



哈爾濱工業大學

Harbin Institute of Technology

认知神经科学原理 实验报告

信息相关电位认知实验
听觉 Oddball 实验

课程名称： 认知神经科学原理

学院： 计算学部

专业： 计算机科学与技术

学号： 1180300811

姓名： 孙骁

实验地点： 格物 213

指导老师： 马琳

学期： 2020 秋季学期

2021 年 1 月 12 日

目录

一、ERPs 定义及性质	3
二、脑认知功能与 ERPs 关联关系	3
三、认知神经科学实验范式及意义	4
3.1 听觉 Oddball 实验及结果分析	4
3.1.1 听觉 Oddball 实验介绍	4
3.1.2 听觉 Oddball 实验结果分析	5
3.2 信息相关电位认知实验及结果分析	11
3.2.1 信息相关电位认知实验介绍	11
3.2.2 信息相关电位认知实验结果分析	12
参考文献	16
附录 A 处理 oddball 听觉实验数据——oddball.py	20
附录 B 处理信息处理实验数据——information_cognition.py	21

一、 ERPs 定义及性质

首先介绍诱发电位 (Evoked Potentials, EPs), 也称为诱发反应, 是指给予神经系统特定的刺激, 或使大脑对刺激的信息进行加工, 在该系统和脑的相应部位产生的可以检出的、与刺激有相对固定时间间隔和特定位相的生物电反应 [1]。

ERP(Event-Related Potential) 即事件相关电位 [2], 是一种特殊的脑诱发电位, 是大脑的特定脑区在受到不同刺激时记录的电位变化。ERP 研究方法是将刺激事件, 包括: 视觉、听觉、体感等物理刺激和心理因素, 在大脑内引起的相关反应, 客观的表达出来, 为观察认知活动在不同时间进程中的脑功能活动状态, 研究人类心理和行为的神经机制, 以及认知的发展、成熟、衰退过程提供可靠的实验技术方法 [3]。

ERP 的性质如下:

1. ERPs 不像普通诱发电位记录神经系统对刺激本身产生的反应, 而是大脑对刺激带来的信息引起的反应 [2]。是在注意的基础上, 与识别、比较、判断、记忆、决断等心理活动有关, 反映了认知过程中大脑的神经电生理改变。
2. ERPs 成分除受刺激物理特性影响的“外源性 (生理性) 成分”, 还包括不受刺激物理特性的影响“内源性 (心理性) 成分”, 与被试的精神状态和注意力有关 [3]。
3. ERPs 属于长潜伏期诱发电位, 测试时一般要求被试者清醒, 并在一定程度上参与其中。

二、 脑认知功能与 ERPs 关联关系

脑认知功能

大脑的认知功能是一种心理过程, 可以使我们接收, 选择, 存储, 转换, 发展和恢复从外部刺激中收到的信息 [4]。这个过程使我们能够更有效地理解世界并与之建立联系。认知功能是一种基于大脑的感知技能, 我们需要执行从最简单到最复杂的任何任务。它们与我们学习, 记忆, 解决问题和注意力的机制有关。

脑认知功能与 ERPs 的关系

借助 ERPs 的分析方法是研究大脑信息处理阶段认知功能的最常用的动态方法之一 [3]。通过使用相应工具测量相关脑区上的 ERP 的数据提供了有关早期感觉感知过程和更高级处理的信息, 包括注意状态, 皮质抑制, 决策反应, 错误监测, 记忆更新和其他认知活动 [5]。基于 ERP 的研究方法是研究典型受试者的规范性认知过程的一种有价值的技术, 同时, ERP 可以用作评估具有神经发育疾病的儿童 (例如 ASD 和 ADHD) [6] 或患有各种精神病的成年个体差异的敏感工具 (例如 PTSD, ASD, SCZ, 物质使用障碍 [SUD] 等) [7] [8]。

尽管功能性神经影像学 (例如功能性磁共振成像 [fMRI] 或正电子发射断层扫描 [PET]) 取得了重大进展, 但因为许多精神病性疾病与可检测到的脑电图反应模式改变有关, ERP 仍然是精神病学中的一种重要的大脑研究方法。ERP 技术在精神病理学中可以用来用作功能诊断目的的有效和敏感的生物标记 (例如阿尔兹海默症) [9]。另一方面, 对神经发育障碍者的 ERP 与普通人的 ERP 之间差异的比较可以有助于更好地理解神经发育障碍和其他精神病学中扰乱的认知功能 [10]。

三、 认知神经科学实验范式及意义

3.1 听觉 Oddball 实验及结果分析

3.1.1 听觉 Oddball 实验介绍

实验目的及意义

探究人在加工小概率刺激时, 大脑的异常反应。经典 Oddball 范式为在一项实验中, 随机呈现作用于同一感觉通道的两种刺激, 刺激出现的概率有很大差别。概率大者我们称之为标准刺激 (standard stimuli), 相当于是整个实验中的背景; 概率小和偶然出现的刺激则为偏差刺激 (deviant stimuli)。

实验流程

实验共包含两种声音刺激, 两种声音刺激除音高外完全相同, 两种声音出现的频率不同 (高音: 低音 = 1: 4)。两种声音随机出现, 被试需对两种声音刺激做出不同的按键反应。

问题

1. 计算标准刺激和偏差刺激对应的准确率以及反应时长, 并进行比较和分析;
2. 分析标准刺激和偏差刺激对应的 ERP 差异, 并测量幅值、潜伏期差异对应的脑区、电极以及时间;
3. 请检索与实验相关 5 篇文献综述与之相关结果。

事件码

11: 2000Hz 声音刺激

22: 500Hz 声音刺激

1: 对 2000Hz 声音进行按键响应

2: 对 500Hz 声音进行按键响应

3.1.2 听觉 Oddball 实验结果分析

问题一

对实验数据进行预处理，得到采样频率为 1000Hz 的预处理实验数据，将 EEG 结构体的 event 属性导出到 excel 中，得到如图 (1) 的结果。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	23287	1										
2	2	23791	2										
3	3	25255	3										
4	11	27299	4										
5	2	28151	5										
6	22	29482	6										
7	1	30327	7										
8	22	31666	8										
9	1	32423	9										
10	22	33699	10										
11	1	34399	11										
12	22	35999	12										
13	1	36703	13										
14	22	38216	14										
15	1	38847	15										
16	22	40282	16										
17	1	40887	17										
18	22	42599	18										
19	1	43198	19										
20	22	44682	20										
21	1	45246	21										
22	22	46932	22										
23	1	47462	23										
24	22	48999	24										
25	1	49486	25										
26	22	51115	26										
27	1	51582	27										
28	11	53232	28										
29	2	53886	29										
30	11	55415	30										

图 1 将 Oddball 实验的事件信息导入 Excel 结果

由于在实验中我对于高频和低频按键记反，所以事件码 1 和 2 正好相反。将 Excel 文件转化为 csv 文件，编写代码计算事件码 11 和事件码 2 之间以及事件码 22 和事件码 1 之间的平均反应时长，以及计算标准刺激以及偏差刺激对应的准确率。

实验的代码见附录A

最终得到的结果如 log 文件所示。

```
1 the number of high frequency(11) is 51
2 the number of low frequency(22) is 249
3 the response accuracy of high frequency(11) is 1.0
4 the response accuracy of low frequency(22) is 1.0
5 the average of reaction time of high frequency(11) is 519.843137254902
6 the average of reaction time of low frequency(22) is 555.6586345381526
```

高音刺激与低音刺激对应的准确率均为 100%，高音刺激的平均反应时间为 519.84ms，低音刺激的平均反应时间为 555.66ms。

问题二

标准刺激叠加后的 ERP 频谱图如图 (2a) 所示, 偏差刺激叠加后的 ERP 频谱图如图 (2b) 所示。采样频率设置为 30%, 画出的脑地形图频率为 6Hz、10Hz、22Hz 和 40Hz, 画出的频带范围是 1~45Hz。

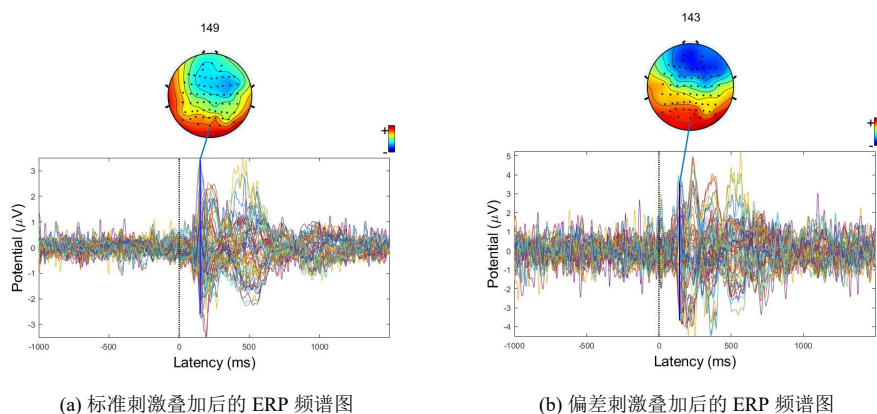


图 2 特征提取部分运行结果

对标准刺激和偏差刺激的 ERP 进行对比, 得到的结果如图 (3) 所示, 点击不同的电极进行分析, 我们发现, 标准刺激在 200~300ms 之间达到峰值, 而偏差刺激在 500~600ms 之间达到峰值, 也就是说, 偏差刺激得到的 ERP 峰值普遍比标准刺激得到的 ERP 峰值滞后 300ms 左右的时间。

对偏差刺激进行分析, 发现在 565ms 时达到波峰值, 幅值约为 $5.0\mu V$, 主要活跃的脑区为左侧顶叶和左侧额叶, 如图 (4) 所示, 查看左侧额叶和左侧顶叶附近的电极, 如图 (5) 所示, 在 565ms 左右也符合上述变化趋势, 特别是电极 FT7(5c)、T7(5f)、C5(5g)、TP7(5i) 的特征很明显。

对标准刺激进行分析, 发现在 450ms 时达到波峰值, 幅值约为 $3.5\mu V$, 主要活跃的脑区为左侧顶叶和左侧额叶, 如图 (6) 所示, 查看左侧额叶和左侧顶叶附近的电极, 如图 (7) 所示, 在 450ms 左右也符合上述变化趋势, 特别是电极 FT7(7b)、T7(7c)、C5(7d)、TP7(7e)、P7(7g) 的特征很明显。

问题三

1. Event-related potentials in the auditory oddball as a function of EEG alpha phase at stimulus onset [11]

本文旨在通过利用正交相位效应的概念, 来研究固定时间间隔的听觉 oddball 实验中刺激作用时 EEG 的 α 波活动的情况与 ERPs 之间的关系。实验让 14 名受试者受到 4 组目标概率为 50% 的 150 刺激, 通过检查受试者对按钮按压的 EEG 的反应, 对 EEG 信号进行滤波处理后, 对每个受试者评估刺激前后的 α 波。结果显示, 在

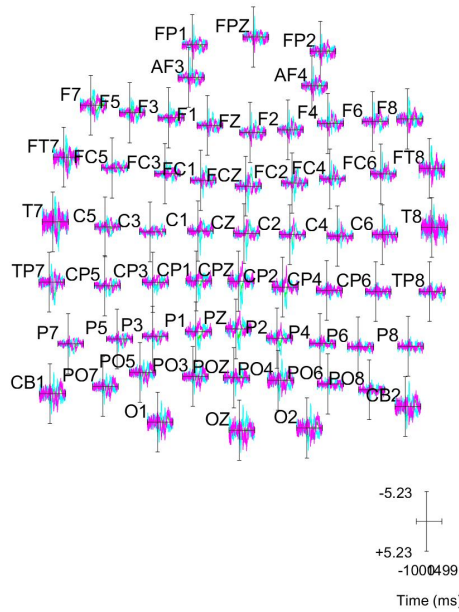


图 3 标准刺激与偏差刺激的 ERP 比较

本次实验中，大脑状态对表现负向刺激的驱动力比正向刺激的驱动力高出 8%，且在 waxing 阶段会比 waning 阶段提高 33%，这与 N1 延迟增加和 N2 振幅减小相关。这些反映了刺激开始时 α 波的频率和振幅的系统变化。因此，在固定时间间隔的的刺激范式中，动态调节 EEG 的成分频率，以便在刺激表现时更好地展示大脑状态，从而有差别地判断刺激过程在 EEG 结果中的相关性。

2. EEG alpha activity and the ERP to target stimuli in an auditory oddball paradigm [12]

本文采用了固定时间间隔刺激的 oddball 实验，以获得 ERP 中 N1P2 和 N2P3 组件的峰-峰幅度与受试者体内受刺激前 α 活性水平的关系。实验采取两种不同的音调作为刺激，概率各为 50%，分别向 14 位受试者提供了 600 个听觉刺激，需要受试者按目标按钮对不同音调做区分。对于每个试验，通过从 8 到 13 Hz 的滤波来评估 Pz 刺激前的 α 活性，并使用 α 波的 RMS 幅度对 Pz 和 Cz 处的 ERPs 进行排序。对 Pz 和 Cz 处的分量振幅与 Pz 处的刺激前 α 波建立关系，发现刺激前潜伏期的自发性脑电图与刺激后的 α 波峰和波谷强烈相关。实验结果证实了中枢神经系统激活与刺激导致 ERP 的变化之间存在密切的关系。

3. Principal components analysis of Laplacian waveforms as a generic method for identifying ERP generator patterns: I. Evaluation with auditory oddball tasks [13]

本文评估了基于 PCA 的 ERP 波形简化与无参考 Laplacian 变换的有效性和可比性，以分离听觉 oddball 实验中与任务和响应相关的 ERP 生成器模式。实验中记录了 66 位惯用右手的成年人在使用音节或音调进行奇数球测试时记录的鼻子参考 ERP，

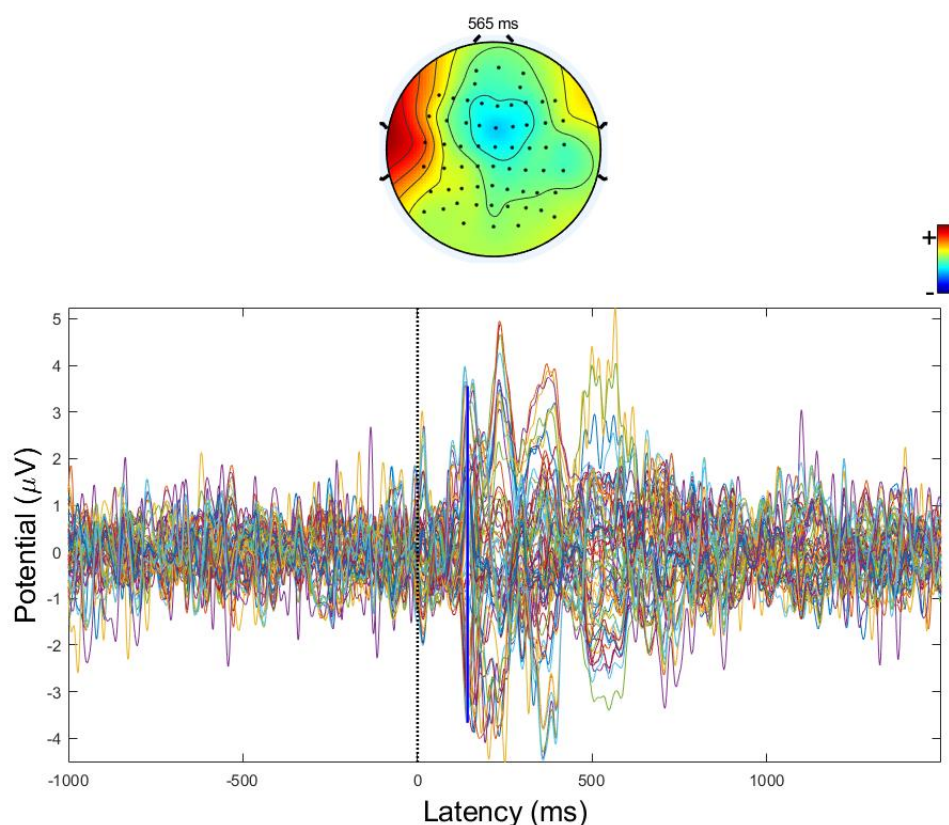


图 4 偏差刺激的峰值

并计算球形样条电流源密度 (CSD) 波形以锐化 ERP 头皮形貌并消除体积传导的影响。ERP 和 CSD 数据作为基于协方差的不受限制的时间 PCA 模型的输入，以分离时间和空间上存在相关性的 ERP 和 CSD。

实验结果显示，相应刺激的 ERP 和 CSD 结果与已知的 ERP 成分相关。例如，可以从包围 Sylvian 裂缝的前槽和后源中中央的 N1 的偶极组织附近明显看到。与 N2 相关的因素以不对称的额外侧（音调：额颞 R>L）和顶颞（语音：顶颞 L>R）汇为特征。一个单一的 ERP 因子总结了顶叶 P3 的活性，以及前叶的阴性反应。相反，两个 CSD 因子在 360ms 和 560ms 达到峰值，将具有前汇的顶叶 P3 源与具有明显局限性 Fz 汇的中心顶叶 P3 源区分开来。与按按钮相比，静默计数的顶叶较小，但左侧颞叶 P3 源较大。

实验结果表明，CSD 变换是 ERP 数据 PCA 的一个有价值的预处理步骤，为普遍存在的参考问题提供了一个独特的、有意义的解决方案。通过减少 ERP 冗余和生成更清晰、更简单的脑地形图，并且不丢失或扭曲任何感兴趣的效果，CSD-PCA 解决方案复刻并扩展了以前与任务和响应相关的发现。

4. ERP measures of stimulus processing during an auditory oddball task in Parkinson's disease:

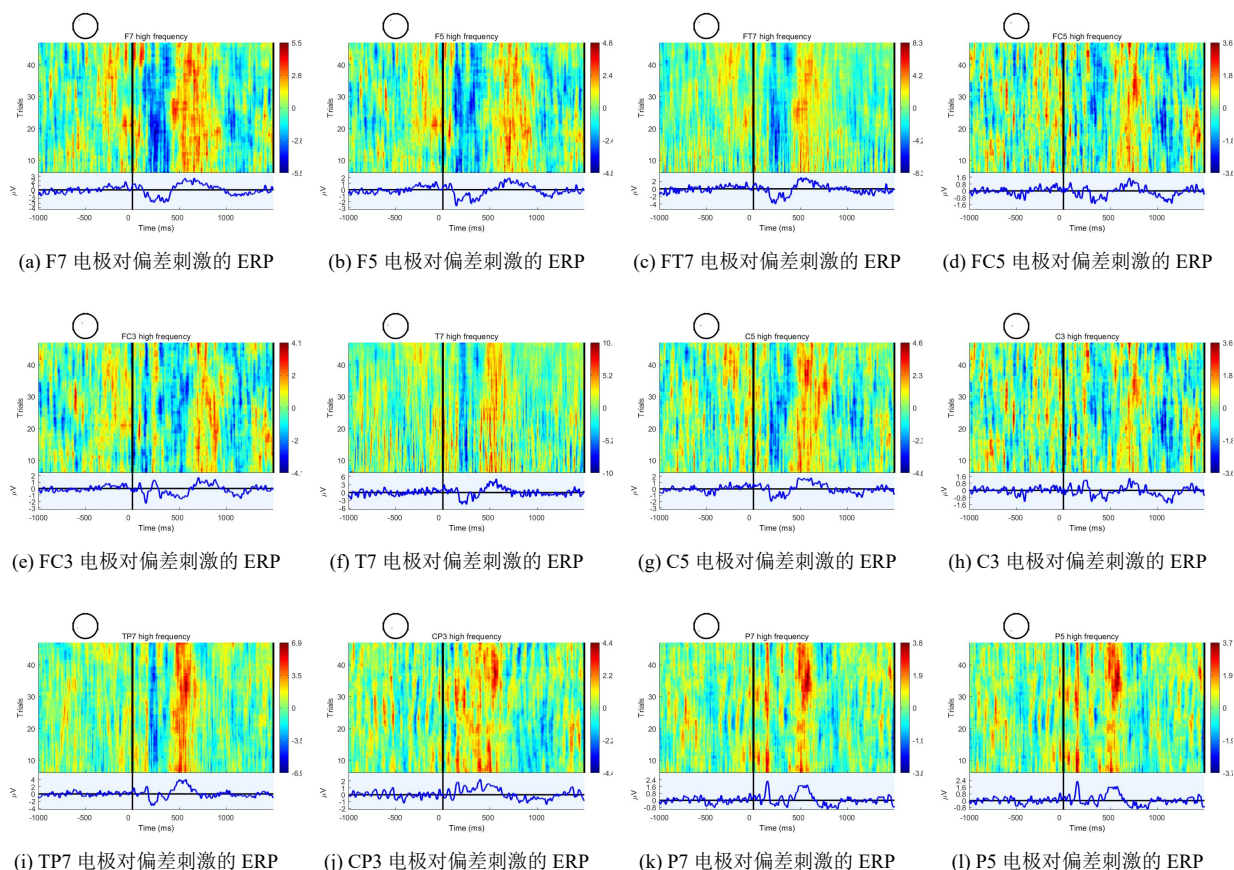


图 5 左侧额叶的左侧顶叶附近电极对偏差刺激的 ERP

Evidence for an early information processing deficit [14]

本文将 17 名非特发性帕金森氏病的非痴呆患者与年龄和性别相匹配的对照组进行听觉 oddball 任务比较。通过无声计数或按下按钮来响应低概率目标音调。帕金森组的 N1 振幅减弱至目标音和非目标音，表明早期信息处理受到损害。相反，P2、N2 和 P3 的幅度并未将患者与对照区分开。计数目标时，帕金森组几个峰值潜伏期 (P2, N2 和 P3) 增加，而通过按按钮确定目标时，只有 N2 被延迟。N2 潜伏期越长，表明刺激分类所需的时间增加。早期 ERP 成分的幅度和潜伏期变化为帕金森氏病的早期信息处理受损提供了证据。

5. Auditory processing in an inter-modal oddball task: Effects of a combined auditory/visual standard on auditory target ERPs [15]

先前的研究在多模式联动任务中对听觉 oddball 实验的受试者的 ERP 进行的研究表明，早期成分受模式内过程的影响，而 ERP 后期的成分 (大约 200~400ms) 受多模式和多模式的影响。由此得出的结论是，听觉目标处理存在不同的阶段，早期的特定于情态的阶段和较晚的取决于上下文的阶段。本文通过研究在异类游戏中同时呈现视觉标准刺激和听觉标准刺激，以确定在此任务中包括视觉标准刺激是否会影响 ERP 达到目标。听觉-视觉 oddball 任务包括定期提出的听觉和视觉标准刺激 (80%

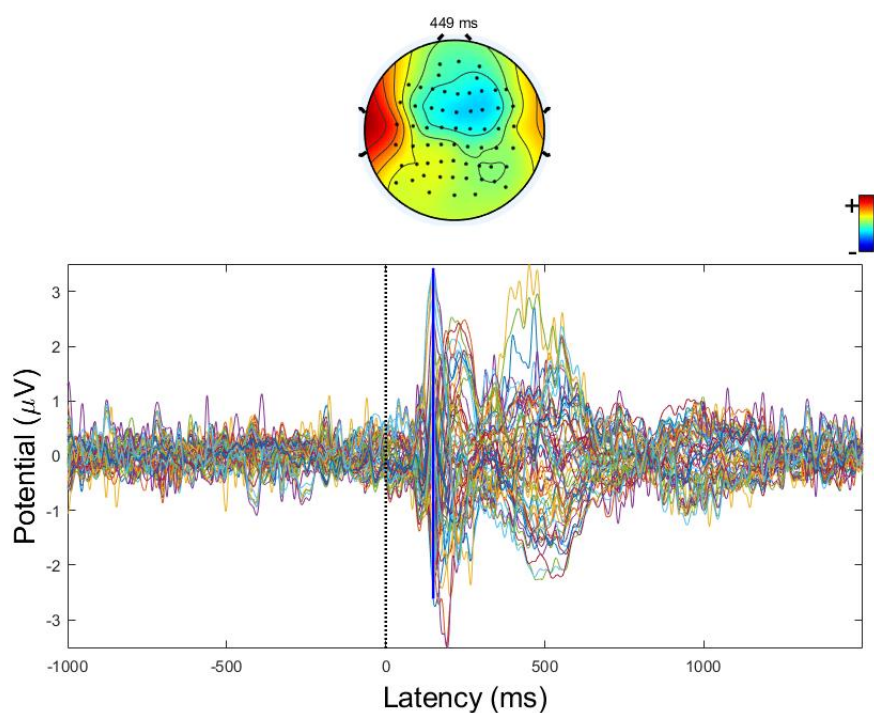


图 6 标准刺激的峰值

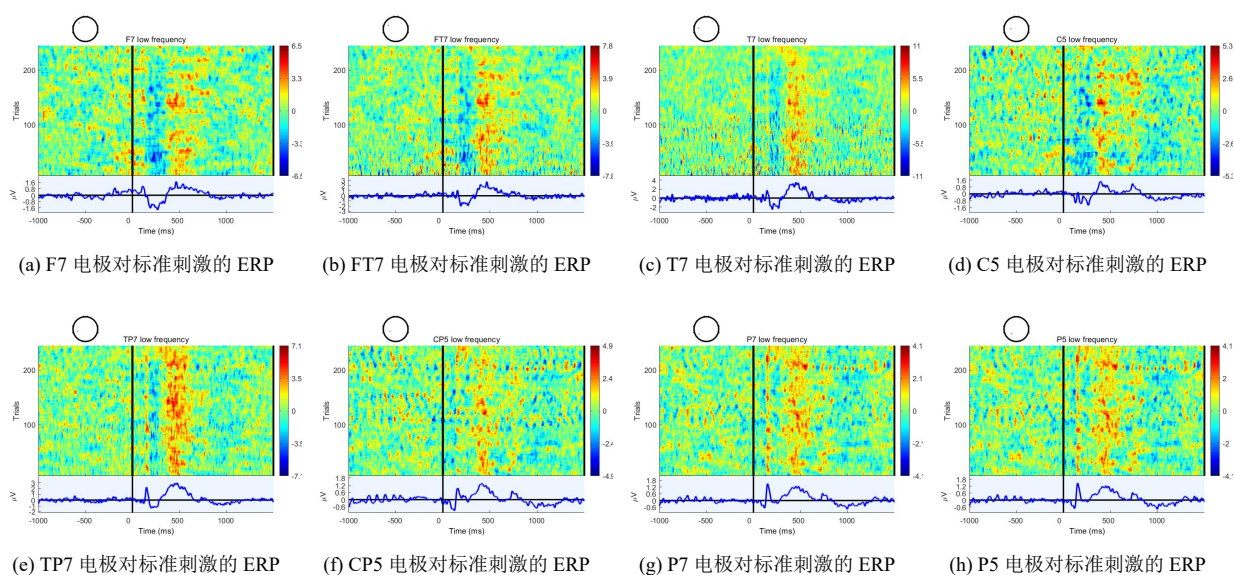


图 7 左侧额叶的左侧顶叶附近电极对标准刺激的 ERP

和不常见的听觉目标 (20%)。将听觉-视觉 oddball 任务中目标的 ERP 与具有相同听觉刺激且没有视觉标准的听觉 oddball 任务中的 ERP 进行了比较。结果表明，早期组件 N100, P200 和 N200 在任务之间没有区别。这与先前研究的结果一致，并证实了长达 200ms 的活动不受视觉标准刺激的影响。后来的 P250, P300 和 P350 组件较

大，并且在听觉-视觉 oddball 任务中显示出较大的脑地形图差异。特别是，P300 和 P350 成分被认为代表了独立的模态和模态内成分。总的来说，这项研究提供了听觉加工在早期情态依赖阶段和后期情境依赖阶段这两个阶段发生的进一步证据。

3.2 信息相关电位认知实验及结果分析

3.2.1 信息相关电位认知实验介绍

实验目的及意义

当刺激材料含有信息时，脑在对刺激中的信息进行加工的过程中，是否会产生一系列信息加工相关的认知成分？信息加工过程中，脑活动的 EEG/ERP 表现是否存在？若存在，则该成分或者表现是否有一定的模式，呈现一定的规律？

实验流程

实验共有 3 轮，每轮 5 个 blocks，每个 block 含有 30 个 trials。实验时长 36min。实验流程如图 (8) 所示

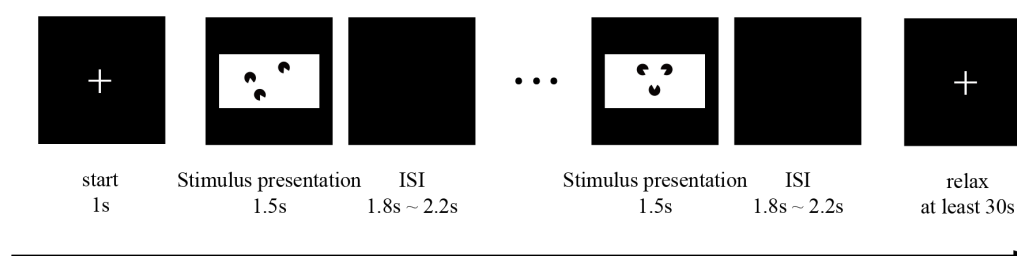


图 8 信息相关电位认知实验流程图

在每个 trial 中，被试需根据当前显示的图像是否含有信息进行判断，含有信息按下键盘方向键左键，不含有信息的按下键盘方向键右键。

问题

1. 请你分析在判断有信息和无信息时，准确率和反应时间有何区别？

准确率：

$$\text{有信息刺激的准确率} = \frac{\text{有信息刺激时，判断为有的数量}}{\text{有信息刺激的数量}} \quad (1)$$

$$\text{无信息刺激的准确率} = \frac{\text{无信息刺激时，判断为无的数量}}{\text{无信息刺激的数量}} \quad (2)$$

反应时间为：在判断正确的 trial 中，反应时刻-刺激发生时刻

2. 请分析

- (a) 在有信息刺激时，花鸟图案刺激引起的 ERP 和几何图像刺激引起的 ERP 有何差异？

(b) 在无信息刺激时，花鸟图案刺激引起的 ERP 和几何图像刺激引起的 ERP 有何差异？（分析 ERP 成分的幅值、潜伏期等差异即可）

3. 请分析在几何图像 trial 中，含信息刺激引起的 ERP 与不含信息刺激引起的 ERP 有何差异？这些差异主要体现在哪些脑区（通过脑地形图说明）？

信息相关电位实验事件码说明

31: 三个扇形图案位置与开口方向完全随机分布，无信息刺激

32: 三个扇形图案位置固定，开口方向固定，有信息刺激

33: 三个扇形图案位置固定、开口方向随机，无信息刺激

41: 四个扇形图案位置与开口方向完全随机分布，无信息刺激

42: 四个扇形图案位置固定，开口方向固定，有信息刺激

43: 四个扇形图案位置固定、开口方向随机，无信息刺激

51: 花鸟图案完全随机分布，无信息刺激

52: 花鸟图案能形成人脸轮廓，有信息刺激

53: 花鸟图案完全随机分布，无信息刺激

1: 判断为有信息刺激

2: 判断为无信息刺激

3.2.2 信息相关电位认知实验结果分析

问题一

对实验数据进行预处理，得到采样频率为 1000Hz 的预处理实验数据，将 EEG 结构体的 event 属性导出到 excel 中，得到如图 (9) 的结果。

将 Excel 文件转化为 csv 文件，手动去掉了等待时间自己乱按的事件码，得到的 data 文件夹下的 information.csv 文件。编写代码计算事件码 31、32、33、41、42、43、51、52 和 53 对应的判断，并计算判断正确时的平均反应时长以及不同 trial 对应的有信息刺激准确率和无信息刺激准确率。

实验的代码见附录B。

最终得到的结果如 log 文件所示。

```
1 ineffective judge at line 2
2 ineffective judge at line 3
3 ineffective judge at line 63
4 ineffective judge at line 251
5 ineffective judge at line 254
6 ineffective judge at line 291
7 three sectors have information correct accuracy:0.988888888888889
8 three sectors have no information correct accuracy:0.988888888888889
```

	A	B	C	D	E	F	G	H	I	J
1	100	3466	1							
2	3	19222	2							
3	11	21282	3							
4	3	50629	4							
5	12	52698	5							
6	101	174763	6							
7	3	179507	7							
8	102	181647	8							
9	103	182546	9							
10	32	184079	10							
11	31	188146	11							
12	31	192212	12							
13	2	193450	13							
14	32	195512	14							
15	1	196522	15							
16	31	198579	16							
17	2	199770	17							
18	32	201829	18							
19	1	202970	19							
20	32	205029	20							
21	1	205954	21							
22	31	208012	22							
23	2	209050	23							
24	31	211112	24							
25	2	212370	25							
26	31	214428	26							
27	2	216130	27							
28	32	218195	28							
29	1	219490	29							
30	32	221545	30							

图 9 将信息相关电位实验的事件信息导入 Excel 结果

```

9 the average of reaction time of three sectors is:946.1797752808989
10 four sectors have information correct accuracy:0.9888888888888889
11 four sectors have no information correct accuracy:1.0
12 the average of reaction time of four sectors is:942.9162011173185
13 flower and bird pattern have information correct accuracy:0.9333333333333333
14 flower and bird pattern have no information correct accuracy:1.0
15 the average of reaction time of flower and bird pattern is:958.0114942528736

```

根据结果来看，在实验中，我并没有出现对某个信息判断出错的情况，但是准确率并未达到 100%。这是因为每个 trial 刚开始的两三幅图片，系统并未等我作出判断就进行了翻页，根据之后对于平均反应时间的测量，估计系统设置自动翻页的时间为 1s，不知道是否为系统的 bug，也希望这里改进一下，等待受试者作出判断后再翻页。

我将未及时进行判断的情况定义为失败的判断，因此得到：

1. 三个扇形图案有信息刺激准确率为 98.89%；
2. 三个扇形图案无信息刺激准确率为 98.89%；
3. 三个扇形图案的平均反应时间为 946.18ms；
4. 四个扇形图案有信息刺激准确率为 98.89%；
5. 四个扇形图案无信息刺激准确率为 100%；
6. 四个扇形图案的平均反应时间为 942.92ms；

7. 花鸟图案有信息刺激准确率为 93.33%;
8. 花鸟图案无信息刺激准确率为 100%;
9. 花鸟图案的平均反应时间为 958.01ms。

从实验结果来看,对于四个扇形图案的有信息刺激判断准确率与三个图案的有信息刺激判断准确率持平,对于四个扇形图案的无信息刺激判断准确率较三个图案的无信息刺激判断准确率有所提升,但是平均反应时间几乎相同;花鸟图案的有信息刺激判断准确率有所下降,平均反应时间与扇形判断相比也更长,可能是由于实验时紧张,担心翻页过快而未来得及进行判断有关,从而导致的准确率下降。

问题二

此处几何图像的刺激以三个扇形的刺激为例。有信息刺激时,三个扇形图案刺激引起的 ERP 结果如图 (10) 所示,选择 100、200、300、400、500、600、700、800、900ms 时的 ERP 叠加图像,如图 (11) 所示,可以看到在刺激发生后的 100~400ms 内,主要活跃的脑区为右侧枕叶,在 700~800ms 之间,主要活跃的脑区为左侧额叶和左侧顶叶;花鸟图案刺激引起的 ERP 结果如图 (12) 所示,选择 100、200、300、400、500、600、700、800、900ms 时的 ERP 叠加图像,如图 (13) 所示,可以看到在刺激发生后的 100~500ms 内,主要活跃的脑区为右侧枕叶和部分左侧枕叶,在 700ms 左右主要活跃的脑区为左侧额叶。二者对比是的 ERP 图像如图 (14) 所示,选择右侧枕叶、左侧额叶处的电极进行分析,右侧枕叶处相应电极的 ERP 叠加结果如图 (15) 所示,可以看到,在 200ms 左右,花鸟图案刺激下的右侧枕叶的 ERP 幅值要高于三个扇形图案引起的刺激,由此可以印证,右侧枕叶是图像认知的能力较为活跃的区域,

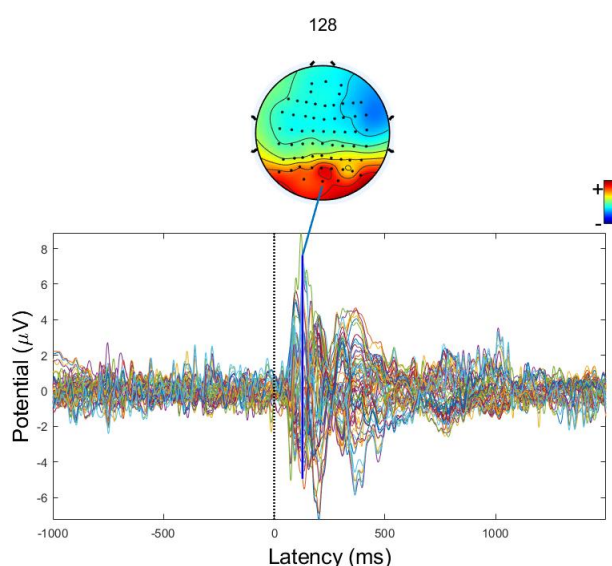


图 10 有信息刺激时,三个扇形图案刺激的 ERP 结果

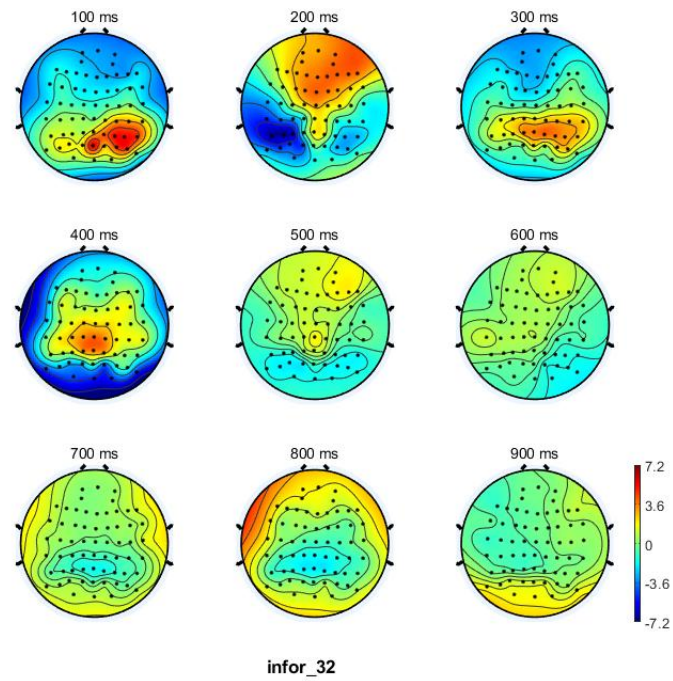


图 11 有信息刺激时，三个扇形图案刺激的 ERP 分时结果

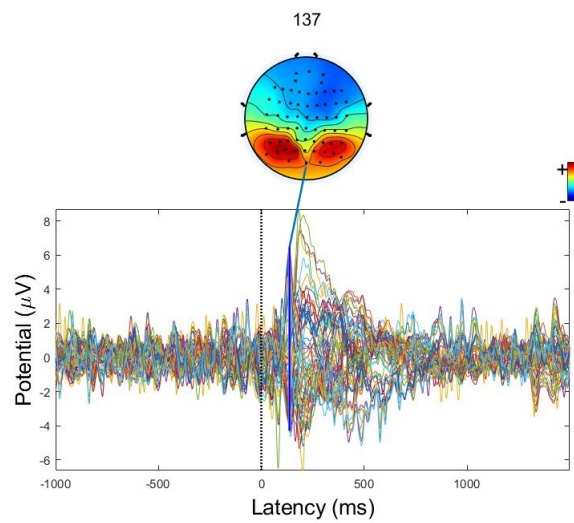


图 12 有信息刺激时，花鸟图案刺激的 ERP 结果

问题三

问题四

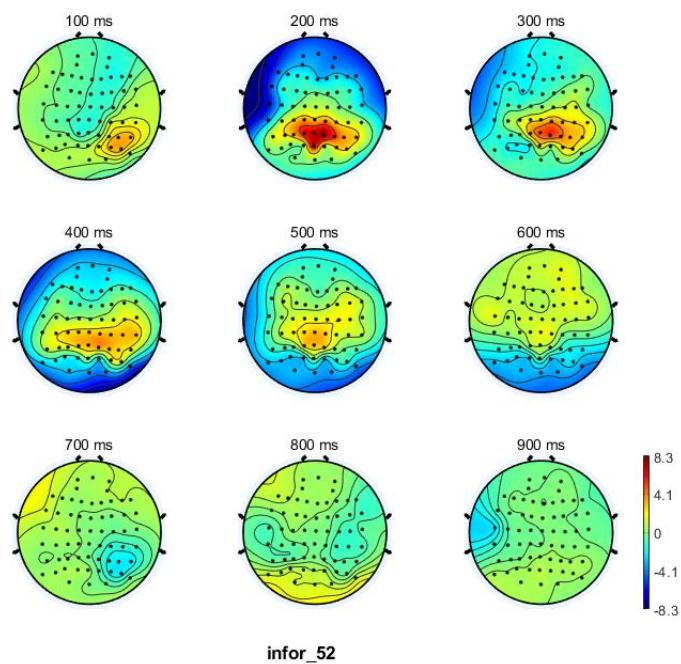


图 13 有信息刺激时，花鸟图案刺激的 ERP 分时结果

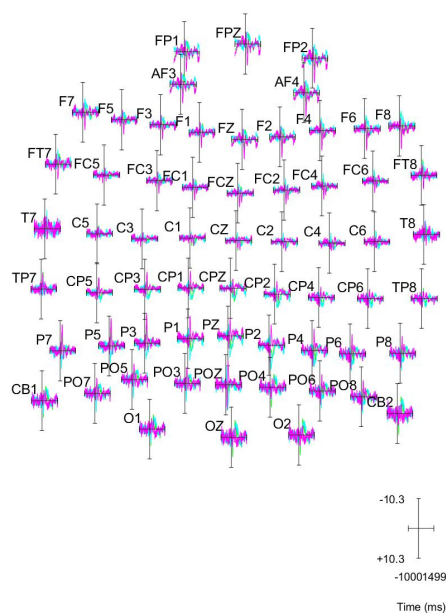
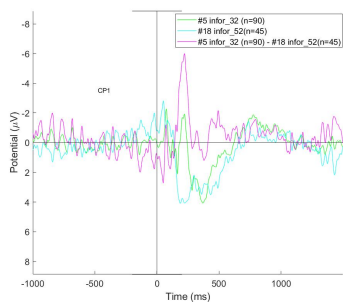
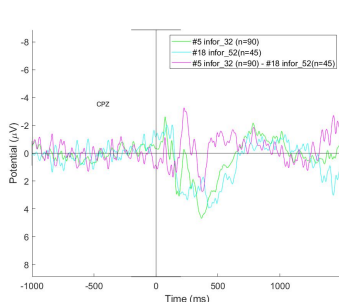


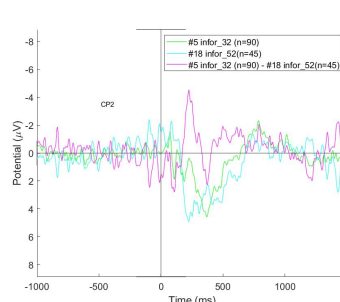
图 14 有信息刺激时，三个扇形图案刺激与花鸟图案刺激的 ERP 比较



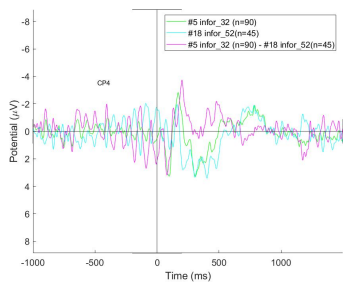
(a) CP1 电极几何图形和花鸟图案 ERP 对比



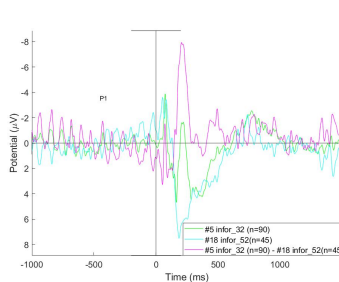
(b) CPZ 电极几何图形和花鸟图案 ERP 对比



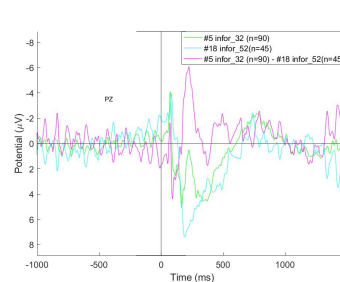
(c) CP2 电极几何图形和花鸟图案 ERP 对比



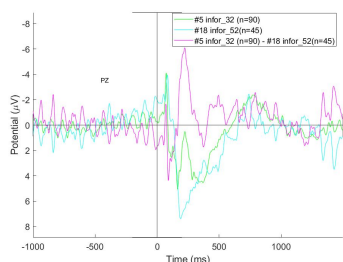
(d) CP4 几何图形和花鸟图案 ERP 对比



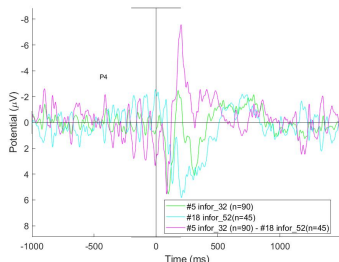
(e) P1 几何图形和花鸟图案 ERP 对比



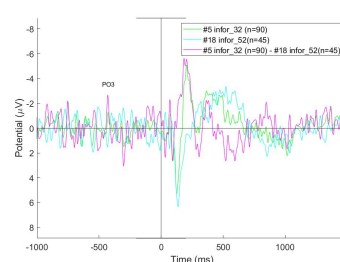
(f) PZ 几何图形和花鸟图案 ERP 对比



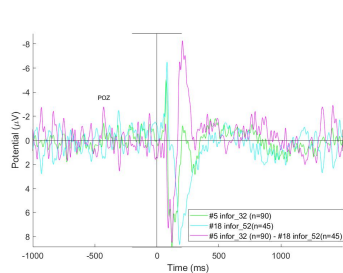
(g) P2 几何图形和花鸟图案 ERP 对比



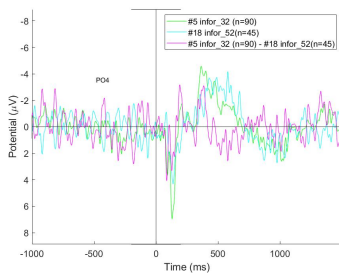
(h) P4 几何图形和花鸟图案 ERP 对比



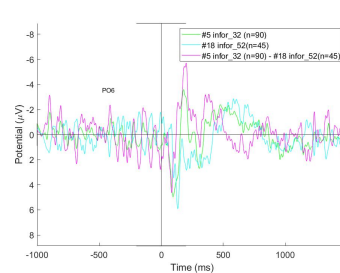
(i) PO3 几何图形和花鸟图案 ERP 对比



(j) POZ 几何图形和花鸟图案 ERP 对比



(k) PO4 几何图形和花鸟图案 ERP 对比



(l) PO6 几何图形和花鸟图案 ERP 对比

图 15 右侧枕叶处相应电极的 ERP 对比结果

参考文献

- [1] A.D. Legatt. Evoked potentials. In Michael J. Aminoff and Robert B. Daroff, editors, *Encyclopedia of the Neurological Sciences (Second Edition)*, pages 228 – 231. Academic Press, Oxford, second edition edition, 2014.

- [2] Shravani Sur and V. K. Sinha. Event-related potential: An overview. *Industrial Psychiatry Journal*, 18:70–73, Jan–Jun 2009.
- [3] Steven J. Luck. *An Introduction to the Event-Related Potential Technique*. The MIT Press, 2014.
- [4] Jiawei Zhang. Cognitive functions of the brain: Perception, attention and memory. IFM LAB TUTORIAL SERIES, 2019.
- [5] Tato Sokhadze, Manuel Casanova, Emily Casanova, Eva Lamina, Desmond Kelly, and Irma Khachidze. Event-related potentials (erp) in cognitive neuroscience research and applications. *NeuroRegulation*, 4:14–27, 03 2017.
- [6] Tato Sokhadze, Lonnie Sears, Allan Tasman, Emily Casanova, and Manuel Casanova. Comparative event-related potential study of performance in visual oddball task in children with autism spectrum disorder, adhd, comorbid autism and adhd, and neurotypical children. *NeuroRegulation*, 6:134–152, 06 2019.
- [7] Tato Sokhadze, Eva Lamina, Emily Casanova, Desmond Kelly, Ioan Opris, Allan Tasman, and Manuel Casanova. Exploratory study of rtms neuromodulation effects on electrocortical functional measures of performance in an oddball test and behavioral symptoms in autism. *Frontiers in Systems Neuroscience*, 12:20, 05 2018.
- [8] Joshua Baruth, Manuel Casanova, Lonnie Sears, and Tato Sokhadze. Early-stage visual processing abnormalities in high-functioning autism spectrum disorder (asd). *Translational neuroscience*, 1:177–187, 06 2010.
- [9] Andras Horvath, Anna Szucs, Gabor Csukly, Anna Sakovics, Gabor Stefanics, and Anita Kamondi. Eeg and erp biomarkers of alzheimer’s disease: a critical review. *Frontiers in bioscience (Landmark edition)*, 23:183–220, Jan 2018.
- [10] Alex Lau-Zhu, Michael P.H. Lau, and Gráinne McLoughlin. Mobile eeg in research on neurodevelopmental disorders: Opportunities and challenges. *Developmental Cognitive Neuroscience*, 36:100635, 2019.
- [11] Robert J. Barry, Jacqueline A. Rushby, Stuart J. Johnstone, Adam R. Clarke, Rodney J. Croft, and Carlie A. Lawrence. Event-related potentials in the auditory oddball as a function of eeg alpha phase at stimulus onset. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 115:2593–601, Nov 2004.

- [12] R. J. Barry, S. Kirkaikul, and D. Hodder. Eeg alpha activity and the erp to target stimuli in an auditory oddball paradigm. *International journal of psychophysiology : official journal of the International Organization of Psychophysiology*, 39:39–50, Dec 2000.
- [13] Jürgen Kayser and Craig E. Tenke. Principal components analysis of laplacian waveforms as a generic method for identifying erp generator patterns: I. evaluation with auditory oddball tasks. *Clinical Neurophysiology*, 117(2):348 – 368, 2006.
- [14] M.J. Wright, G.M. Geffen, and L.B. Geffen. Erp measures of stimulus processing during an auditory oddball task in parkinson’s disease: Evidence for an early information processing deficit. *Parkinsonism and Related Disorders*, 2(1):13 – 21, 1996.
- [15] Christopher R. Brown, Adam R. Clarke, and Robert J. Barry. Auditory processing in an inter-modal oddball task: Effects of a combined auditory/visual standard on auditory target erps. *International Journal of Psychophysiology*, 65(2):122 – 131, 2007.

附录 A 处理 oddball 听觉实验数据——oddball.py

```
1 from logger import *
2 import pandas as pd
3 import os
4
5 file_path = '../data/'
6 sys.stdout = Logger(sys.stdout)
7
8
9 def calculate_accuracy():
10     odd_data = pd.read_csv(file_path + 'oddball.csv')
11     code_num = [int(i) for i in odd_data['事件码']]
12     react_time = [int(i) for i in odd_data['反应时间']]
13     num_11 = 0
14     num_22 = 0
15     num_11_correct = 0
16     num_11_incorrect = 0
17     num_22_correct = 0
18     num_22_incorrect = 0
19     num_11_react_time = 0
20     num_22_react_time = 0
21     for i in range(len(code_num)):
22         if code_num[i] == 11:
23             num_11 += 1
24             num_11_correct += 1 if code_num[i + 1] == 2 else num_11_incorrect + 1
25             # print(f'react time is {react_time[i + 1] - react_time[i]}')
26             num_11_react_time += react_time[i + 1] - react_time[i]
27         elif code_num[i] == 22:
28             num_22 += 1
29             num_22_correct += 1 if code_num[i + 1] == 1 else num_22_incorrect + 1
30             # print(f'react time is {react_time[i + 1] - react_time[i]}')
31             num_22_react_time += react_time[i + 1] - react_time[i]
32     print(f'the number of high frequency(11) is {num_11}')
33     print(f'the number of low frequency(22) is {num_22}')
34     print(f'the response accuracy of high frequency(11) is {num_11_correct / num_11}')
35     print(f'the response accuracy of low frequency(22) is {num_22_correct / num_22}')
36     print(f'the average of reaction time of high frequency(11) is {num_11_react_time / num_11}')
37     print(f'the average of reaction time of low frequency(22) is {num_22_react_time / num_22}')
38
39
40 def main():
41     calculate_accuracy()
42
```

```

43
44 if __name__ == '__main__':
45     main()

```

附录 B 处理信息处理实验数据——information_cognition.py

```

1  from logger import *
2  import pandas as pd
3
4  file_path = '../data/'
5  sys.stdout = Logger(sys.stdout)
6
7
8  def calculate_accuracy():
9      odd_data = pd.read_csv(file_path + 'information.csv')
10     code_num = [int(i) for i in odd_data['事件码']]
11     react_time = [int(i) for i in odd_data['反应时间']]
12     num_31, num_32, num_33, num_41, num_42, num_43, num_51, num_52, num_53 = 0, 0, 0,
13         0, 0, 0, 0, 0, 0
14     num_3_correct, num_3_incorrect, num_4_correct, num_4_incorrect, num_5_correct,
15         num_5_incorrect = 0, 0, 0, 0, 0, 0
16     num_3_react_time, num_4_react_time, num_5_react_time = 0, 0, 0
17     effective_judge, ineffective_judge = 0, 0
18     for i in range(len(code_num)):
19         if code_num[i] == 31:
20             num_31 += 1
21             if code_num[i + 1] == 1:
22                 effective_judge += 1
23                 print(f'judge wrong at line {i + 2}')
24             elif code_num[i + 1] == 2:
25                 effective_judge += 1
26                 num_3_incorrect += 1
27                 num_3_react_time += react_time[i + 1] - react_time[i]
28             else:
29                 ineffective_judge += 1
30                 print(f'ineffective judge at line {i + 2}')
31         elif code_num[i] == 32:
32             num_32 += 1
33             if code_num[i + 1] == 1:
34                 effective_judge += 1
35                 num_3_correct += 1
36                 num_3_react_time += react_time[i + 1] - react_time[i]
37             elif code_num[i + 1] == 2:
38                 effective_judge += 1

```

```

37         print(f'judge wrong at line {i + 2}')
38     else:
39         ineffective_judge += 1
40         print(f'ineffective judge at line {i + 2}')
41 elif code_num[i] == 33:
42     num_33 += 1
43     if code_num[i + 1] == 1:
44         effective_judge += 1
45         print(f'judge wrong at line {i + 2}')
46     elif code_num[i + 1] == 2:
47         effective_judge += 1
48         num_3_incorrect += 1
49         num_3_react_time += react_time[i + 1] - react_time[i]
50     else:
51         ineffective_judge += 1
52         print(f'ineffective judge at line {i + 2}')
53 elif code_num[i] == 41:
54     num_41 += 1
55     if code_num[i + 1] == 1:
56         effective_judge += 1
57         print(f'judge wrong at line {i + 2}')
58     elif code_num[i + 1] == 2:
59         effective_judge += 1
60         num_4_incorrect += 1
61         num_4_react_time += react_time[i + 1] - react_time[i]
62     else:
63         ineffective_judge += 1
64         print(f'ineffective judge at line {i + 2}')
65 elif code_num[i] == 42:
66     num_42 += 1
67     if code_num[i + 1] == 1:
68         effective_judge += 1
69         num_4_correct += 1
70         num_4_react_time += react_time[i + 1] - react_time[i]
71     elif code_num[i + 1] == 2:
72         effective_judge += 1
73         print(f'judge wrong at line {i + 2}')
74     else:
75         ineffective_judge += 1
76         print(f'ineffective judge at line {i + 2}')
77 elif code_num[i] == 43:
78     num_43 += 1
79     if code_num[i + 1] == 1:
80         effective_judge += 1
81         print(f'judge wrong at line {i + 2}')
82     elif code_num[i + 1] == 2:
83         effective_judge += 1

```

```

84         num_4_incorrect += 1
85         num_4_react_time += react_time[i + 1] - react_time[i]
86     else:
87         ineffective_judge += 1
88         print(f'ineffective judge at line {i + 2}')
89 elif code_num[i] == 51:
90     num_51 += 1
91     if code_num[i + 1] == 1:
92         effective_judge += 1
93         print(f'judge wrong at line {i + 2}')
94     elif code_num[i + 1] == 2:
95         effective_judge += 1
96         num_5_incorrect += 1
97         num_5_react_time += react_time[i + 1] - react_time[i]
98     else:
99         ineffective_judge += 1
100        print(f'ineffective judge at line {i + 2}')
101 elif code_num[i] == 52:
102     num_52 += 1
103     if code_num[i + 1] == 1:
104         effective_judge += 1
105         num_5_correct += 1
106         num_5_react_time += react_time[i + 1] - react_time[i]
107     elif code_num[i + 1] == 2:
108         effective_judge += 1
109         print(f'judge wrong at line {i + 2}')
110     else:
111         ineffective_judge += 1
112         print(f'ineffective judge at line {i + 2}')
113 elif code_num[i] == 53:
114     num_53 += 1
115     if code_num[i + 1] == 1:
116         effective_judge += 1
117         print(f'judge wrong at line {i + 2}')
118     elif code_num[i + 1] == 2:
119         effective_judge += 1
120         num_5_incorrect += 1
121         num_5_react_time += react_time[i + 1] - react_time[i]
122     else:
123         ineffective_judge += 1
124         print(f'ineffective judge at line {i + 2}')
125 print(f'three sectors have information correct accuracy:{num_3_correct / num_32}')
126 print(f'three sectors have no information correct accuracy:{num_3_incorrect /
127        (num_31 + num_33)}')
127 print(f'the average of reaction time of three sectors is:{num_3_react_time /
128        (num_3_correct + num_3_incorrect)}')
128 print(f'four sectors have information correct accuracy:{num_4_correct / num_42}')

```

```

129     print(f'four sectors have no information correct accuracy:{num_4_incorrect /
130           (num_41 + num_43)}')
131     print(f'the average of reaction time of four sectors is:{num_4_react_time /
132           (num_4_correct + num_4_incorrect)}')
133     print(f'flower and bird pattern have information correct accuracy:{num_5_correct /
134           num_52}')
135     print(f'flower and bird pattern have no information correct
136           accuracy:{num_5_incorrect / (num_51 + num_53)}')
137     print(
138         f'the average of reaction time of flower and bird pattern is:{num_5_react_time /
139           (num_5_correct + num_5_incorrect)}')
140
141     def main():
142         calculate_accuracy()
143
144     if __name__ == '__main__':
145         main()

```