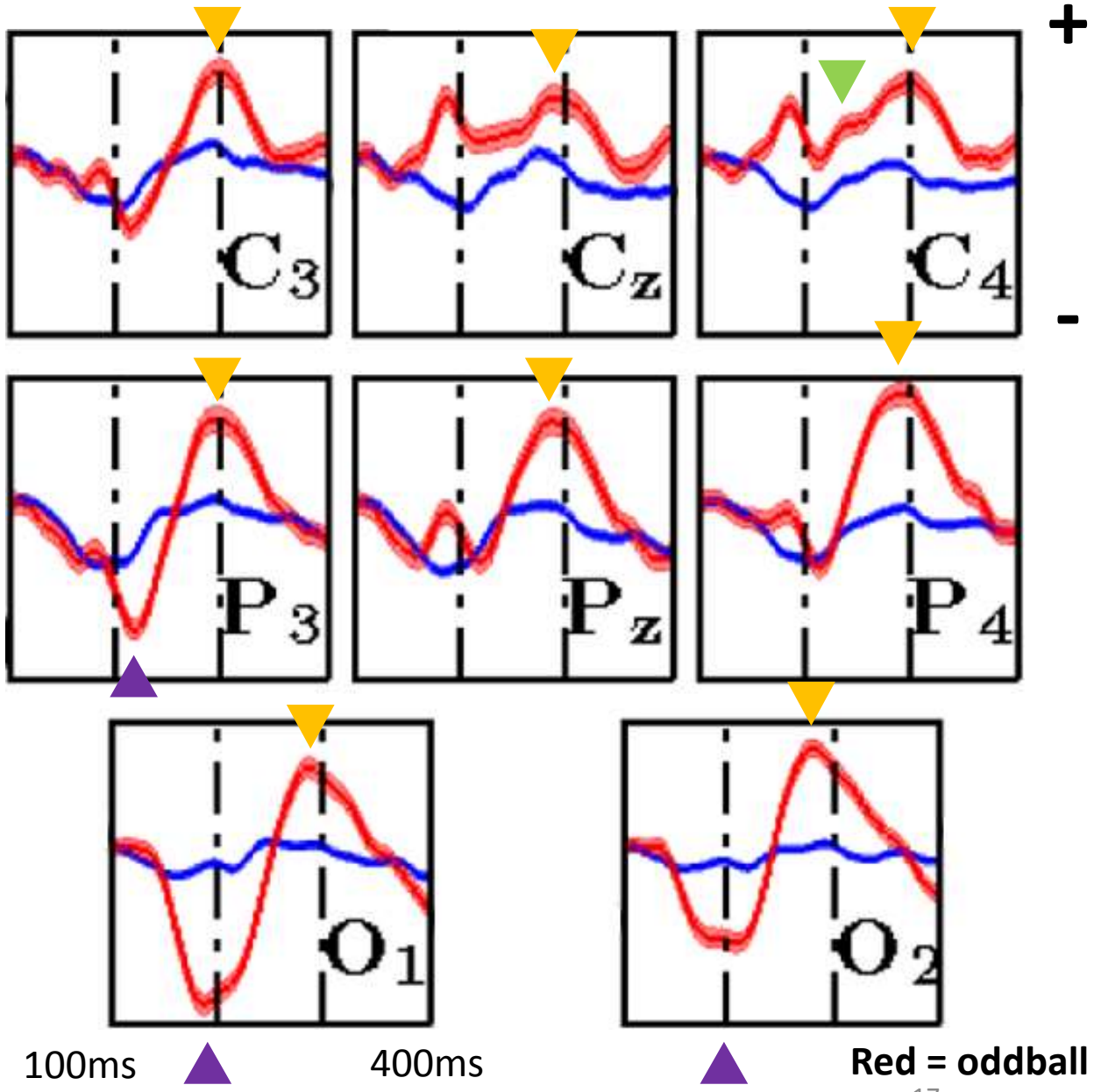


P300 Speller



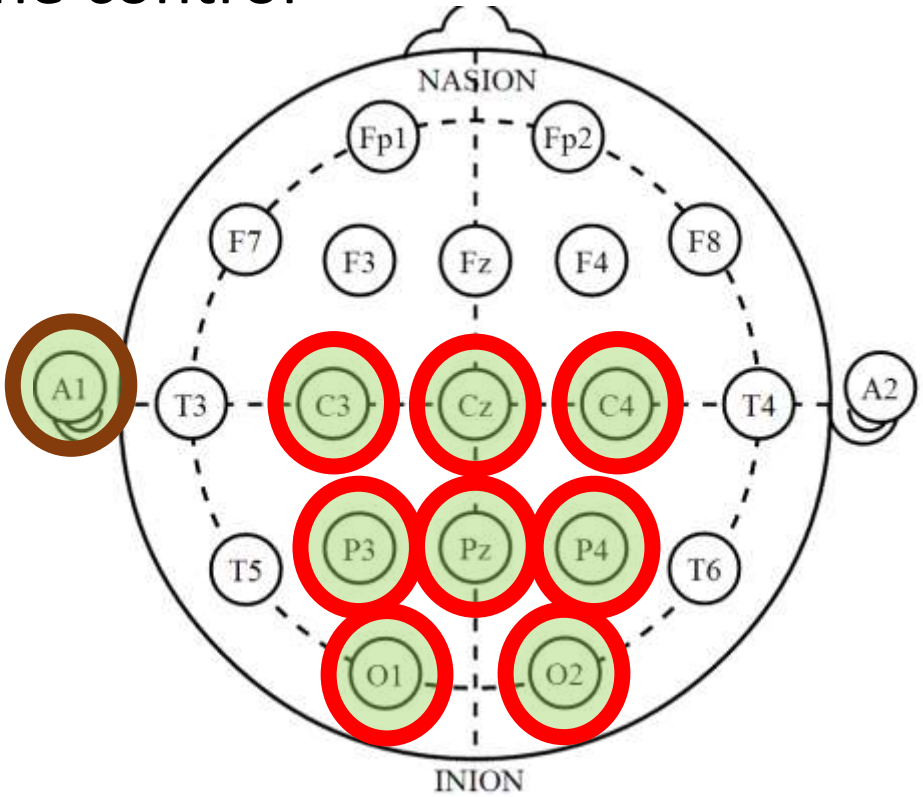
Our BCI Speller

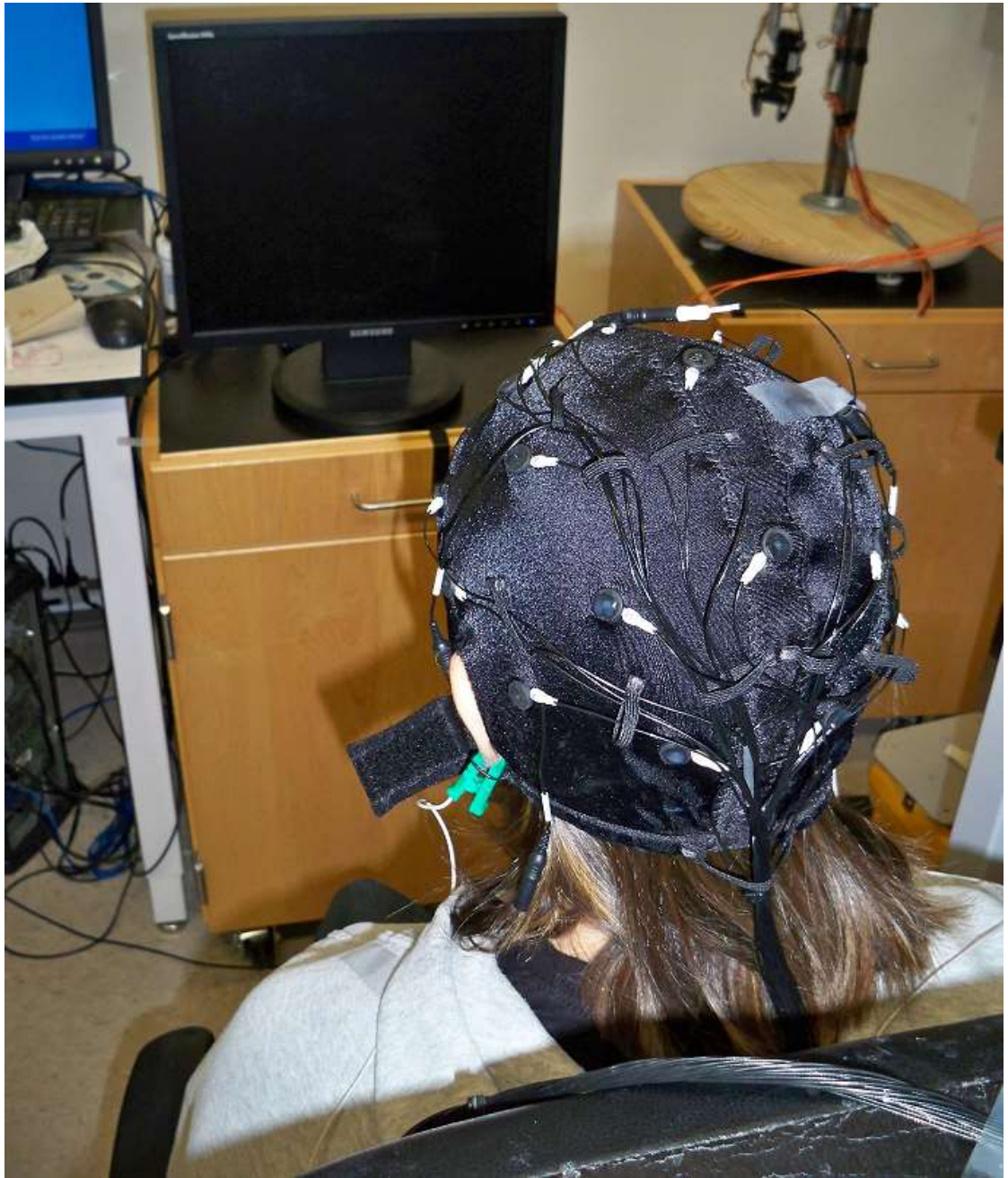
- Uses a combination of attention (P3a), oddball (P300/P3b), and visual search (N200/N2c) signals
- Does not expect any particular waveforms, completely data-driven
- Makes a decision based on only one trial of data



Experiment setup

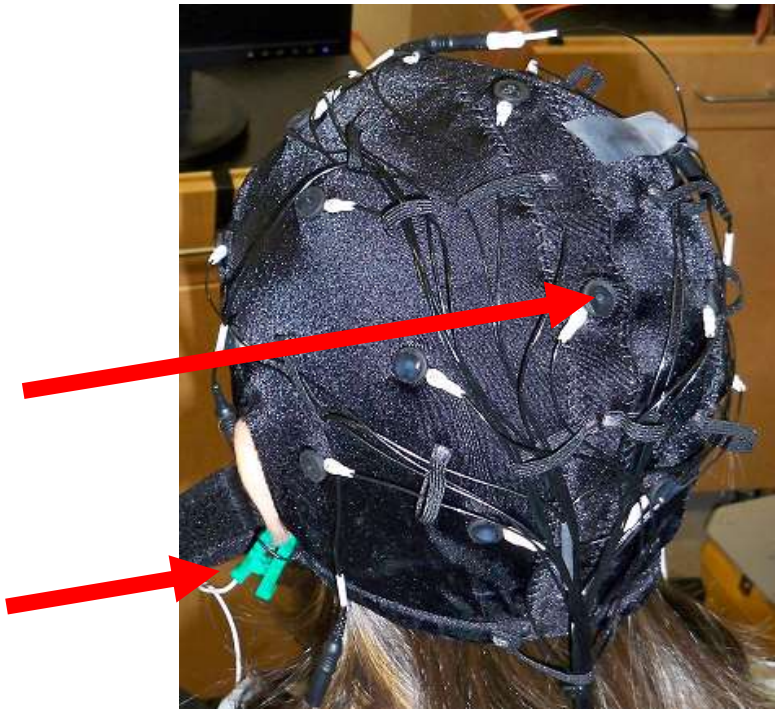
1. Human subject
2. EEG cap, 8 chan. (~10 minutes)
3. Training procedure (6-7 minutes)
4. Computer builds a classifier
5. Online control





Equipment

- **EEG cap**: NeuroScan Quik-Cap with Ag/AgCl electrodes (Compumedics USA, Charlotte, NC)
- **EEG reference**: Ear clip with Ag/AgCl electrode (Compumedics USA)



MP150 DAQ

EEG amplifiers: EEG100C (Biopac Systems, Goleta, CA)

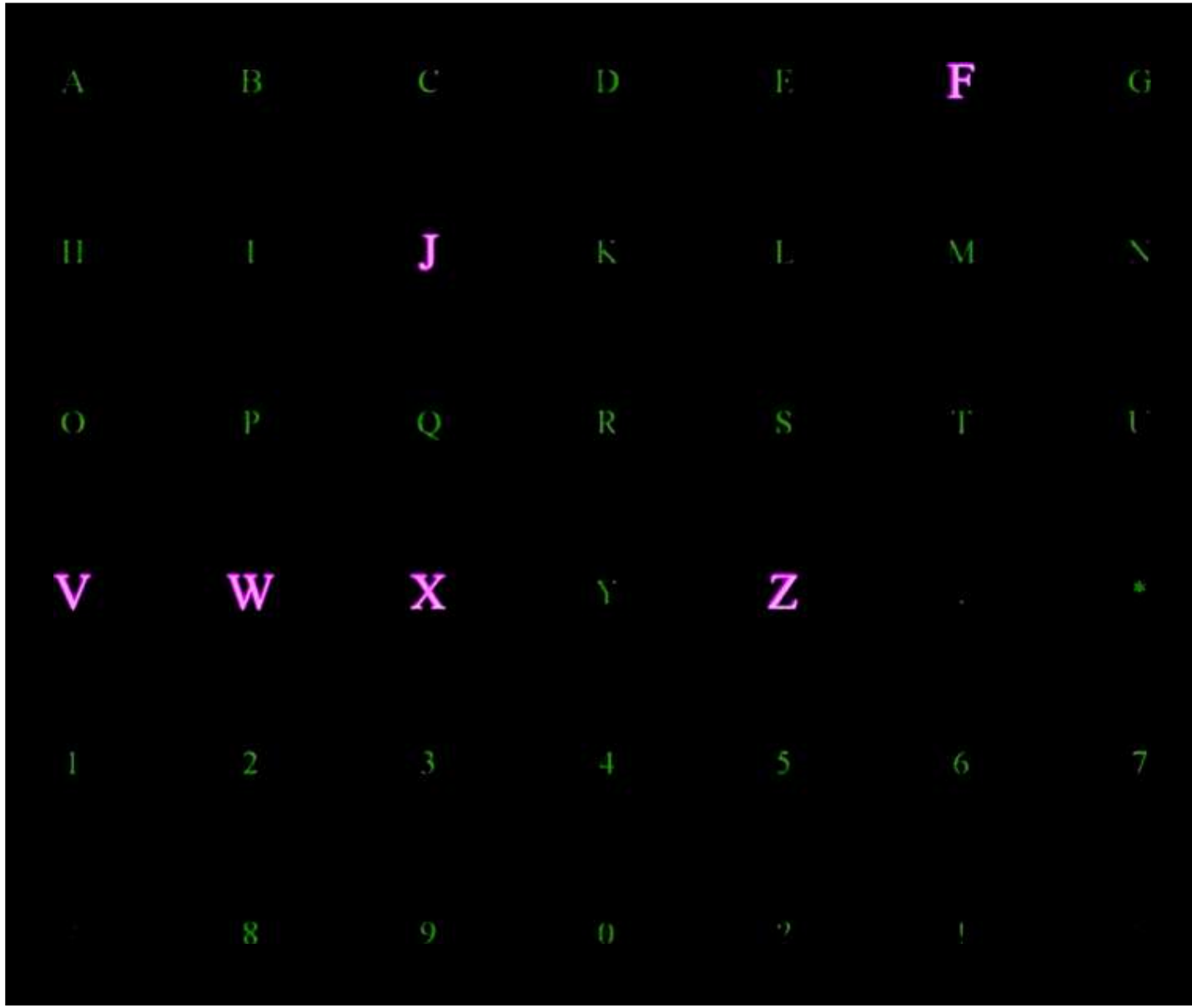
EEG data acquisition device (DAQ): MP150 (Biopac Systems)

Training procedure

- Comprises 10 mini-sessions
- In each 30-s mini-session, subject is asked to pay attention to one character from the following

  A G < > Y D V 0 * K

- This character is called the “oddball” or the “target”
- 8-10 s pause between mini-sessions
- Hence 6-7-minute training time



Timing terminology for stimulations

Stimulus duration (For our BCI-Speller, always 60% of the ITI)

Stimulus offset

Stimulus onset

Stimulation ON

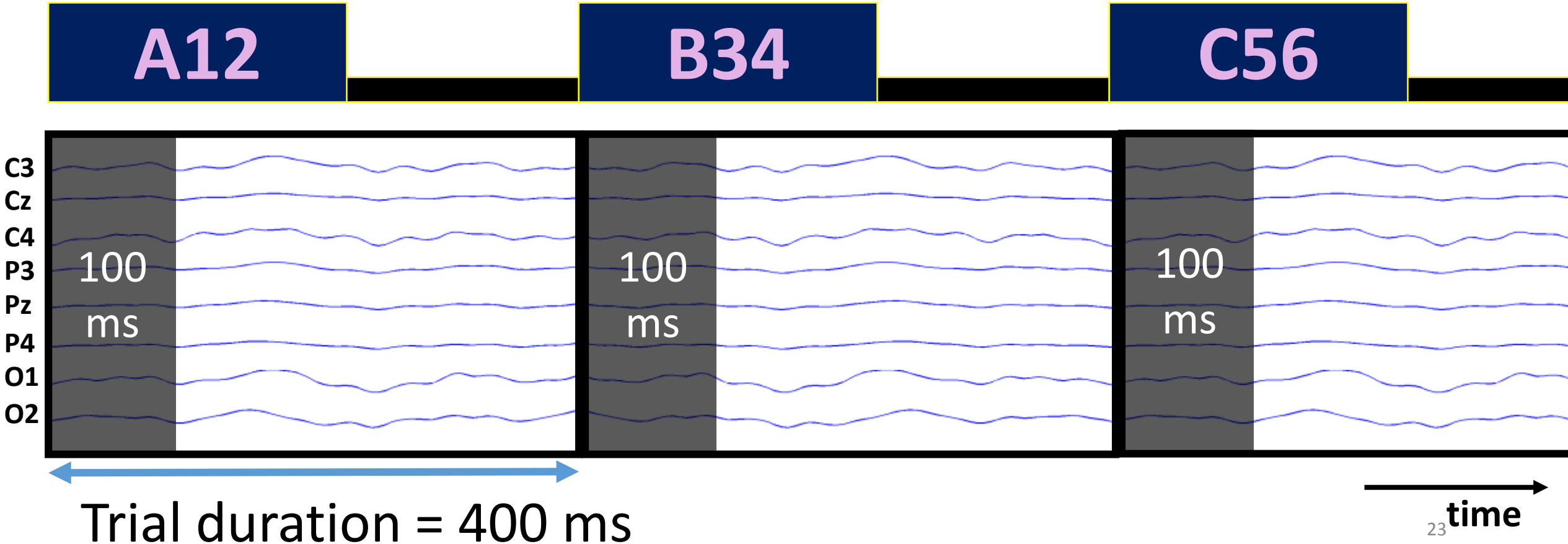
Stimulation OFF

3 different ITIs are offered:

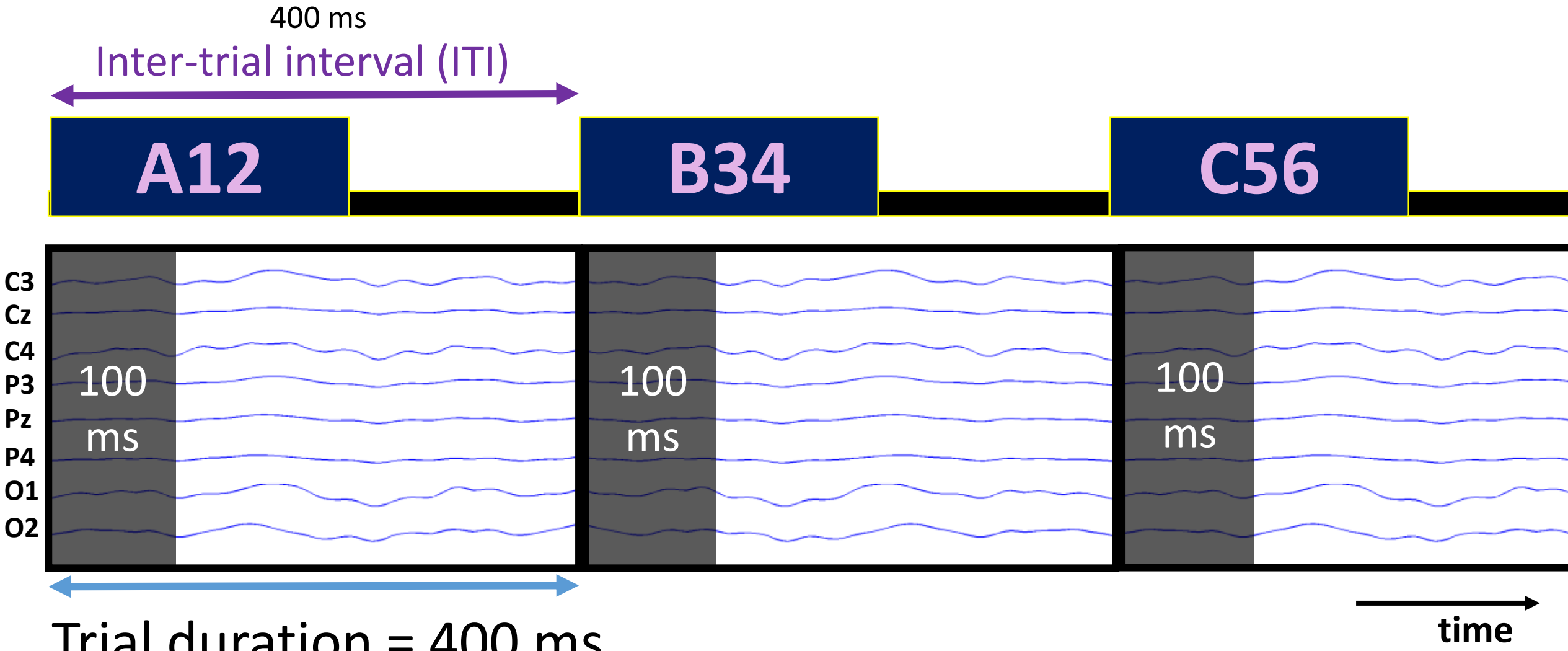
- 400 ms (slow speed)
- 240 ms (medium speed)
- 160 ms (fast speed)

Training procedure – data acquisition

- EEG sampled at 200 Hz from 8 channels
- 400 ms post-stimulus acquired as one ‘trial’. 400 ms = 80 samples
- First 100 ms discarded (visual transmission delay)



Slow speed



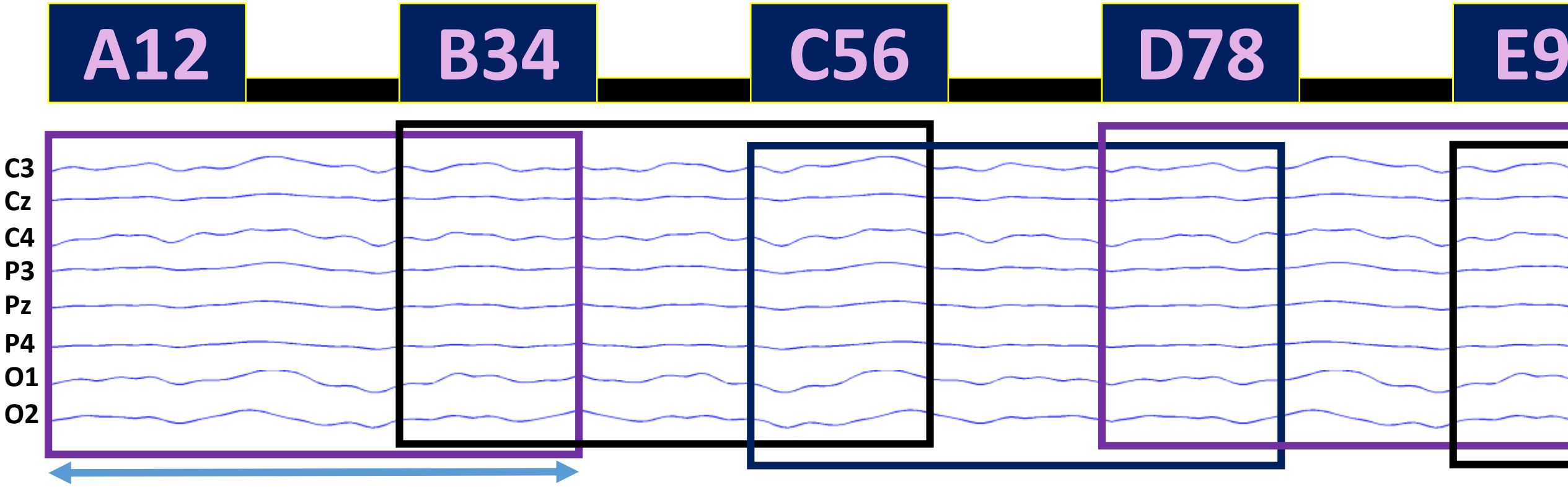
Trial duration = 400 ms

Medium/fast speed

240 ms (medium speed)
160 ms (fast speed)

Inter-trial interval (ITI)

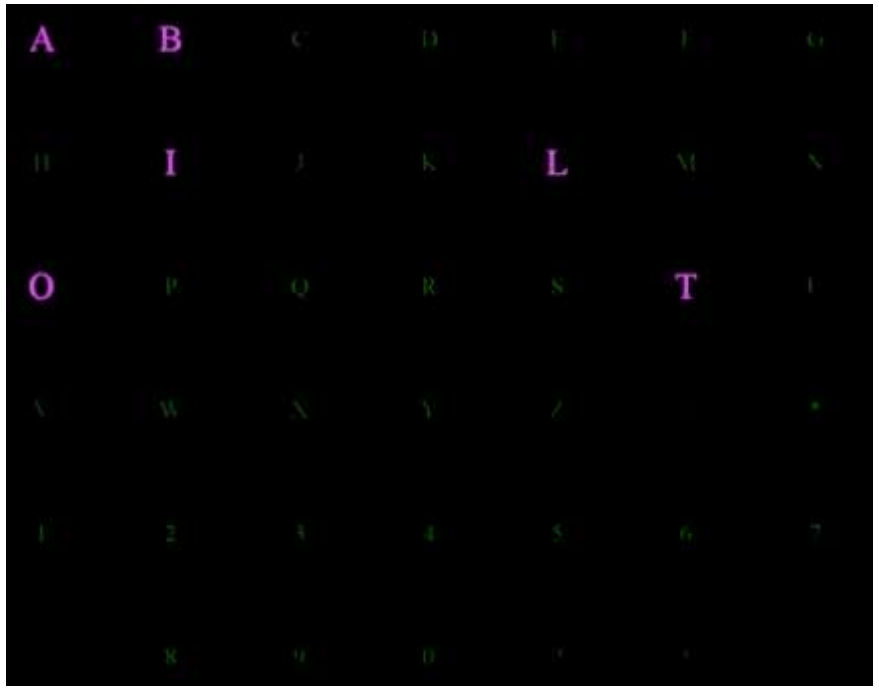
Trial duration remains the same.
Trials overlap

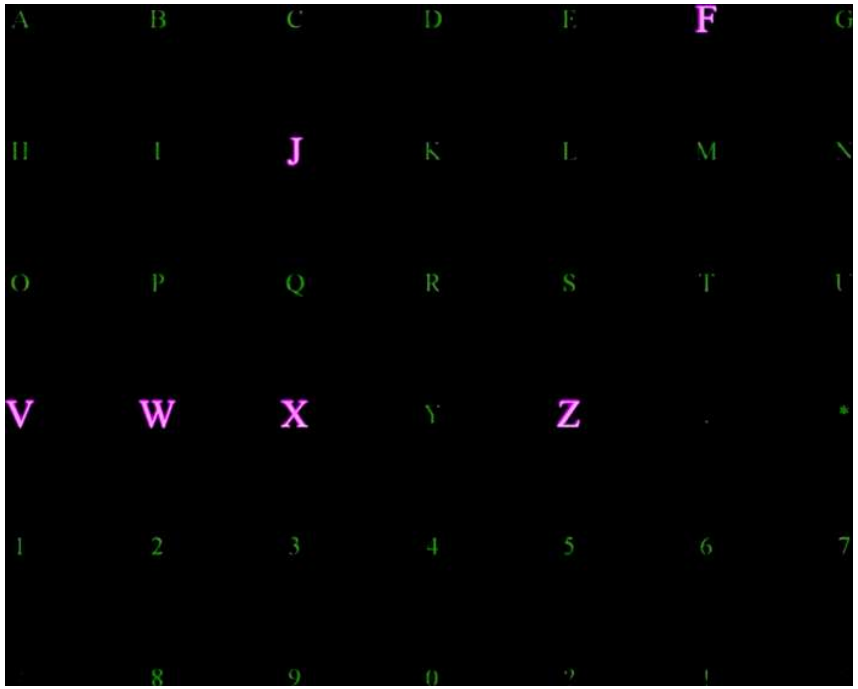


Trial duration = 400 ms

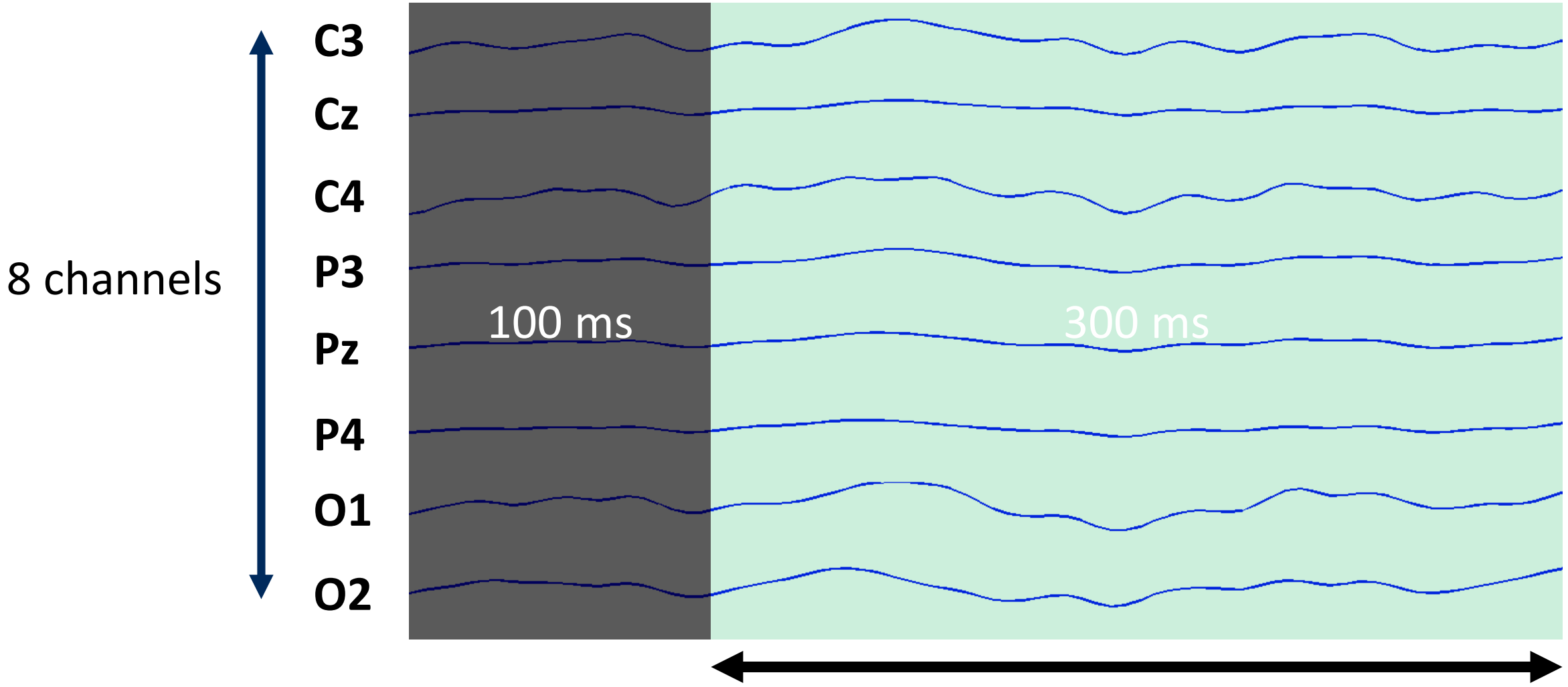
Training procedure – data labeling

- A group of letters can either contain or not contain the oddball.
- Trials recorded when the oddball is present is labeled **ODD**, and **EVEN** otherwise. ODD and EVEN are **classes**.





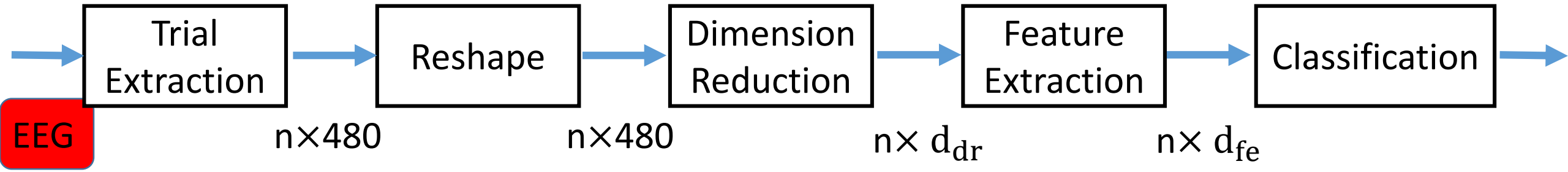
Data dimension



This is 1 trial of EEG data.

Feature extraction and classification Motivation

1. There are many redundant and irrelevant data in EEG.
2. We only need a subset of **features** from the EEG that can tell us whether or not oddball is presented to the subject.
3. Feature extraction is to extract only the relevant subset of EEG data that help us with the task of discriminating between **ODD** and **EVEN** classes.



Feature extraction and classification
Preparing data

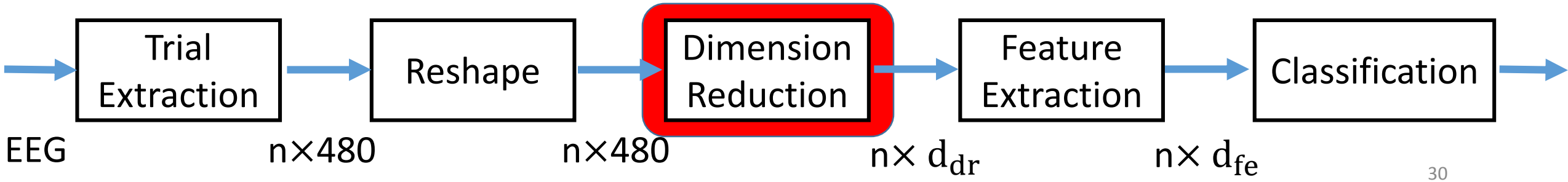
1. Each trial of data (8 ch × 60 sp) is reshaped into a vector (1 × 480).
2. Trial vectors from each class form two matrices:
   1. Oddball matrix ($\mathbf{n_o} \times 480$) containing only oddball trials
   2. Evenball matrix ($\mathbf{n_E} \times 480$) containing only evenball trials
3. Feed both matrices into <u>dimension reduction</u> and then <u>feature extraction</u> algorithms

Dimension reduction
Motivation

1. Many linear discriminant feature extraction techniques do not work well with **small-sample-size problems** (more dimensions than samples).
2. Even at fast speed, we only get 267 oddball samples, but we have 480 dimensions.
3. A more general, systematic way to reduce the dimension is required.

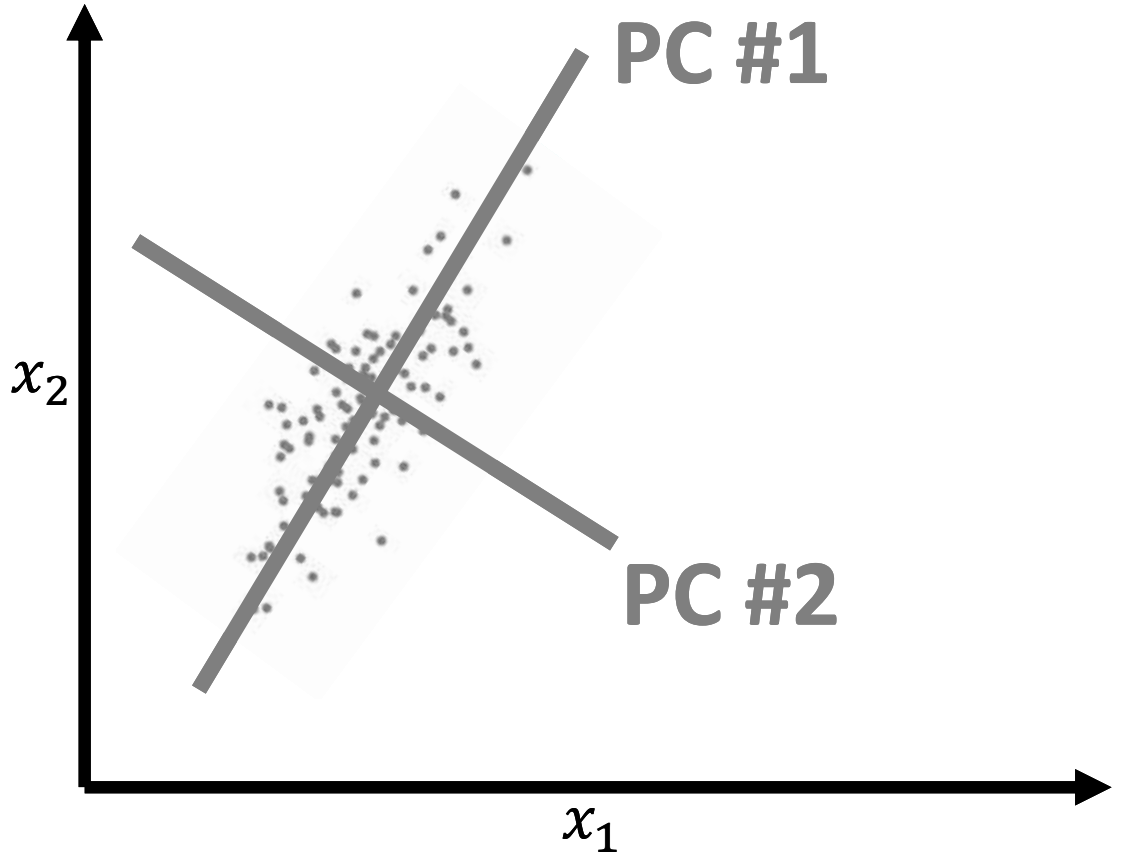
PCA

CPCA



Principal Component Analysis (PCA) Overview

- PCA is a procedure that transforms data into linearly uncorrelated components (principal components), sorted by variances in descending order
- Similar to:
  - Singular value decomposition of the data
  - Eigenvalue decomposition of the covariance
- You can only have up to **n-1** or **d** principal components, whichever is smaller
  - n = number of trials
  - d = number of original dimensions



Dimension reduction (DR) by PCA Algorithm (SVD method)

X is data. E[X] = 0.

1. Calculate SVD: $\mathbf{X} = \mathrm{U}\,\Sigma\,\mathrm{V}^{\mathrm{T}}$. U is (n × n), S is (n × d), V is (d × d)
2. **PCA**:
   - Retain only the *n-1* highest singular values. Call this σ'.
   - Square and scale the singular values. $s = (\sigma')^2/(\mathrm{n}-1)$, i.e. **latents**
     Retained vectors in V **= PC coefficients**.
3. **Dimension reduction**:
   - Retain only latent values above the mean
   - Retained vectors = $\boldsymbol{\Phi}$**, the retained PC coefficients**

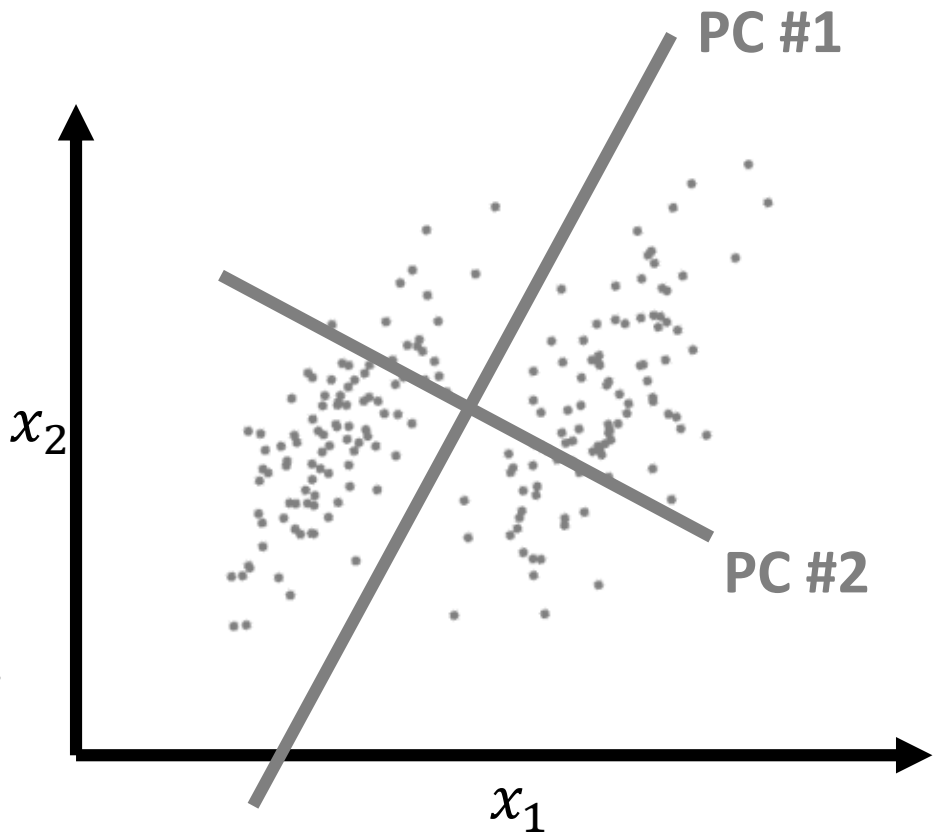
If we multiply **X** by $\boldsymbol{\Phi}$, we reduce the dimension of **X**.

Dimension reduction by Classwise PCA (CPCA) Motivation

1. Ordinary PCA does not consider labels. It does not know whether data is for oddball.

2. Ordinary PCA may fail to separate data into the correct labels due to variances.

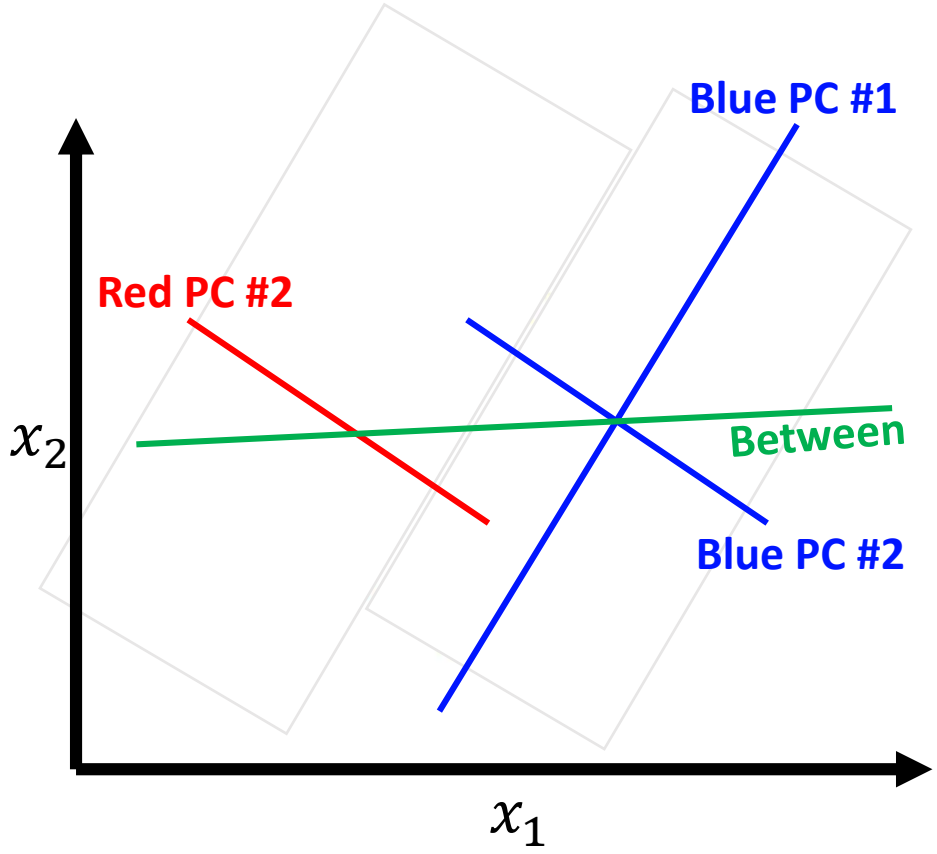
3. Problem can be alleviated by adding between-class information and treat each class separately



Ref: Das K, Nenadic Z (2009). An efficient discriminant-based solution for small sample size problem.

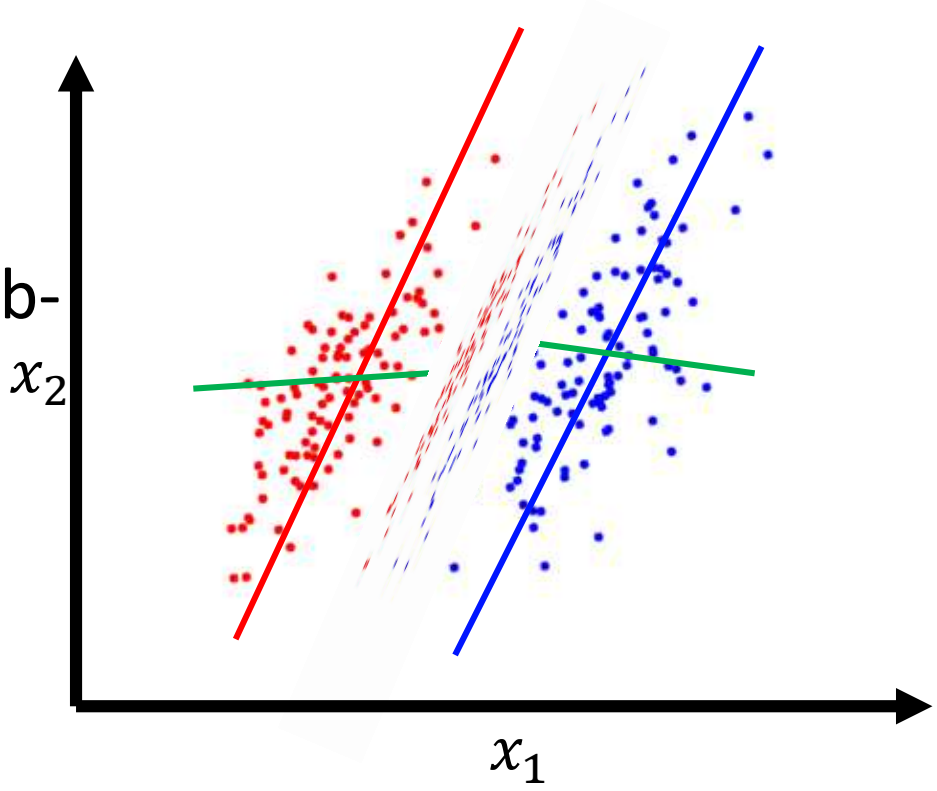
Dimension reduction (DR) by CPCA Algorithm

1. Perform PCA + DR on **each class**
   - Red cloud → Red PC
   - Blue cloud → Blue PC
2. Between-class PC coefficient (green line)
3. Each class has its own PC #1s and the green line
4. Orthonormalize separately:
   - Red and green lines
   - Blue and green lines
5. Result: Classwise PC coefficients
   - Red + green lines = PC coeff. for red subspace
   - Blue + green lines = PC coeff. for blue subspace



Feature extraction
Motivation

- CPCA retains 20-40 dimensions.
- Covariance estimates.
- Picking 1-3 highest latents (from CPCA) may be sub-optimal
- Need an algorithm that is specifically designed to separate classes
- Easier to visualize the data in 1-3 dimensions
- LDA and AIDA



Linear discriminant analysis (LDA) Algorithm

Assume Gaussian distribution for class data, maximize the ratio of between-class covariance to within-class covariance.

1. Calculate within-class covariance of class data X_C. (X_C is already reduced by CPCA.)

$$\Sigma_w = \sum_C P_C \, X_C X_C^T$$

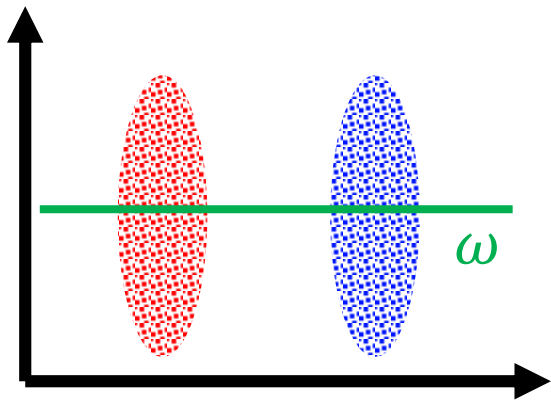
P_C is the prior probability of C.
$P_C = n_C/n$

2. Calculate between-class covariances

$$\Sigma_b = \sum_C P_C(\mu_C - \mu)(\mu_C - \mu)^T$$

3. Find the matrix ω in $\underset{\omega}{\operatorname{argmax}} \frac{\omega^T \Sigma_b \omega}{\omega^T \Sigma_w \omega}$ by Generalized EVD:

$$(\Sigma_w^{-1}\Sigma_b)\omega = \Lambda\omega$$



4. The normalized matrix $\|\omega\|$ is the **LDA coefficient**

- Note: The dimension of ω must be less than number of classes

Approximate information discriminant analysis (AIDA) Algorithm

Based on mutual information between data and class variable.

1. Whitening transform:

$$\mathrm{W} = (\Sigma_{\mathrm{w}})^{-1/2}$$
$$\mathrm{S} = \mathrm{W}(\Sigma_{\mathrm{w}} + \Sigma_{\mathrm{b}})\mathrm{W}$$
$$\mathrm{S_C} = \mathrm{W}\,\mathrm{X_C}\mathrm{X_C^T}\,\mathrm{W}$$

W = Whitening operator
Σ_{W} = Within class covariance
Σ_{b} = Between-class covariance
$\mathrm{P_C}$ = Prior probability

S = Whitened total covariance
$\mathrm{S_C}$ = Whitened class covariance

2. Approximate μ measure:

$$\mathrm{Z} = \log \mathrm{S} - \sum_{\mathrm{C}} \mathrm{P_C} \log \mathrm{S_C}$$

3. Solve the EVD with only the *m* largest eigenvalues (m = output dimension)

$$\mathrm{Zv} = \lambda \mathrm{v}$$

4. The **AIDA coefficient** is:

$$\left\| \mathrm{v^T W} \right\|^{\mathrm{T}}$$

Ref: Das K, Nenadic Z (2008). Approximate information discriminant analysis: A computationally simple heteroscedastic feature extraction technique
Nenadic Z (2007). Information discriminant analysis: feature extraction with an information-theoretic objective

Classification by Bayes rule Formula

Data are reduced to **features** of 1-3 dimensions and can be classified:

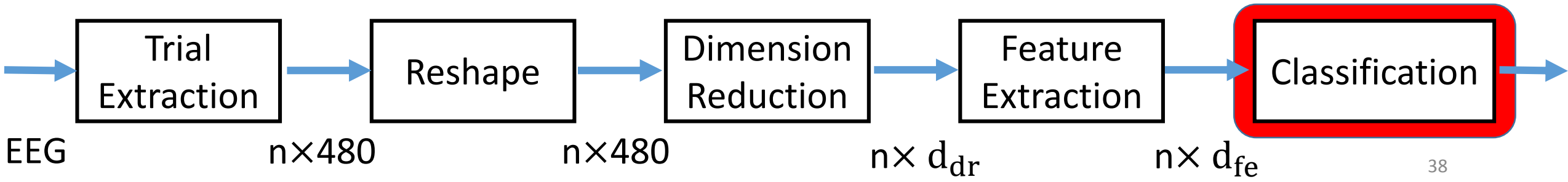
$$f_C = X\,\Phi_C\,T_{DA}$$

The posterior probability of class C given data feature f is:

$$P(C|f) = \frac{p(f|C)\,P_C}{p(f)}$$

- $p(f) = \sum_C p(f|C)P_C$ is the total probability of f
- P_C is the prior prob. of C
- $p(f|C)$ is the **likelihood** of f given C.

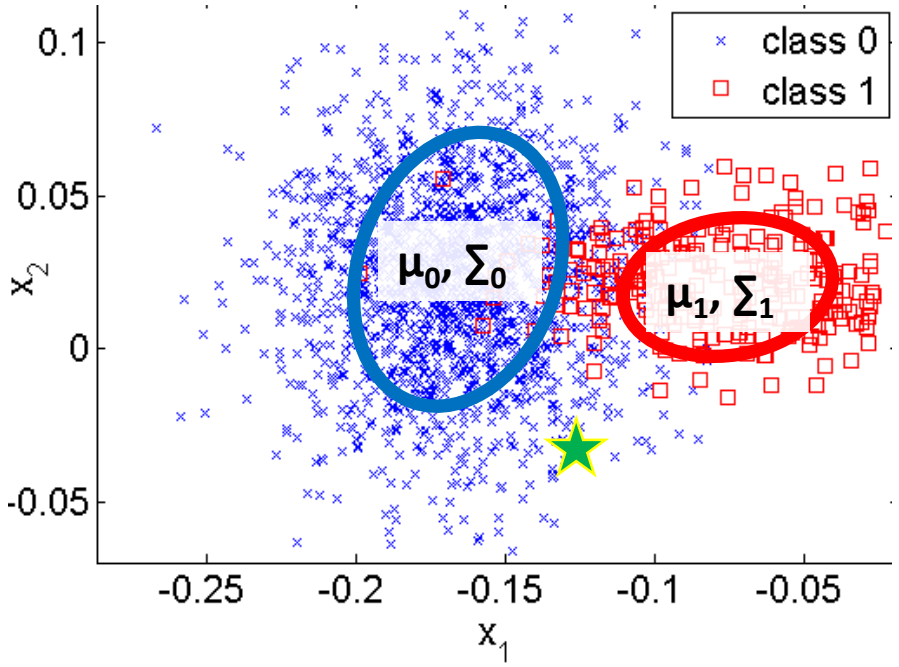
In BCI Speller, $\boldsymbol{P_C = (\frac{1}{7}, \frac{6}{7})}$



Classification by Bayes rule
Likelihood function

We use Gaussian (normal) distribution for the likelihood. It is essentially the Mahalanobis distance between f and μ_C.

$$p(f|C) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_\star|}} \exp\left((f - \mu_C)\Sigma_\star^{-1}(f - \mu_C)^T\right)$$



d = 2

C = 0 or 1

Classification by Bayes rule
Quadratic and linear classifiers

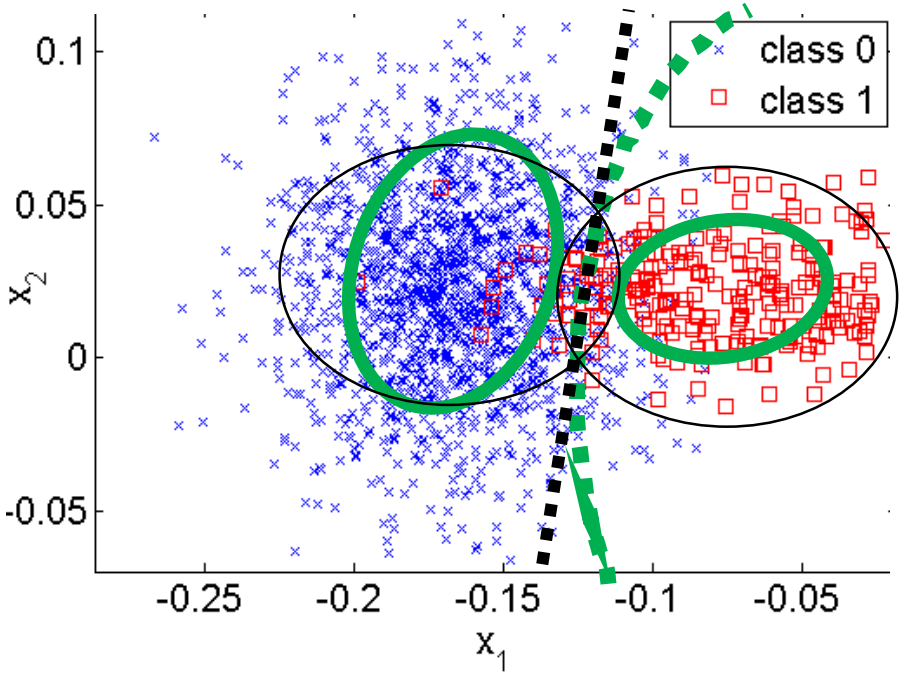
Note the variance $\Sigma_\star$ in the likelihood function can either be the class covariance or the total covariance.

- For quadratic classifier,

$$\Sigma_\star = \Sigma_C$$

- For linear classifier,

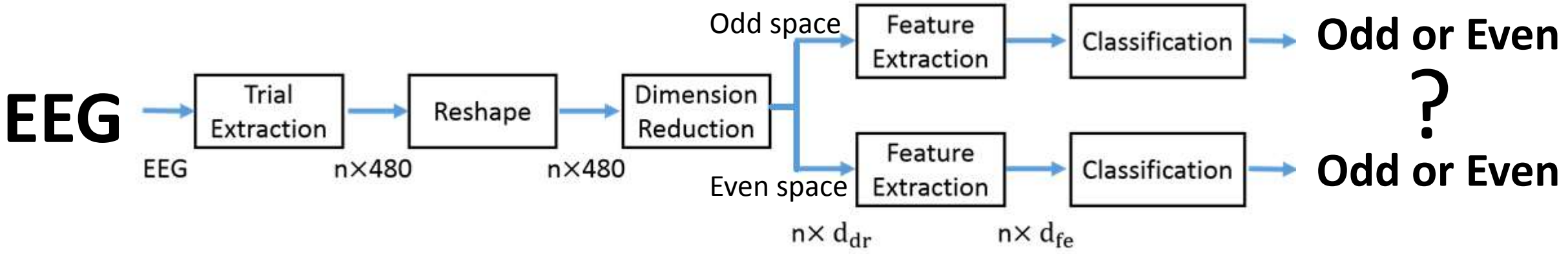
$$\Sigma_\star = \sum_C \mathrm{P_C}\, \mathrm{X}_C \mathrm{X}_\mathrm{C}^\mathrm{T}$$



- Quadratic is more specific to each class (since classes have different covariances)
- Linear is more robust (more trials to estimate covariance)

Classification
Classwise decision rule

- We get the posterior probabilities for each class, e.g. $\mathrm{P}(\mathrm{Oddball} \mid \mathrm{f})$ and $\mathrm{P}(\mathrm{Evenball} \mid \mathrm{f})$ from Bayes rule, **from each class subspace.**
  - Remember there are as many Φ_C as there are classes, and each forms a subspace.
  - We need a way to make it compatible with ordinary Bayes classifier.



Classification
Classwise decision rule

The class with the highest posterior, anywhere, wins.

Example:

s_o = **oddball subspace**
s_e = **evenball subspace**

$$\mathrm{P}(\mathrm{Odd}|\mathrm{f},\mathrm{s_o}) = 0.1, \qquad \mathrm{P}(\mathrm{Even}|\mathrm{f},\mathrm{s_o}) = 0.9$$
$$\mathrm{P}(\mathrm{Odd}|\mathrm{f},\mathrm{s_e}) = 0.6, \qquad \mathrm{P}(\mathrm{Even}|\mathrm{f},\mathrm{s_e}) = 0.4$$

"Even" class wins

To consolidate the posteriors, normalize highest posterior from each class:

$$\mathrm{P}(\mathrm{Odd}|\mathrm{f}) = \frac{0.6}{1.5} = 0.4, \qquad \mathrm{P}(\mathrm{Even}|\mathrm{f}) = \frac{0.9}{1.5} = 0.6$$

Classification
Classwise decision rule

With the posterior probabilities calculated, we can decide on the class label of the EEG data:

$$\text{EEG trial is} \begin{cases} \text{Oddball} : \mathrm{P}(\text{Odd}|\mathrm{f}) > \mathrm{P}(\text{Even}|\mathrm{f}) \\ \text{Evenball} : \text{Otherwise} \end{cases}$$

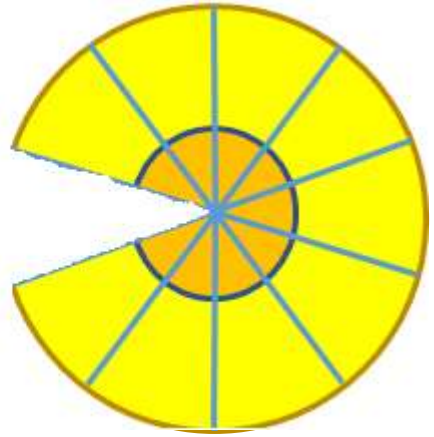
Maximum a posteriori (MAP) rule

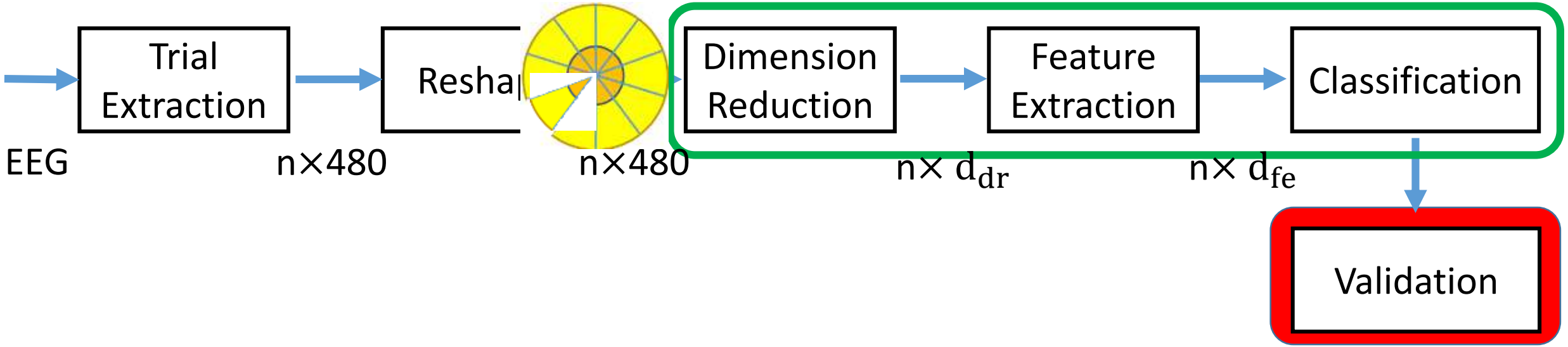
Recap:

1. One trial of data **X** = (8 channels × 60 sample points).
2. Feature extracted per class subspace: $\mathrm{f_c} = \mathrm{X}\Phi_\mathrm{C}\mathrm{T_{DA}}$
3. Posterior probabilities from classwise classification: $\mathrm{P}(\mathrm{C}|\mathrm{f})$
4. Decision from posteriors is made

Cross-validation (CV)

- Cross-validation is a technique to estimate the performance of the classifier on new data, without obtaining new data.
- We use stratified 10-fold cross-validation. In each CV iteration,
  1. 10% of trials from each class are removed from the training data
  2. The remaining 90% are used to build the classifier (green box)
  3. The classifier decodes those 10% trials into class labels
  4. The numbers of times it gets correct/wrong are tallied





Cross-validation output

- The output of a cross-validation run is a confusion matrix:

$$\begin{bmatrix} P(C_1|C_1) & \cdots & P(C_m|C_1) \\ \vdots & \ddots & \vdots \\ P(C_1|C_m) & \cdots & P(C_m|C_m) \end{bmatrix}$$

  where m is the total number of unique classes, and $P(C_X|C_Y)$ is read as "probability of decoding a new trial from Class Y as Class X".

- Diagonal terms are the probabilities of correct decoding
- The total probability of correct decoding is the sum of the diagonal weighted by priors:

$$P_{correct} = \sum_i P_{C_i} \, P(C_i|C_i)$$

BCI-Speller experiment

Six subjects (2F, 22-56 y.o.) participated over 3 days. Each day consisted of training and online procedures for all 3 speeds (slow, medium, fast)

Table 5.1: Demographic data of the study participants.

Subject	Gender	Age	Prior BCI experience	Native English speaker
A	F	23	3 hours	Yes
B	M	40	10 hours	No
C	M	29	1 hour	Yes
D	M	22	Naïve	Yes
E	M	24	Naïve	Yes
F	F	56	10 hours	Yes

BCI-Speller experiment training procedure results

1. Training performance (cross-validation) ranged ~90-97% (chance level = 85.7%)

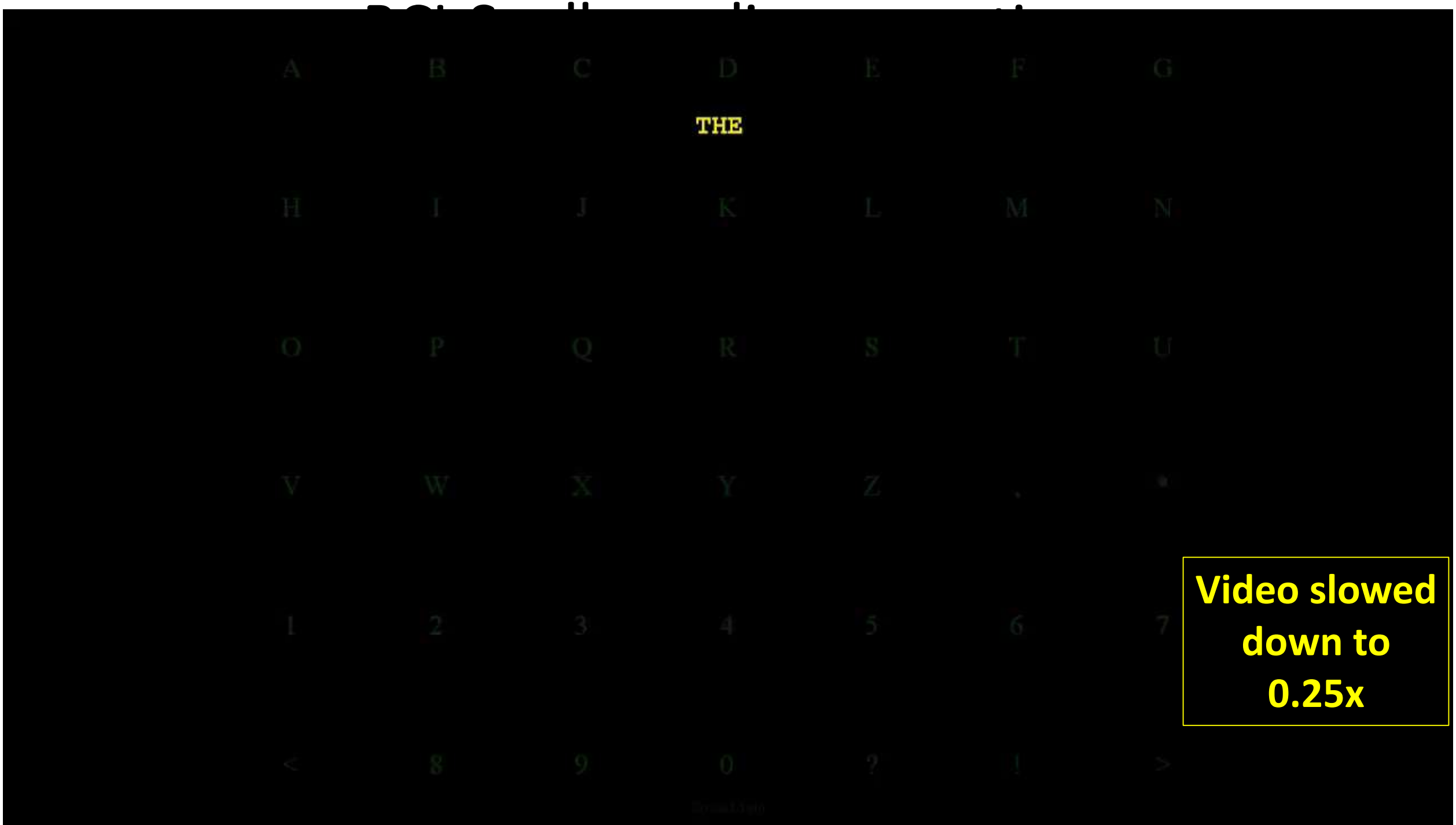
2. Online task: They were then asked to *correctly* copy-spell:
   THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG*
   over 1-3 online sessions per speed on each day.

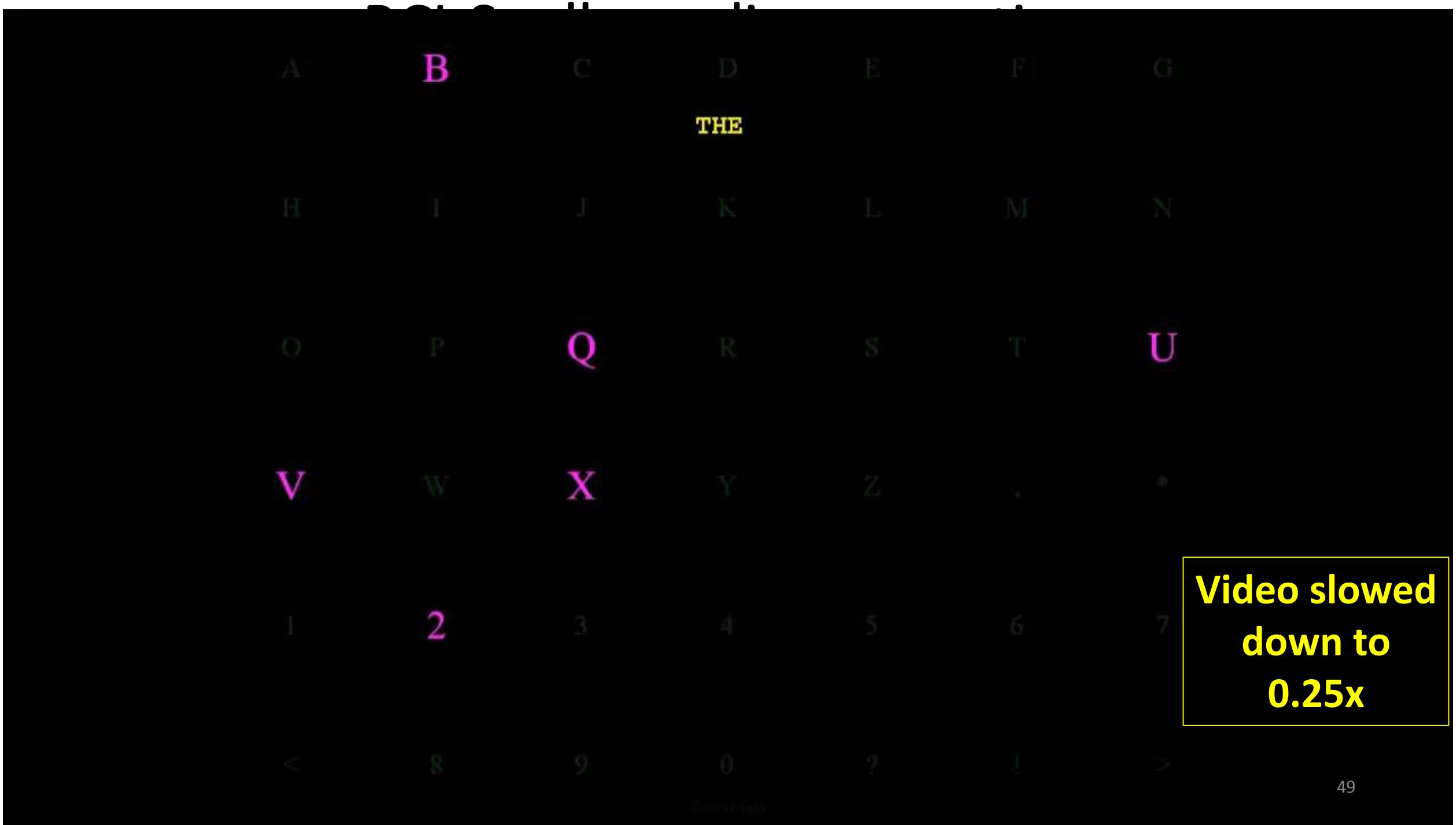
Average±s.d.

Best

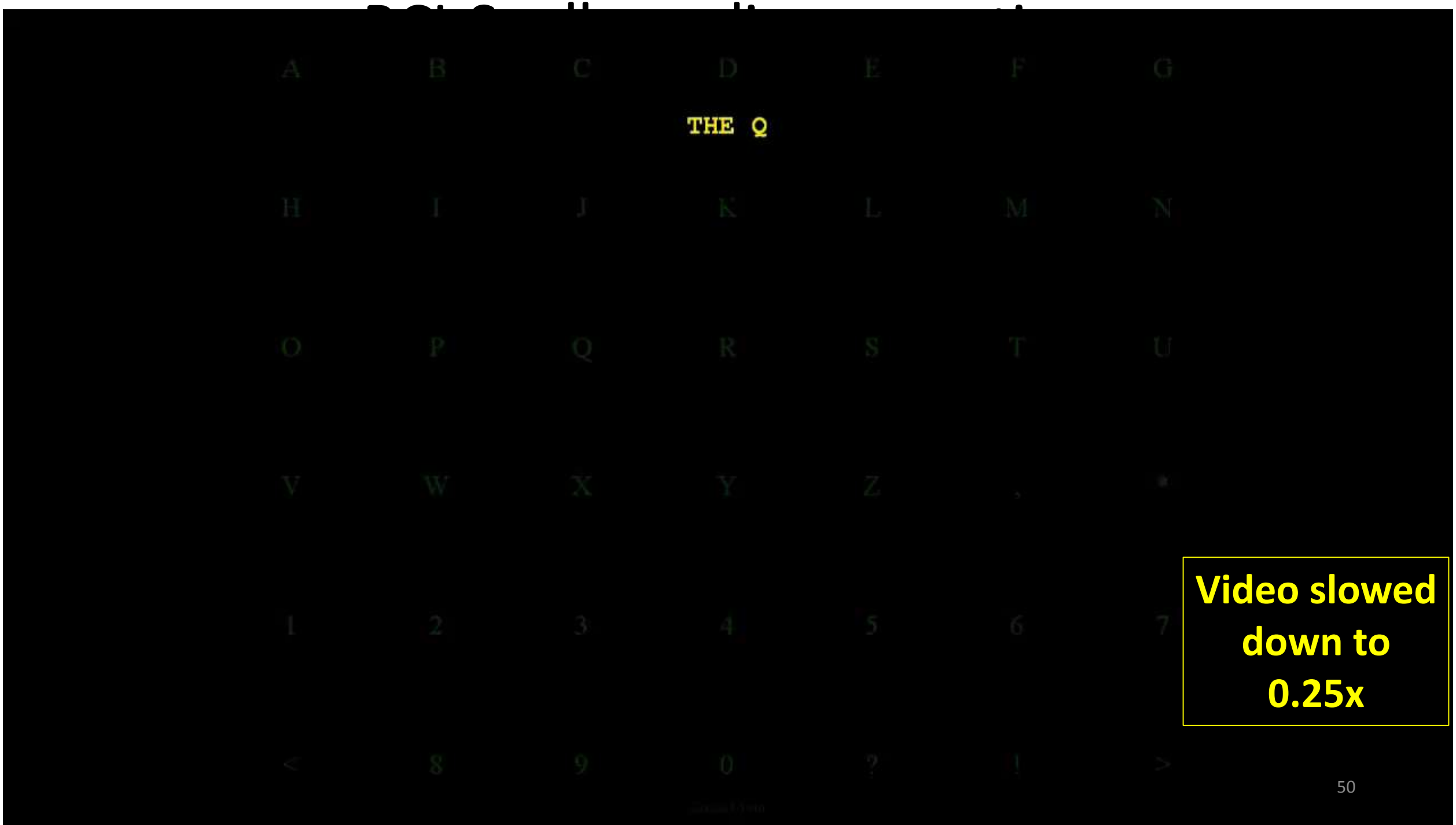
Subject	Interface speed (ITI, ms)		
	400	240	160
A	95.7±1.5% 97.0%	94.1±1.1% 94.9%	93.4±1.6% 94.4%
B	95.2±1.9% 97.4%	93.9±2.5% 96.2%	94.3±2.9% 96.6%
C	93.9±2.5% 96.6%	93.7±2.8% 96.9%	93.5±1.9% 95.5%
D	91.2±0.8% 92.1%	90.5±0.2% 90.7%	91.1±1.0% 92.2%
E	91.2±1.5% 92.3%	92.8±2.2% 95.1%	90.5±2.3% 92.4%
F	94.2±0.9% 95.3%	93.3±0.9% 94.2%	93.8±0.3% 94.0%

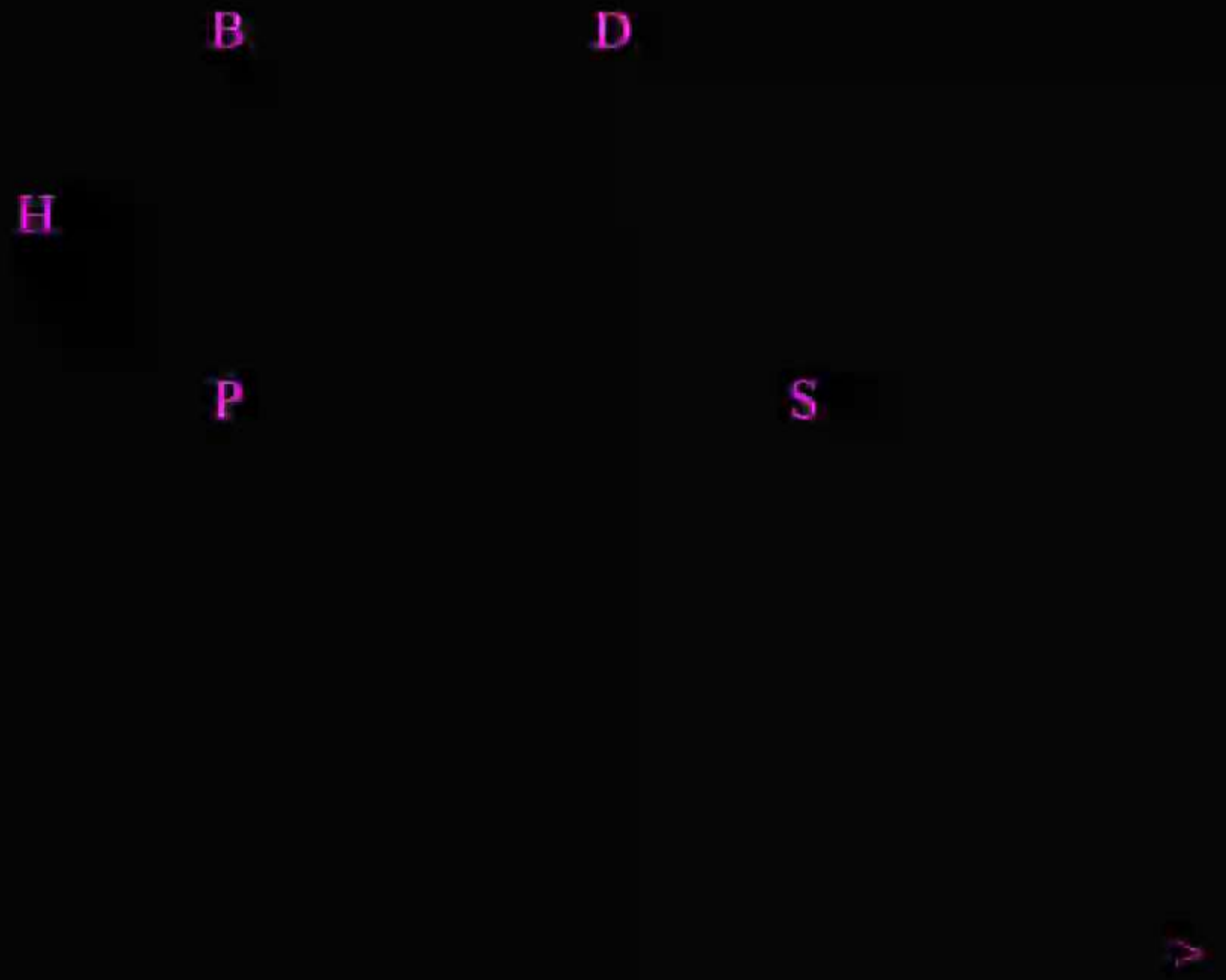
* marks personal best and ** overall best.


A B C D E F G
THE
H I J K L M N
O P Q R S T U
V W X Y Z . *
1 2 3 4 5 6 7
< 8 9 0 ? ! >
Video slowed down to 0.25x



Video slowed down to 0.25x


THE Q
Video slowed down to 0.25x


B
D
H
P
S

Information transfer rate

Information transfer rate

$$ITR = \frac{N_c}{T} \log_2 |A|$$

- $|A| = 42$ (number of different characters that can be chosen)
- N_C = number of characters correctly entered (44 for the quick brown fox…)
- T = time taken (seconds)

Useful when comparing between different speller designs (more/less characters on screen).

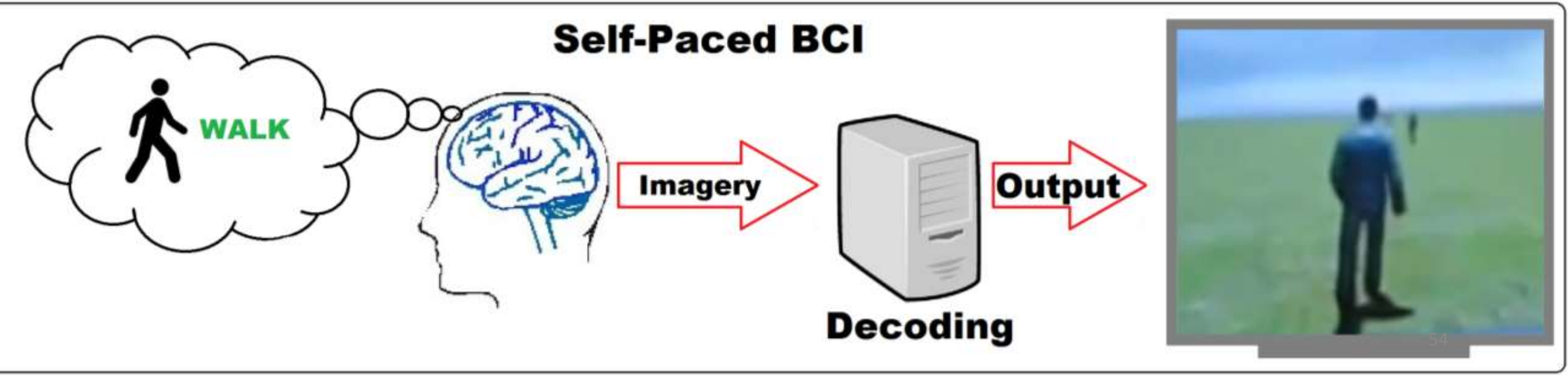
Note that T excludes the 3-s pause times.

Input design	ITR range (bit/s)
Our BCI Speller (4.7 s/char – 10.7 s/char)	**1.77 – 3.04** (4 – 7 wpm)
Other BCI spellers	0.22 – 1.03 (0.5 – 3 wpm)
Typing by hand	9.01 – 12.9 (23 – 33 wpm)

BCI Gait

BCI Gait

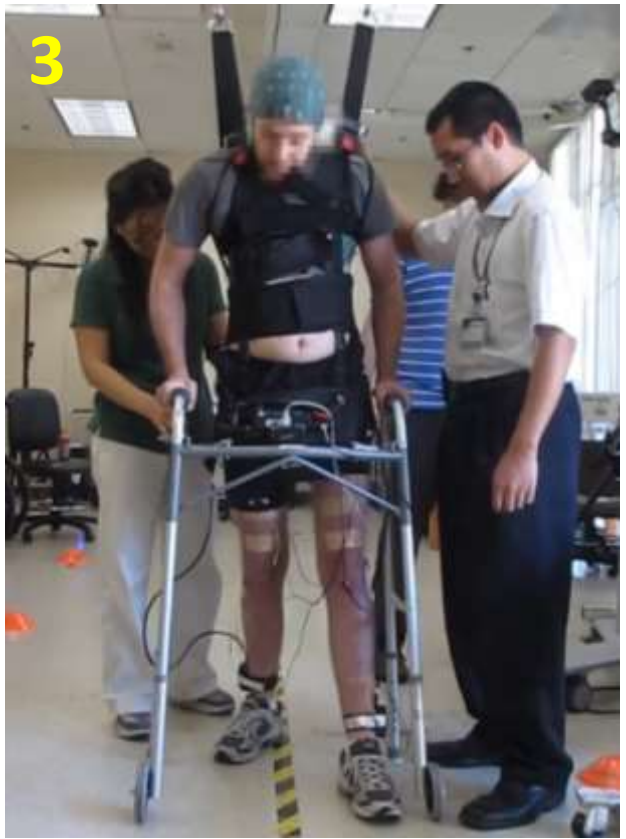
- BCI-Gait uses the changes in the sensorimotor rhythms (SMR) on the motor areas of the brain to control a walking device
- **Self-paced –** Instead of stimulus-response, the user changes their own brain signals at their own pace to control the BCI in real time
- Idling and walking kinesthetic motor imageries (KMI) – intuitive control strategy



BCI Gait

Three stages of implementation

1. **BCI-Avatar**: Control a virtual reality walking simulation
2. **BCI-RoGO**: Control a robotic gait orthosis over a treadmill
3. **BCI-Parastep**: Control a functional electrical stimulation (FES)-based walking prosthesis



BCI Gait experiment protocols

BCI-Avatar	BCI-RoGO	BCI-Parastep
1. Training procedure	1. Training procedure	1. Training procedure
2. Build classifier	2. Build classifier	2. Build classifier
3. Calibration	3. Calibration	3. Calibration
4. Online control of an avatar in a video game	4. Online control of a robotic gait orthosis	4. Online control of a FES-based walking prosthesis

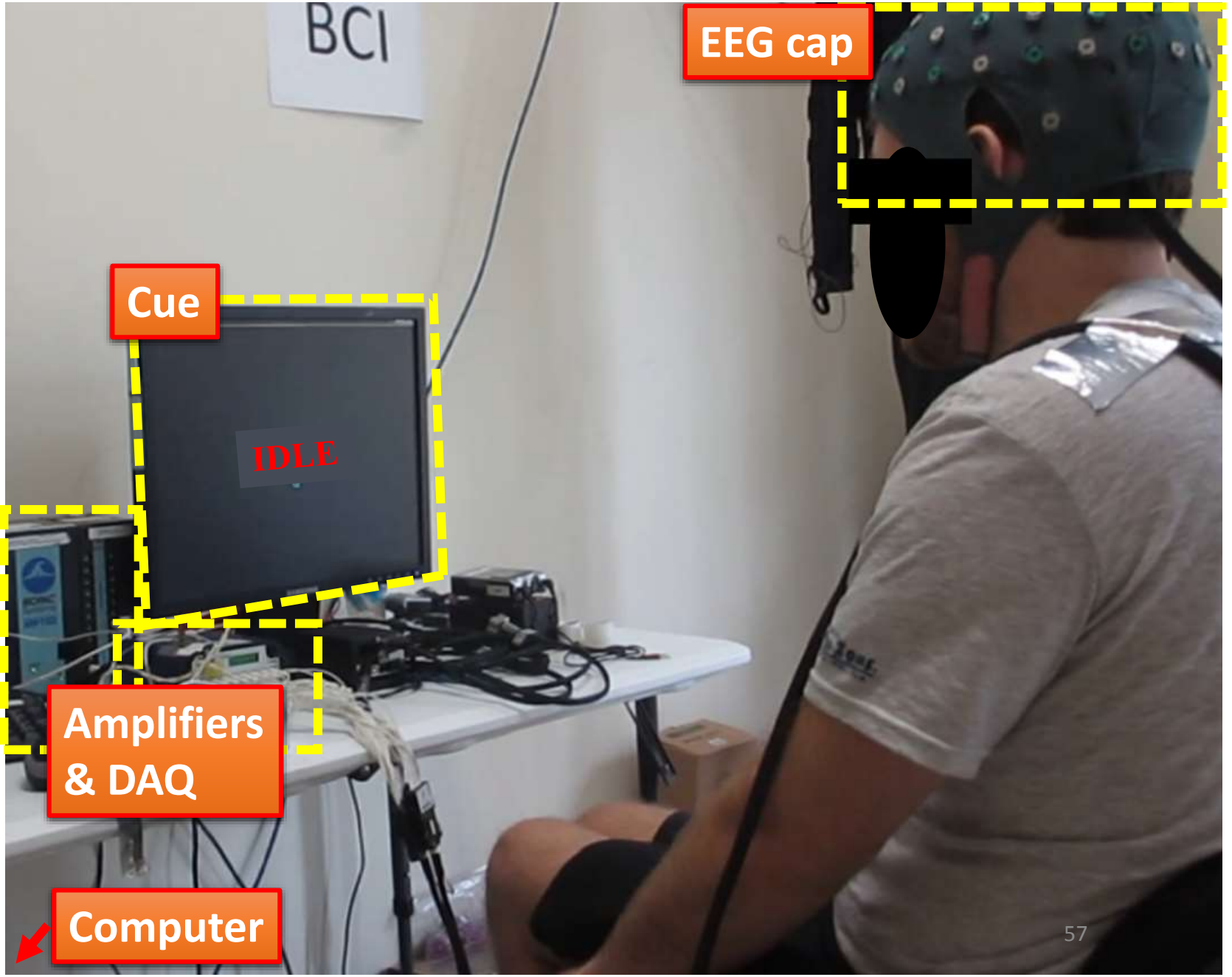
Parastep® (Sigmedics Inc., Fairborn, OH)

Training procedure – protocol

10 x

1. Cue `Walk' (30 s)
   Kinesthetic motor imagery (KMI) of walking
2. Cue `Idle' (30 s)
   Imagine idling

(total 10 minutes)



Subject demographics (BCI-Avatar only)

Table 6.1: List of participants with demographic data and prior BCI experience relevant to the task. SCI status scored according to American Spinal Injury Association (ASIA) Impairment Scale.

Subject	Gender	Age	BCI experience	SCI status
A1	M	40	~1 hr	–
A2	M	29	~1 hr	–
A3	F	23	~1 hr	–
A4	F	57	0 hr	–
A5	F	24	0 hr	–
A6	M	21	0 hr	–
A7	M	25	0 hr	–
A8	M	32	0 hr	–
S1	F	27	0 hr	T8 ASIA B, 11 yr post injury
S2	M	34	0 hr	T11, ASIA A, 8 yr. post injury
S3	M	46	0 hr	T1, ASIA B, 4 yr. post injury
S4	M	43	0 hr	C5, Syringomyelia, 14 yr. post onset
S5	M	59	0 hr	T1, ASIA B, 2 yr. post injury
S6	M	21	0 hr	T11, ASIA B, 1 yr. post injury

Able-Bodied

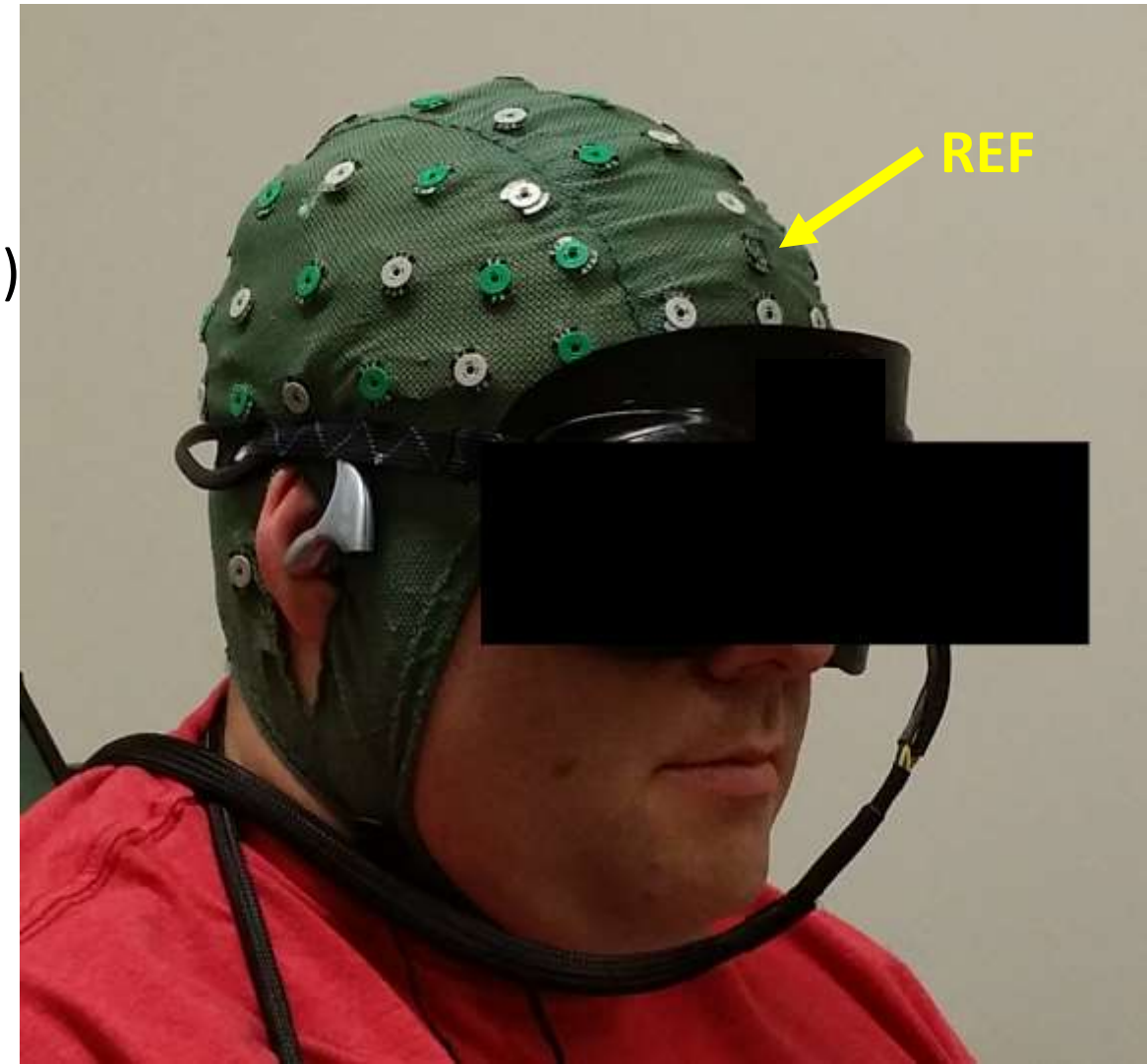
Paraplegia, tetraplegia

Equipment

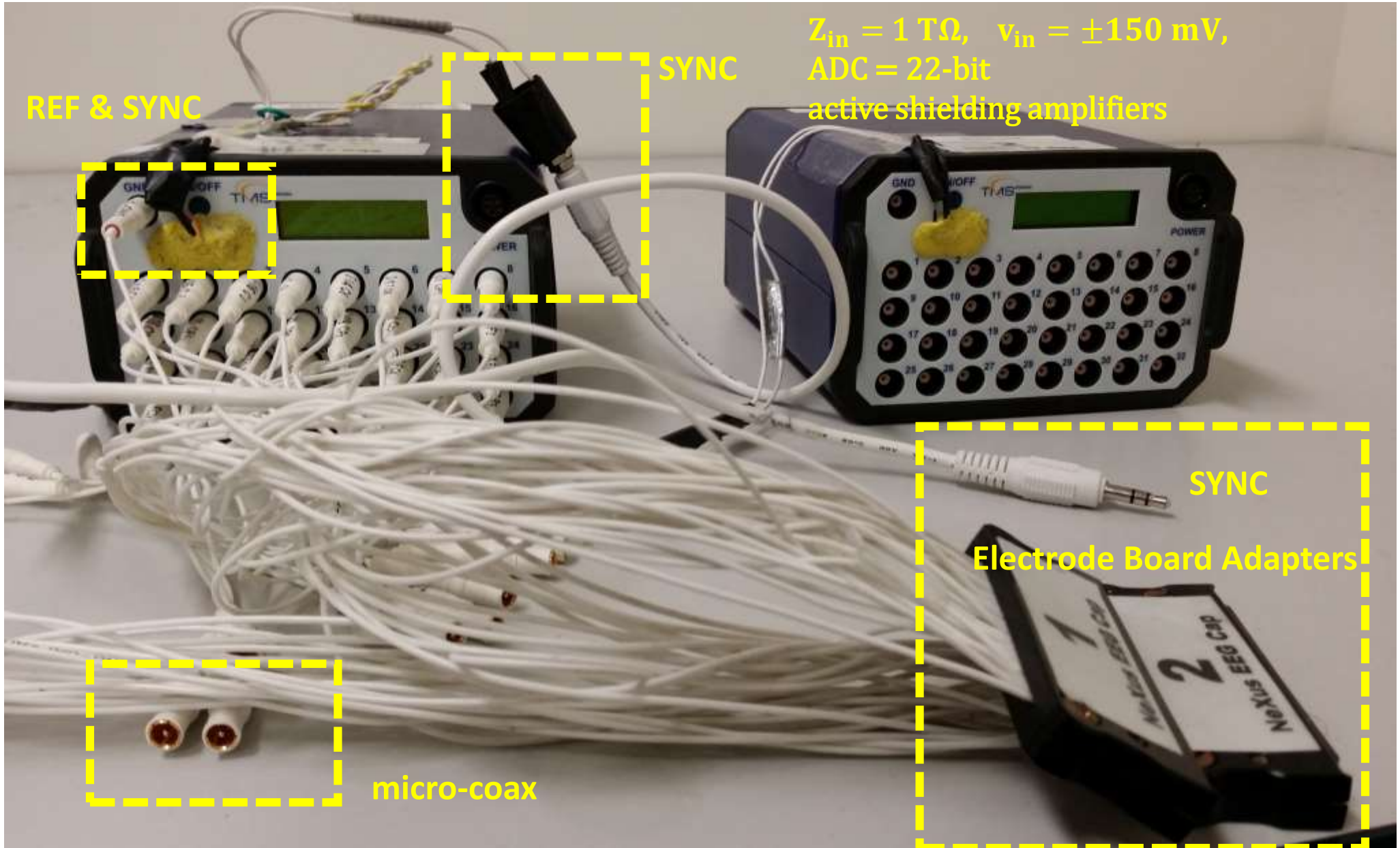
- **EEG cap**: NeXus EEG Cap
  (Medi Factory, Heerlen, The Netherlands)
- **EEG reference**: Between AFz and FPz electrodes
- **EEG amplifiers**: NeXus-32
  (MindMedia, Roermond-Herlen, The Netherlands)
- **EEG data acquisition device (DAQ)**: Same as EEG amplifiers
- **DAQ to receive cue signal:** MP150
  (Biopac Systems)

EEG: Sampled at 256 Hz

Cue: Sampled at 2000 Hz



NeXus EEG amplifiers and DAQ



Timing and synchronization

Problem:

EEG and textual cues are not recorded together.

EEG amps have no aux ports.

How to label ***Idle vs. Walk*** segments on EEG?

Solution:

1. A **timing signal** from the computer
   Walk cue = Audio tone
   Idle cue = No audio
2. A **synchronization signal** between NeXus EEG and MP150

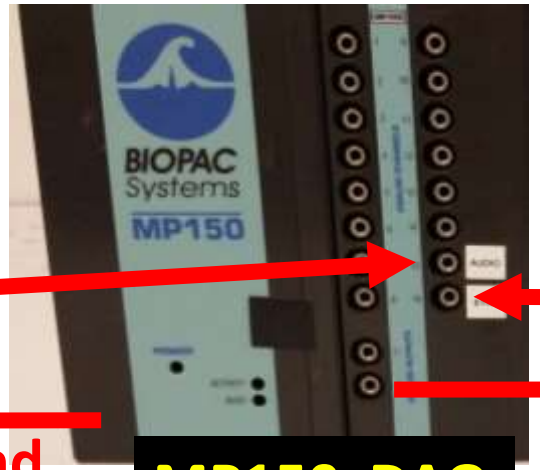
Timing and synchronization

Problem:

EEG and textual cues are not recorded together.

How to label ***Idle vs. Walk*** segments on EEG?

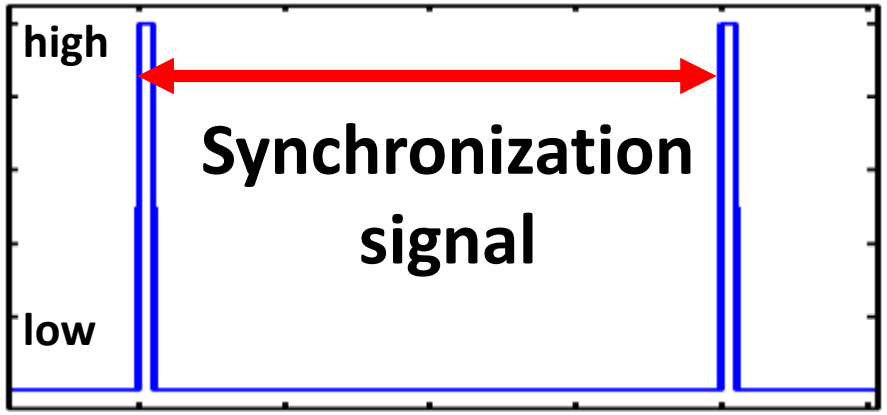
Cue and timing signals



Sync

Timing and sync signals

MP150 DAQ

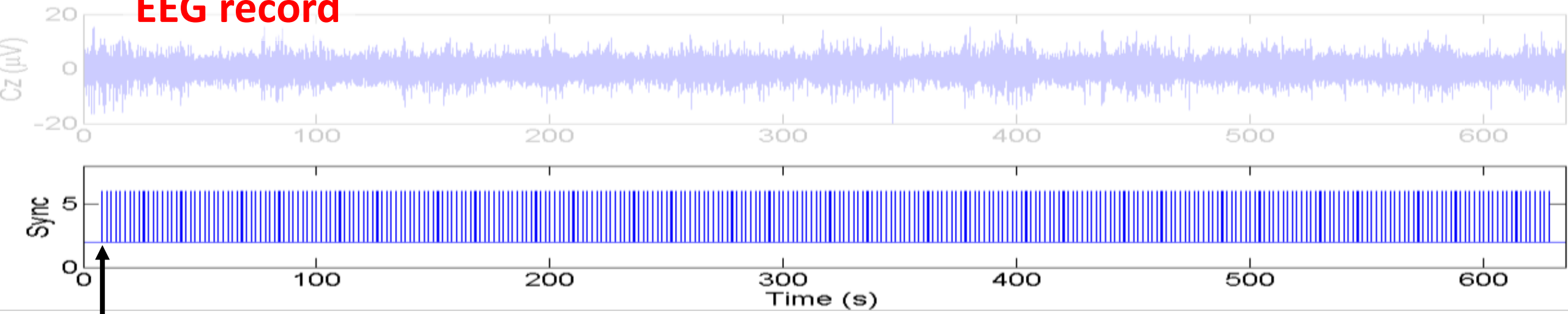


EEG and sync signals (optical fiber)

NeXus EEG amplifiers

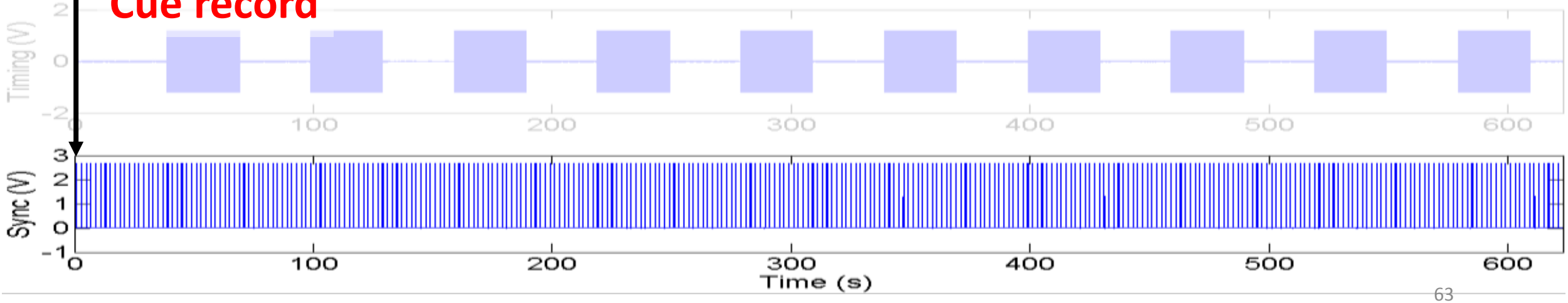
Timing and synchronization

EEG record

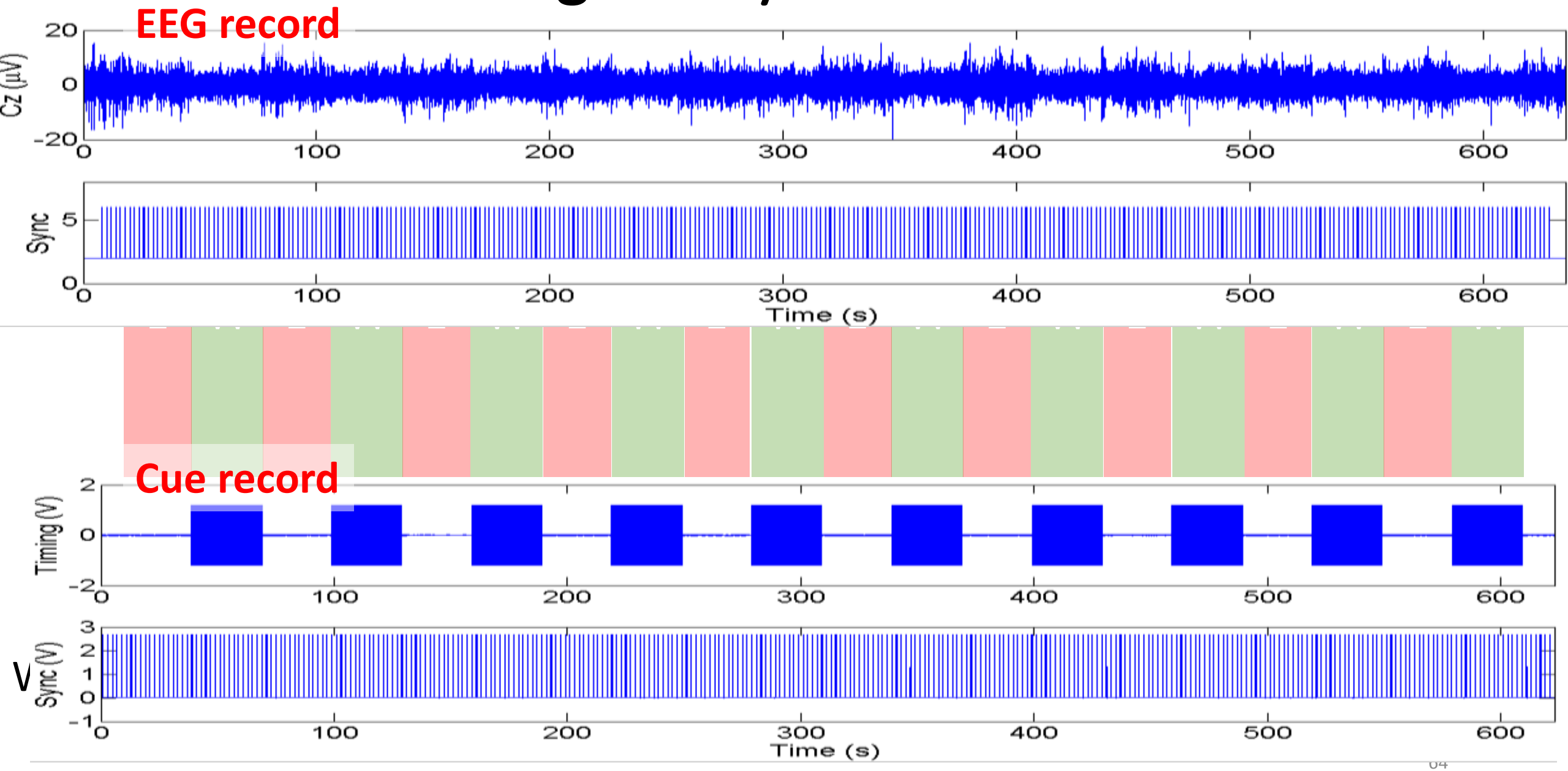


The first square pulses are aligned on both records

Cue record

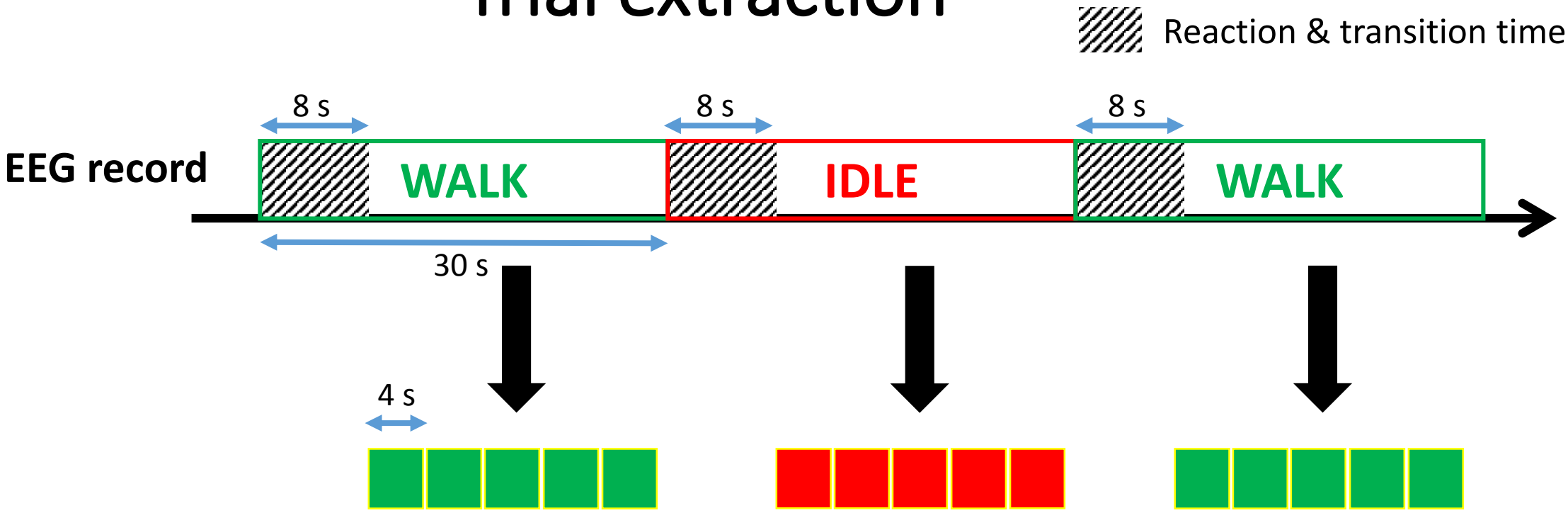


Timing and synchronization



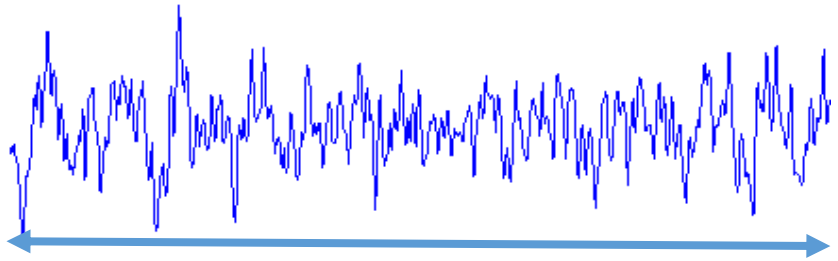
V

Trial extraction

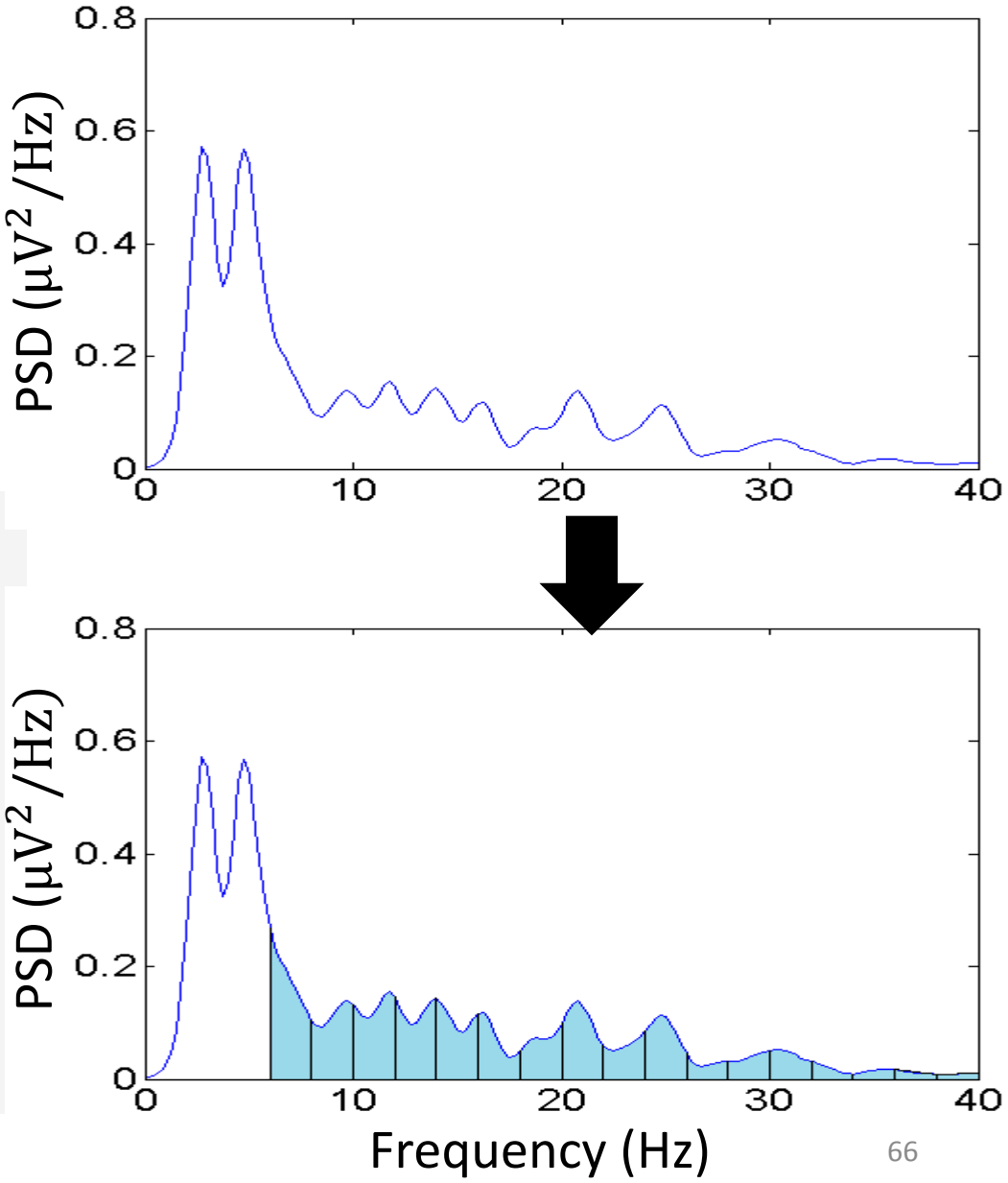


1. EEG record is divided into 30-s Idle and 30-s Walk segments
2. The first 8 s in each segment are discarded
3. Remaining segments are further split into five 4-s small segments
4. A total of 50 Idle trials and 50 Walk trials are extracted

Data processing

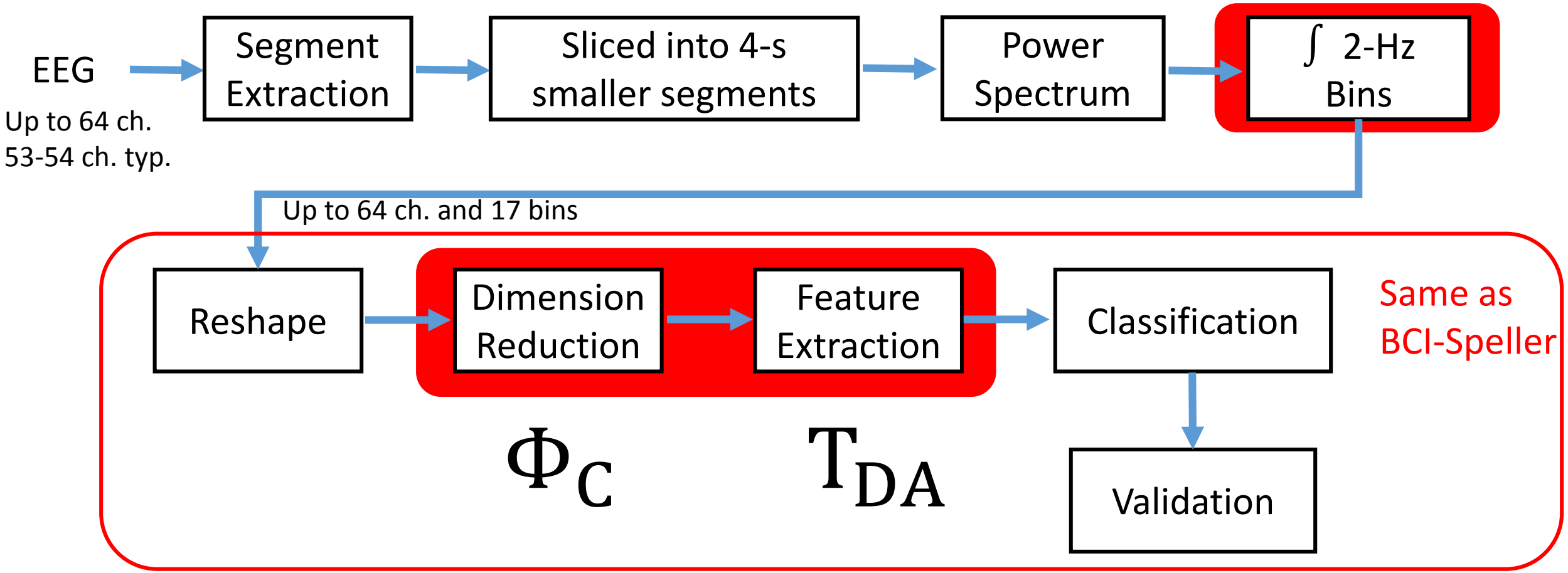


FFT

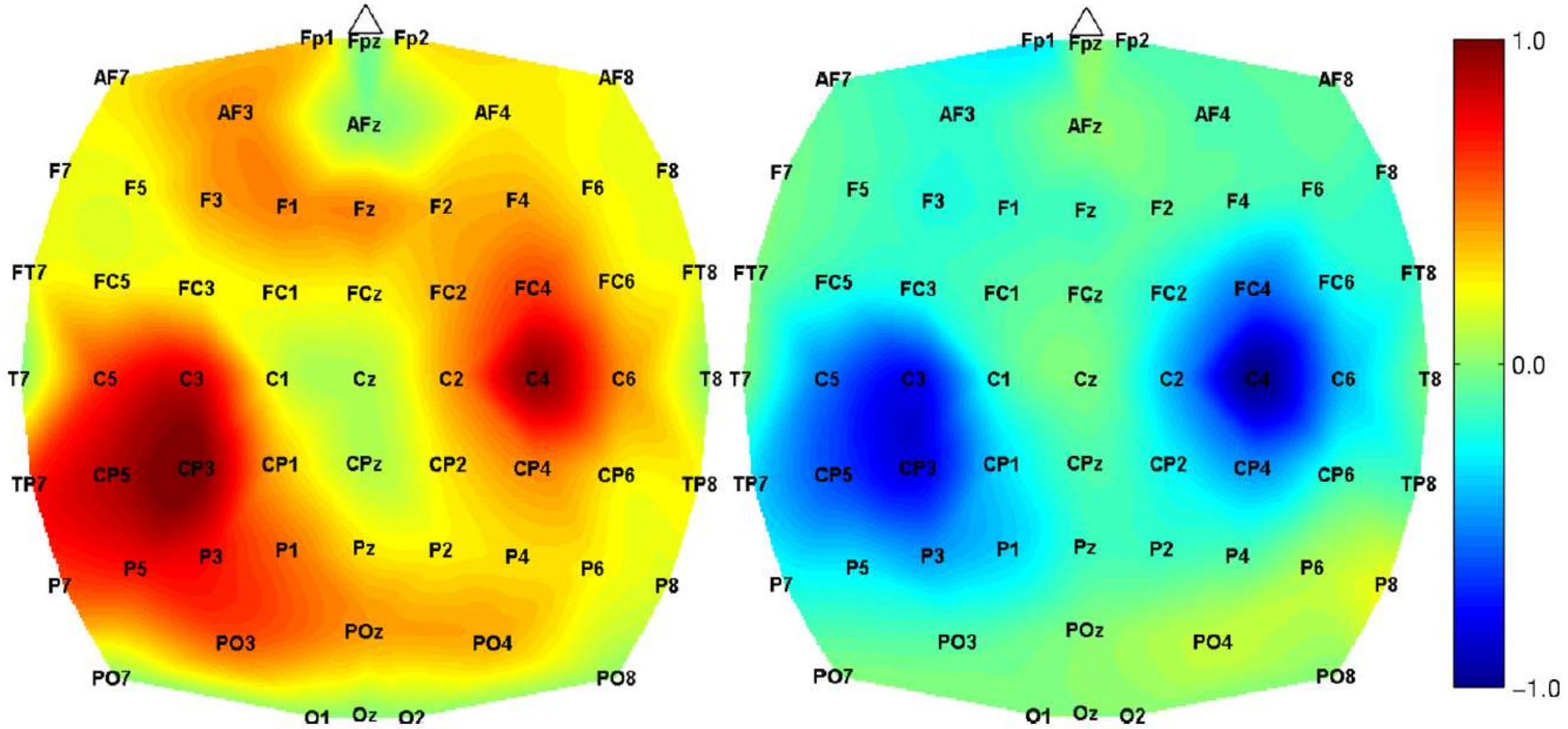


1. Each small segment is transformed to frequency domain using Fast Fourier Transform (FFT)
2. Power spectrum is integrated over 2-Hz bins, centered around 7, 9, … 39 Hz.
3. Resulting in 17 sample points per channel.

Feature extraction, etc.

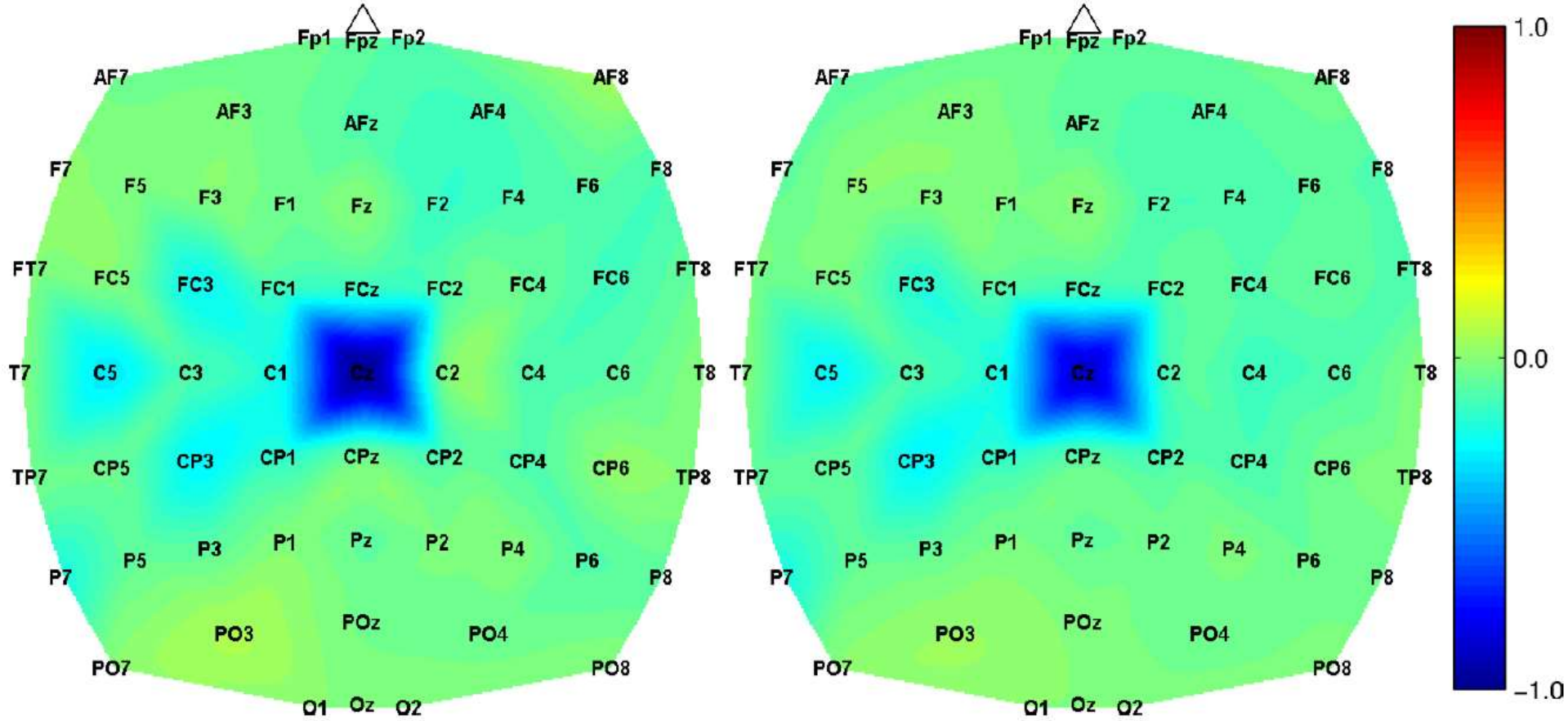


Feature extraction filter (AB)



Feature extraction filter in the 12-14 Hz bin for Subject A2 (29M).
Higher magnitudes (deep blue/red) = more important.
The CPCA technique generates one filter for each class. Left: Idle subspace, Right: Walk subspace.
Training performance: 86.6% (50% chance).

Feature extraction filter (SCI)



Feature extraction filter in the 14-16 Hz bin for Subject S1 (27F, SCI, T8, ASIA B, 11 yr post)

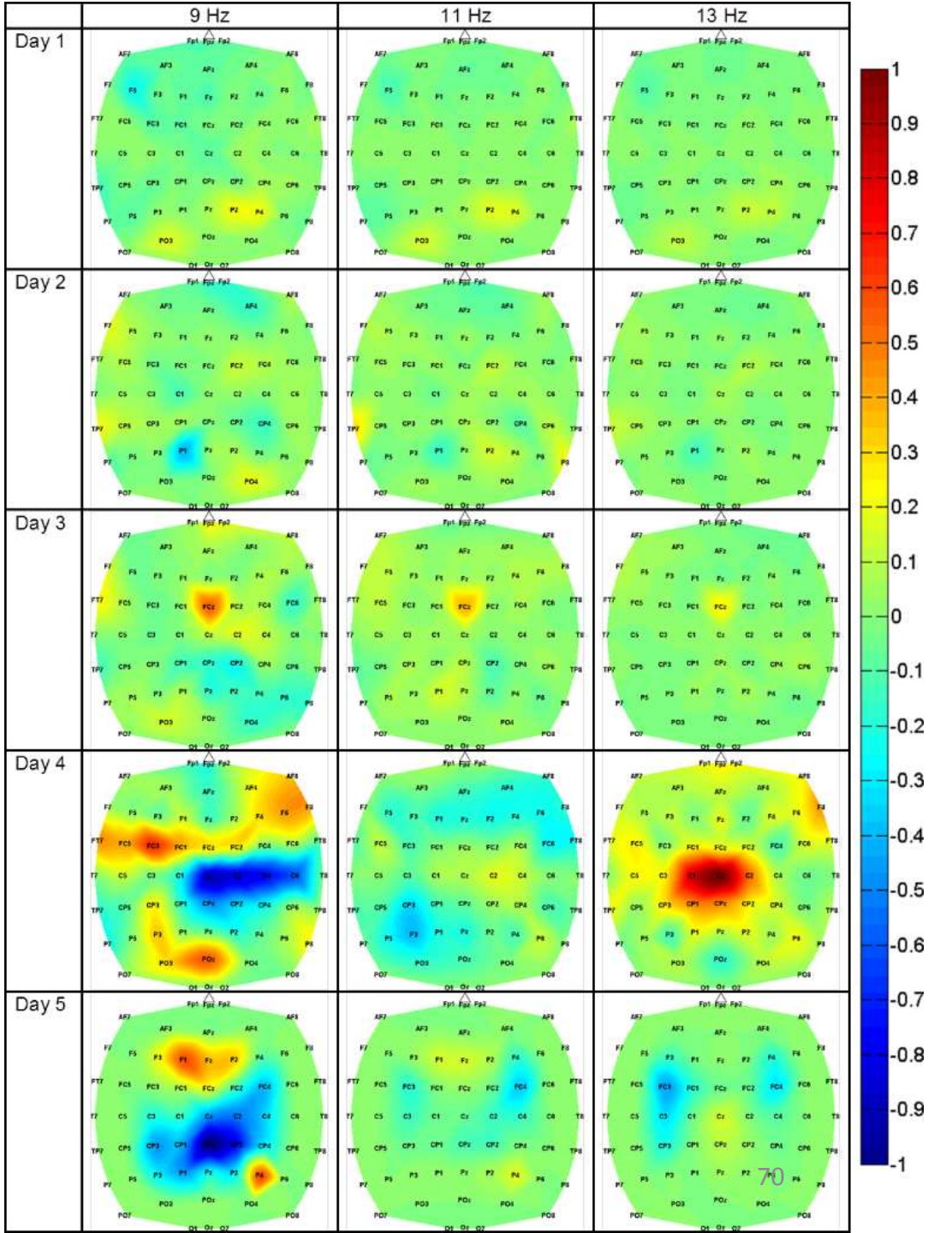
Training performance: 94.5%

Relearning process?

The FE filters for Subject S3 (46M, SCI, T1, ASIA B, 4 yr post) evolved over the 5 days.

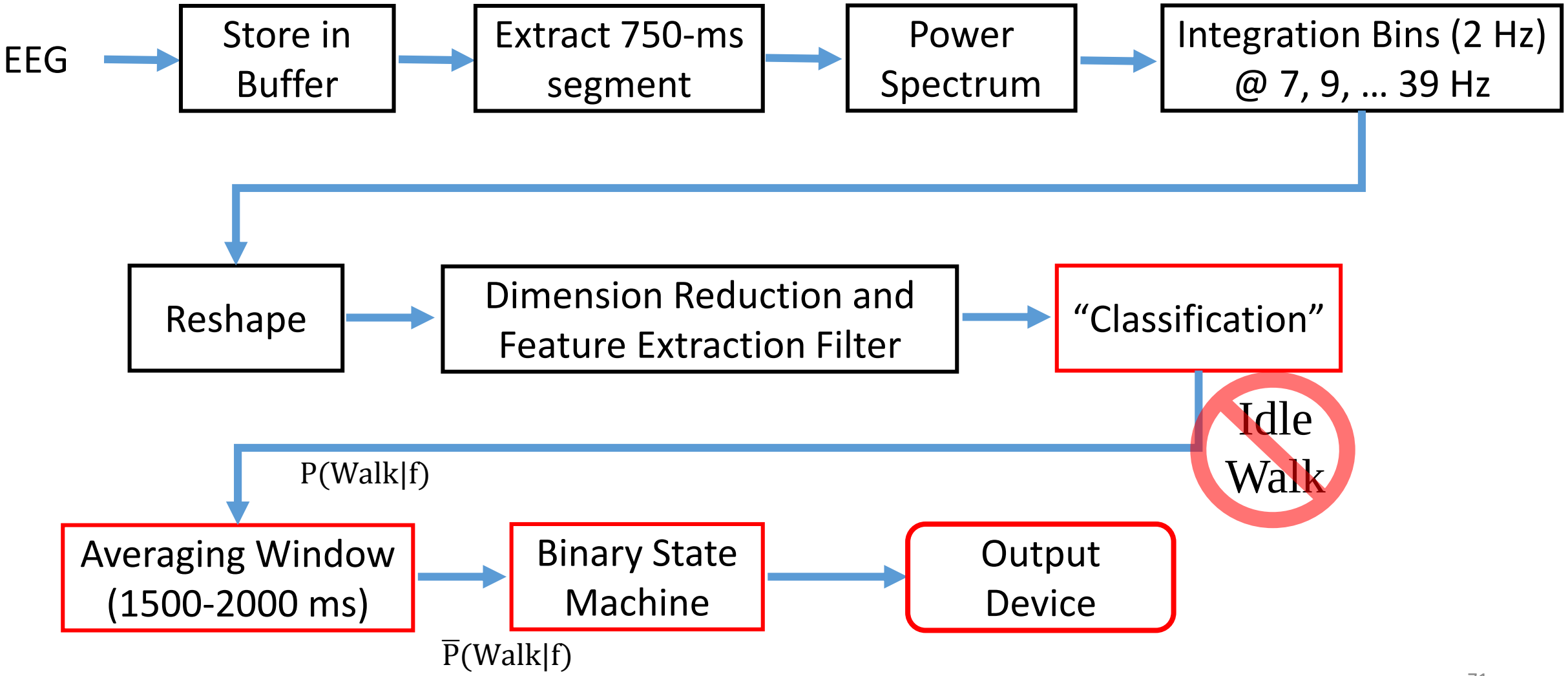
- No strong features in μ band in the first 3 days
- On the 4th day, foot and left hand representation areas "lit up"
- Training performance jumped from 62% to 92%

Day	$P(\text{correct}\,\vert f^\star)$	p-value	RC	Freq. band
1	$62.2 \pm 1.8\%$	6.00×10^{-3}	53	4-40 Hz
2	$62.0 \pm 1.8\%$	1.05×10^{-2}	53	4-40 Hz
3	$60.5 \pm 2.0\%$	1.76×10^{-2}	53	6-40 Hz
4	$91.6 \pm 1.7\%$	1.60×10^{-19}	54	4-40 Hz
5	$82.5 \pm 1.6\%$	6.55×10^{-12}	$25^\dagger$	6-20 Hz
Avg.	$71.8 \pm 14.3\%$	6.82×10^{-3}		

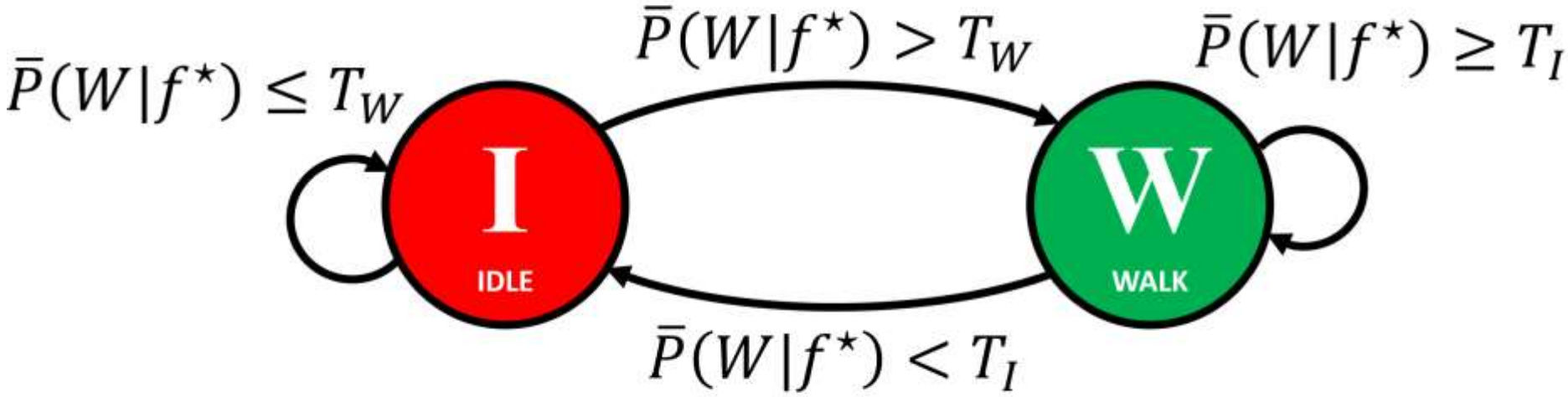


BCI Gait: Online procedure flowchart

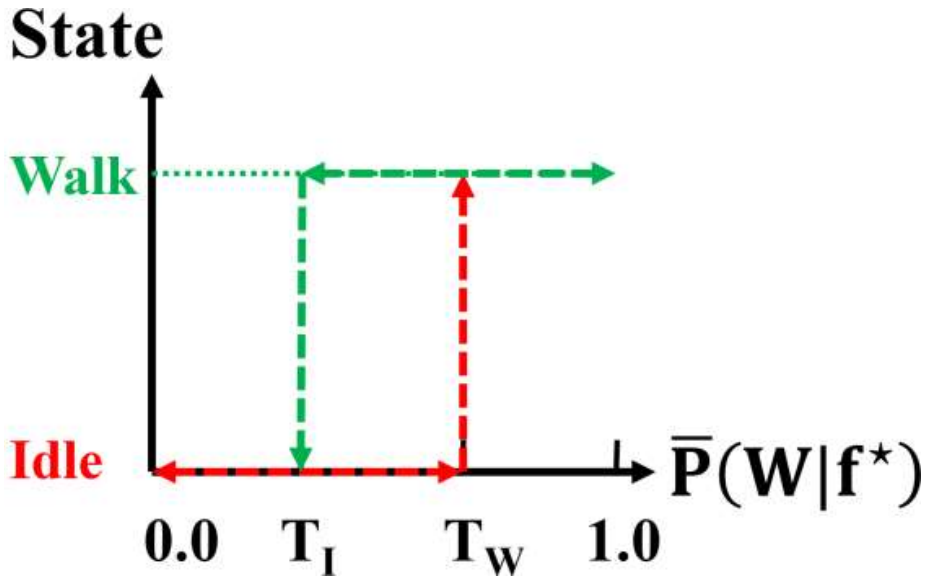
Every 500 ms, these steps are performed:



Binary state machine



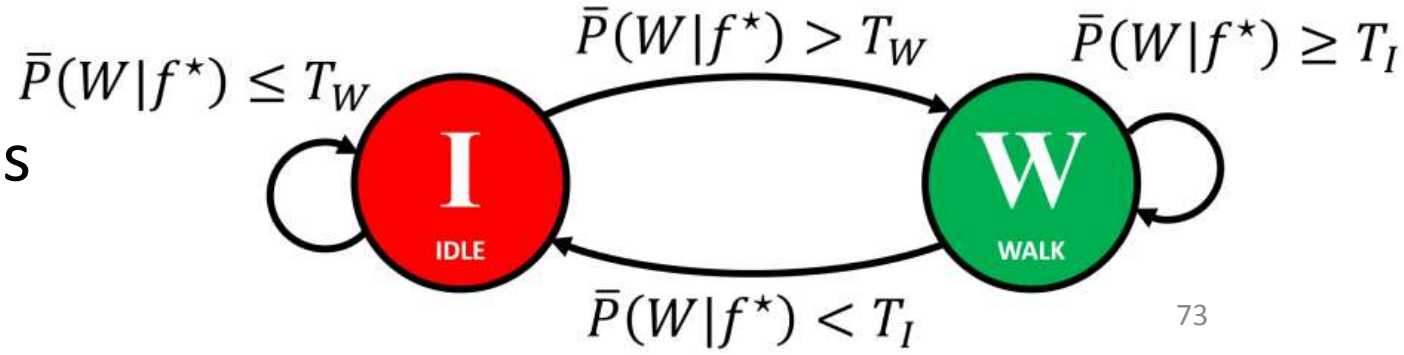
1. **Input**: P(W|f*) averaged over 1.5-2.0 s
   Output: Idle or Walk, controls the output device
2. Two parameters:
   Idle Threshold (T_I)
   Walk Threshold (T_W)
3. Concentrated effort vs. maintenance of imagery



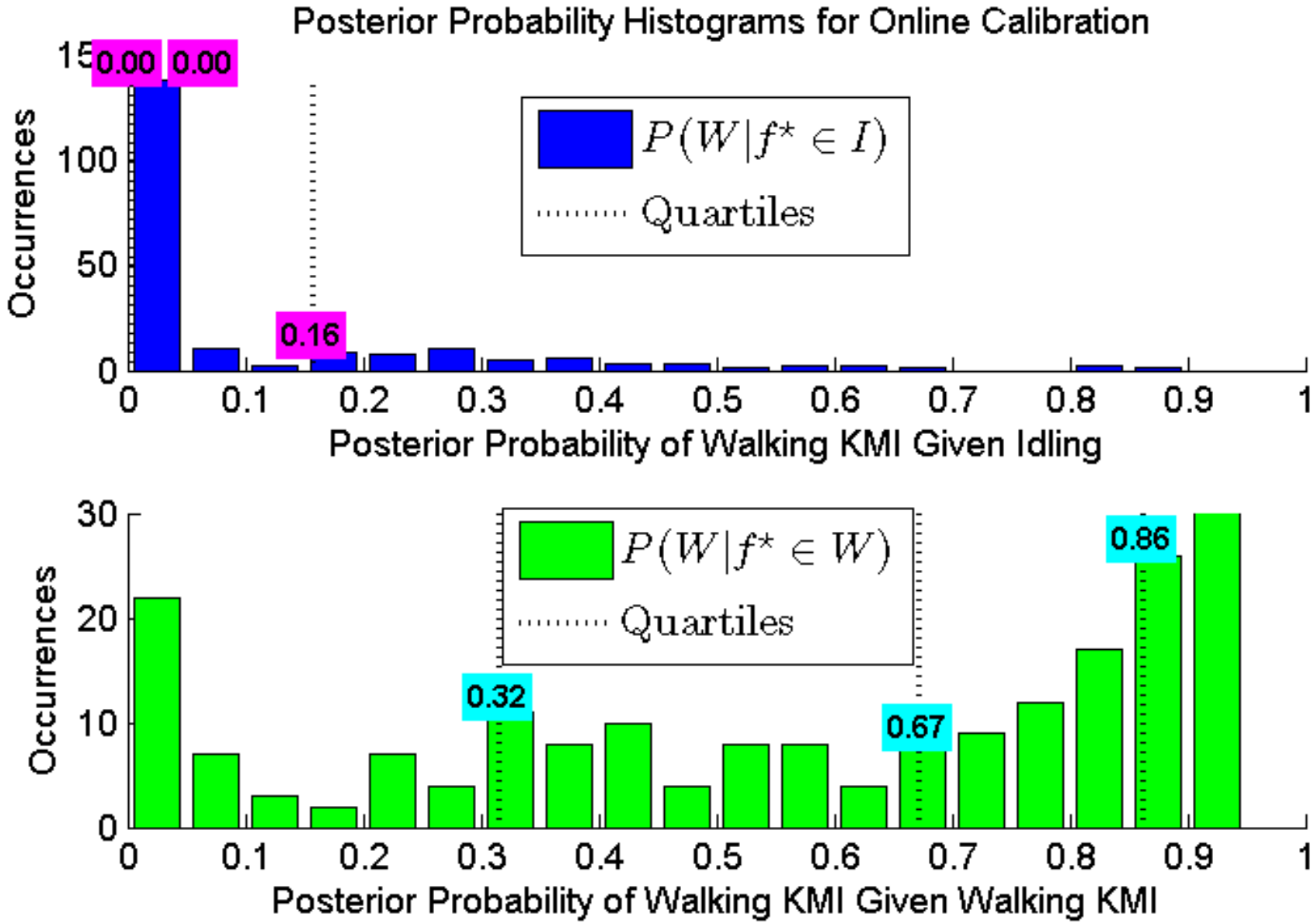
Calibration procedure: determine $\mathbf{T_I}$ and $\mathbf{T_W}$ Steps

1. The BCI is put into online mode, with the following exceptions:
   - The output device is disconnected
   - The subject follows verbal cues from an experimenter
2. The experimenter instructs the subject to perform idling and walking KMI
   - The timing of each KMI is noted
3. $\bar{P}(W|f)$ from the calibration session is split into $\bar{P}(W|f \in I)$ (when instructed to idle) and $\bar{P}(W|f \in W)$ (when instructed to imagine walking).
4. Histograms are plotted.

- Calibration procedure ~ 4 minutes



Calibration histograms



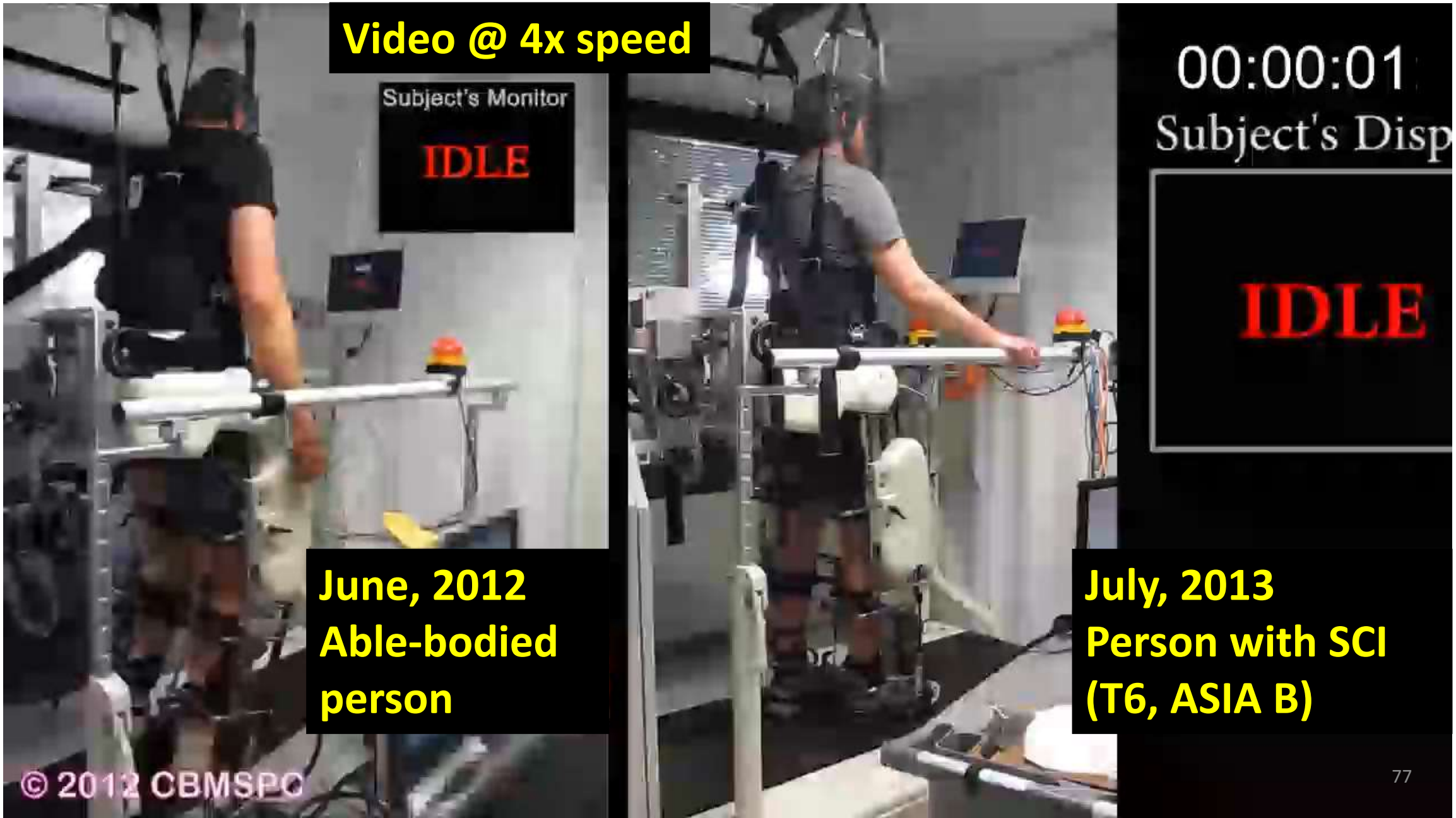
BCI Avatar: Online task

BCI Avatar: Online performance

Subj. S1 (Mar, 2010) SCI (T8, ASIA B, 11 yr post)
(Live demo, Reeve-Irvine Research Center)

Subj. S4 (Sep, 2012) Chiari malformation (C5, 14 yr post)

BCI RoGO: Online performance



Conclusions and future works

Review of objectives

- Goal: Create high-performance BCI using novel data processing techniques
- Specifically: Apply non-linear (piecewise linear) feature extraction, classification, and validation techniques to develop BCIs
- Result:

Design goal	BCI-Speller	BCI-Gait
Short training time	Yes (6-7 min)	Yes (10-15 min)
Intuitive control	Yes	Yes
High performance	Yes (4-5 s/chr)	Yes (high correlation)
Adaptive	(not tested)	Yes (AB, paraplegia, tetraplegia)
Versatile	(not pursued)	6+ output devices

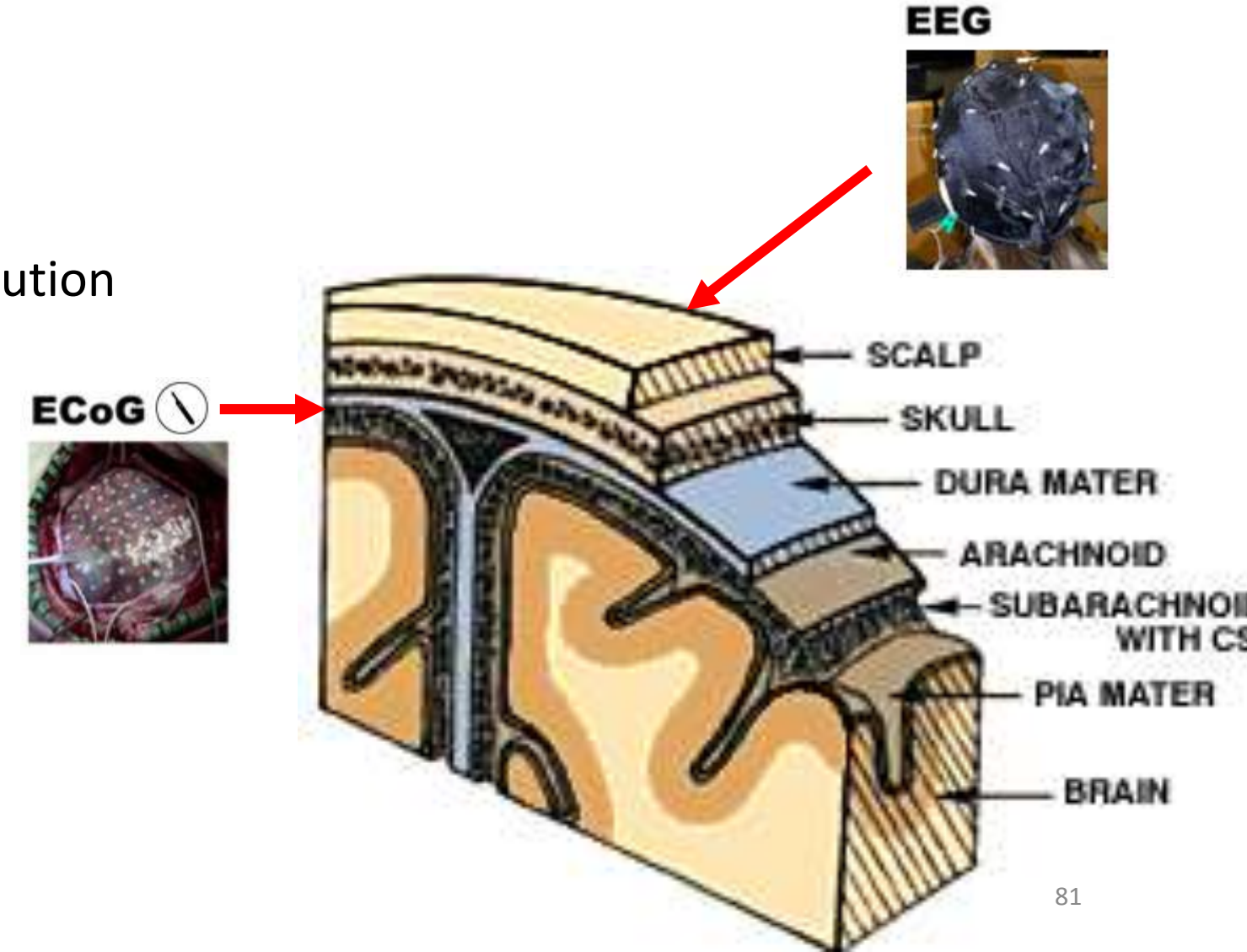
BCI-Gait: Overground walking

- Training time:
  - 17 physiotherapy sessions
  - 7-11 BCI-Avatar training sessions
- Performance:
  - Offline: 100%
  - Online: $\rho > 0.9$

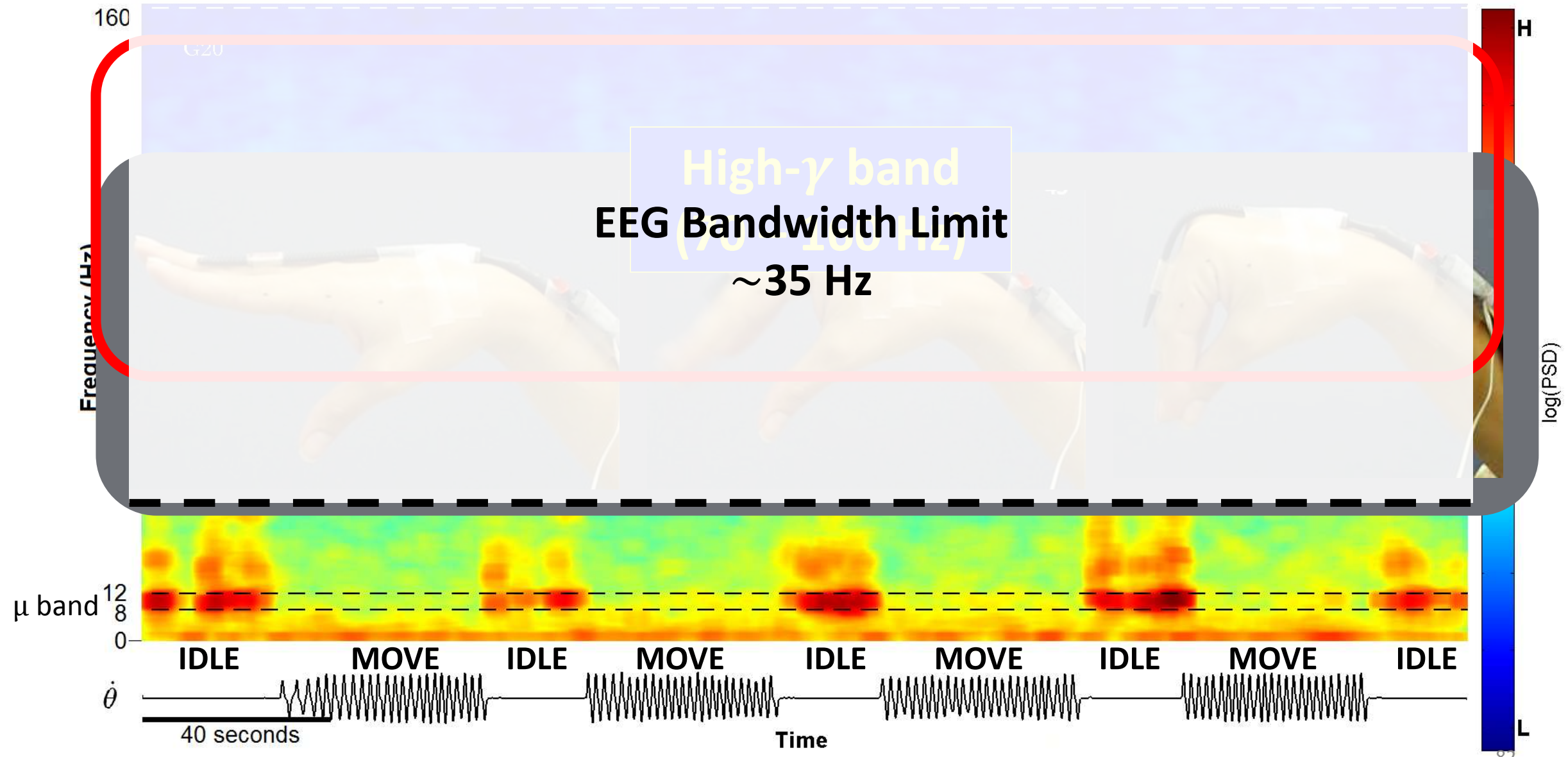
iMove Lab

Electrocorticography (ECoG)

- EEG based system
  - EEG cap
  - Artifacts
  - Dries up
  - Bulky equipment
  - Limited bandwidth and resolution
- ECoG
  - Record underneath the dura
  - Better spatial resolution
  - Higher bandwidth
  - Long-lasting
  - Immune to artifacts

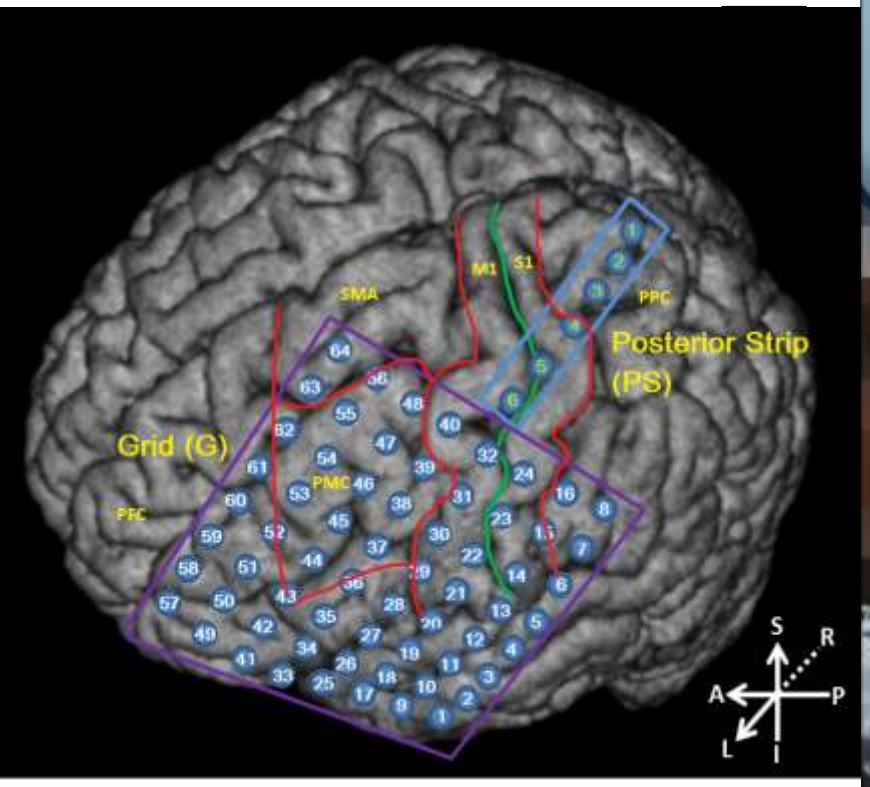


ECoG: Better bandwidth and SNR



ECoG robotic arm control

- State decoder using the same BCI-Gait software
- Trained in 3 minutes



Future works

Fully-enclosed ECoG-based BCI implant

- Bio-compatible
- Long lasting
- Minimal infection risk
- Better recording resolution
- More accurate decoding

Long-term solution to restore functional independence for people with severe paralysis

Acknowledgements

CBMSPC
Center for BioMedical Signal Processing and Computation

Zoran Nenadic, D.Sc.

An H. Do, M.D.

Christine E. King

Andrew Schombs

Aggie A. F. Szymanska

Colin M. McCrimmon

Everardo Camacho Jr.

Former/rotation members

Ahmad Abiri

Sergey Osechinskiy

RANCHO LOS AMIGOS
NATIONAL REHABILITATION CENTER

Rehabilitation Engineering
Philip Requejo, Ph.D.

Neurology
Susan J. Shaw, M.D.
David E. Millett, M.D.

Neurosurgery
Charles Y. Liu, M.D.

Neurodiagnostic Technician
Angelica Nguyen
Christel Jean

Facilities and equipment
Long Beach Veterans Affairs Medical Center
Robotic gait orthosis

UCIRVINE

Biology
Debra Mauzy-Melitz, Ph.D.

Mech. Aerospace Eng. & Biomed. Eng.
David J. Reinkensmeyer, Ph.D.

Neurology
Luis A. Chui, M.D. (ret.)
Steven C. Cramer, M.D.
Jack J. Lin, M.D.
Mona Sazgar, M.D.

Neurosurgery
Frank P. K. Hsu, M.D.

Physical Therapy
Cathy C. Y. Chou

Reeve-Irvine Research Center
Oswald Steward, Ph.D.
Kelli Sharp, D.P.T.

Funding

Roman Reed Foundation
hope for spinal cord research

NSF

SCIRE

Spinal Cord Injury Fund