

基于非侵入式脑机接口的运动康复训练技术与系统赛项

参赛说明

一、 实验范式

本实验数据由博睿康 256 导联湿电极设备采集，设备采样率为 1000Hz。如图 1 所示，本实验包含：左/右手屈伸运动想象、左/右手握拳运动想象四类，如图 1 所示。

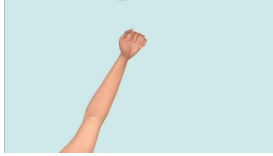
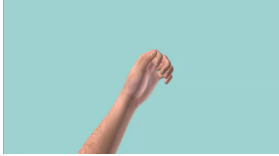

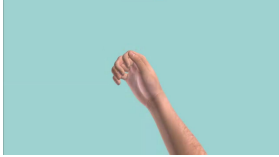
	
左手屈伸运动想象提示	左手握拳运动想象提示
	
右手屈伸运动想象提示	右手握拳运动想象提示

图 1 实验分类范式

每组数据 (session) 包含 4 个 block，其中每个 block 之间有一定的休息时间。每位受试者的前两个 block 的数据作为训练数据，该受试者的后两个 block 的数据作为测试数据，具体如图 2 所示。

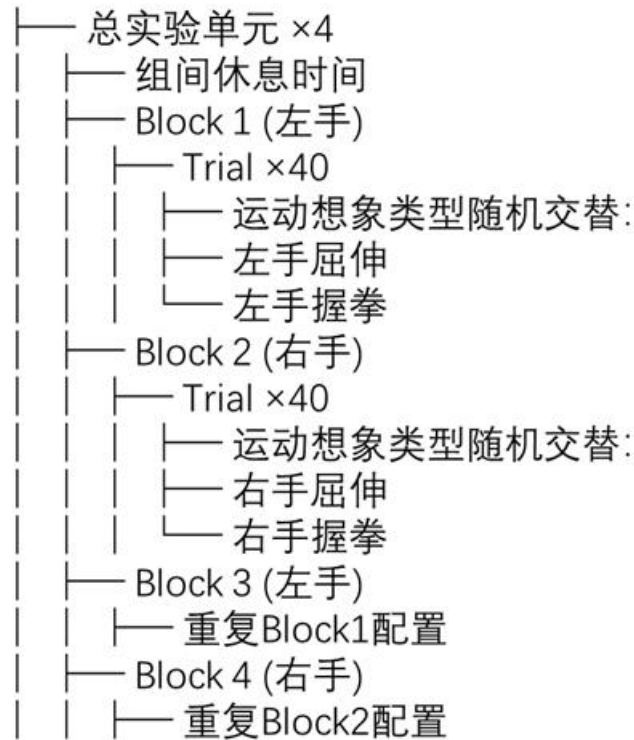


图2 实验范式数据流程

每个 block 包含 40 个 trial，即 1 个 session 共计 4 个 block，160 个 trial。以下为每个 block 的详细内容：

(1) Block1 与 Block3：让被试执行左手屈伸运动想象、左手握拳运动想象，在单个 block 中两个动作随机交替出现。

(2) Block2 与 Block4：让被试执行右手屈伸运动想象、右手握拳运动想象，在单个 block 中两个动作随机交替出现。

在每个 session 数据中，最开始有 30s 睁眼时间，此时看着屏幕“+”不做运动想象。30s 睁眼结束后紧接着提示开始闭眼 30s。闭眼结束后被试按任意键开始运动想象。整体 session 流程如图 3 所示。

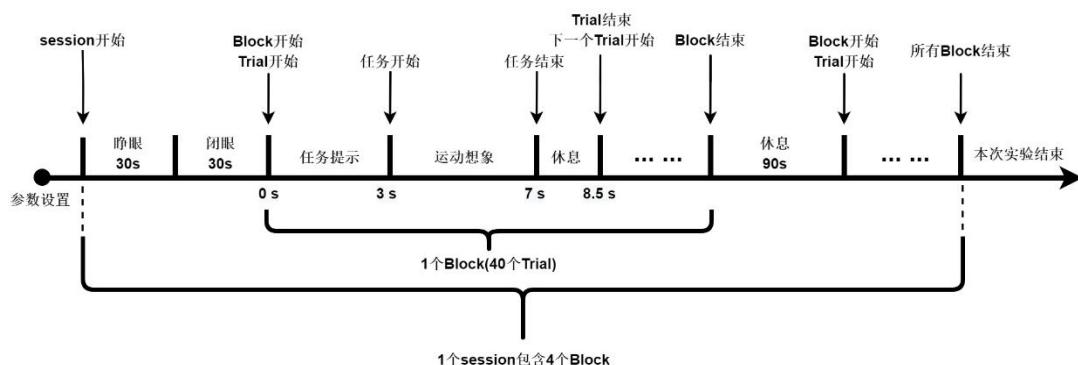


图3 整体 session 流程图

每个 trial 包含 3s 的任务提示、4s 的运动想象以及 1.5s 的休息，如图 4 所示。

本次比赛采用的数据包含患者数据。

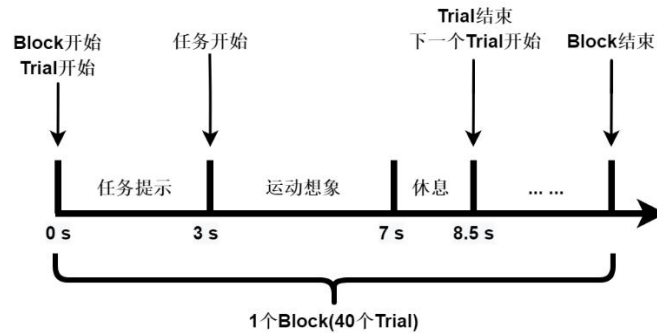


图 4 单个 trial 运动想象流程图

实验数据使用博睿康 256 导联湿电极设备采集（实际使用其中 77 导联），原始采样率为 1000Hz，单次 Trial 中数据部分仅保留运行想象任务开始至运动想象任务结束之间的 4s 数据，任务提示与休息时间数据均置为 0。

对于提供给参赛方的数据未做降采样或其他滤波处理。

具体 trigger 标签定义如表 1、2 所示。整个实验范式包含 Session 开始标 250、Session 结束标 251，睁眼标 7、闭眼标 8。

每个 block 开始标 242，结束标 243，block 之间休息标 9。

每个 trial 包含 6 种 trigger，打标详情如下：

0s：刺激开始，打 trial 开始标签（240）。

0~3s：被试开始准备运动想象，3s 时打任务标签（1/2/3/4）。

3~7s：被试根据提示进行相应运动想象。

7~ns：被试休息，7s 打结束标签（241）。

表 1 block 标签信息

标签	242	243
含义	开始	结束

表 2 trial 标签信息

标签	240	241	1	2	3	4
含义	开始	结束	左手屈伸	右手屈伸	左手握拳	右手握拳

（测试数据中 1234 会统一以“200”标签给出），参赛方只需关注 trial 标签即可）

标签总结：

单个 trial：3 个（240+类别标签+241）

单个 block：123 个（242+3*40+243+9）

表 4 电极区域-导联名称

电极区域	导联名称
额极区 (Fp)	Fp1, Fpz, Fp2
额区前部 (AF)	AF7, AF3, AFz, AF4, AF8
额区 (F)	F5, F3, F1, F2, F4, F6, F7, F8, F9, F10
额中央区 (FC)	FC5, FC3, FC1, Fz, FCz, FC2, FC4, FC6
中央区 (C)	C5, C3, C1, Cz, C2, C4, C6
额颞区 (FT)	FT7, FT8, FT9, FT10
颞区 (T)	T7, T8, T9, T10
颞顶区 (TP)	TP7, TP8, TP9, TP10
中央顶区 (CP)	CP5, CP3, CP1, CPz, CP2, CP4, CP6
顶区 (P)	P7, P5, P3, P1, P2, P4, P6, P8, P9, P10
顶枕区 (PO)	PO5, PO3, Pz, POz, PO4, PO6, PO9, PO7, PO8, PO10
枕区 (O)	O1, Oz, O2
耳电极 (A)	A1, A2

数据流采用模拟在线方式提供。每调用一次数据读取方法，可获得一个新数据包，数据包中包含 40ms 的实验 EEG 数据(最后一个数据包长度可能小于 40ms)，以及在该数据包记录过程中收到的 trigger 信息。在同一 block 中，数据包按照时间顺序依次发送。若测试数据中包含多组 block 数据，则一组 block 数据发送完毕后，数据读取方法被再次调用时，将会开始下一组 block 数据的发送。而当所有实验数据发送完毕后，程序终止标记 finished_flag 将被置为 1。参赛算法检测到 finished_flag 为 1 后，需要自行结束 run()方法执行。需要指出，由于实验数据来自真实 EEG 信号，每个 block 中最后一个数据包的长度可能不是一个定值。在算法开发过程中请特别注意。另外，极个别被试的某些导联可能没有采集到数据，但并不影响算法进行分类。

二、 算法规范

参赛算法调用数据读取方法获取脑电数据。数据读取方法被调用一次，比赛系统会返回一个新数据包，参赛算法可以对新数据包进行缓存并处理。当算法认为接收到的数据足以满足判决条件时，需要调用反馈方法向比赛系统报告识别结果。比赛系统根据反馈正确率，综合计算出最终准确率。

参赛算法需要同时满足以下几个约束条件：

1、 试次起止约束：

在对单一试次数据的检测识别过程中，参赛算法需要在接收到该试次 `trigger` 之后开始检测，并且在接收到成绩提交点 `trigger` 前进行反馈报告。否则，报告结果将被判定为无效的识别结果。

2、 单试次最大数据长度约束：

本项目对于单一试次最长检测时间需小于 4 秒。从试次的想象 `trigger` 信号起，参赛算法最多采集 4 秒的 EEG 数据（不含 4 秒），否则该试次识别结果将被视为无效，该试次结果为-1。

3、 算法终止约束：

当接收到数据包中 `finished_flag = true` 时，意味着所有实验数据均已发送完毕，参赛算法需要停止处理并自行退出。

三、 赛题框架

1. 参赛者用例

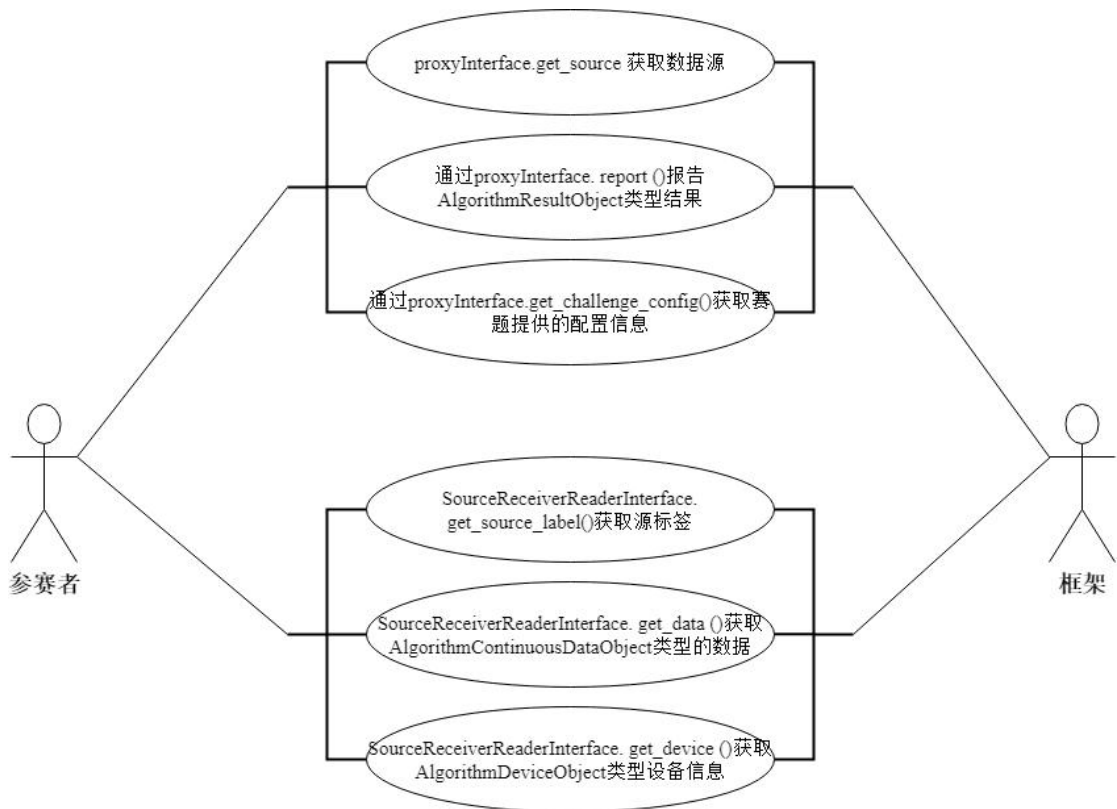


图 4 参赛者用例

2. 系统主体框架

(1) ProxyInterface 题目接口

该接口是面向参赛者的赛题接口，主要负责题目与参赛算法之间的数据传递及结果报告。参赛者可以通过该接口获取比赛数据，并通过该接口报告识别结果。比赛题目需要根据参赛算法获取比赛数据的次数，以及报告结果综合给出比赛评分。具体评分规则由出题单位自行拟定。

(2) ChallengeInterface 赛题接口

该接口是面向出题人的赛题接口，主要负责实现赛题的清洗数据和计算得分，框架通过该接口实现对赛题的调用。

(3) AlgorithmInterface 算法接口

通过该接口比赛题目可以对参赛算法进行验证计算。该接口由参赛者具体实现。在执行过程中，参赛算法需要通过 `proxyInterface.get_source` 获取数据源，继而通过 `SourceReceiverReaderInterface.get_source_label()` 获取源标签，`SourceReceiverReaderInterface.get_data()` 获取 `AlgorithmContinuousDataObject` 类型的数据 `SourceReceiverReaderInterface.get_device()` 获取 `AlgorithmDeviceObject` 类型设备信息，还可以通过 `proxyInterface.get_challenge_config()` 获取赛题提供的

配置信息。最终通过 `proxyInterface.report()` 报告 `AlgorithmResultObject` 类型结果，`AlgorithmResultObject` 类型结果中，需要根据赛题要求返回指定类型的数据。在本赛题中，最终需要返回字符串类型的标签，左手屈肘对应 ‘1’，右手屈对应 ‘2’，左手握拳对应 ‘3’，右手握拳对应 ‘4’。

同时，参赛者需要控制算法的计算复杂度，否则当运行时间超过预定长度时，则该算法比赛成绩将被视为无效。

3. 参赛者相关接口函数

参赛者需要根据 “`Algorithm\method\interface\AlgorithmInterface.py`” 中的 `AlgorithmInterface` 类实现算法类，并将算法类名称和路径写入 “`Algorithm\config\AlgorithmConfig.yml`” 文件中。

参赛者需在算法类中重写以协程的形式重写 `run` 方法，并在获取设备信息、获取数据和汇报结果时使用 `await` 关键字异步执行。框架运行时，会主动调用 `run` 方法。因此参赛者需要在 `run` 方法中读取数据，进行计算，并汇报结果。

其中，参赛者可以通过调用 `_proxy` 属性的 `get_source` 方法，并传入 `str` 类型的数据源名称参数，获取数据源（`SourceInterface` 实例）。如果存在多个源，可以依次读取。获取到数据源之后，可以使用 `await` 关键字异步调用其 “`get_device`” 方法获取设备信息（`AlgorithmDeviceObject` 实例），调用其 “`get_data`” 方法获取一个数据包（`AlgorithmDataObject` 实例）。

在需要汇报结果时，参赛者可以实例化一个 `AlgorithmResultObject` 类型的实例，并将结果赋值给该对象的 “`result`” 属性。随后将该实例作为参数传入并调用 “`_proxy`” 属性的 `report` 方法。

以下是关于上文中出现的类的说明。

(1) `ProxyInterface`

算法类与框架交互的代理类，在算法类实例化时，框架会实例化一个此类的对象放入算法类的 “`_proxy`” 属性中。

1) `get_source` 方法

无传入参数。返回一个 `SourceInterface` 对象。用于算法从框架获取数据源对象。

2) `report` 方法

必须传入且仅传入一个 `AlgorithmResultObject` 对象，无返回值。

(2) SourceInterface

1) get_source_label 方法

无传入参数。返回 str 类型的数据源名称。

2) get_data 方法

无传入参数。返回一个 AlgorithmDataObject 对象。用于算法从数据源获取数据。调用时需使用 await 关键字异步调用。

3) get_device 方法

无传入参数。返回一个 AlgorithmDeviceObject 对象。用于算法从数据源获取数据采集设备的设备信息。调用时需使用 await 关键字异步调用

(3) AlgorithmDeviceObject

1) data_type 变量

str 类型；数据源中的数据类型。

2) channel_number 变量

int 类型；数据包的通道数。

3) sample_rate 变量

float 类型；数据的采样率。

4) channel_label 变量

str 类型构成的 list 类型；数据各个通道的标签

5) other_config_map 变量

dict 类型；其他设备配置信息

(4) AlgorithmDataObject

1) start_position 变量

int 类型；数据包内数据的起始位置

2) data 变量

numpy.ndarray 类型；数据内容，每行表示一个导联，最后一行通道为 trigger 通道。

3) subject_id 变量

str 类型；当前数据包的被试编号

4) finish_flag 变量

bool 类型；处理结束标志位，当检测到该标志位为 true 时，算法需终止运行

(5) AlgorithmResultObject

1)result 变量

str 类型；算法汇报的结果。

4. 提交样例

参考配套代码。

参赛者可通过修改 Algorithm 文件夹中的代码完成算法,为了避免未知错误,请勿在主目录内添加文件夹。完成后重新打包程序(包含 AlgorithmImplement 文件夹和 config.toml) --> 分组 --> 具体分组 --> 计算单元 --> 定义计算单元 --> 上传程序包 --> 提交到比赛 --> 选择比赛 --> 部署 -->完成比赛。

部署完成后在具体到比赛中的排行榜中查看 比赛成绩;

需要注意的是,为防止参赛者修改代码框架作弊,保护评分程序会完全覆盖参赛者的代码(除了 AlgorithmImplement 目录和 config.toml)在提交到比赛 --> 部署时,启动的实际为评分程序 + 参赛者的 AlgorithmImplement 目录,其余运行配套代码均为服务器内置程序(包括 main.py 等文件,服务器内置评分程序与范例中程序框架基本相同,但包含评分功能和读取服务器比赛数据功能)。

5. 评分方式

对于每个试次的预测结果,根据预测正确程度分配分值。具体分配方案如下:

- 完全正确(预测类别与真实类别完全一致): 得 100 分。
- 仅侧别正确(预测与真实同属左手或同属右手,但具体动作类别不一致): 得 50 分(部分得分)。
- 完全错误(预测的左右侧别与真实不符): 得 0 分。

其中,每个 trial 提交一次结果。最终的总成绩将由所有被试的全部 block 的成绩加总得出。

特别需要指出:

当正确率 P 小于 $1/M$ 时,当前次提交对应的 ITR 强制为 0。

6. 性能评估方法

参赛算法通过数据读取方法获取新数据包。当所得数据包内含有 trigger 信号时,评分系统将自动开始记录算法识别过程中所使用 EEG 信号的长度,直至反馈方法被调用。从 trigger 开始到反馈方法被调用时所获取的 EEG 数据长度将作为该试次的模拟试次时长。而平均准确率将根据算法反馈结果与真实刺激的一致性进行计算。

需要特别指出，在本比赛项目中每一个包含 **trigger** 的数据包，其依然被视为是前一试次的数据。而新试次数据是从包含 **trigger** 数据包的下一个数据包开始计算。因此参赛算法不可在获取到包含 **trigger** 信号的数据包时立刻反馈，而最早需获取到下一数据包后才可反馈。

7. 结果反馈异常处理

1) 单次报告

在一个试次时间内，参赛算法需要反馈一次结果。

2) 结果未反馈

若在一个试次时间内，参赛算法未反馈结果，则判决结果将被记录为误判。

3) 结果反馈超时

结果反馈时，从当试次 **trigger** 开始计算参赛算法已经获取了超过当前限制时间的 EEG 数据，则本次判决结果将被记录为误判。

4) 算法执行超时

为满足脑-机接口系统实时处理需求，本项目同时对参赛算法的计算复杂度有一定要求。本比赛项目将会根据比赛数据量大小确定一个计算时间。若算法复杂度过高导致系统运行超时，则该算法比赛成绩将被视为无效。

8. 奖项设置

设特等奖、一等奖、二等奖、三等奖，三个奖项获奖总赛队数量为 20 支队，选取初赛前 10 支队伍参加决赛。

决赛中根据分数排名，设特等奖 1 个、一等奖 2 个、二等奖 3 个，三等奖 14 个，并颁发获奖证书。