2023 EEG WORKSHOP

GROUP 6
FINAL PRESENTATION

組員：許珈綺、林健驊、陳昱如、黃茂勛

August 18, 2023

Outline

- **Course and Assignments**
  - Time-domain Analysis
  - Time-frequency Analysis
  - Source Analysis
  - Machine Learning Framework
- **Overall Project Demo**
  - Qualitative and quantitative analysis
- **Conclusion**
  - Challenges and Solutions
- **References**

TIME-DOMAIN ANALYSIS

Event Related Potentials(ERP)

Introduction

Event-Related Potentials : measuring brain wave with specific event-related stimulus.

1. ERP Image
2. Time Domain
3. Topography Mapping
4. Signal-to-noise Ratio

Dataset : ERP_Two-target Oddball.cnt

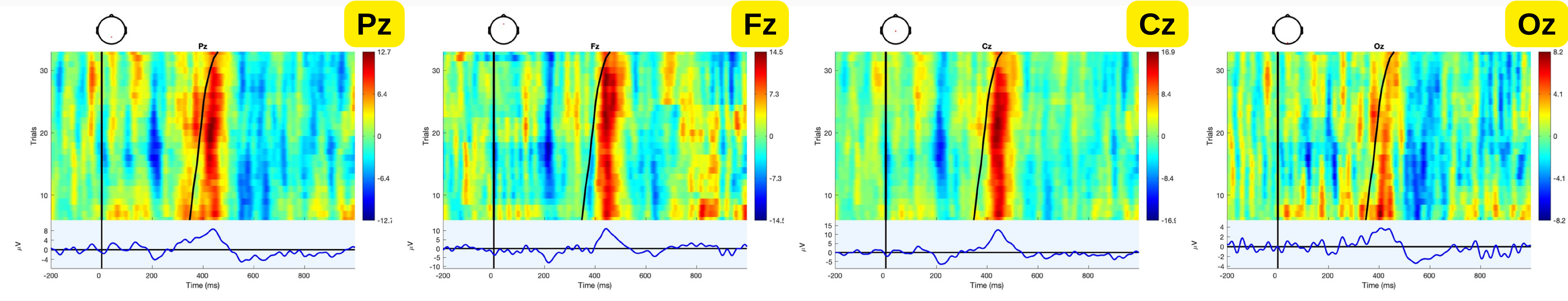
Pre-processing : Channel location -> Filter the data -> Change sampling rate -> Re-reference the data -> Select data -> Extract Epochs

ERP Images

Plot -> Channel ERG image

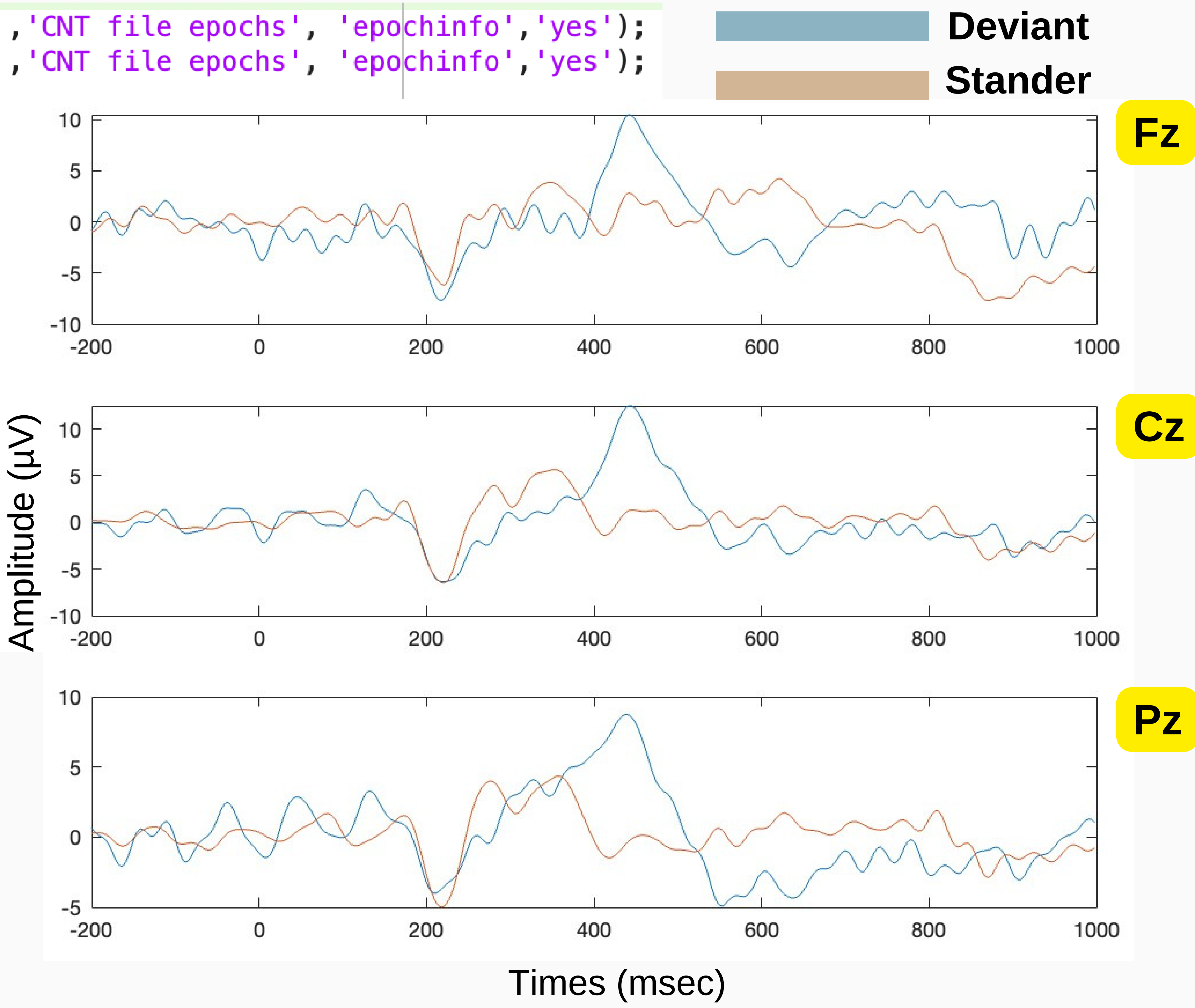
USE :

Sorting trials by reaction time (event = 5), it can show the latency of the rt event.



Time Domain

```
Deviant = pop_epoch(EEG,{'3'}, [-0.2 1], 'newname' ,'CNT file epochs', 'epochinfo','yes');
Stander = pop_epoch(EEG,{'2'}, [-0.2 1], 'newname' ,'CNT file epochs', 'epochinfo','yes');
dev_data = Deviant.data;
stn_data = Stander.data;

clf
ch = [5 15 25];
for n = 1:3
    x = dev_data(ch(1, n), :, :);
    y = stn_data(ch(1, n), :, :);
    x_ = mean(x, 3);
    y_ = mean(y, 3);

    subplot(3, 1, n);
    plot(Deviant.times,x_);
    hold on
    plot(Stander.times,y_);
end
```



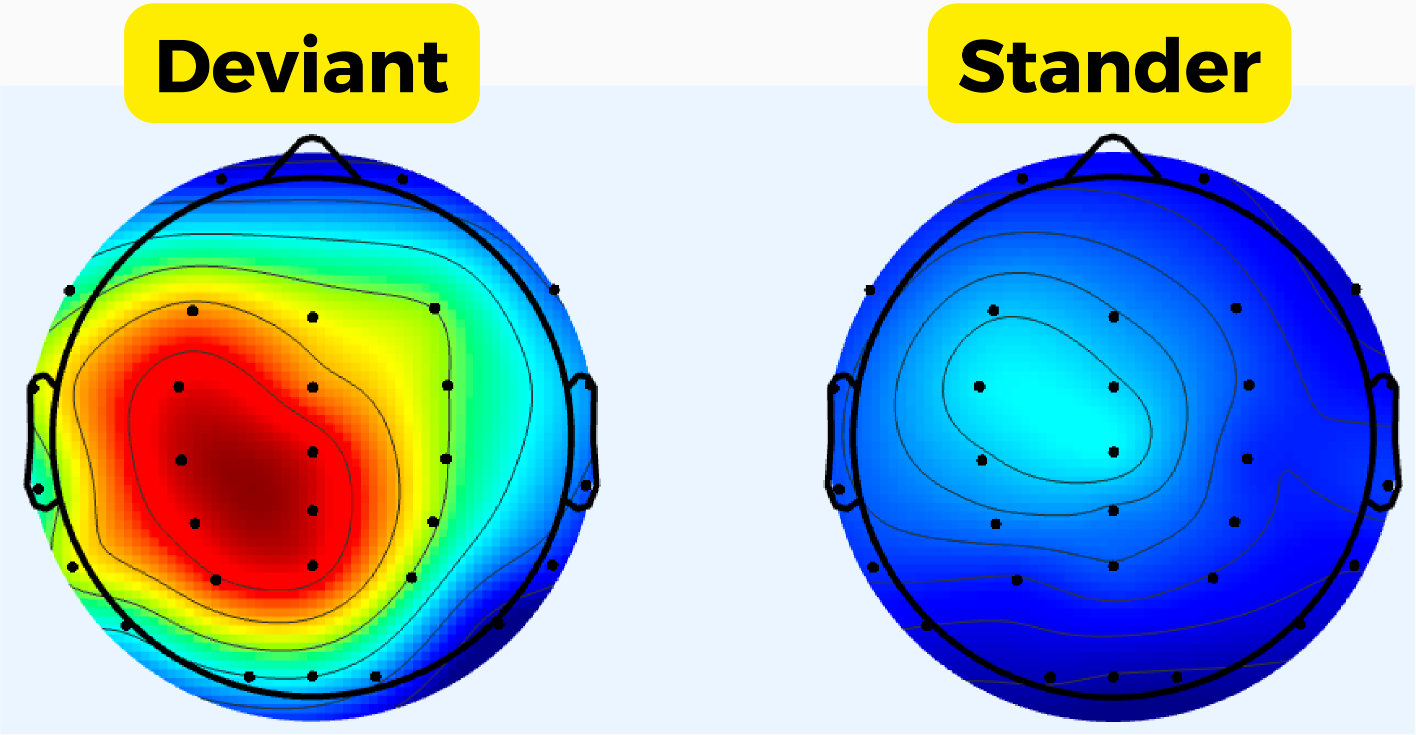
USE :

Show the amplitude / epoch of Deviant and Stander in Fz, Cz, Pz.

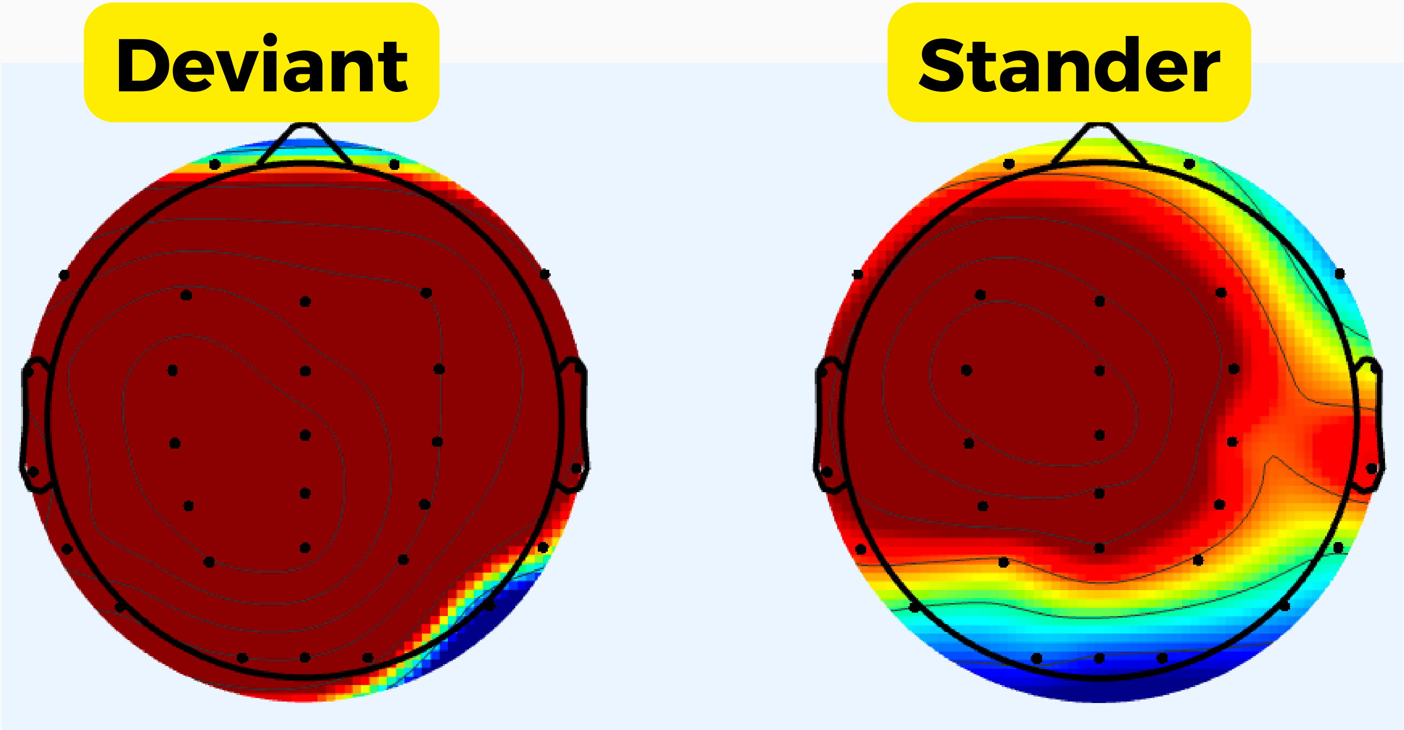
Topographic mapping

```matlab
a = dev_data(:, 251:351, :);
b = stn_data(:, 251:351, :);

% 平均 trail
a_ = mean(a, 3);
b_ = mean(b, 3);

% 平均取出的times
a_ = mean(a_, 2);
b_ = mean(b_, 2);

A = cat(2, a_, b_);
minA = min(A,[],"all");
maxA = max(A,[],"all");

a_nom = (a_ - minA) / (maxA - minA);
b_nom = (b_ - minA) / (maxA - minA);

figure;
subplot(1, 2, 1);
topoplot(a_nom, EEG.chanlocs,'maplimits',[0 1]);
subplot(1, 2, 2);
topoplot(b_nom, EEG.chanlocs,'maplimits',[0 1]);
```

USE : Display the active spatial.



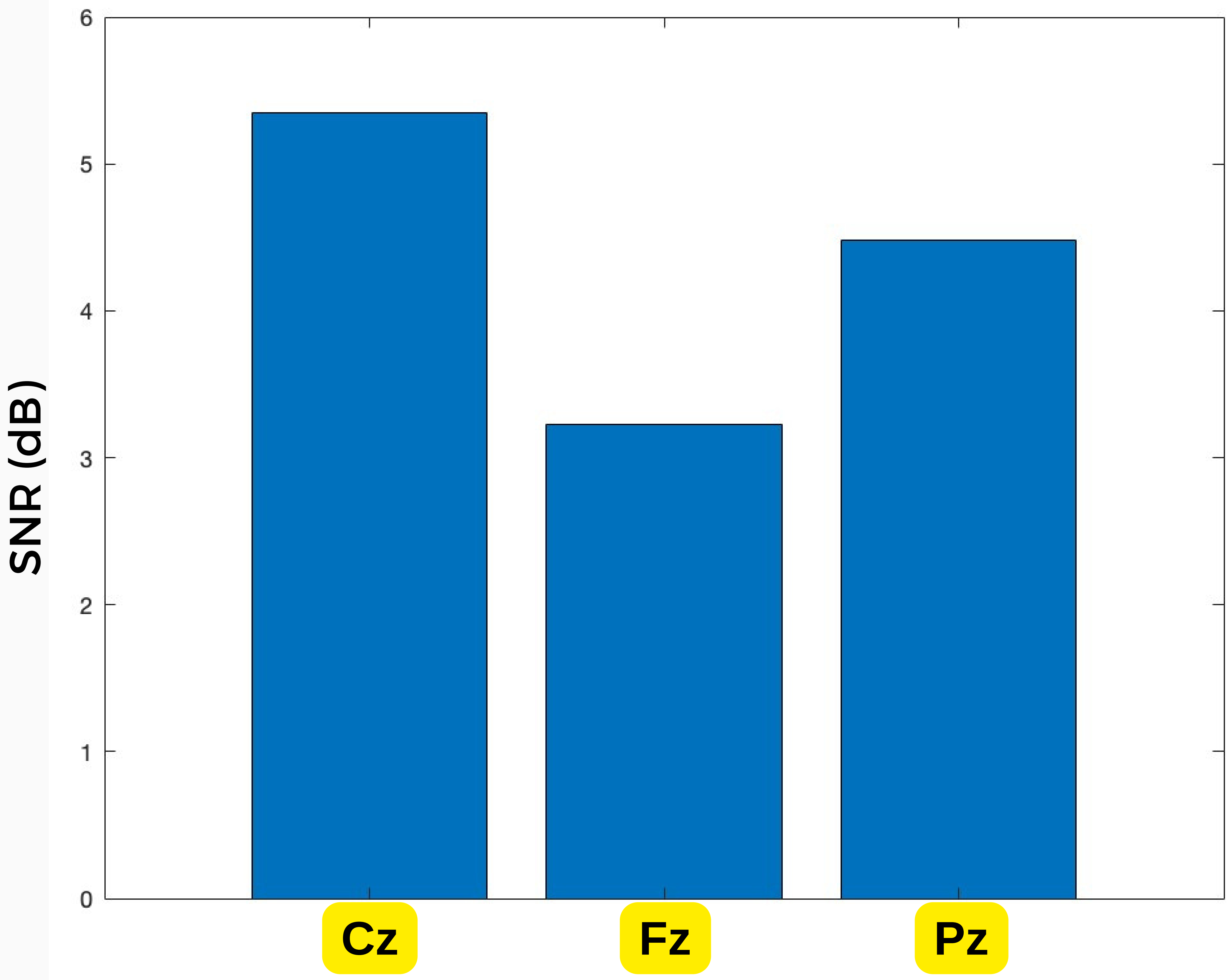
Normalization



None-Normalization

Signal-to-Noise Ratio (SNR)

```
p300 = dev_data([5 15 25], 251:351, :);
baseline = dev_data([5 15 25], 1:101, :);

p300_ = mean(p300, 3);
baseline_ = mean(baseline, 3);

SNR = mean(p300_, 2) ./ std(baseline_, 0, 2);
cats = categorical(["Fz","Cz","Pz"]);

bar(cats, SNR);
xlabel('SNR');
```



USE :

Show which channel performance is better.

TIME-FREQUENCY ANALYSIS

Steady State Visually Evoked Potentials (SSVEP)

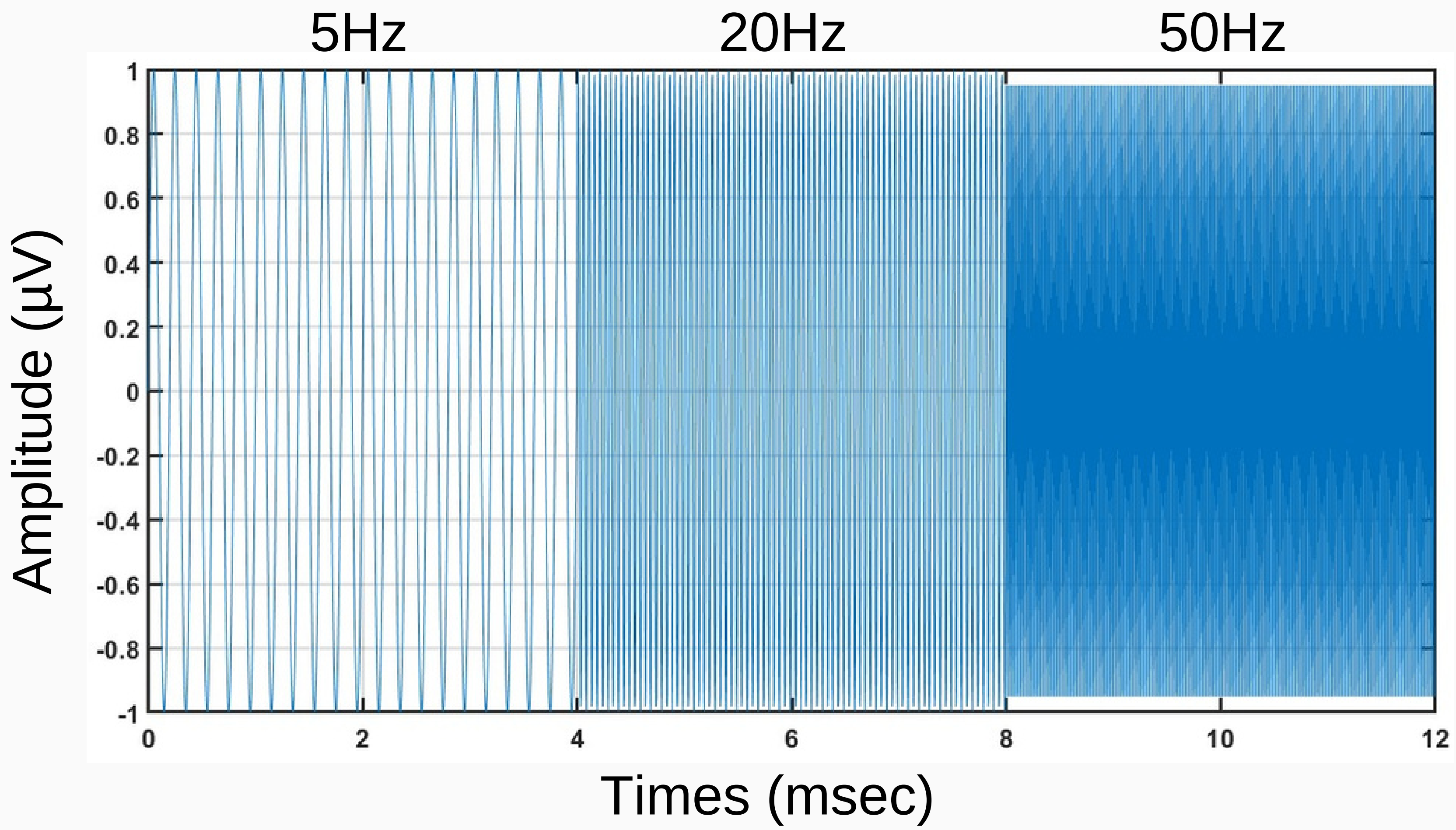
Introduction

Steady State Visually Evoked Potentials

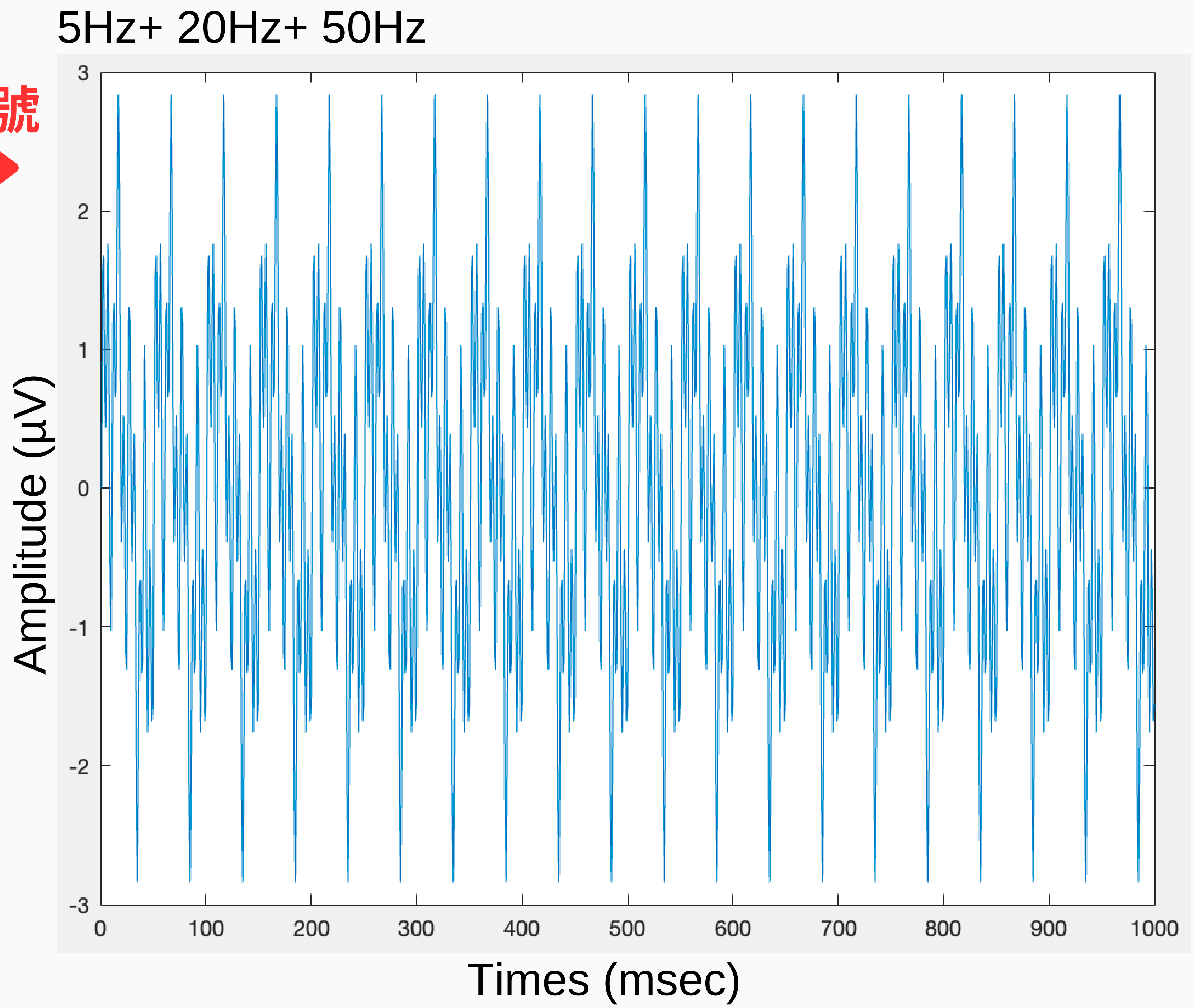
- 大腦對於特定頻率的視覺刺激會誘發相對應頻率的訊號
  - First harmonic
  - Second harmonic
- 著重在單一頻率上的分析 : 10Hz / 11Hz / 12Hz / 13Hz / Nan

Dataset : SSVEP_Five-target_flickers.set

Practice 1:
Time Analysis based on Sine Wave

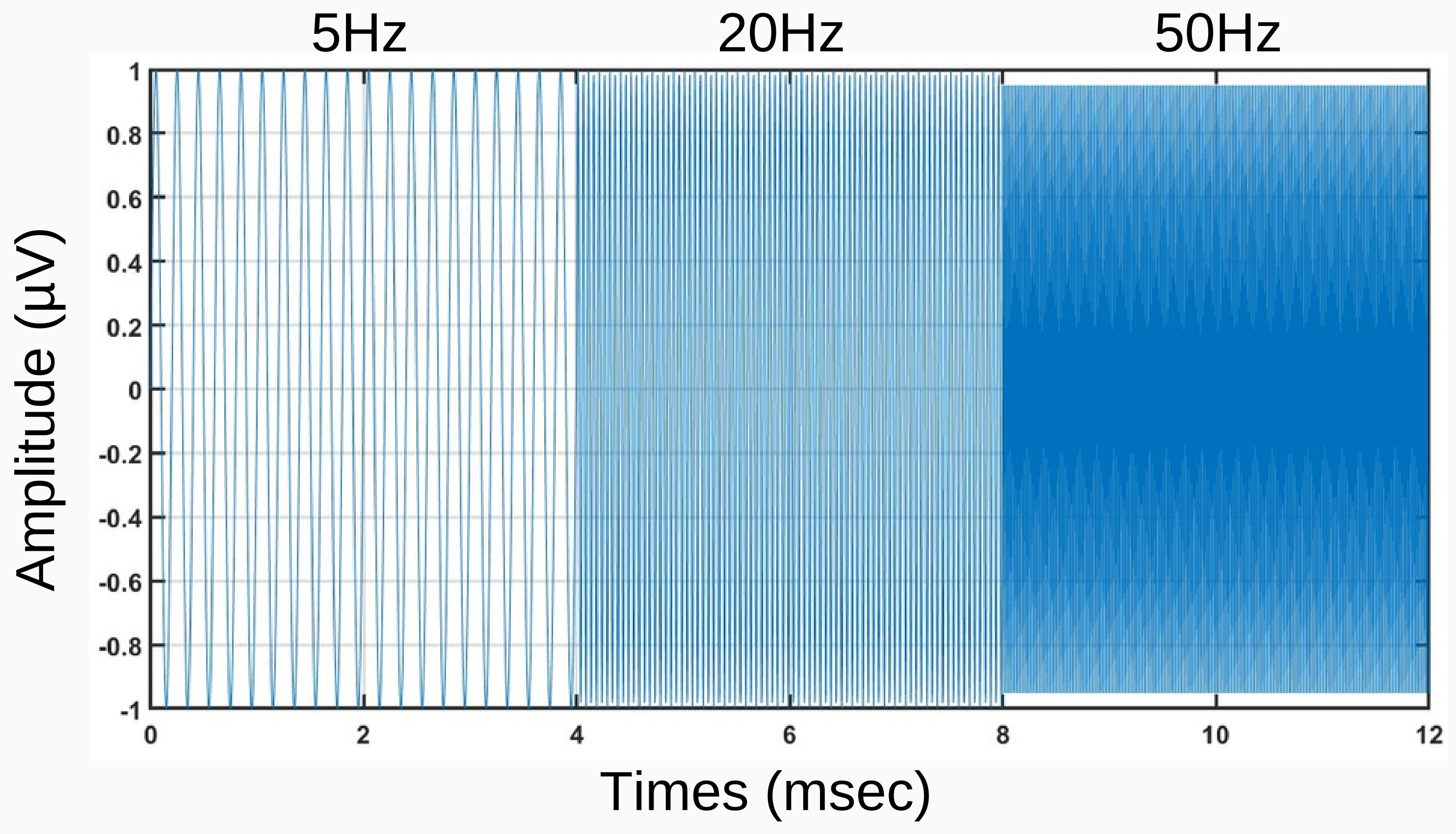


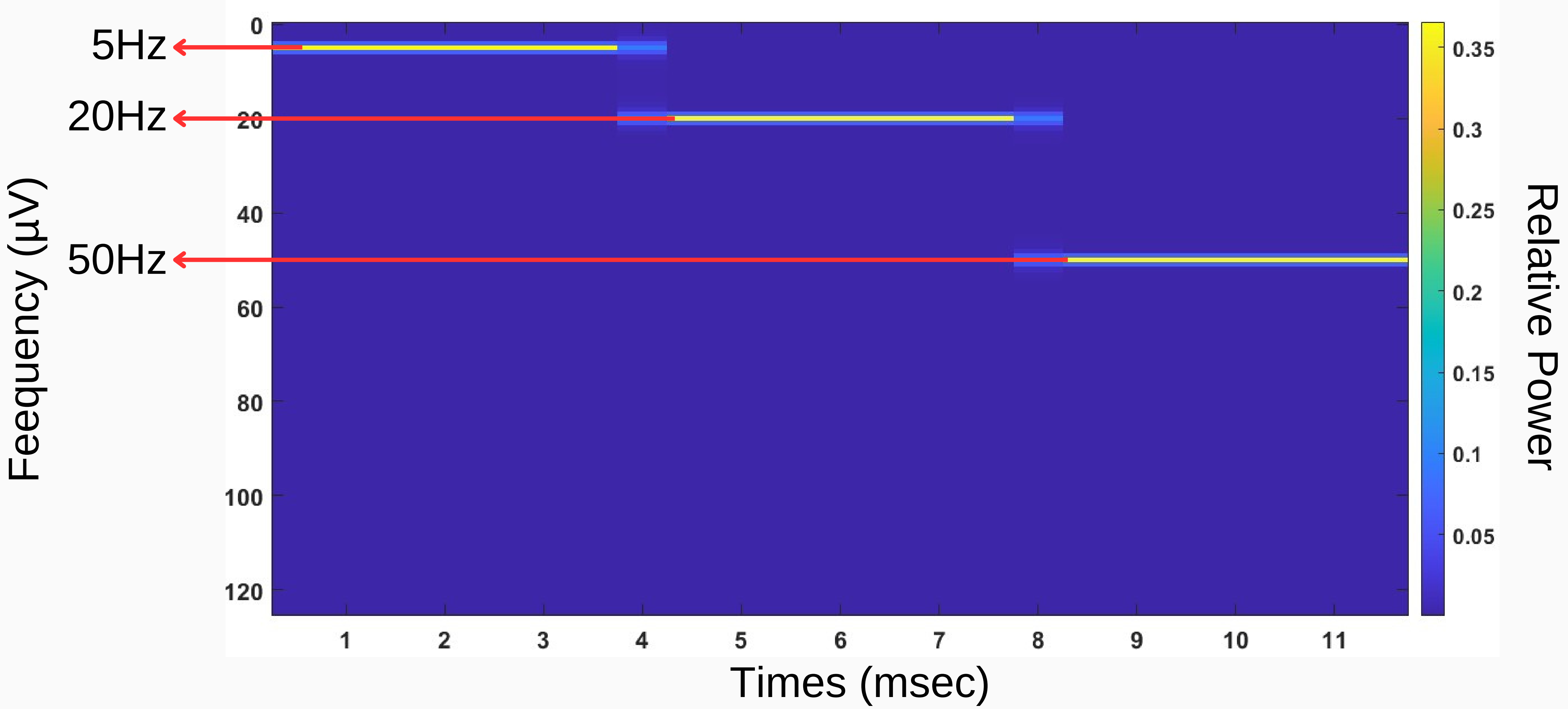
生理訊號
包含許多
不同頻率的訊號



無法透過時域解析出不同頻率的訊號

Practice 2: Time-Frequency Analysis based on Sine Wave



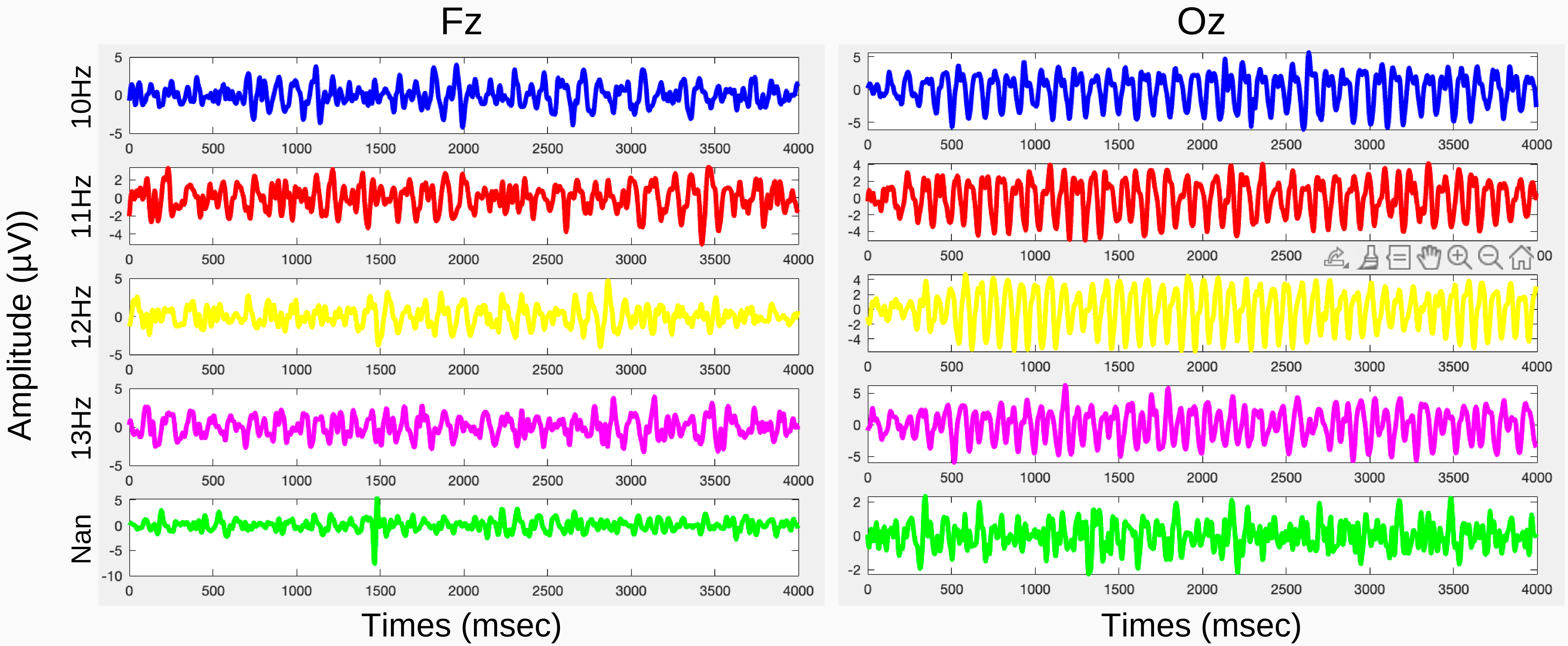


振幅的變化 → 能量的變化

透過時域轉頻域分析出每個時間點下不同頻率的功率強度

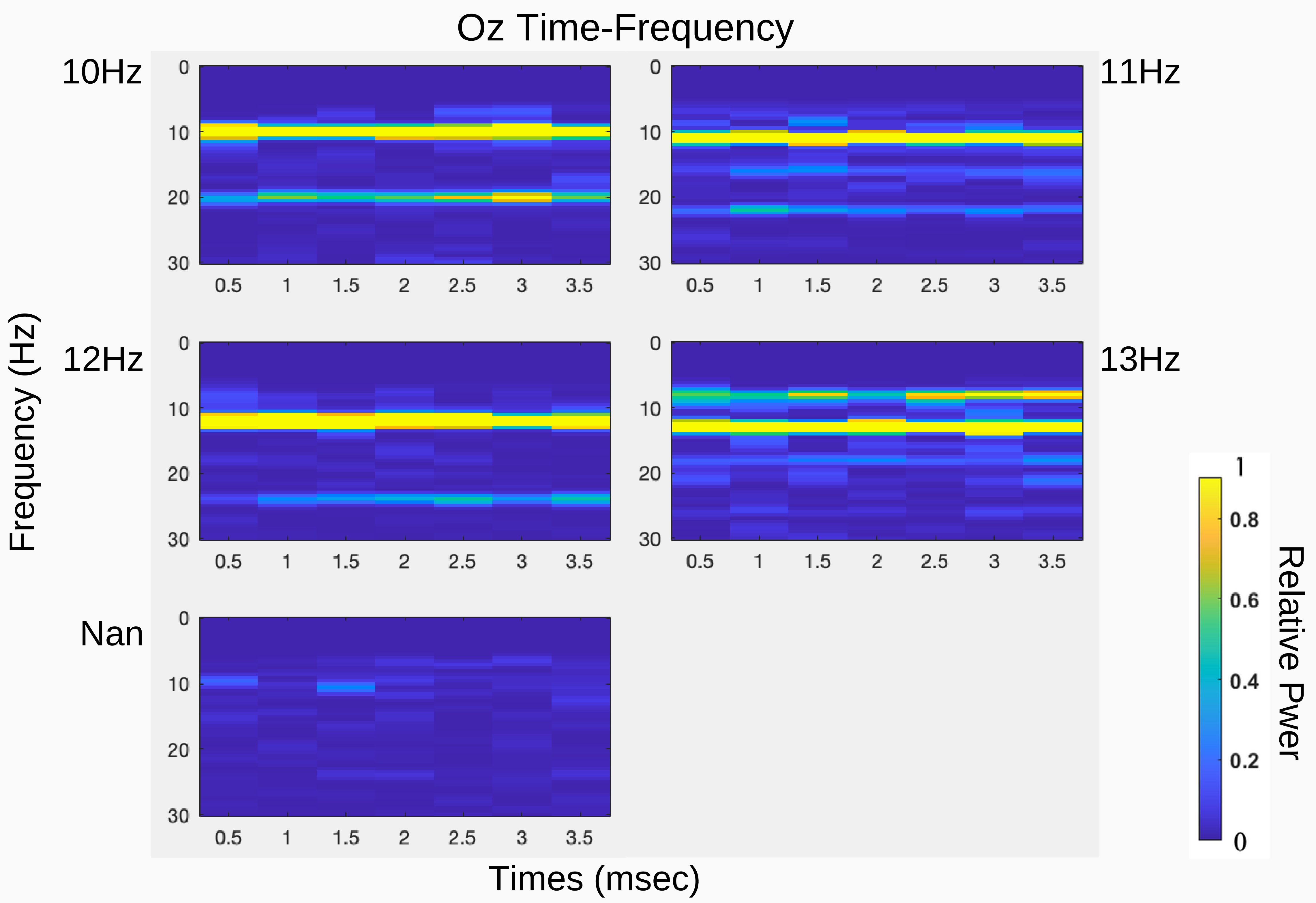
Practice 3:
SSVEP Temporal Profile

- 針對SSVEP實驗無法用時域分析看出不同target之間的差異

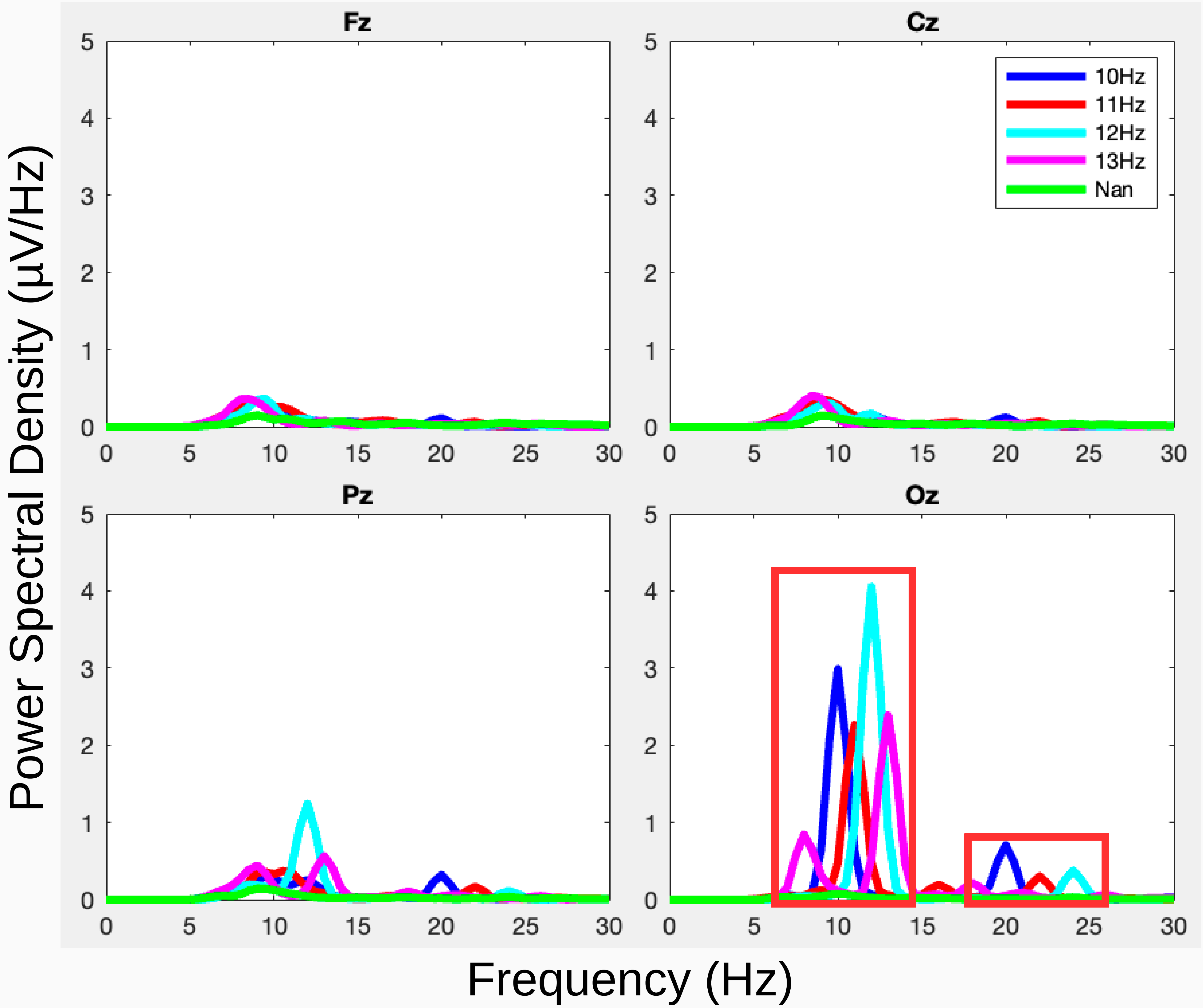


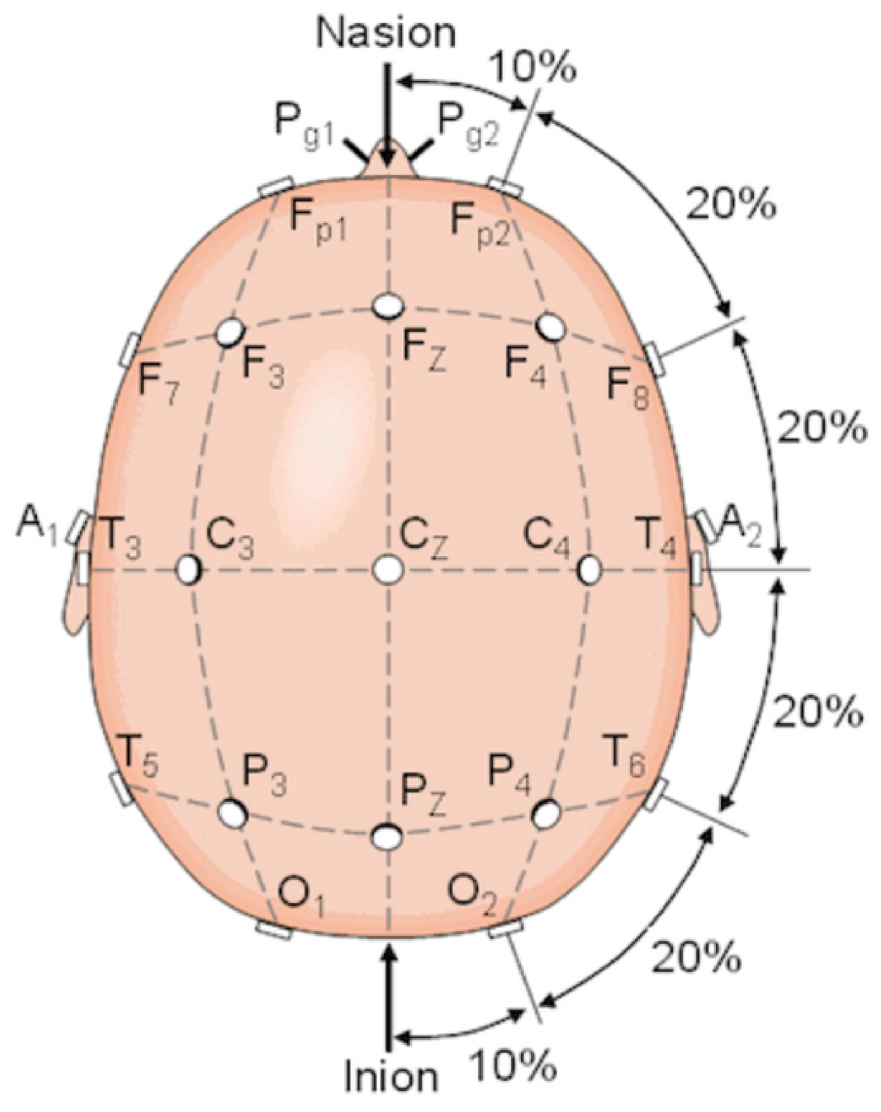
Practice 4: Time-Frequency Analysis

- 透過頻域分析可以看出不同target下的first harmonic和second harmonic



Practice 5: SSVEP Spectral Profile



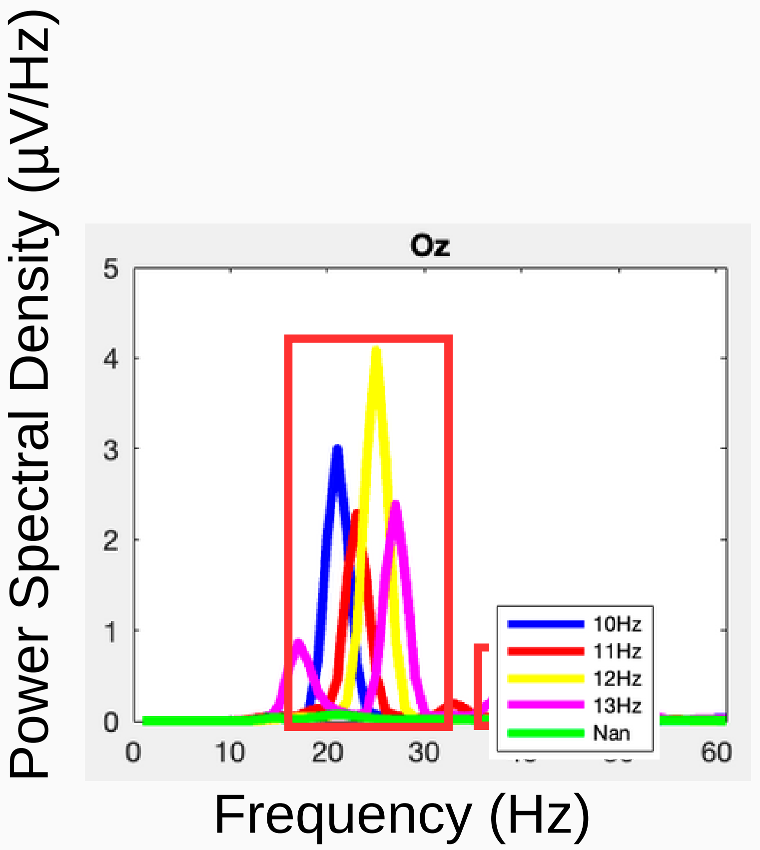


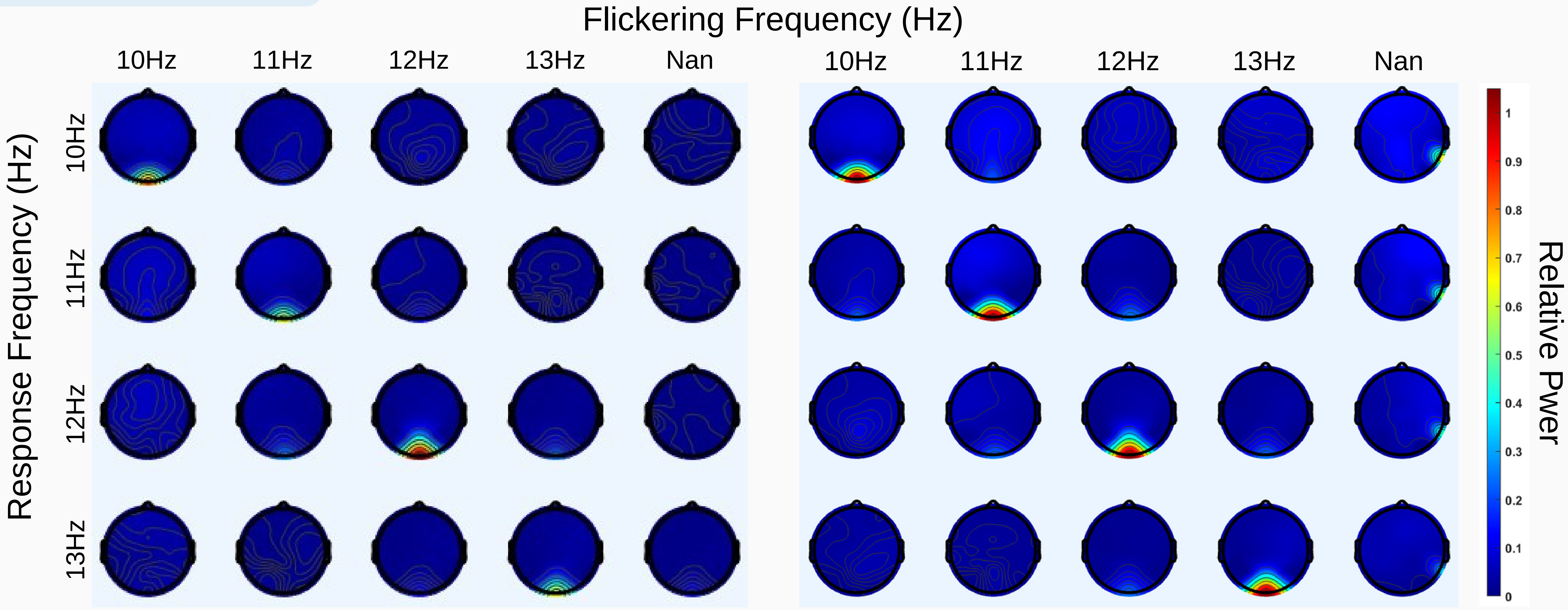
- Occipital lobe(枕葉)
  - 視覺功能中樞

→ Oz在四個event下的peak都較高

Practice 6:
SSVEP Topographic Mapping

腦區的能量變化

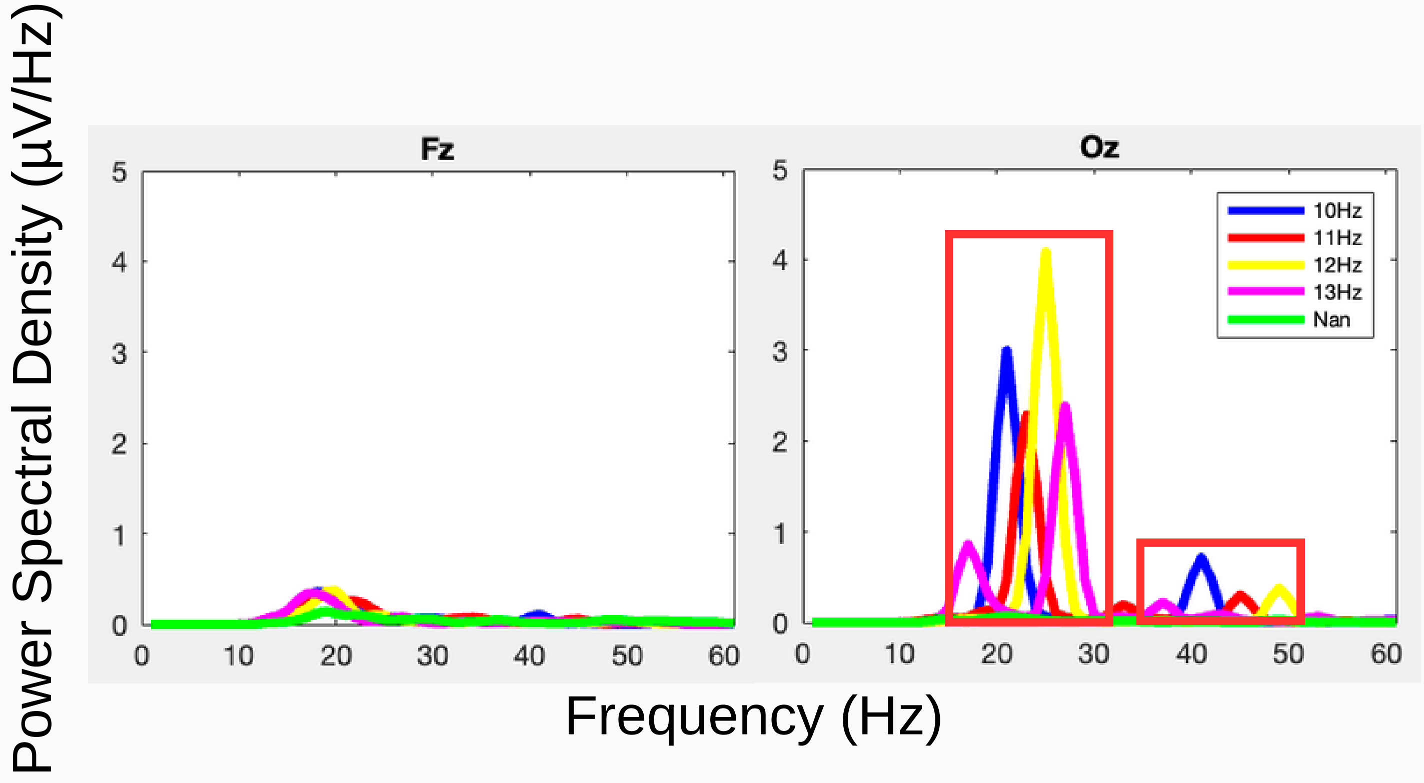


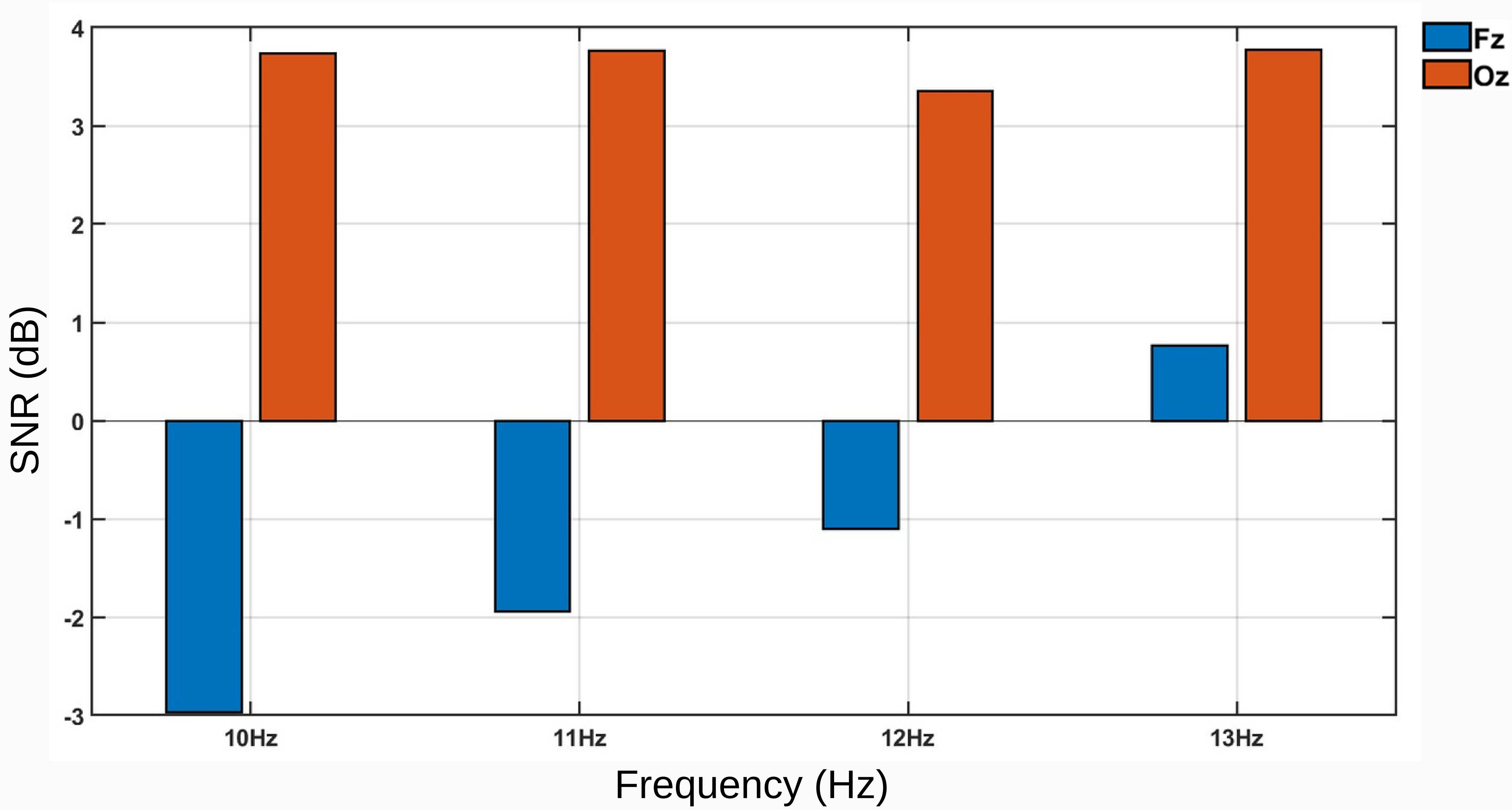


只針對10Hz、11Hz、12Hz、13Hz頻帶做正規化

針對全部頻帶做正規化

Practice 7:
SSVEP Signal-to-Ratio (SNR)

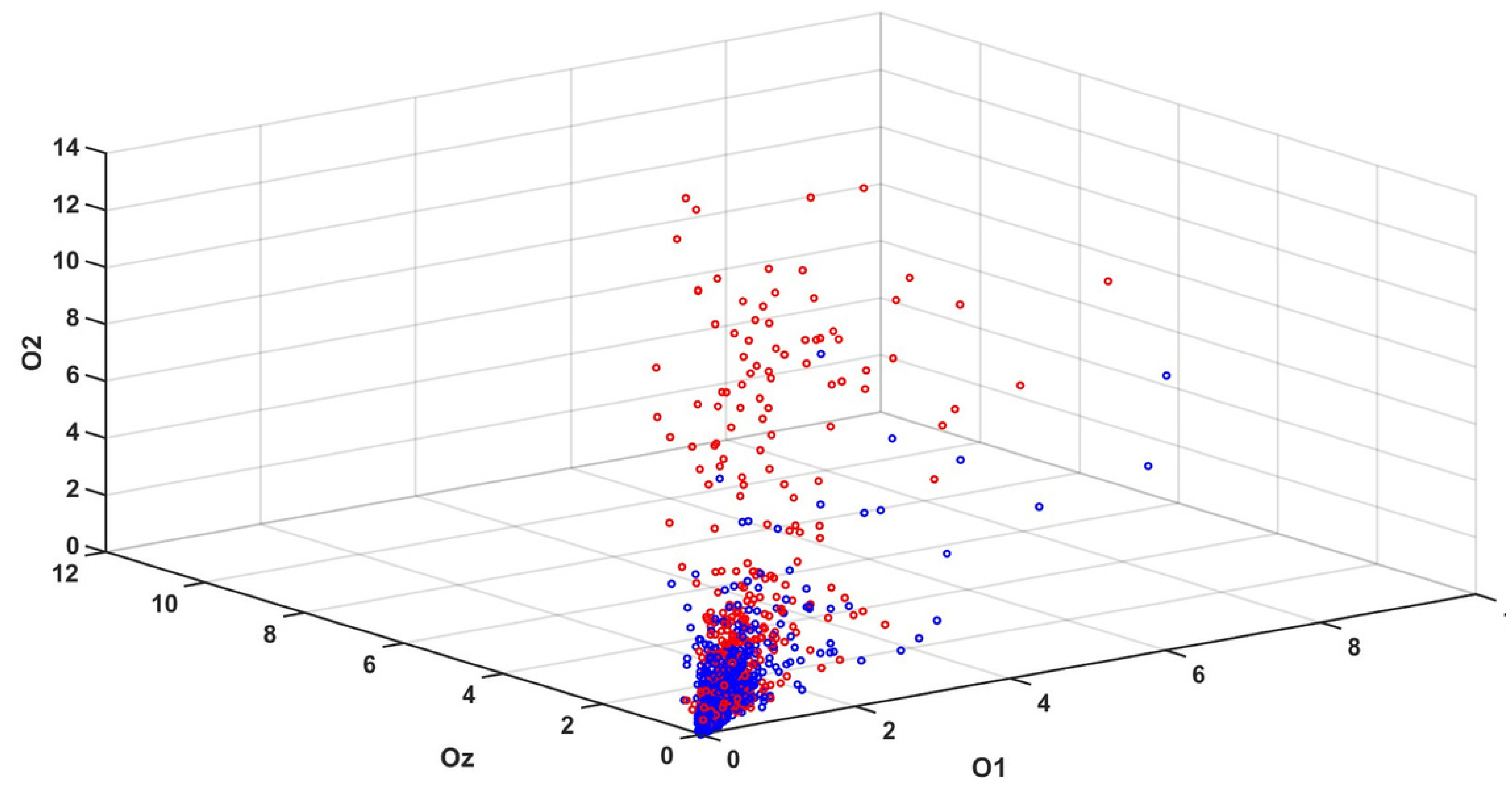




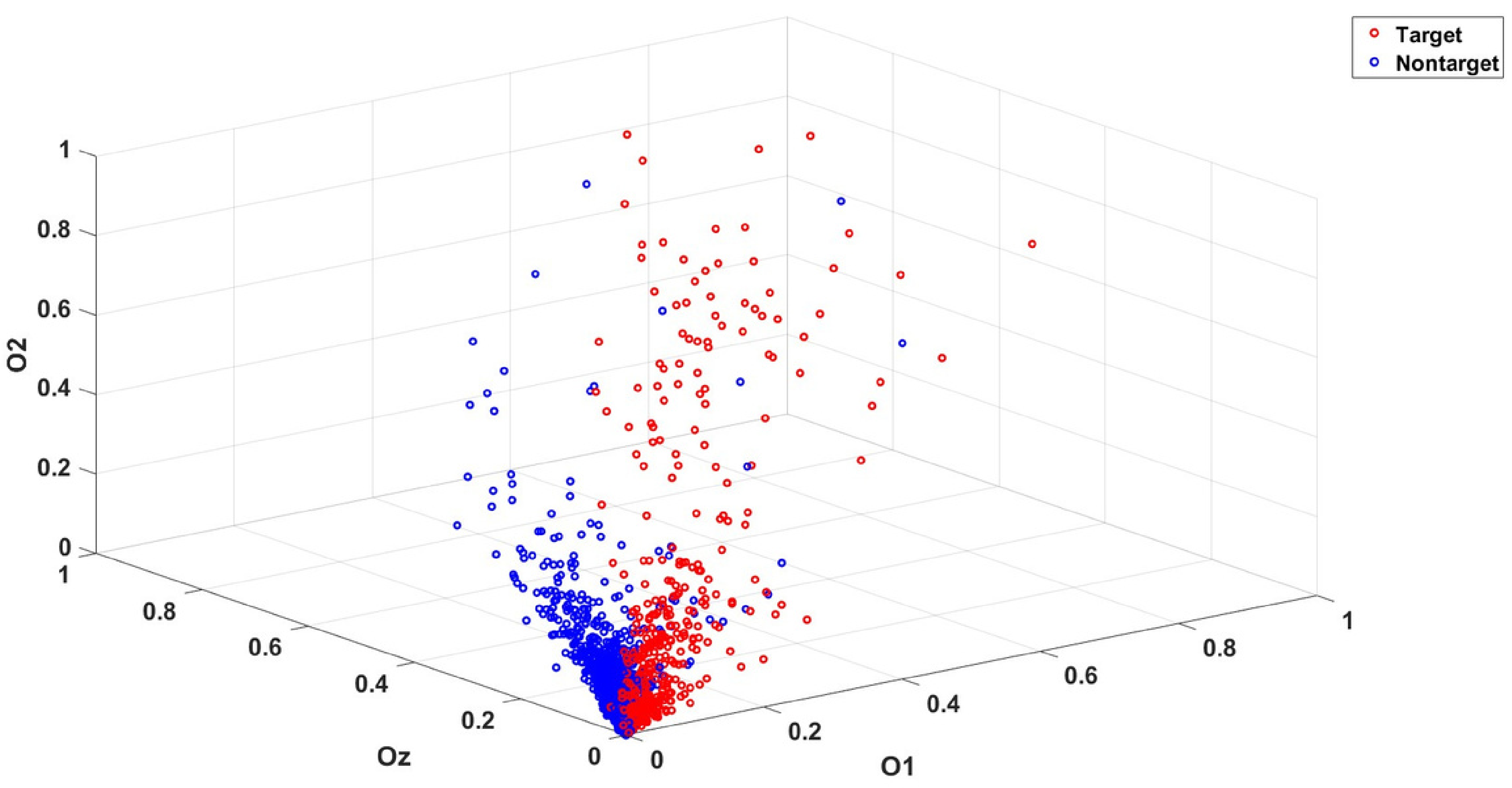
Practice 8:
SSVEP Feature Space

All Trials

Non-normalization

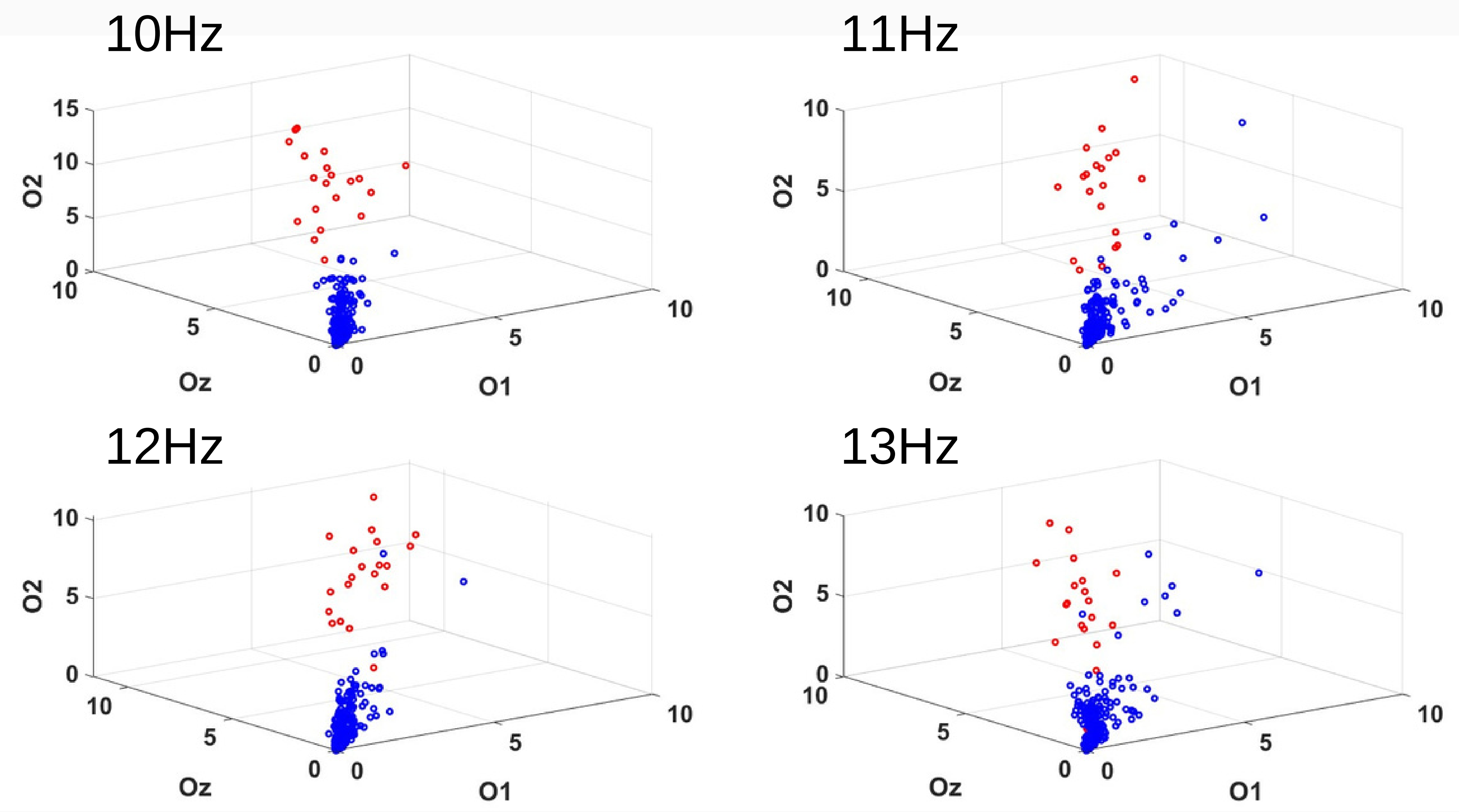


normalization

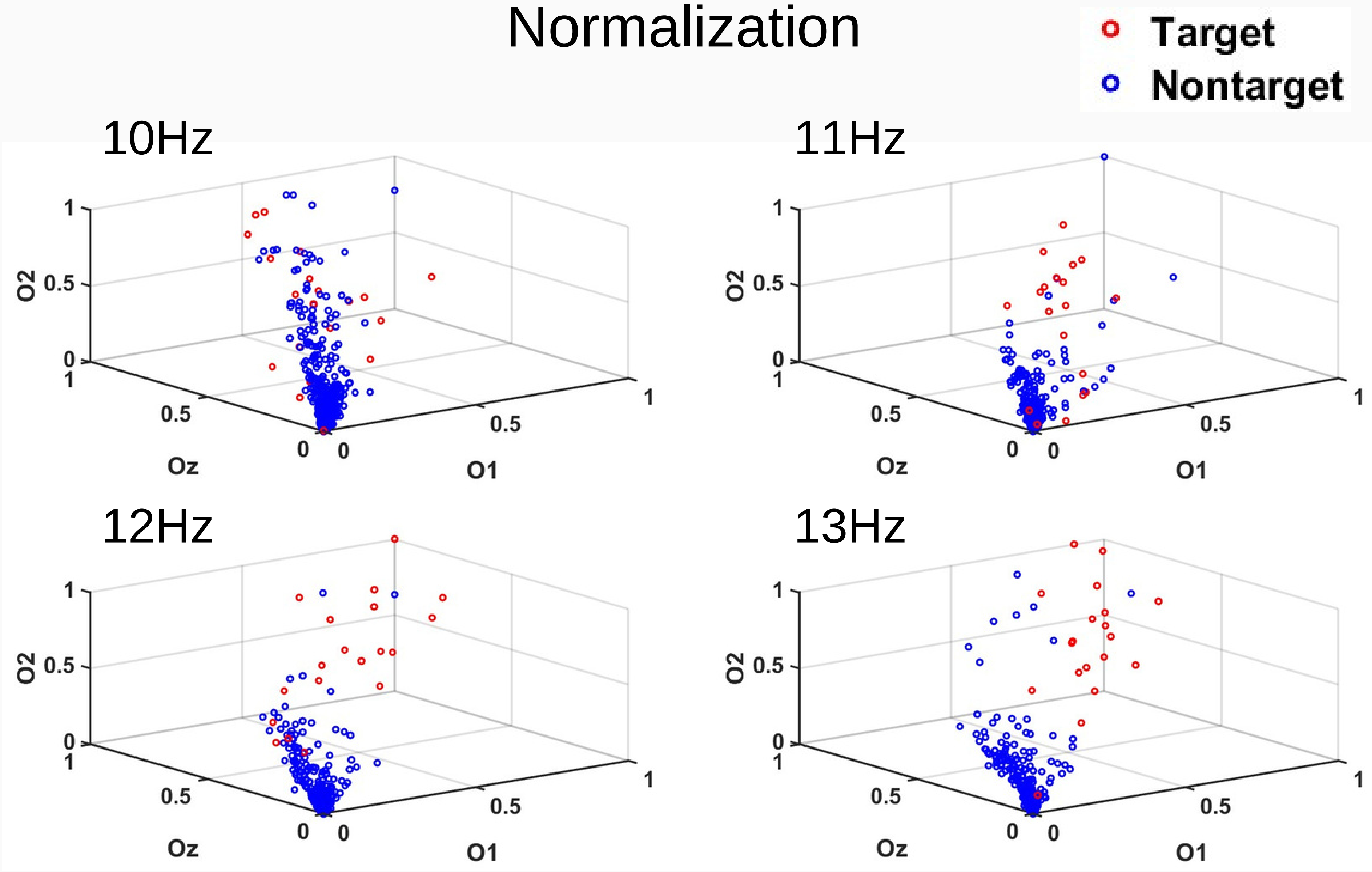


Practice 8:
SSVEP Feature Space

Non-normalization



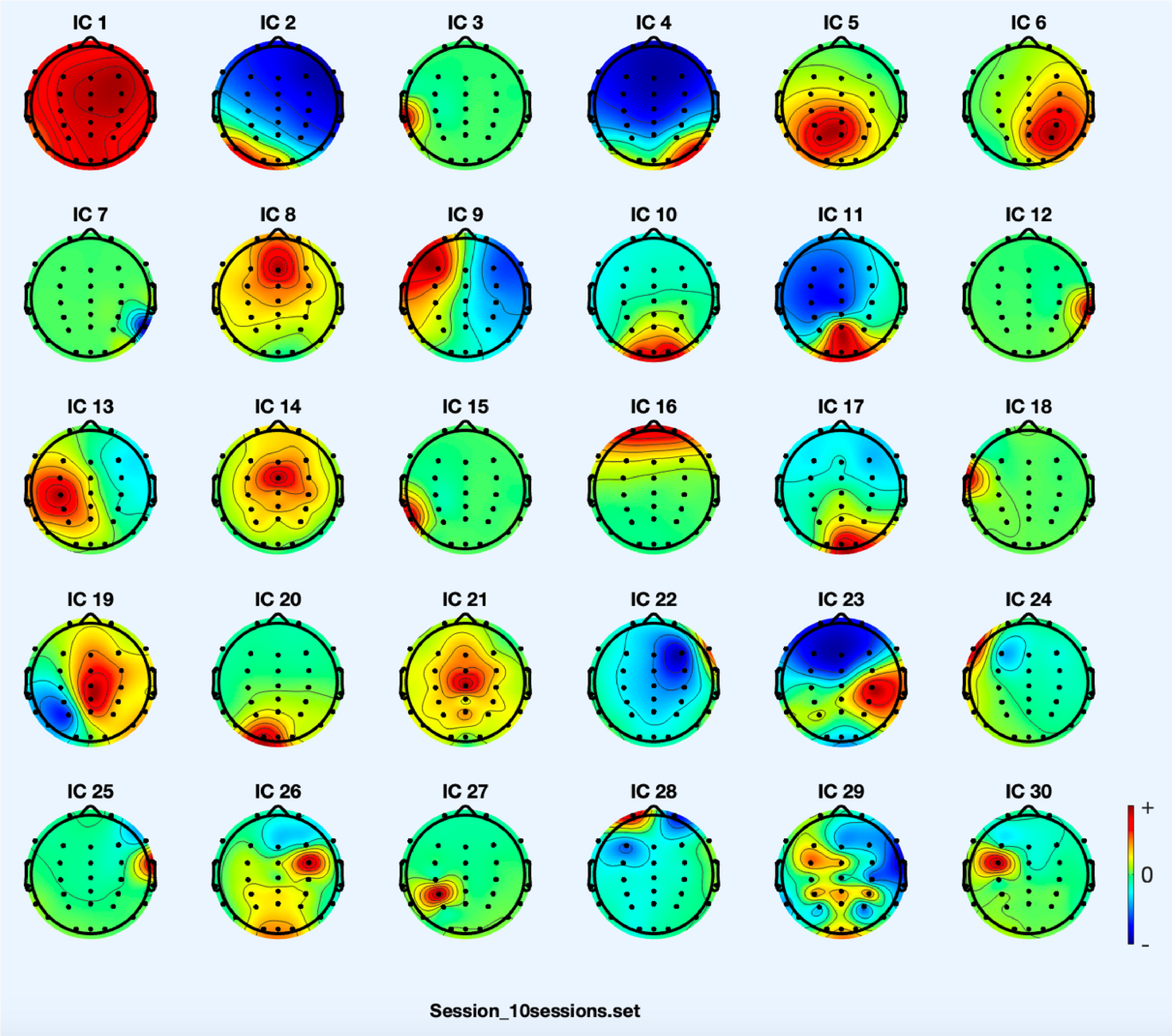
Normalization



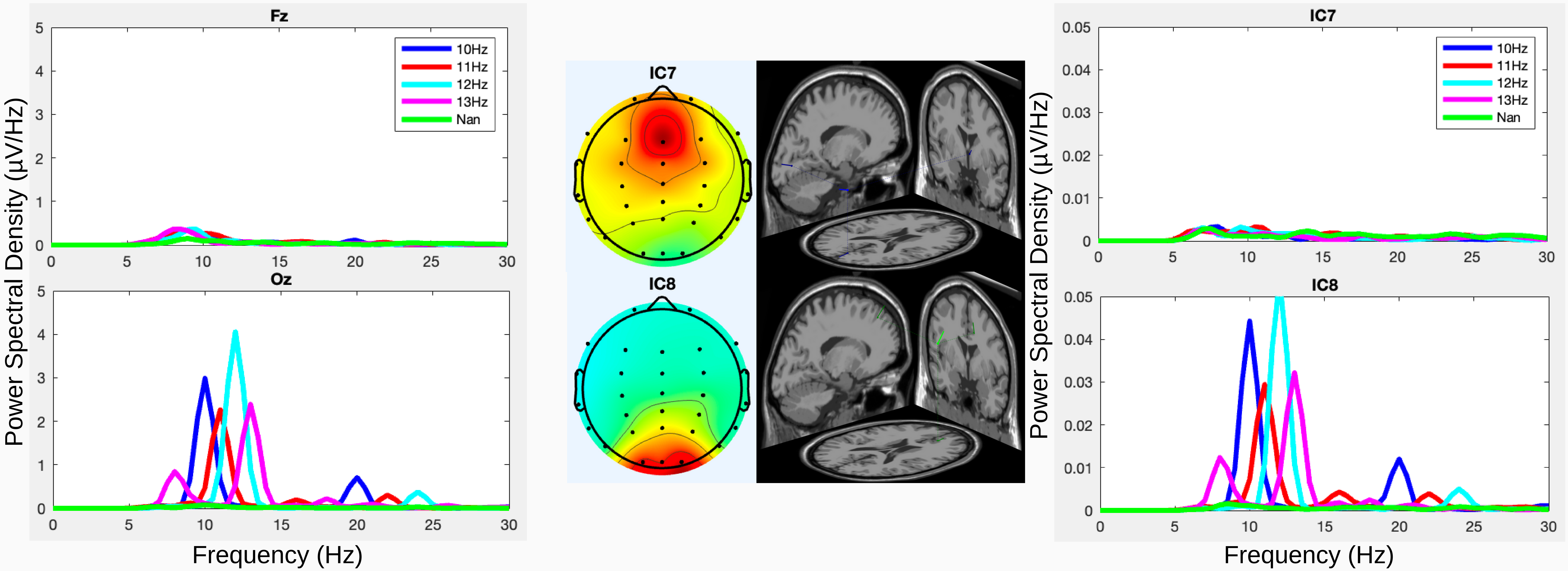
SOURCE ANALYSIS

Steady State Visually Evoked Potentials (SSVEP)

Independent Component Analysis (ICA)

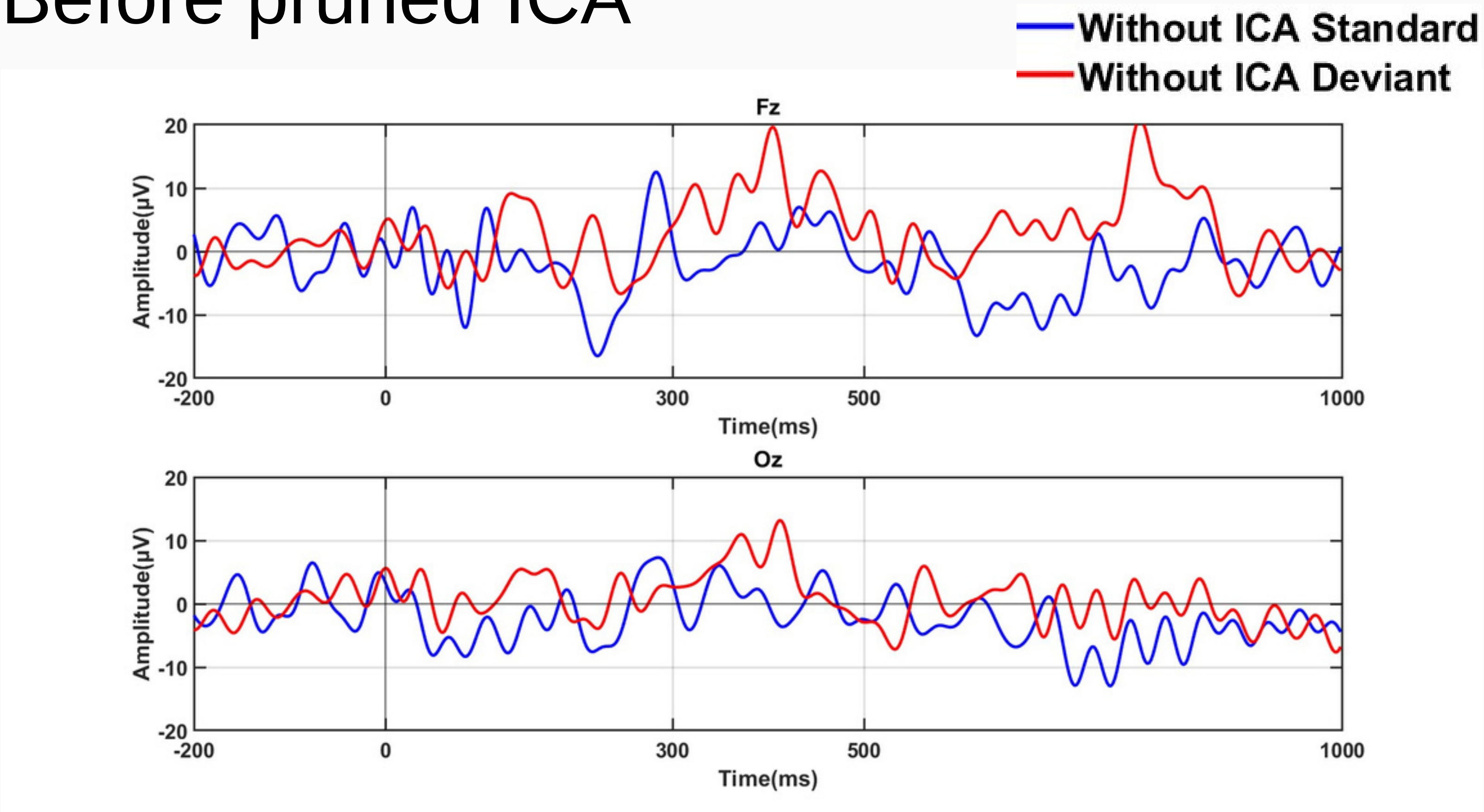


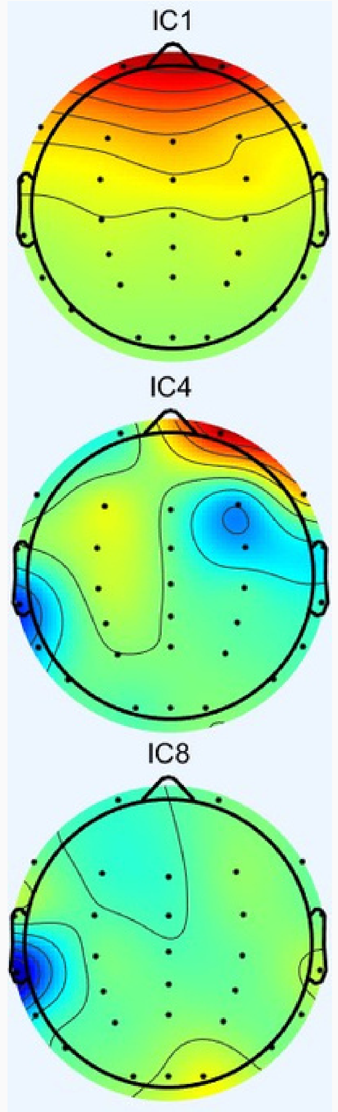
Compare SSVEP Reponses using Channel-Level vs. Source-Level Analysis



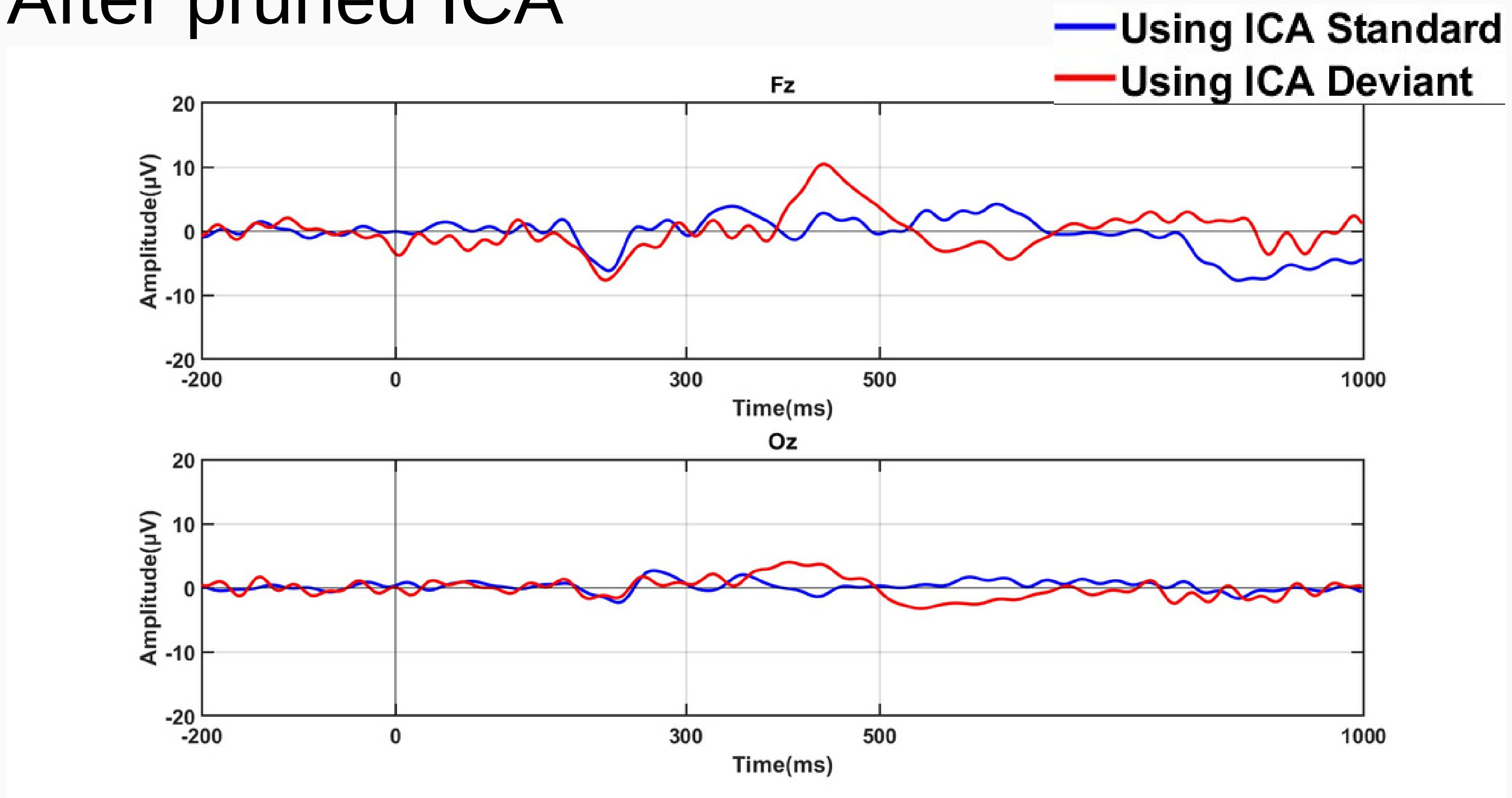
Compare ERP
Before and After ICA Artifact Removal

Before pruned ICA

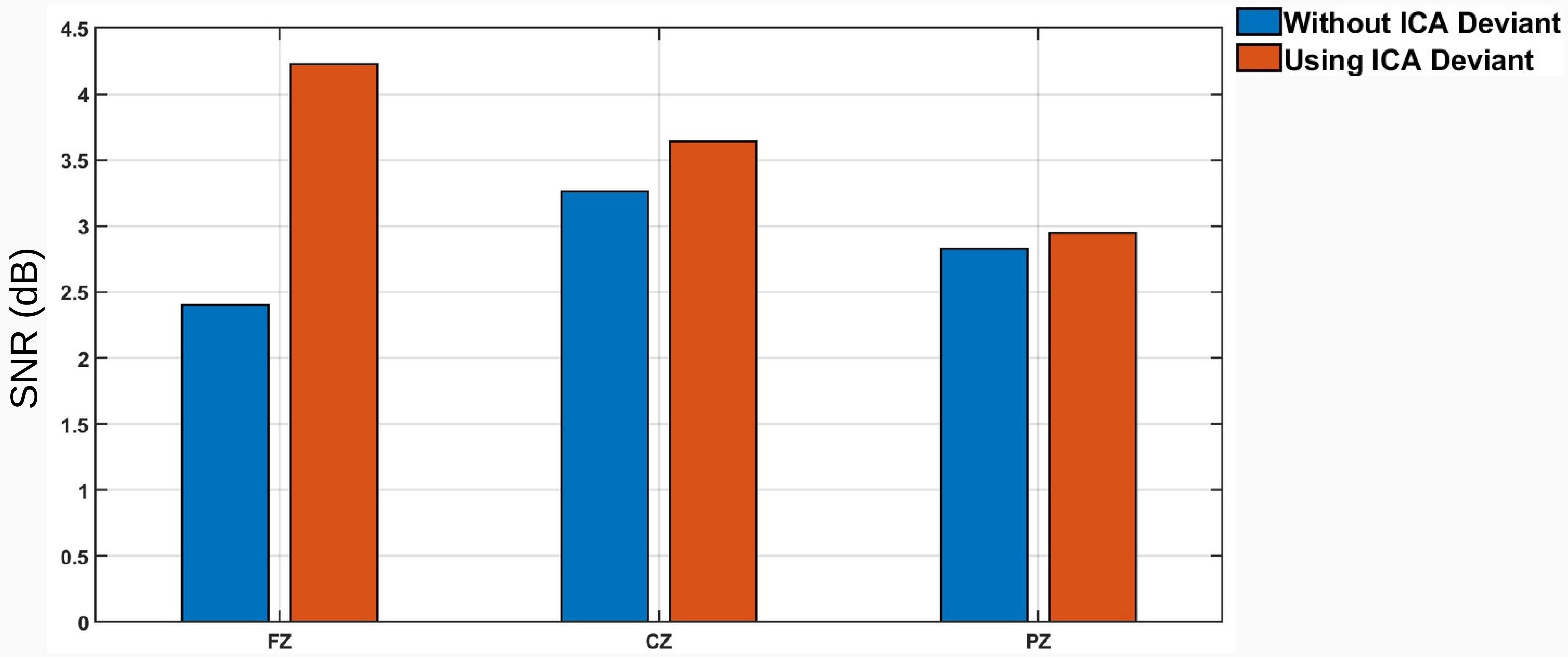




After pruned ICA



Compare SNR
Before and After ICA Artifact Removal



MACHINE LEARNING FRAMEWORK

Steady State Visually Evoked Potentials (SSVEP)

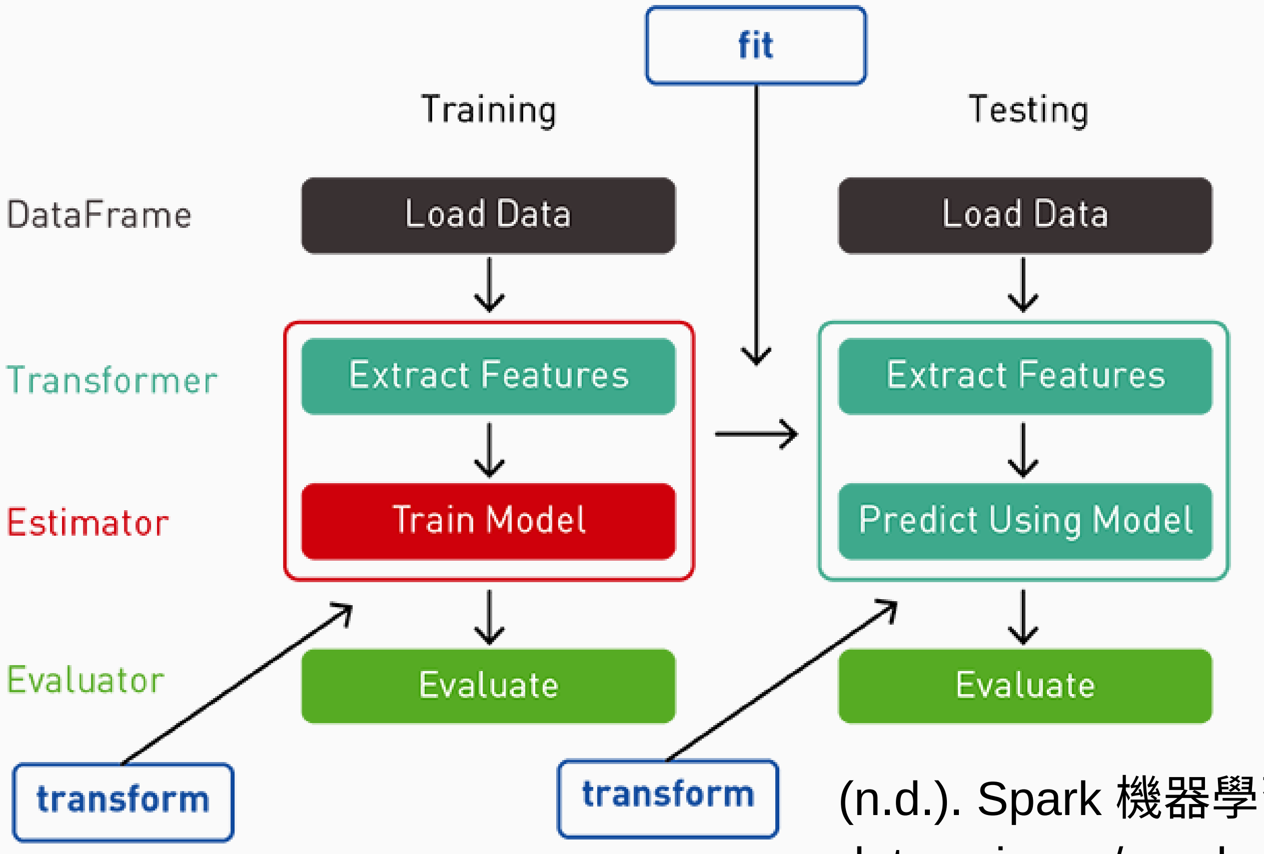
Introduction

從資料中去學習一個模式，來去預測新的任務

ML is the study of algorithms that

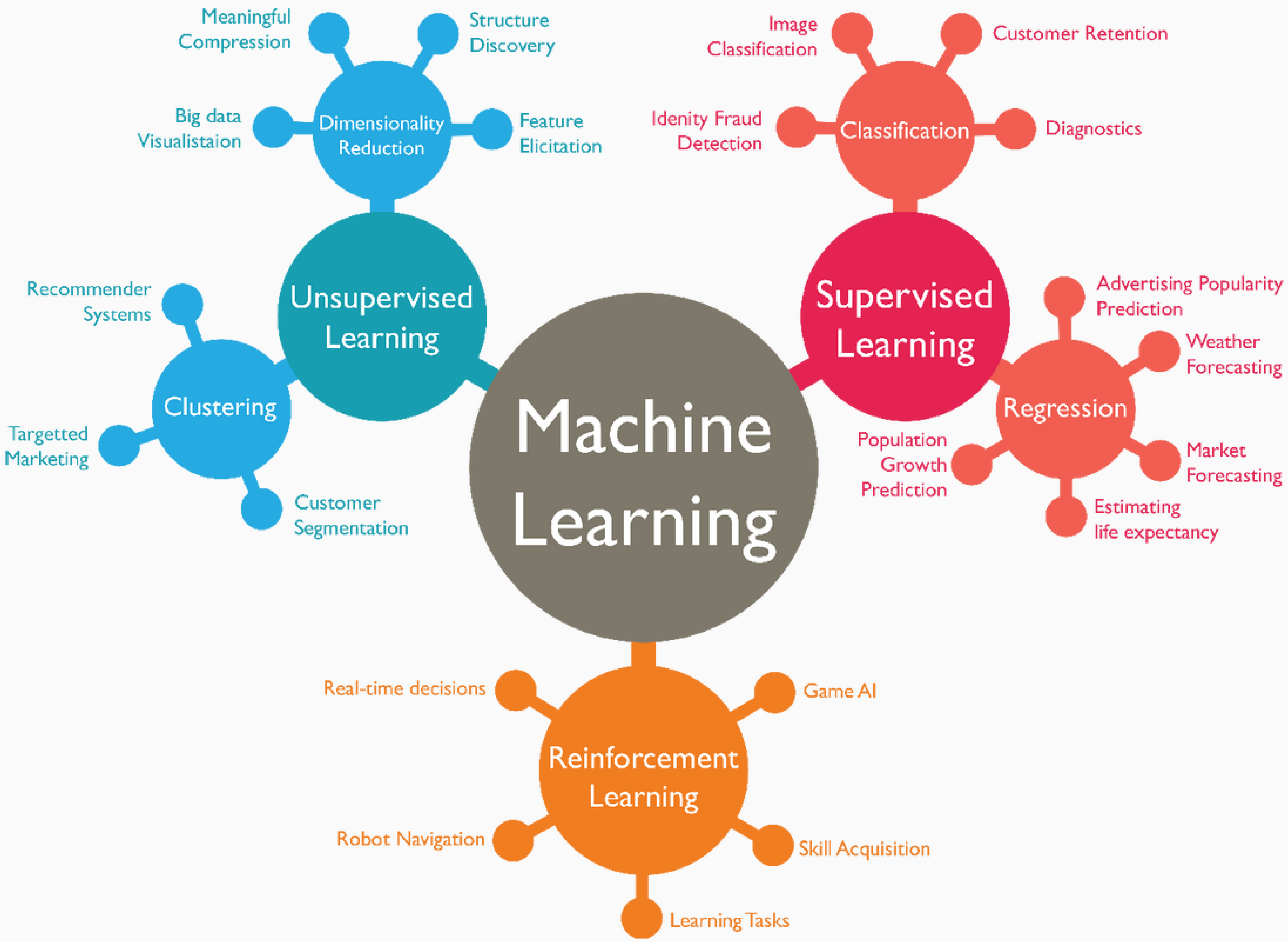
- improve Performance P
- at some task T
- with experience E

A well-defined task is given by <P,T,E>



(n.d.). Spark 機器學習預測分析教學. NVIDIA. https://www.nvidia.com/zh-tw/ai-data-science/spark-ebook/predictive-analytics-spark-machine-learning/

https://wordstream-files-prod.s3.amazonaws.com/s3fs-public/machine-learning.pngt

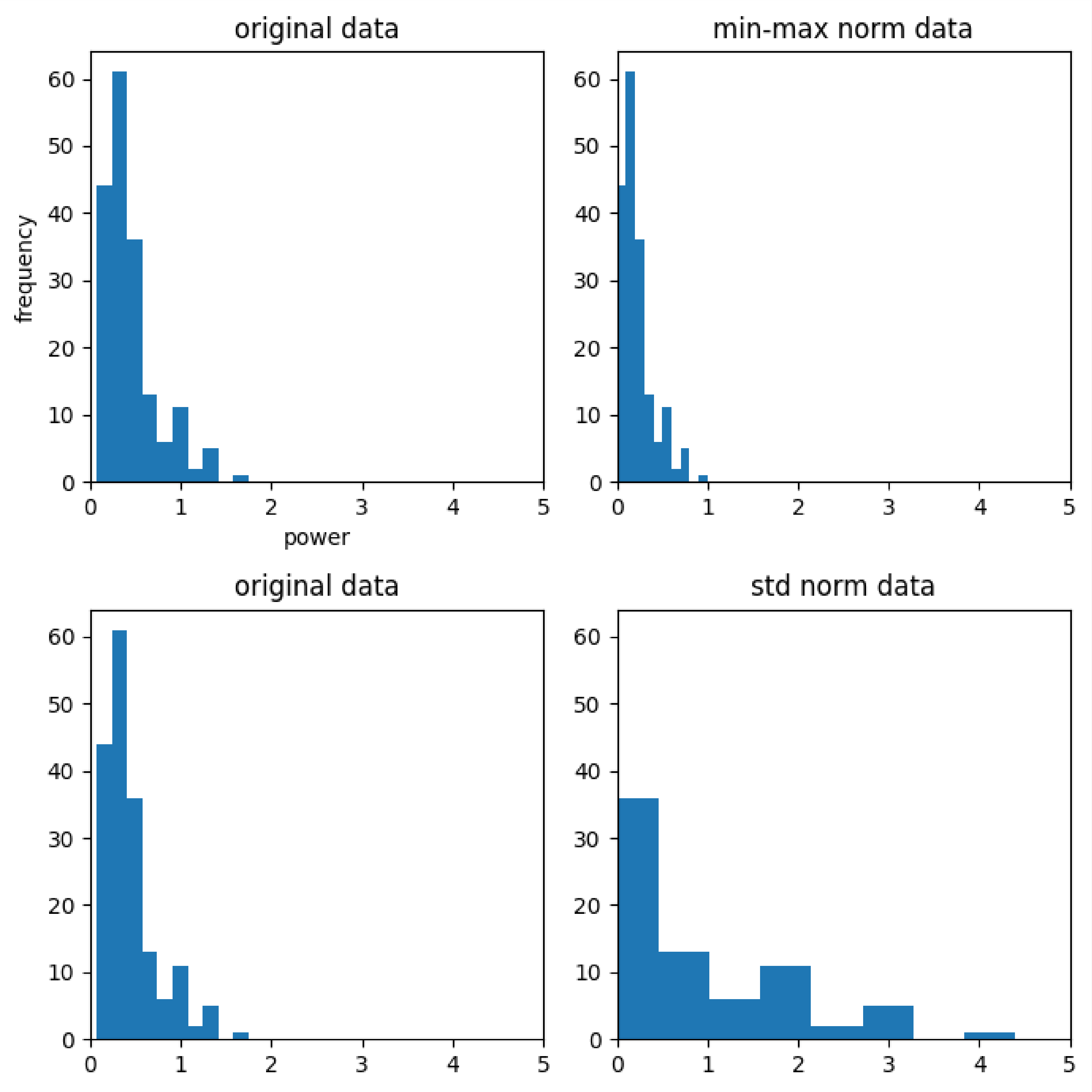


Practice & Analysis

We have some problems with MATLAB... , so we switch to Python instead!

- **Loading SSVEP data and use unsupervised model (tSNE) to visualize data**
- **Filter the data and select the target channel source.**
  Then, perform short-time Fourier's transform.
- **Use supervised model (KNN) to cluster the trials into K clusters**
  - Split data into training and testing dataset
  - Cross Validation (Leave-one Trial Out)
  - Evaluation (accuracy, recall, precision, confusion matrix)
- **Normalize**
- **CCA (the correlation b / w 2 different datasets)**
- **Optimize model**

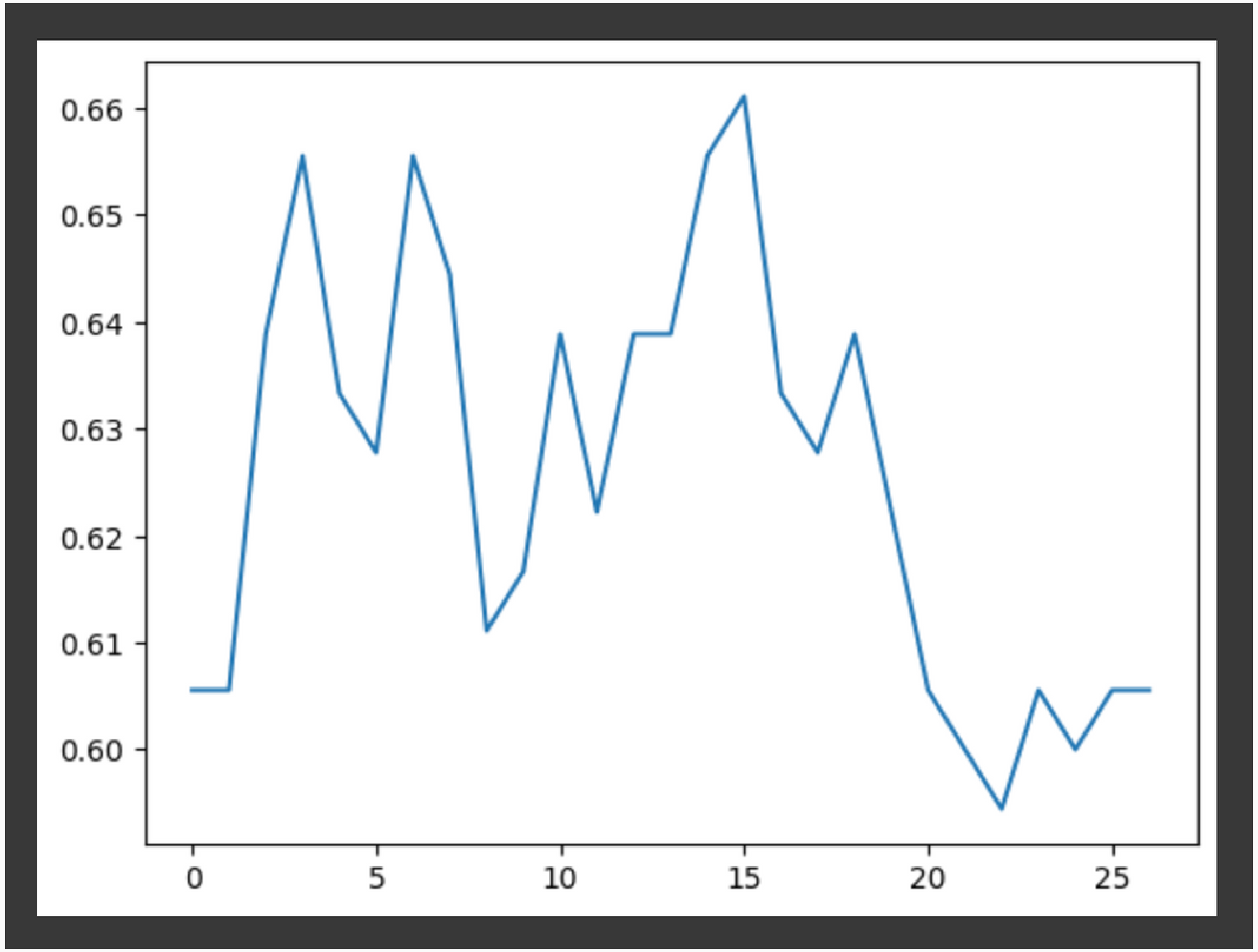
Practice & Analysis Normalization



- If you want to center your data around 0 and standardize it's variance, use standard normalization.
- If you want to scale your data to a specific range (e.g., between 0 and 1), use min-max normalization.
- Normalized via min-max won't affect the original distribution
- Normalized via standard will change the distribution

Practice & Analysis Example - LTO

```python
Y_pred_all = []
for k_index in range(len(k_values)):
  k = k_values[k_index]

  # initialize variables to store all the predicted labels
  Y_pred_all = []

  # implement LTO (Leave-one Trial Out)
  for i in range(0, len(X_cca)):
    # be careful to the dimension here
    X_train = np.concatenate((X_cca[:i, :], X_cca[i+1:, :]), axis=0)
    Y_train = np.concatenate((Y[:i], Y[i+1:]), axis=0)
    X_test = X_cca[i, :]
    Y_test = Y[i]

    # combine all the pred
    # leave one trial to test
    knn_model = KNeighborsClassifier(n_neighbors=k)
    knn_model.fit(X_train, Y_train)
    Y_pred = knn_model.predict(X_test.reshape(1, X_test.shape[0]))
    Y_pred_all.append(Y_pred[0])

  # compute overall accuracy
  overall_acc = accuracy_score(Y, Y_pred_all)
  all_acc.append(overall_acc)
```



OVERALL PROJECT DEMO

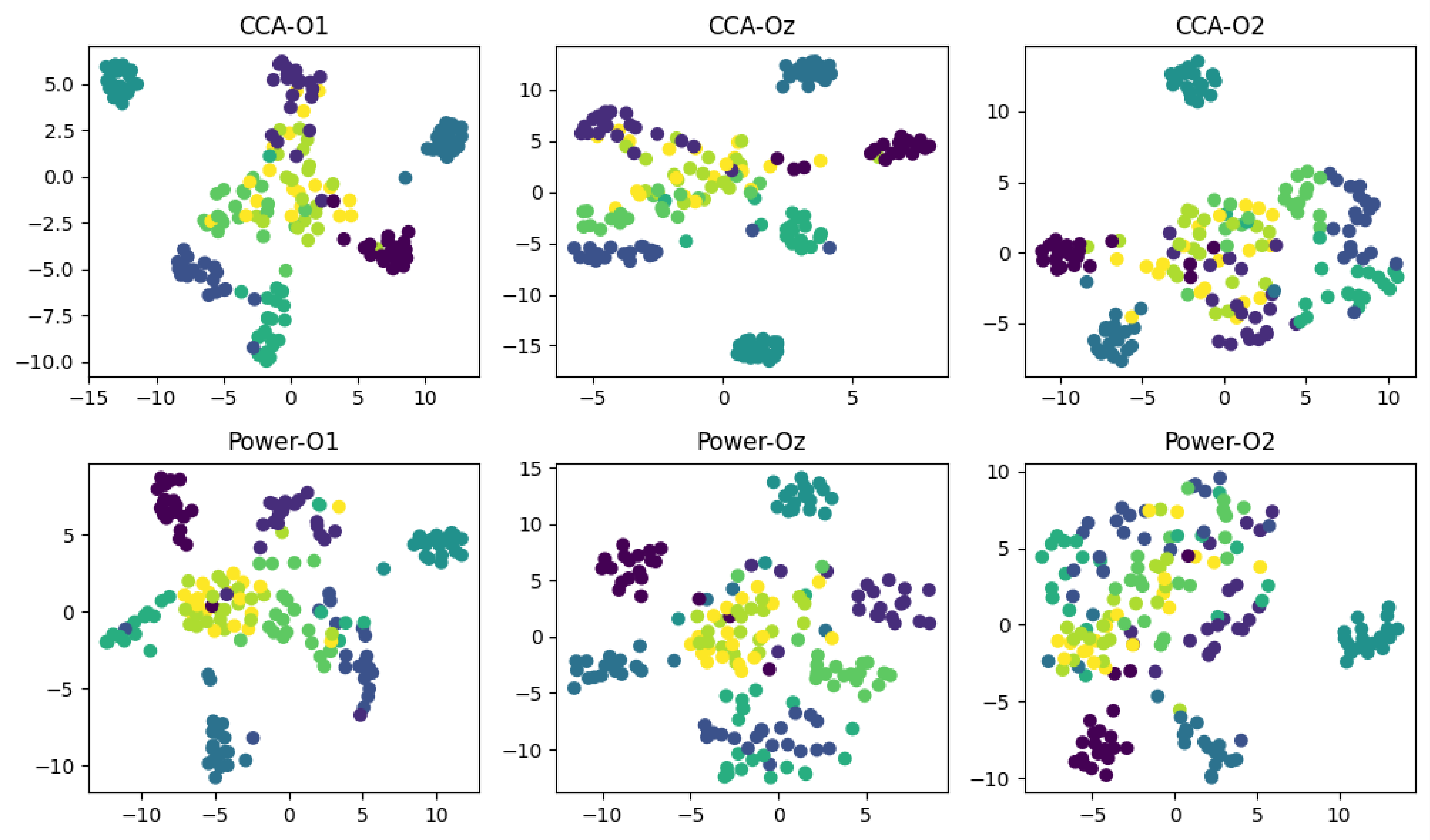
Analysis Strategy

1. **Compare the performance between different ML models (SVM & KNN)**
    a. 使用不同性質的 ML model，來比較性能，並探討影響效能的原因
2. **Compare the performance between applying CCA and Power datas**
    a. 使用不同的 Source Data，來比較其分類成效
3. **Compare the performance between choosing Oz, O1+Oz+O2 channels and O1, O2 channels**
    a. 前面 20 張腦譜圖都是在 Occipital 功率比較強，所以我們想嘗試在 Channel 把 O1 跟 O2 加進去是否會影響分類成效
    b. 不同 Channels 間的差別

Analysis Strategy

比較基準	model	dataset	channel
ML model	**SVM & KNN**	CCA	Oz
Dataset	KNN	**CCA & Power**	Oz
Channel	KNN	CCA	**Oz & O1+Oz+O2 & O1 & O2**

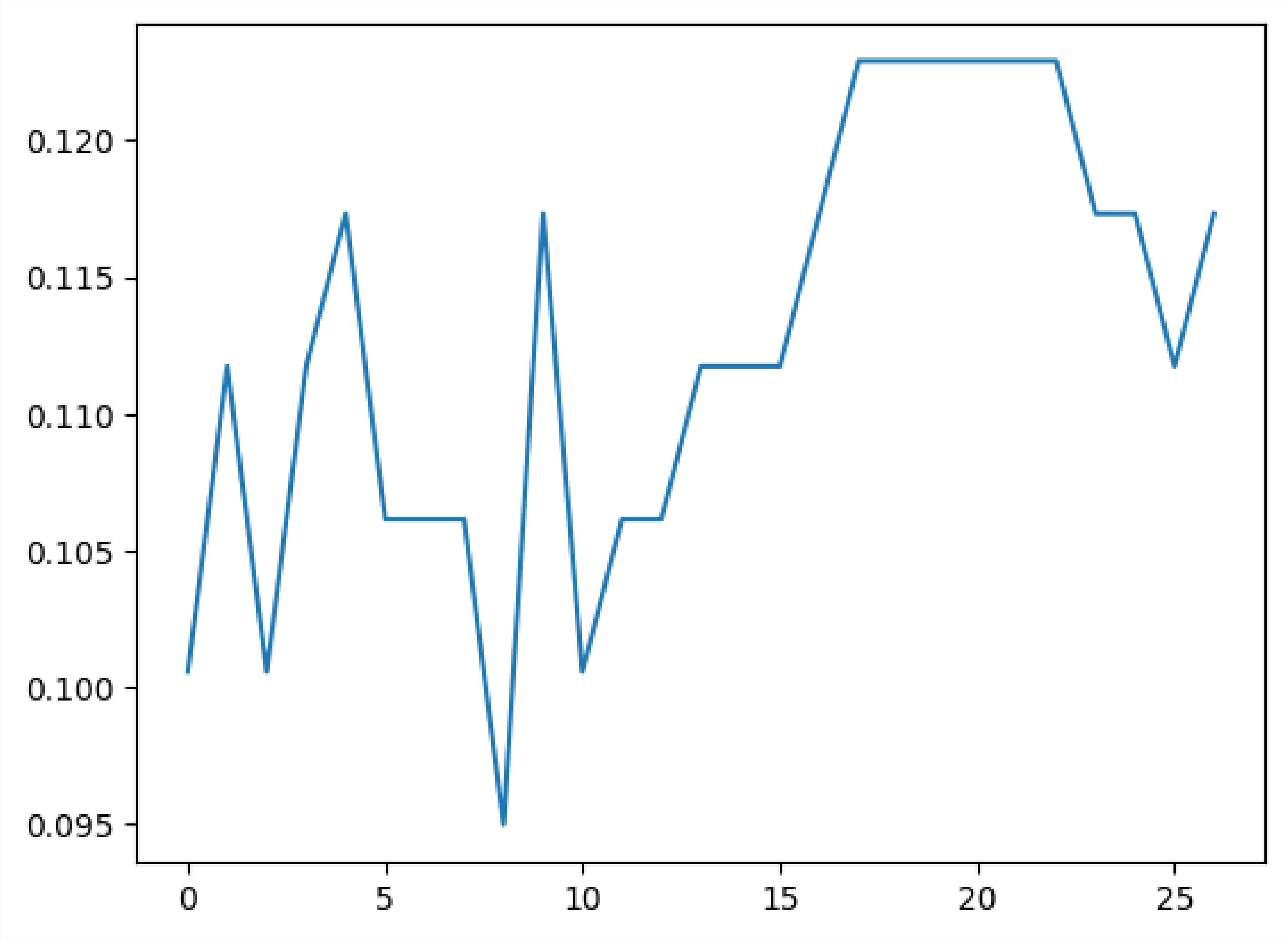
Data exploration



1. 定性分析
2. TSNE 分群 (Feature Space)
3. 確認這樣的 Features 選擇是否足夠清晰分群
4. 在訓練前對資料的探勘

Data exploration

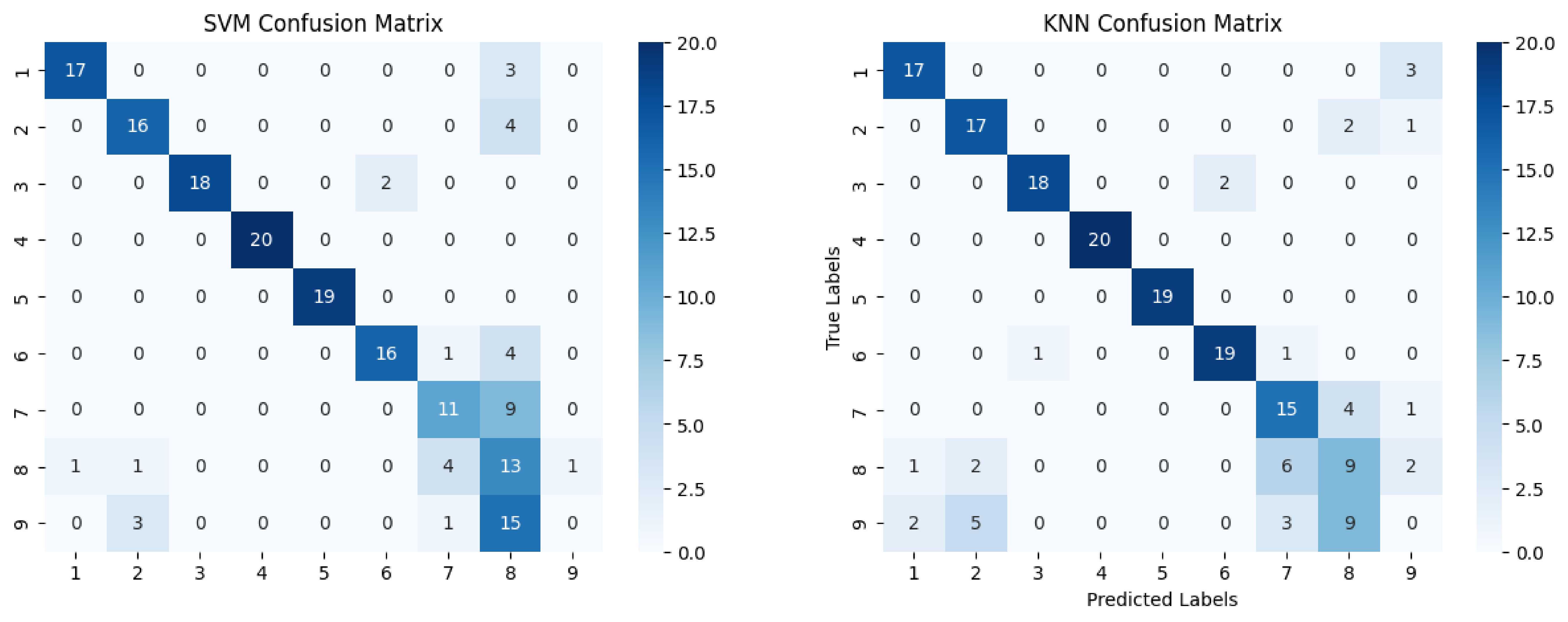
```python
y_k_pred_all = []
all_k_acc = []
for k_index in range(len(k_values)):
  k = k_values[k_index]

  # initialize variables to store all the predicted labels
  y_k_pred_all = []

  # implement LTO (Leave-one Trial Out)
  for i in range(len(X_ccaOz)):
    # be careful to the dimension here
    X_train = np.concatenate((X_ccaOz[:i, :], X_ccaOz[i+1:, :]), axis=0)
    Y_train = np.concatenate((Y[:i], Y[i+1:]), axis=0)
    X_test = X_cca[i, :]
    Y_test = Y[i]

    # combine all the pred
    # leave one trial to test
    knn_model = KNeighborsClassifier(n_neighbors=k)
    knn_model.fit(X_train, Y_train)
    y_k_pred = knn_model.predict(X_test.reshape(1, X_test.shape[0]))
    y_k_pred_all.append(y_k_pred[0])

  # compute overall accuracy
  overall_acc = accuracy_score(Y, y_k_pred_all)
  all_k_acc.append(overall_acc)
```



Case 1 :
Different ML Model Analysis

1. Training strategy : CV-LTO
2. **定量分析**

```
SVM Accuracy: 0.73
KNN Accuracy: 0.75

SVN Recall: 0.73
KNN Recall: 0.75

SVM Precision: 0.73
KNN Precision: 0.71
```

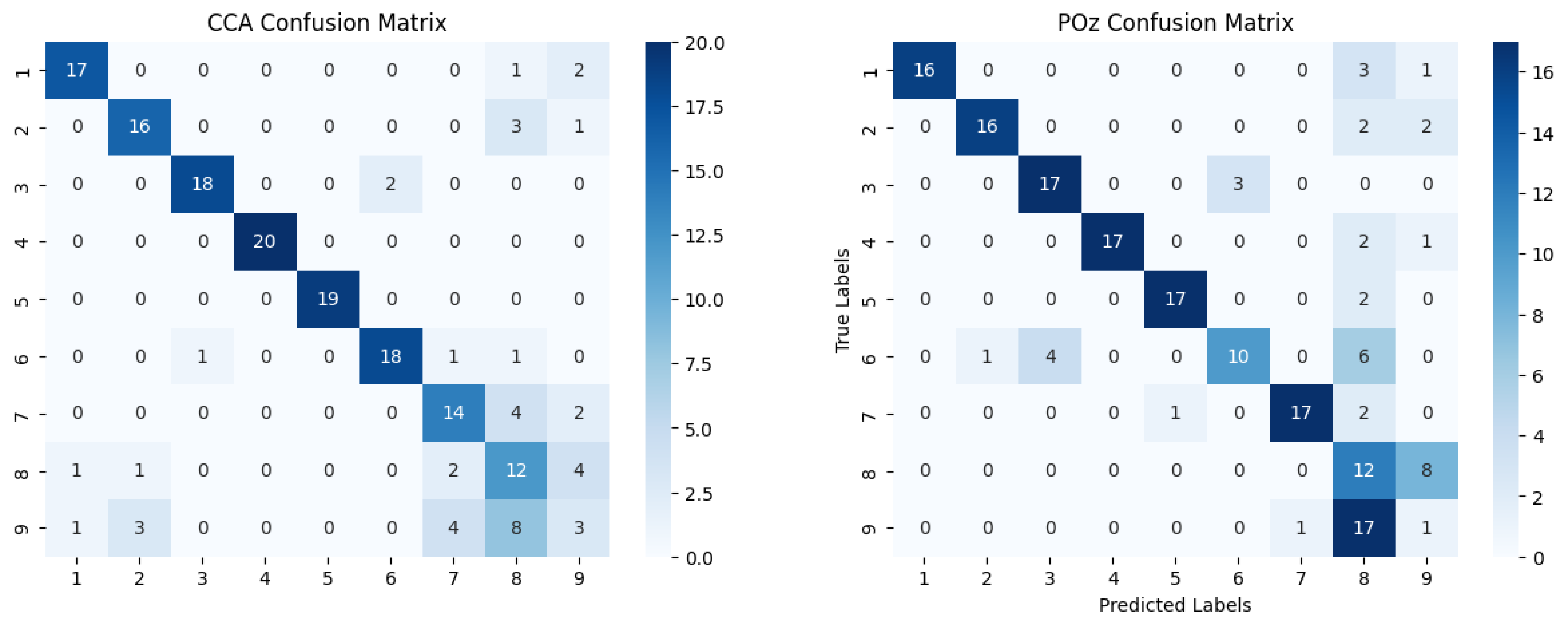


Case 1 : Different ML Model Finding

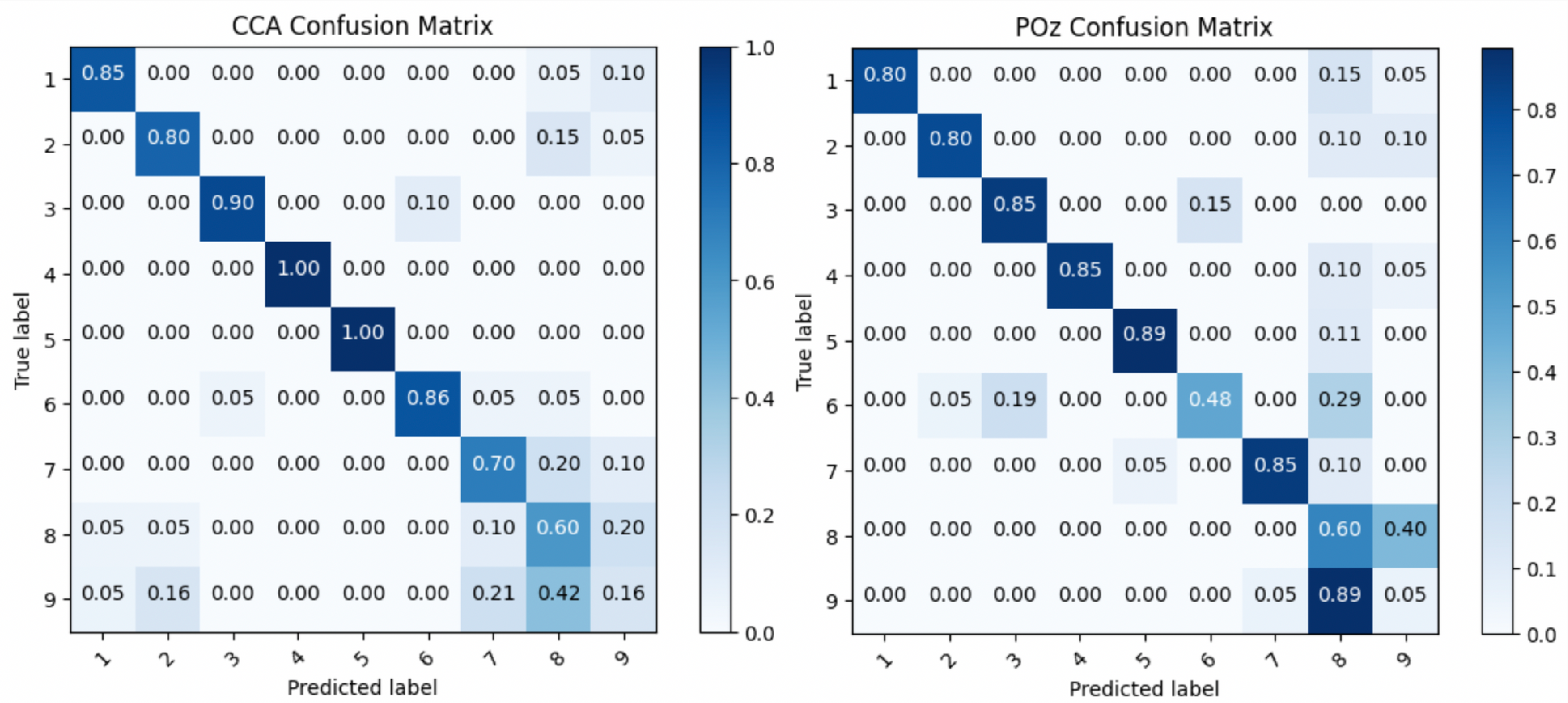
- You might find that SVM performs better when dealing with complex decision boundaries and noisy data.
- While KNN might work well in cases where the data distribution is clear and the decision boundary is irregular.
- 造成 KNN 效能較好的可能原因：
    - Data Size 較小
    - Data 分群界線明顯且為非線性分佈
- SVM參數尚未做最佳化 (kernel, ...)

Case 2 :
Different Dataset Analysis

```
CCA Accuracy: 0.77
POz Accuracy: 0.69

CCA Recall: 0.77
POz Recall: 0.69

CCA Precision: 0.77
POz Precision: 0.75
```



Normalized

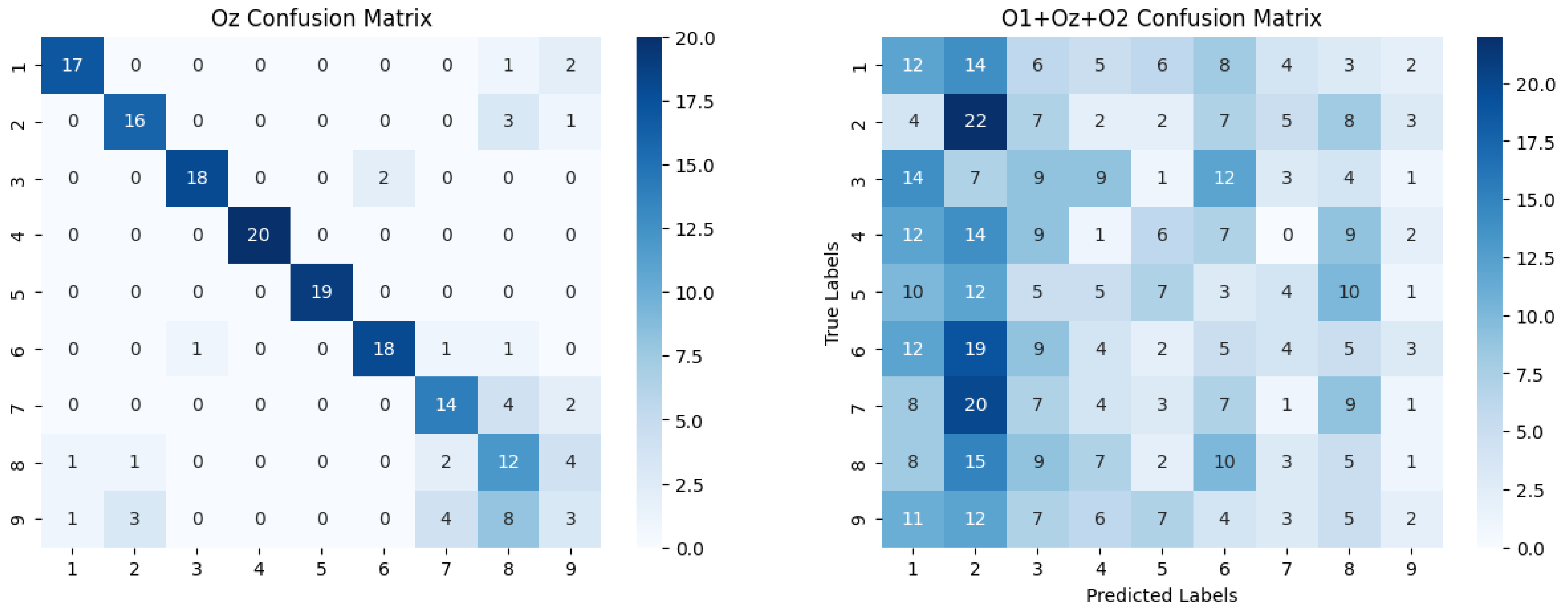

CCA Confusion Matrix
0.85 0.00 0.00 0.00 0.00 0.00 0.00 0.05 0.10
0.00 0.80 0.00 0.00 0.00 0.00 0.00 0.15 0.05
0.00 0.00 0.90 0.00 0.00 0.10 0.00 0.00 0.00
0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00 1.00 0.00 0.00 0.00 0.00
0.00 0.00 0.05 0.00 0.00 0.86 0.05 0.05 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.70 0.20 0.10
0.05 0.05 0.00 0.00 0.00 0.00 0.10 0.60 0.20
0.05 0.16 0.00 0.00 0.00 0.00 0.21 0.42 0.16
True label
Predicted label
1 2 3 4 5 6 7 8 9
1.0 0.8 0.6 0.4 0.2 0.0
POz Confusion Matrix
0.80 0.00 0.00 0.00 0.00 0.00 0.00 0.15 0.05
0.00 0.80 0.00 0.00 0.00 0.00 0.00 0.10 0.10
0.00 0.00 0.85 0.00 0.00 0.15 0.00 0.00 0.00
0.00 0.00 0.00 0.85 0.00 0.00 0.00 0.10 0.05
0.00 0.00 0.00 0.00 0.89 0.00 0.00 0.11 0.00
0.00 0.05 0.19 0.00 0.00 0.48 0.00 0.29 0.00
0.00 0.00 0.00 0.00 0.05 0.00 0.85 0.10 0.00
0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.60 0.40
0.00 0.00 0.00 0.00 0.00 0.00 0.05 0.89 0.05
True label
Predicted label
1 2 3 4 5 6 7 8 9
0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

Case 2 : Different Dataset Finding

- 使用做過 CCA 的資料(CCA) 效果比 PSD 資料好的可能原因 :
  - CCA had calculated the correlation coefficient with sin wave and cos wave, so the data might be more distinctiveness than PSD data.
  - CCA 在判斷與 Sin 波和 Cos 波的相似度（與其他 dataset 的 correlation），間接地在某程度上抑制了雜訊
  - CCA 是選擇 Correlation 最高的 Coefficient，間接地做了較好的 Feature 選擇

Case 3 : Different Channels Analysis

```
Oz Accuracy: 0.77
O1+Oz+O2 Accuracy: 0.12

Oz Recall: 0.77
O1+Oz+O2 Recall: 0.12

Oz Precision: 0.77
O1+Oz+O2 Precision: 0.11
```

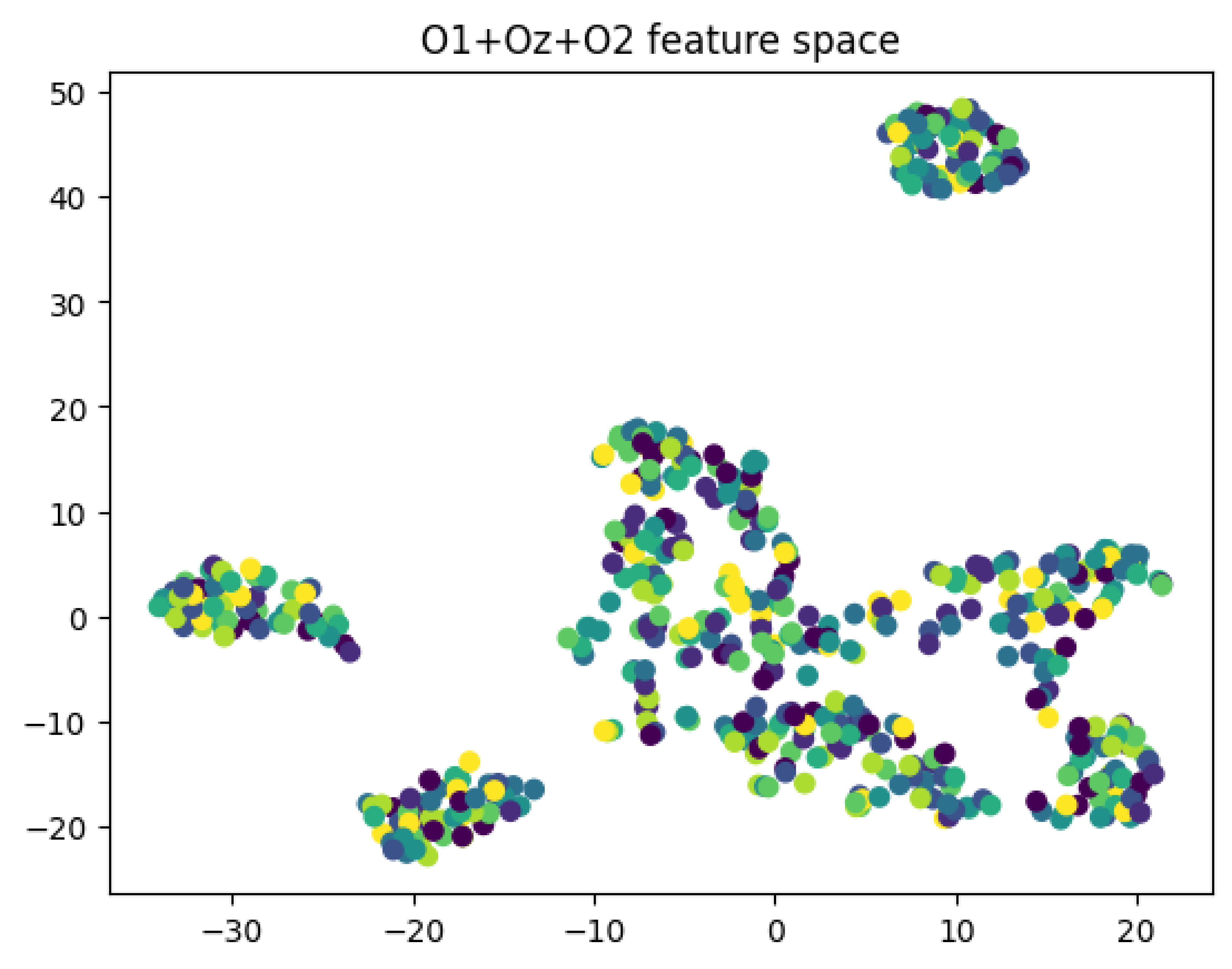


Case 3 :
Different Channels Finding

- 結果的 Accuracy 較低，或許是因為合併後的資料沒有比較一致的 Feature 行為，在 Feature Space的分佈不一樣。
- 將 Channel 的 Feature 接在 Stimulation Frequency 後面 (8 feat. -> 9 feat.)，但效果依舊不好。
- 若將Oz、O1、O2合併成 3D 的 vector，拿去做 FFNN 訓練做 Classification，可能 Accuracy 會有更好的表現。
- 檢查O1+Oz+O2的Feature Space (定性分析)

- 那如果只挑不同Channels呢？

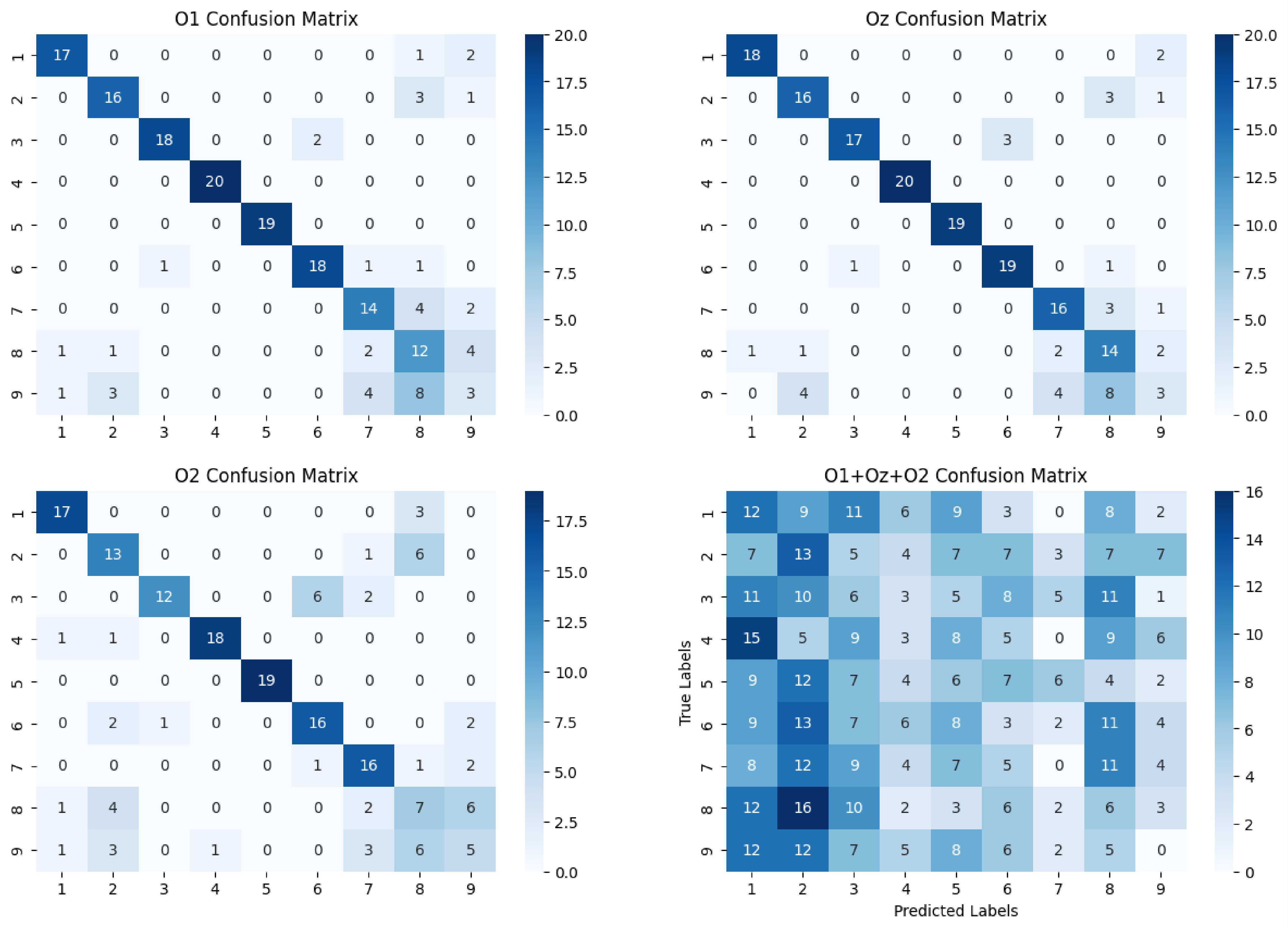
Case 3 : Different Channels Finding



- 在 Feature選擇上可能要重新調整
- 有更好的Featured可以做分群

Case 3 : Different Channels Analysis

```
O1 Accuracy: 0.79
Oz Accuracy: 0.77
O2 Accuracy: 0.69
O1+Oz+O2 Accuracy: 0.09

O1 Recall: 0.79
Oz Recall: 0.77
O2 Recall: 0.69
O1+Oz+O2 Recall: 0.09

O1 Precision: 0.79
Oz Precision: 0.77
O2 Precision: 0.70
O1+Oz+O2 Precision: 0.07
```



Challenges and Solutions

- The latest releases of the Matlab doesn't support "predict" function :
  - Use other tool and language (python) to build the model.
- The plot can't really tell the result :
  - Do normalization.
- CCA data in TSNE has weird distribution
  - Add more Sin and Cos wave to calculate and compare correlation coefficients (original 2waves --> 6 waves)

Reference

(n.d.). EEGLAB. EEGLAB. https://eeglab.org

(n.d.). MATLAB. MathWorks. https://www.mathworks.com/?s_tid=gn_logo

Code File

GitHub Link

GROUP 6

THANK YOU

August 18, 2023

2023 EEG WORKSHOP

GROUP 6

Q&A

August 18, 2023

2023 EEG WORKSHOP