Brain-Computer Interfaces

Chaklam Silpasuwanchai

Asian Institute of Technology

chaklam@ait.asia, chaklam.com

Overview

Sources

- https://sccn.ucsd.edu/wiki/Introduction_To_Modern_Brain-Computer_Interface_Design
- Delorme, A., Sejnowski, T., and Makeig, S. (2007). Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. Neuroimage, 34(4), 1443-1449.
- Malmivuo, J., and Plonsey, R. (1995). Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields. Oxford University Press.
- Rashid, M., Sulaiman, N., Majeed, A. P. A., Musa, R. M., Nasir, A. F. A., Bari, B. S., and Khatun, S. (2020). Current Status, Challenges, and Possible Solutions of EEG-Based Brain-Computer Interface: A Comprehensive Review. Frontiers in neurorobotics, 14.

What is a BCI?

- the idea of using brain signals for communication and control.
- three main components: (1) **Signal acquisition** (e.g., EEG, fNIRS), (2) **Signal Processing** (i.e., feature extraction, classification), (3) **Signal Translation** (e.g., mapping the signal to particular usage).
- an interdisciplinary field concerning signal processing, machine learning, human-computer interaction, cognitive/neuroscience.

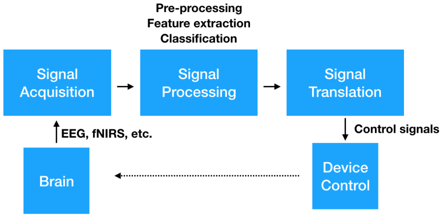


Figure: Basic procedure of a BCI

Underlying Neural Processes

- All BCIs have to operate on observable effects of brain activity
- Except for fMRI and fNIRS, they operate on effects of neural firing processes
- EEG, MEG, and ECoG can only detect *large-scale* neural dynamics
- For example, 50,000 neurons firing in near-synchrony

Underlying Neural Processes

- Largest contributors to the EEG are the pyramidal cells
- Radially oriented in the cortex (orthgonal to the surface)
- Electromagnetic fields of co-aligned and co-activated neurons add up
- `https://www.youtube.com/watch?v=AIvlNNFQLEk`

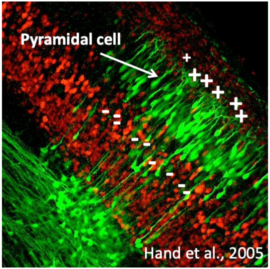


Figure: Hand et al., 2005

Large-Scale Neural Processes

- When would 50,000 neurons fire near-synchronously?
  - An external even triggers a cascade of related neural processes (e.g., in perception)
  - An internal event triggers a cascade of related neural processes (e.g., a sudden aha!)
  - Neural populations enter a synchronized steady-state firing pattern (e.g., idle oscillations)

Why BCI is so hard?

- The field of BCI has been established since the 1960s - 1970s with a lot of promise. However, contrary to the belief that "most obvious areas" are done (e.g., Neuralink), **BCI is still at its infancy.**
- Most of the time, BCI suffers from two key scientific challenges:
  1. **Variability** - individuals variability in brain signals. These variability is task-specific and user-specific. Thus all BCI systems must be calibrated before they can be used. Current research focuses on developing a transferable model of BCI that work across users.
  2. **Signal-to-noise ratio** - concerns the removal of artifacts while preserving the "weak" EEG signal. Effective signal processing and machine learning is key here. All approaches here are fundamentally statistical and computational.

Three main types of BCI (Zander et al. 2009)

1. **Active** BCI: A BCI of which users consciously control/manipulate their thoughts to control an application, e.g., *motor imagery*
2. **Reactive** BCI: A BCI of which output from brain activity arise from external stimulation, independent of users' conscious thoughts, e.g., *P300*, *SSVEP*
3. **Passive** BCI: A BCI of which focused on utilizing our daily brain signals (without users' conscious thoughts) to enrich our daily life/interaction, e.g. *spectral analysis*

BCI paradigms

BCI paradigms define how signals are being induced from the brain:

- **Motor Imagery**: a dynamic state during which an individual mentally simulates a physical action
- **SSVEP**: steady state visually evoked potentials are signals that are natural responses to visual stimulation at specific frequencies
- **P300**: a positive deflection in the EEG signal that appears approximately 300 ms after the presentation of an attended stimulus
- **Spectral analysis**: analysis in terms of a spectrum of frequencies (e.g., alpha beta, gamma)

Motor imagery

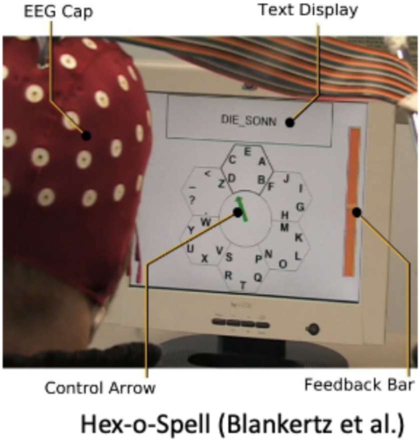


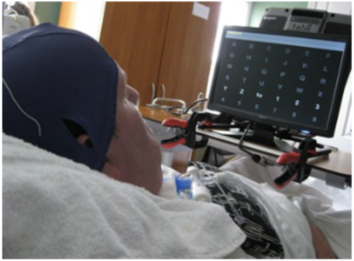
Figure: Motor Imagery based speller

Motor imagery

Brain2Robot
(Fraunhofer FIRST)

Figure: Motor Imagery based prosthetic arm

P300



P300 Speller

Figure: P300 speller

https://www.youtube.com/watch?v=y31GJVnSSsg

SSVEP

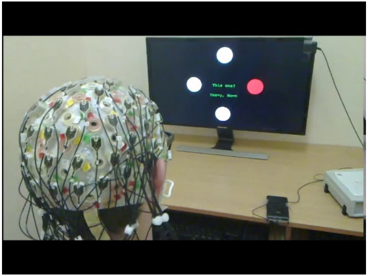


Figure: SSVEP

https://www.youtube.com/watch?v=t96rl1SFHlI

Spectral analysis

Farwell et al. 2000

Figure: Lie detection

Spectral analysis

Welke et al., 2011

Figure: Cognitive load detection

Future Applications

- **Communication** - focuses on developing speller for locked-in patients so they can communicate with their caregivers. FB is making a speller for normal people to type even faster!
- **Control** - focuses on using BCI to controlling prostheses device, for home automation, for gaming and etc.
- **Therapy/Rehabilitation** - focuses on using neurofeedback for rehabilitating users' attention or emotion.
- **Monitoring** - an area concerning affect/intent/cognitive state detection using the brain signals. Workload, fatigue, breaking intent, lie detection are some example states.
- **Cross-Modal learning** - converting EEG to image/speech.

Available tools

- **BioSig** - oldest open-source BCI toolboxes for offline processing (no GUI)
- **BCI2000** - written in C++ and mainly for online processing (acquisition, running experiments) but lack offline processing (e.g., algorithms)
- **OpenViBE** - written in C++; gives a block programming (for non-programmers). Requires Lua knowledge to extend/customize.
- **BCILAB** - matlab-based; lots of algorithms. Requires Matlab knowledge to extend/customize. Little support for acquisition systems but can tie to Lab Streaming Layer (LSL), a low-level technology that allow exchange of time series data between devices.
- **EEGLAB** - matlab-based; lots of algorithms. Requires Matlab to extend/customize
- **MNE-python + other python libraries** - python-based; for programmers. Lots of examples online. Work well with other python libraries. Requires python knowledge to extend/customize.

Event-Related Potentials (ERPs)

- Averaging EEG activity relative to an event results in primarily event-induced activity (trial-to-trial variability averaged out)

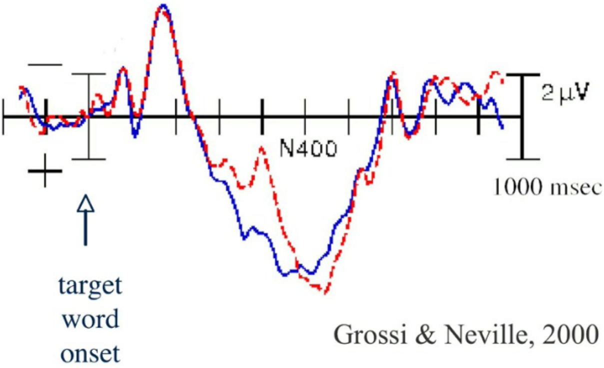


Figure: ERP of N400

Event-Related Potentials (ERPs)

- Single-trial ERPs are much harder to identify

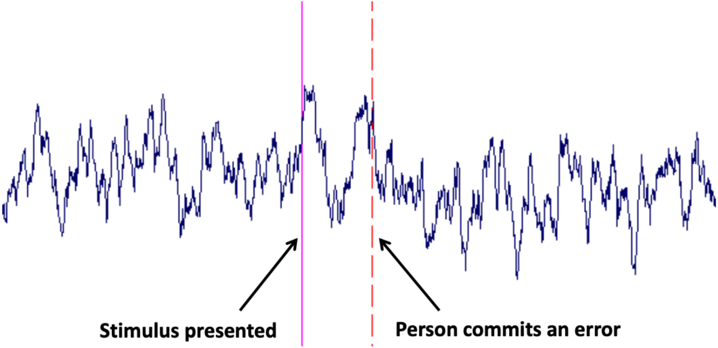


Figure: Single-trial ERPs

Pros and Cons

Pros:

- Computationally simple
- Decades-old literature
- Used in P300 and N400

Cons:

- Jittered and non-phase locked activities are lost
- Limited analysis possibilities (e.g., connectivity)
- Unclear biological mechanism

Frequency-domain analyses

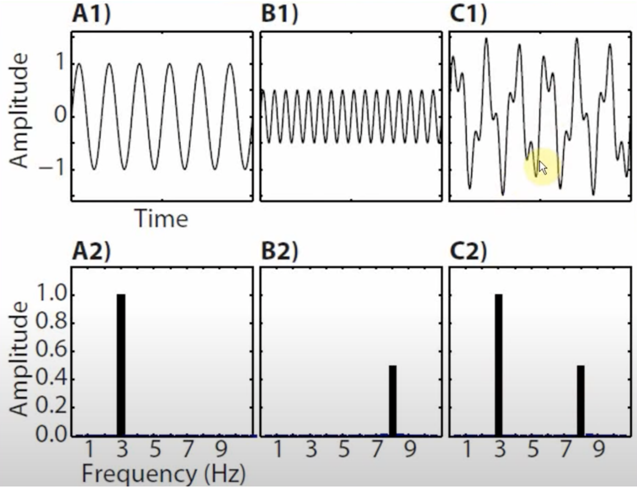


Figure: Changing time to frequency domain

Fourier Transform

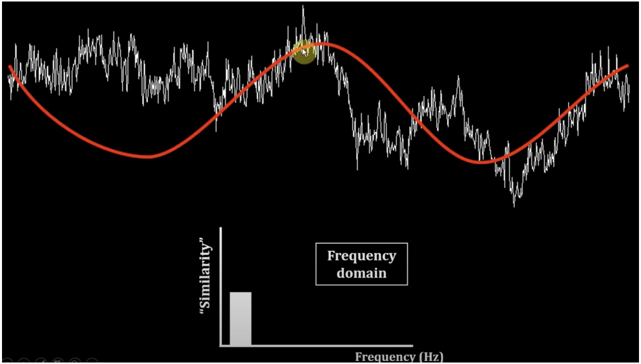


Figure: Dot product of particular sine wave with the EEG signal will give the corresponding similarity index. For example, a sine wave of 2Hz dot product with the EEG will give the frequency component of 2Hz.

Pros and Cons

Pros:

- Computationally fast
- Ubiquitous in science and engineering
- Useful for oscillatory processes, e.g., delta (0-4Hz), theta (4-7Hz), alpha (8-13Hz), beta(12-30Hz), and gamma(25-100Hz)
- Used in SSVEP, emotion recognition, cognition recognition

Cons:

- Lost time information
- Only good for stationary data. Is EEG stationary?

Time-frequency domain analyses

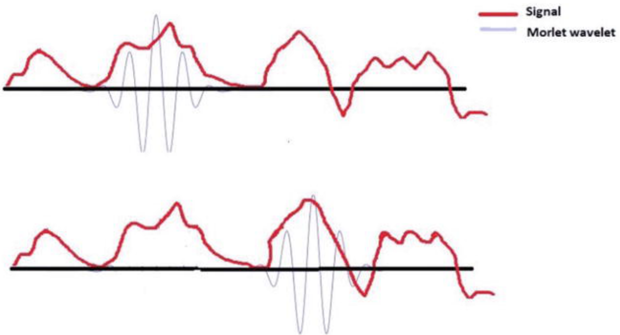


Figure: Instead of using a never ending sine wave, we can create a specific wavelet that captures only a specific time window.

Time-frequency domain analyses

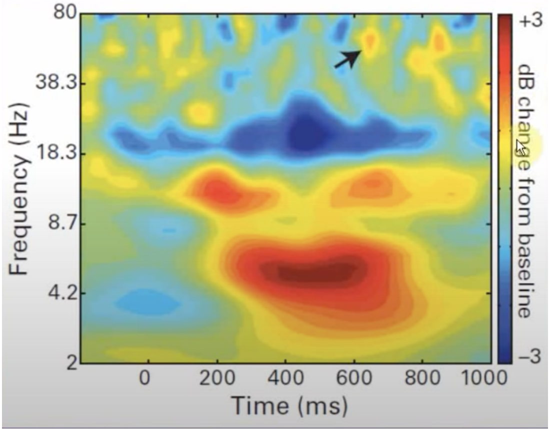


Figure: This will result in a spectrogram with both time and frequency information.

Time-frequency domain analyses

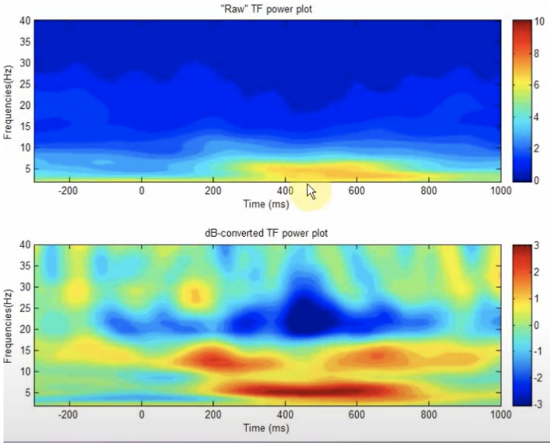


Figure: Often wise to perform baseline normalization. It transforms all data to same scale, disentangle task-related from background activity, and more like to be normally distributed thus good for classification.

Pros and Cons

Pros:

- Almost the best..Thus common neural network often transform EEG to spectrograms first, and then use CNN

Cons:

- Reduced temporal precision but it is a parameter we can fine tune....

Spatial Filters

- Transform a multi-channel signal $X(n)$ such that $Y(n)$ depends only on $X(n)$; most spatial filters are linear, i.e., $Y(n) = MX(n)$ for some matrix M
- Linear spatial filters can approximately invert volume conduction and remap channel signals to approximate source signals - this is the main use in BCIs
- Examples: Re-referencing, Surface Laplacian, Independent Component Analysis (CIA), Common Spatial Patterns (CSP)

Common Spatial Patterns

- Spatial filters designed to recover motor-cortex source activity, calculated via the Common Spatial Patterns algorithm

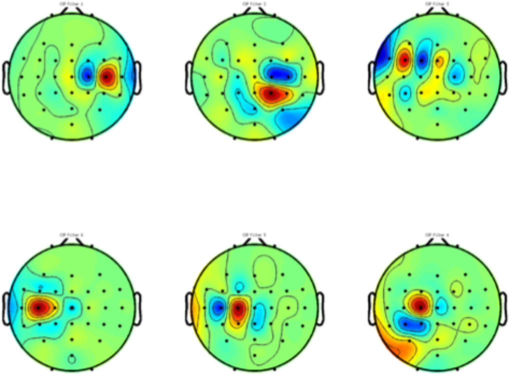
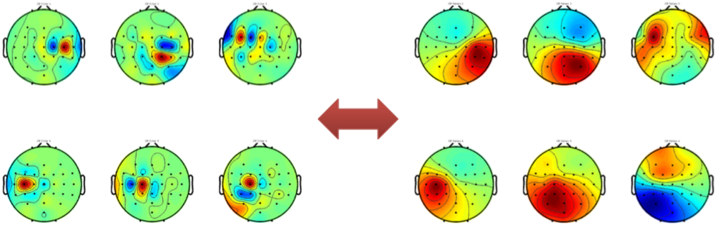


Figure: Common Spatial Patterns

Spatial Filters vs. Forward Projections

- Spatial filters are not the same as forward project maps of some source signal - they are the inverse operation



$$S(n) = \boldsymbol{W}X(n)$$

$$X(n) = \boldsymbol{W}^{-1}S(n)$$

Figure: Inverse operation of one another

Common Spatial Filters

```python
# Assemble a classifier
lda = LinearDiscriminantAnalysis()
csp = CSP(n_components=4, reg=None, log=True, norm_trace=False)

# Use scikit-learn Pipeline with cross_val_score function
clf = Pipeline([('CSP', csp), ('LDA', lda)])
scores = cross_val_score(clf, epochs_data_train, labels, cv=cv, n_jobs=1)

# Printing the results
class_balance = np.mean(labels == labels[0])
class_balance = max(class_balance, 1. - class_balance)
print("Classification accuracy: %f / Chance level: %f" % (np.mean(scores),
                                                          class_balance))

# plot CSP patterns estimated on full data for visualization
csp.fit_transform(epochs_data, labels)

csp.plot_patterns(epochs.info, ch_type='eeg', units='Patterns (AU)', size=1.5)
```

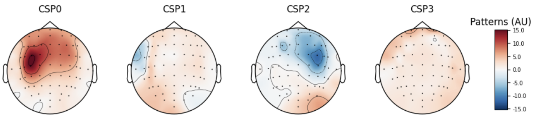


Figure: Common Spatial Filters

Topological Analyses

- Topological analyses create scalp maps that allow for rough source localization
- Topological analyses are NOT source localization

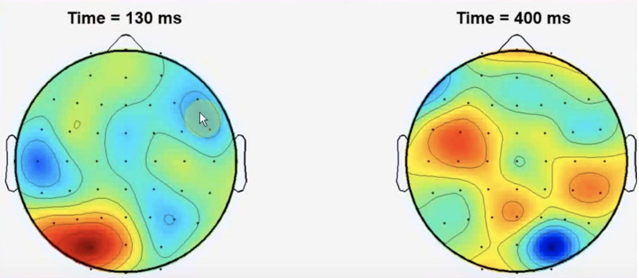


Figure: Left: Likely the participants are looking using the right eye; Right: Likely the participants are using the right hand to click.

More Scalp Maps

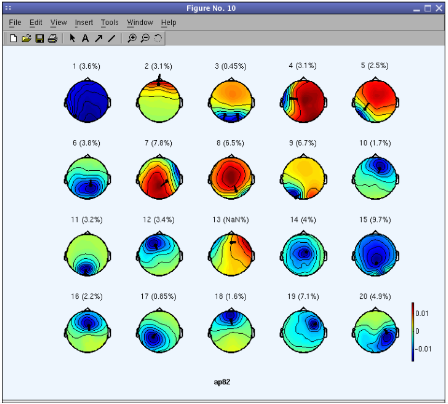


Figure: Scalp maps

Python MNE with scalp maps

```
all_times = np.arange(-0.2, 0.5, 0.03)
evoked.plot_topomap(all_times, ch_type='mag', time_unit='s',
                    ncols=8, nrows='auto')
```

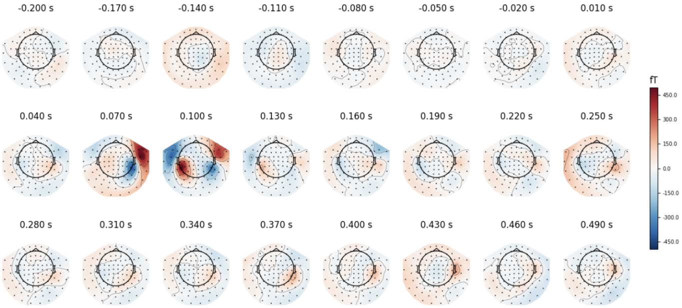


Figure: Plot evoked topographies using MNE

Volume Conduction

- Neural activity is conducted through the brain volume to the scalp which is instantaneous, and characterized by a electromagnetic field
- Volume conduction is linear, thus each sensor measures a (weighted) sum of each neuron's activity

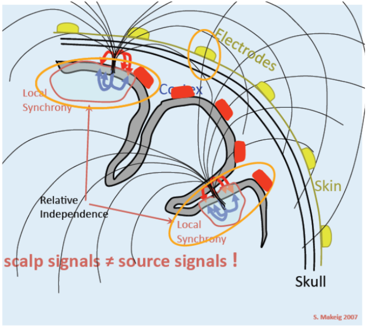


Figure: Neural activity

Volume Conduction

- Note: the point-spread function from a source patch to the scalp is extremely broad

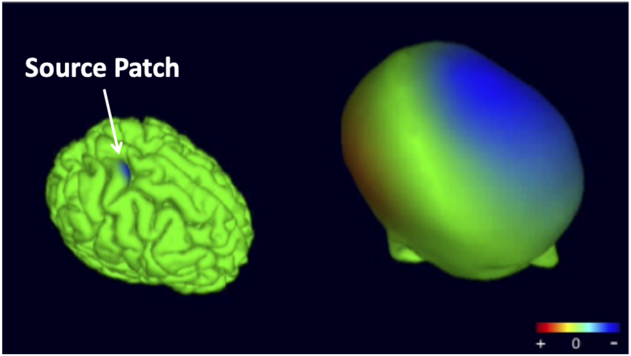


Figure: Akalin Acar et al., 2011

Dipole Fitting

- Electromagnetic field sustained by a compact collection of neurons can be modeled as a single equivalent dipole
- This facilitates localization of the field source

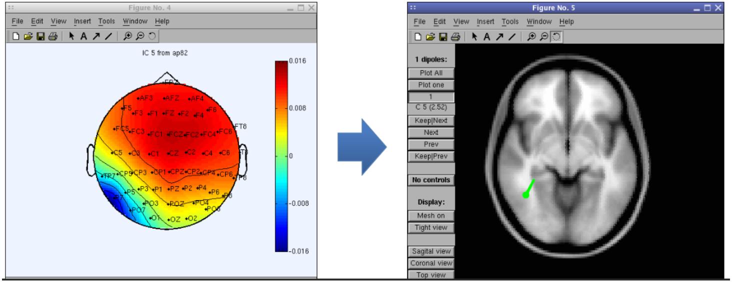


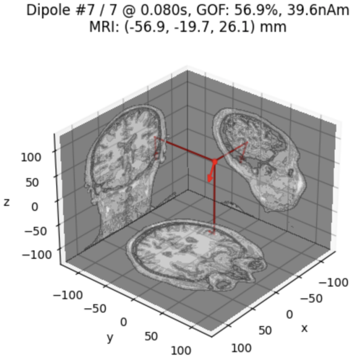
Figure: Source localization

Dipole Fitting

```
evoked = mne.read_evokeds(fname_ave, condition='Right Auditory',
                          baseline=(None, 0))
evoked.pick_types(meg=True, eeg=False)
evoked_full = evoked.copy()
evoked.crop(0.07, 0.08)

# Fit a dipole
dip = mne.fit_dipole(evoked, fname_cov, fname_bem, fname_trans)[0]

# Plot the result in 3D brain with the MRI image.
dip.plot_locations(fname_trans, 'sample', subjects_dir, mode='orthoview')
```



Dipole Modeling Problems

- High-quality fits are hard to achieve
  - Requires knowledge about sensor locations
  - Require assumptions about conductivities of scalp, skull, cerebrospinal fluid (CSF), brain tissue
  - Requires knowledge of the folding of the cortex (candidate dipoles) unless simplistic spherical model is used
  - Some brain tissues has anisotropic conductance (white matter)
  - Scalp maps are usually not perfect (arise from data processing) - fit accuracy suffers
  - Scalp maps can be a sum of multiple dipole sources - requires a distributed source model

Distributed Source Modeling

- Allow to recover and image distributed cortical support of given scalp maps

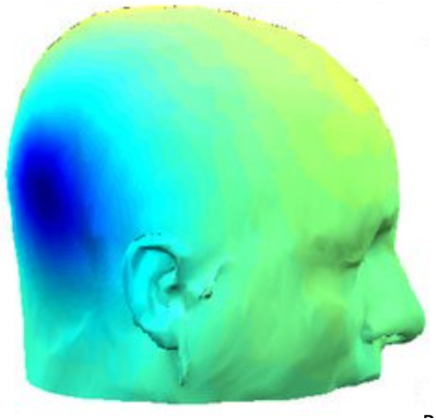
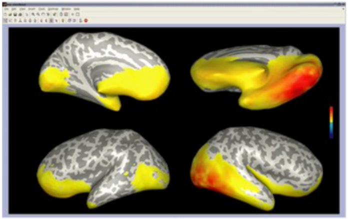


Figure: Ray Ramirez (Scholarpedia)

Distributed Source Modeling

- Wide range of methodologies and underlying assumptions (sLORETA, Beamforming, Sparse Bayesian Learning, ...)
- Prone to finding only locally optimal solutions



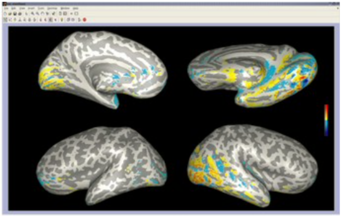


Figure: LCMV Beamforming (Left), Anatomically Constrained Beamforming (Right)

Distributed Source Modeling

- Wide range of methodologies and underlying assumptions (sLORETA, Beamforming, Sparse Bayesian Learning, ...)
- Prone to finding only locally optimal solutions

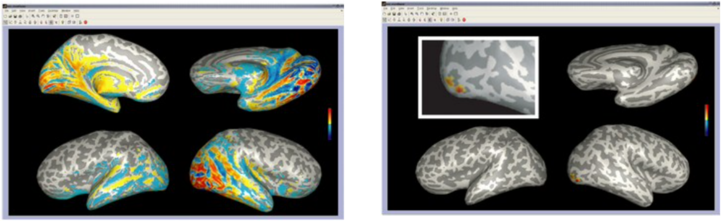


Figure: sLORETA (Left), Sparse Bayesian Learning (Right)

Non-Brain Artifacts

- Often far outscale the brain processes in the EEG (when present)
- **Internally** generated: neck, face, and eye muscles, eye dipoles, heart activity
- **Externally generated: 50/60Hz line noise, EM spikes from equipment**
- **Sensor**-related: DC offset drifts, cable sway, thermal noise, quantization noise

Muscle Artifacts

- High-frequency / broadband, large amplitude

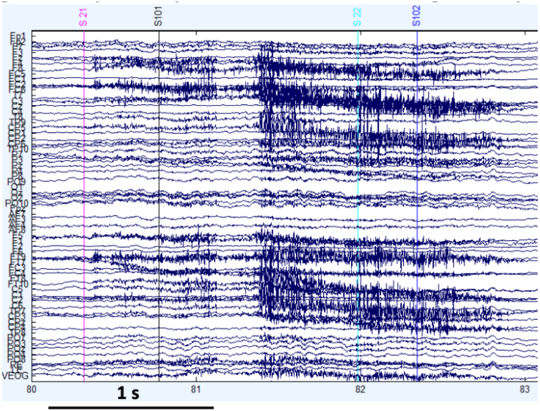


Figure: Muscle Artifacts

Muscle Artifacts

- Scalp projections are spatially stereotyped

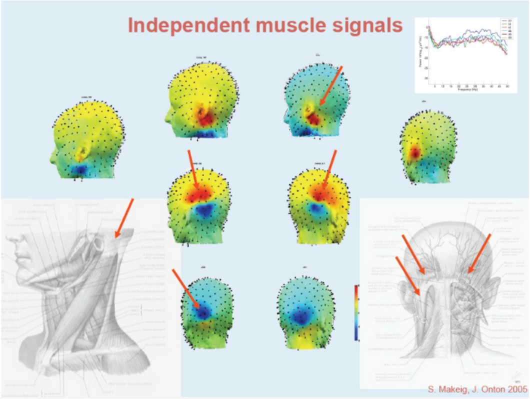


Figure: Muscle Artifacts

Eye Blinks

- Large low-frequency peak and rebound, mainly frontal
- Can also incurs non-linear effects in occipital cortex

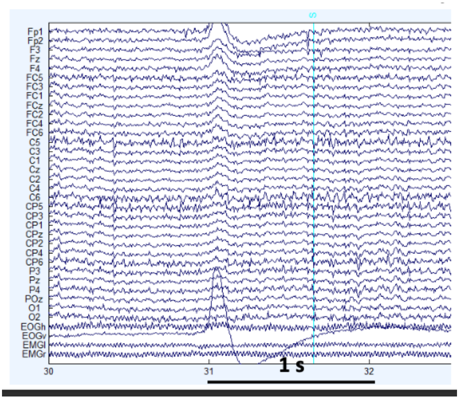


Figure: Eye Blinks Artifacts

Bandpass Filters

- Easiest way to remove artifacts is simply to select only the frequency of interest

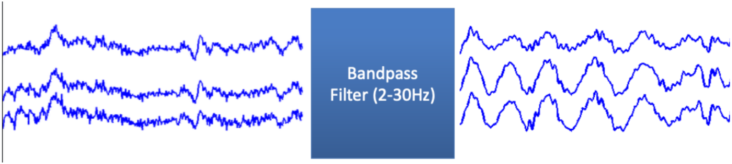


Figure: Bandpass filter

Independent Component Analysis

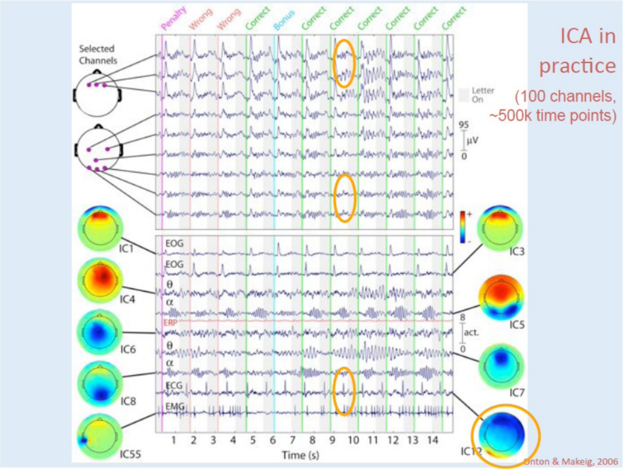


Figure: Onton and Makeig 2006

Deep Neural Network

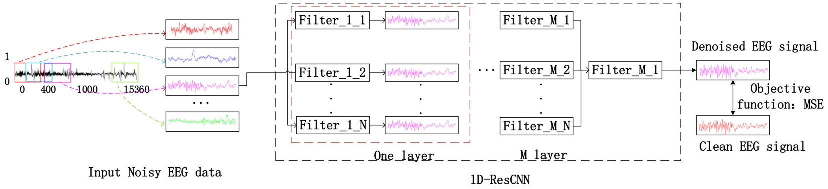


Figure: https://www.sciencedirect.com/science/article/abs/pii/S0925231220305944

Event-Related Coherence

- Event-Related Coherence between two signal components

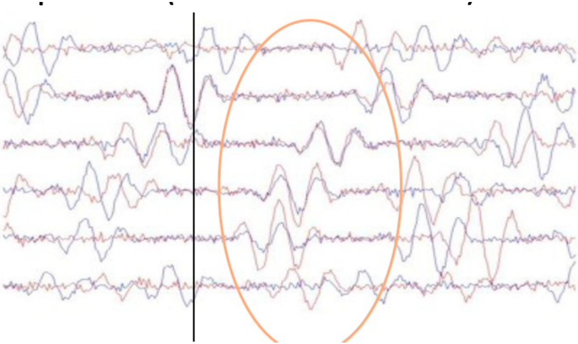


Figure: Makeig 2007

Connectivity

- Sophisticated measure of interaction between multiple signals
- By getting the source estimates along the time, we can construct the connectivity plot

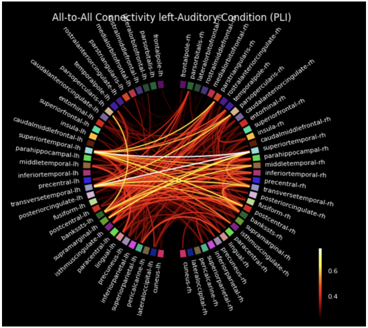


Figure: Python MNE

Measurement Sites

- Standardized location system (10-20 system)
- Saves a lot of hassle vs. custom labels
- Often defined in 8, 16, 32, and 64 channels

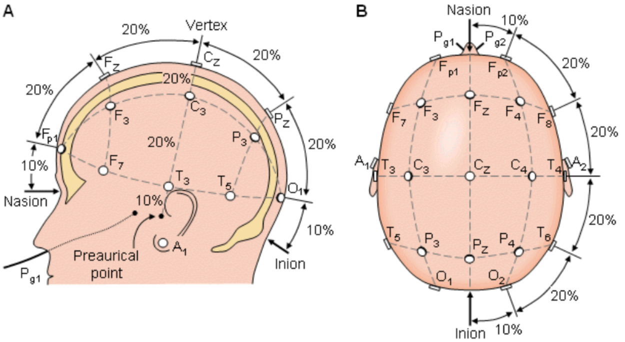


Figure: International 10-20 system (Malmivuo and Plonsey, 1995)

Measurement Sites

```
ten_twenty_montage = mne.channels.make_standard_montage('standard_1020')
print(ten_twenty_montage)
```

```
fig = ten_twenty_montage.plot(kind='3d')
fig.gca().view_init(azim=70, elev=15)
ten_twenty_montage.plot(kind='topomap', show_names=False)
```

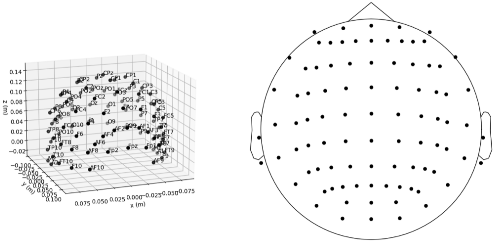


Figure: Montage objects have a plot() method for visualization of the sensor locations in 3D; 2D projections are also possible by passing kind='topomap'

Where to put electrodes?

- Some notable large-scale brain features are the hemispheres, lobes, gyri and suci
- No nerves coming to frontal lobe thus it's less obvious what they do

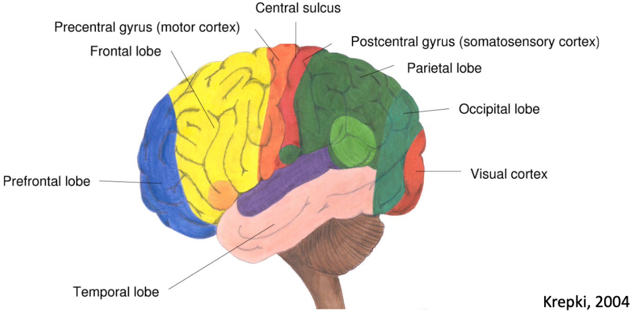


Figure: Major anatomical regions

Functional Mapping

- For most regions more or less well known functional associations exist - the motor cortex is one of the best examples

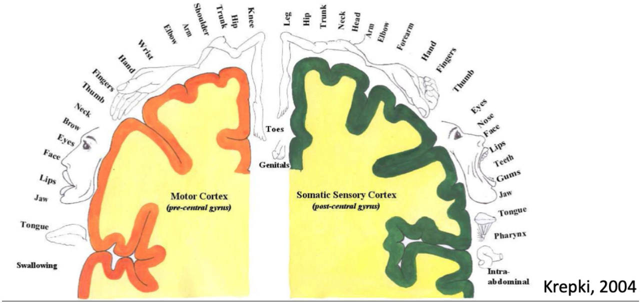
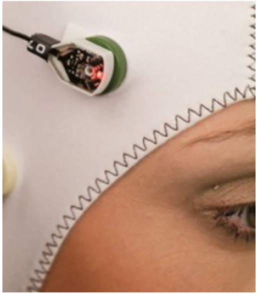


Figure: Homunculus: Functional mappings of the motor cortex

EEG Sensor Designs

- Most EEG systems are gel-based
- Nowadays mostly using active electrodes (i.e., with amplifiers)

Passive, gel-based
(EasyCap)



Active, gel-based
(Brain Products)

Figure: EEG gel-based sensor

EEG Sensor Designs

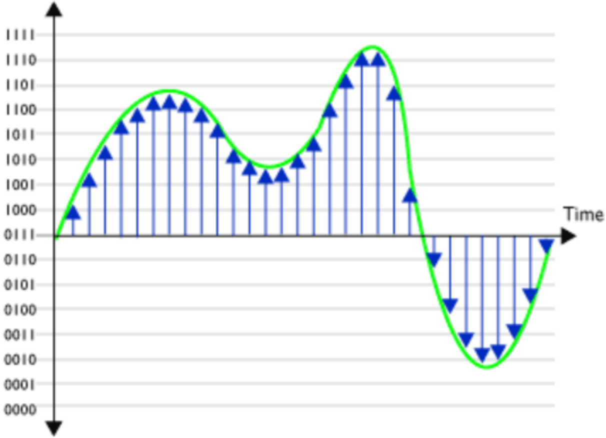
- Dry systems are emerging quickly

Pins (g.SAHARA) | Spring-loaded Pins (NCTU) | Foam-based sensors (NCTU)

Figure: EEG dry-based sensor

Digitization

- After amplification (e.g. 50000x), signal is low-pass filtered using an analog filter, then digitally sampled at fixed rate



Sampling Theorem

- If the signal is band-limited below the Nyquist frequency B (i.e., contains no higher frequency than B), it can be exactly reconstructed using the interpolation function:

$$g(t) = \frac{sin2\pi Bt}{2\pi Bt} \tag{1}$$

$$s(t) = \sum_{n=-\infty}^{\infty} s\left(\frac{n}{F_s}\right) g\left(\frac{t-n}{F_s}\right) \tag{2}$$

- The Nyquist Frequent is $\frac{1}{2}$ sampling rate

OpenBCI connection

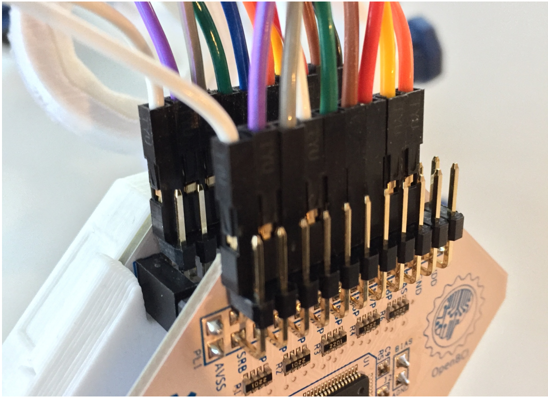


Figure: Cyton with Daisy configuration. The Y splitter (white) on SRB, connecting to earlobe. Another reference wire on BIAS (black), connecting to another earlobe. Earlobe is used as reference because it has no muscles or neurons and therefore very low electrical signals.

OpenBCI connection

Figure: Gold cup electrodes can be used to connect to other part of the body to the Cyton and Daisy board. First scope the electrode paste into the electrode. Then paste accordingly.

OpenBCI connection

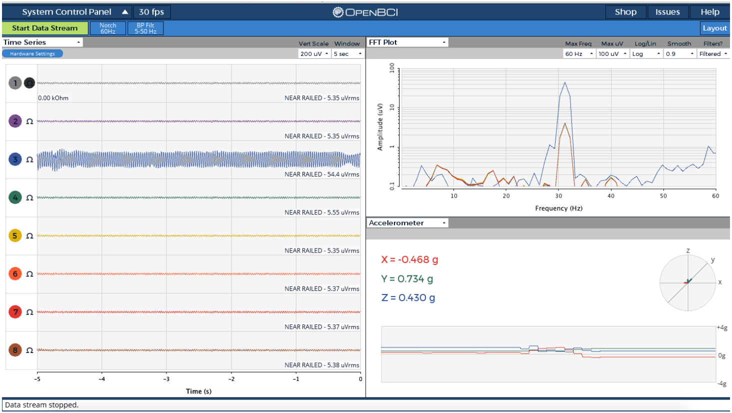


Figure: When the Ohm icon is toggled on, the board sends a small current through the selected channel to obtain the impedance value. For this reason, you won't be able to stream data on a channel and obtain the impedance value simultaneously.

OpenBCI connection

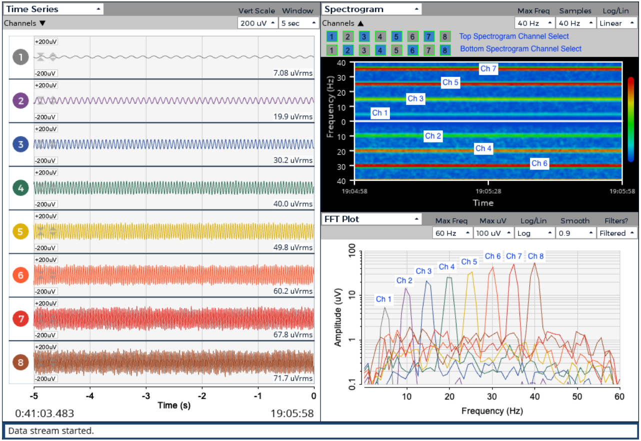


Figure: A dual spectrogram display which allows users to see changes in FFT data over time

OpenBCI connection

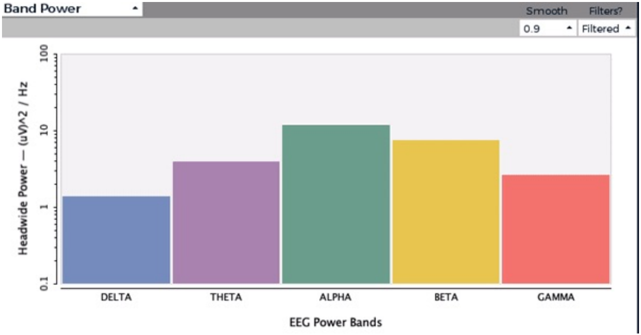


Figure: One easy way to test the system is to ask users to close their eyes, the alpha should be low

OpenBCI connection

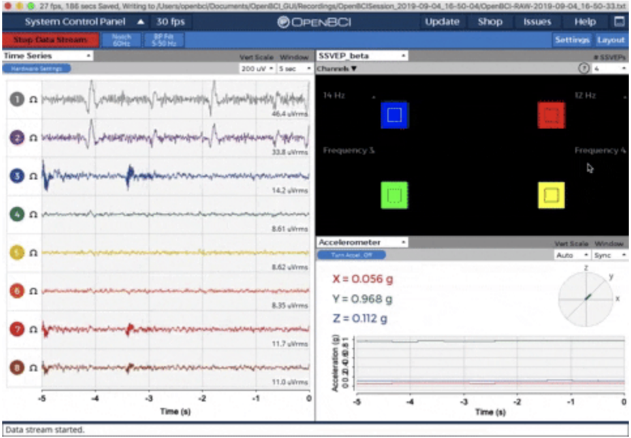


Figure: SSVEP widget

OpenBCI connection

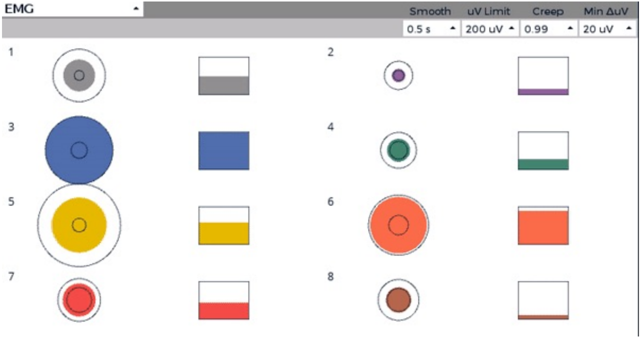


Figure: EMG widget: if you relax, the value will be 0, and if you flex, the value will go to 1

OpenBCI connection

LSL

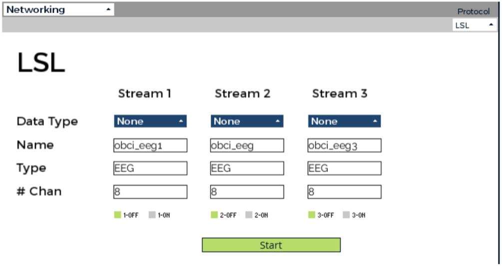


Figure: Lab Streaming Layer is a system for synchronizing streaming data for live analysis or recording. LSL is a good way to send your OpenBCI stream to applications like Python.

Example: Using EMG to stop/start music

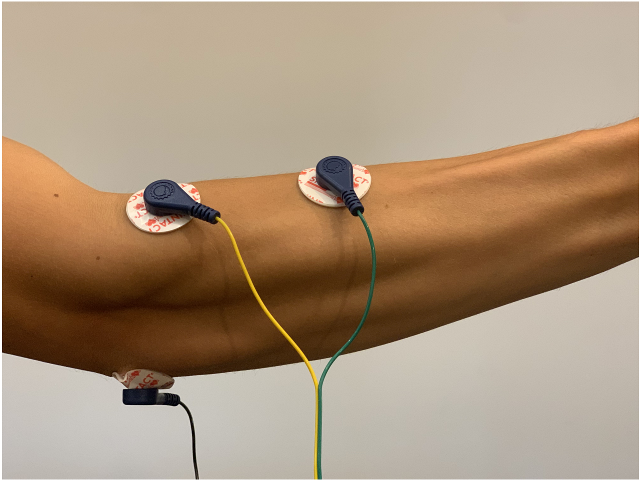


Figure: Connect to top N1pin and bottom N1pin for measuring potential difference. The elbow is connected to the bottom AGND pin for reference.

Example: Using EMG to stop/start music

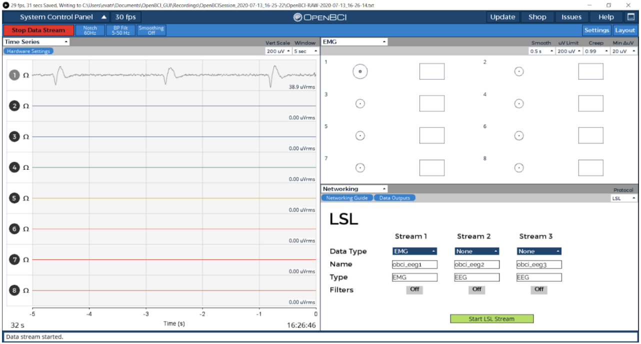


Figure: Turn on the EMG and LSL widgets.

Example: Using EMG to stop/start music

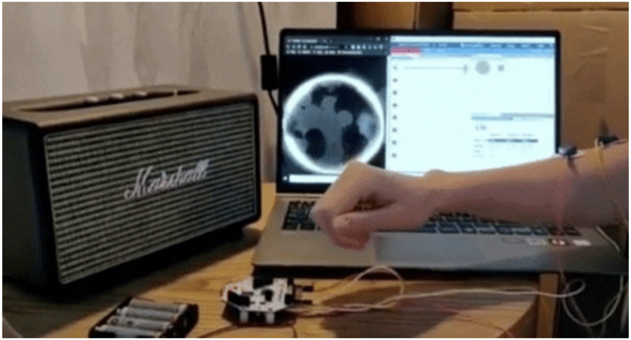


Figure: EMG to start/stop music

Basic CNN

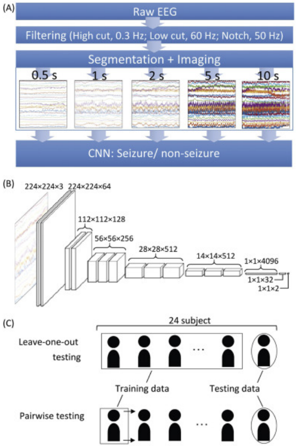


Figure: https://www.sciencedirect.com/science/article/pii/S2213158219300348

EEGNet (2018)

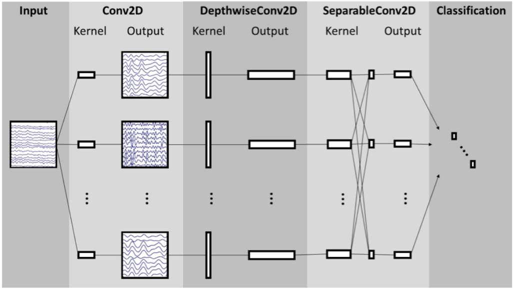


Figure 1: Overall visualization of the EEGNet architecture. Lines denote the convolutional kernel connectivity between inputs and outputs (called *feature maps*) . The network starts with a temporal convolution (second column) to learn frequency filters, then uses a depthwise convolution (middle column), connected to each feature map individually, to learn frequency-specific spatial filters. The separable convolution (fourth column) is a combination of a depthwise convolution, which learns a temporal summary for each feature map individually, followed by a pointwise convolution, which learns how to optimally mix the feature maps together. Full details about the network architecture can be found in Table 2.

Figure: `https://arxiv.org/abs/1611.08024`

Cross-Modal Learning (2020)

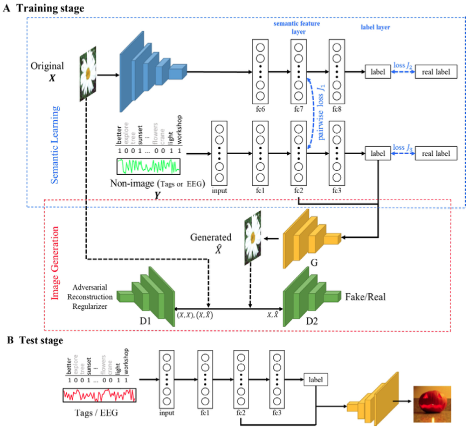


Fig. 2. The framework of our proposed cross-modal generation method. It contains two parts: the semantic learning part and the image generation part. We use the image modality and the non-image modality in the training stage. In the test stage, we only use non-image modality to generate image modality.

Figure: https://www.sciencedirect.com/science/article/abs/pii/S0031320319303863

Good Resources

- Use **Python MNE** - is made by neuroscientists. Documentation is also VERY well written. They have (1) source estimation, (2) machine learning, (3) connectivity, (4) awesome data visualization modules. You will learn A LOT of neuroscience here.
- Although we did not use **MATLAB**, it has many good tutorials that will allow us to better understand how EEG works - `https://sccn.ucsd.edu/eeglab/index.php`
- Use **LSL** protocol with **pyOpenBCI (python way)** or **OpenBCI GUI (GUI way)** to acquire data from the OpenBCI board.
- Use **pygame**, **tkinter** or **javascript** if you want to make BCI application. You have to use **threads** a lot so learn it.
- EEG **datasets**: `https://github.com/meagmohit/EEG-Datasets`
- Best **youtube** channel for learning EEG - Mike X Cohen - `https://www.youtube.com/channel/UCUR_LsXk7IYyueSnXcNextQ`. Our lab also has his book - **Analyzing Neural Time Series Data**.
- Another easy-to-read yet very good book is **Practical Approach to Electroencephalography** by Libenson. Once you learn it, you can read **Rowan's Primer of EEG** (our lab also has it) which teaches you how to read EEG.
- Read **EEGNet** and **EEG-ChannelNet** if you are interested in deep learning + EEG

Future Research

The key questions in BCI remains unchanged and is still in the **infancy** stage. The problem remains **unroboust accuracy**, **impracticality** due to many restrictions (e.g., many artifacts, many variations in users), does not really know what is **EEG** (e.g., where does it come from, what different features mean). Here are some topics I often come across in recent years.

- **Mobile** real-time EEG
- **Feature** Extraction (e.g., functional connectivity)
- **Fusion** model (e.g., LSTM + GCNN)
- **Artifact Removal / Source Localization**
- **Multimodal** EEG + other sources of input (e.g., fNIRS, facial, EMG, EKG)
- **One shot learning** from few training data (e.g., only 1 minute of EEG recordings! or only 1 electrode!)
- **Participant independent** model
- **Cross-modal learning** - EEG $\leftrightarrow$ audio/image/text
- **Data Augmentation**
- Understanding the underlying **neural processes** via EEG

What's next

Read my slide on **Empirical** and these textbook resources:

- Mackenzie, Chapter 4-5, **Scientific Foundations, Designing HCI Experiments**, Human Computer Interaction: An Empirical Research Perspective, 1st ed. (2013)
- Zhao, **How to Design Controlled Experiments in HCI?** `https://www.slideshare.net/shilman/controlled-experiments-shengdong-zhao`

Questions