

基于非侵入式脑机接口的运动康复训练技术赛项说明

一、 实验范式

每组实验包含左手运动想象任务数据，右手运动想象任务数据，和双脚运动想象任务数据，如图 1 所示。本次比赛提供每位受试者的 3 个 block 的数据作为训练数据，该受试者的 3 个 block 的数据作为测试数据，每个 block 包含 30 个 trial（左手、右手、双脚各十个随机出现）；每个 trial 包含 2s 的任务提示、4s 的运动想象以及 2s 的休息，如图 1 所示。本次比赛采用的数据包含正常人数据与患者数据。

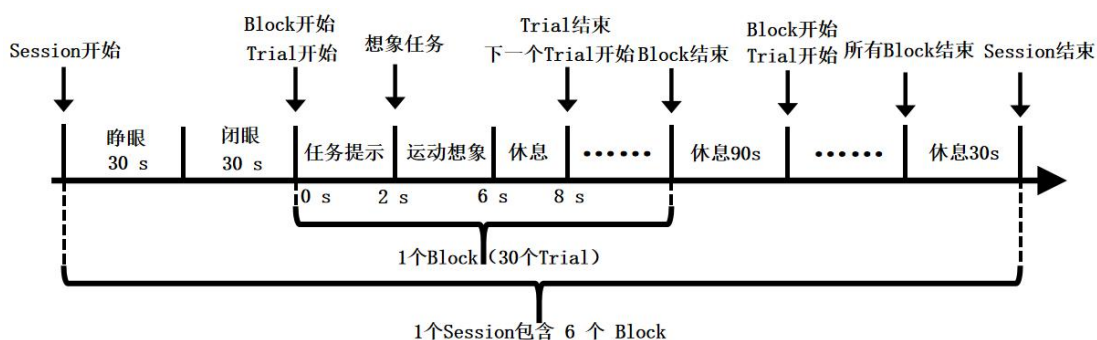


图 1 实验过程

实验数据使用博睿康 64 通道脑电采集设备采集，第 65 导联为 trigger 信息，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的数据均置为 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 定义如表 1、2、3 所示。整个实验范式包含 Session 开始标 250、Session 结束标 251，Block 开始标 242、Block 结束标 243。

训练数据中每个 trial 包含十种 trigger，成绩提交点标 241，第一段数据开始：左手标 11、右手标 21、双脚标 31，第二段数据开始：左手标 12、右手标 22、双脚标 32，第三段数据开始：左手标 13、右手标 23、双脚标 33。如图 2 所示。

测试数据中每个 trial 包含四种 trigger，运动想象开始标 200，已想象 2s 标

202, 已想象 3s 标 203, 成绩提交点标 241。如图 3 所示。

其余为系统预留标。

表 1 整体实验 Trigger 定义

定义	Session 开始	Session 结束	Block 开始	Block 结束
Trigger 号	250	251	242	243

表 2 训练数据 Trigger 定义

定义	提交点	左手 第一段	右手 第一段	双脚 第一段	左手 第二段	右手 第二段	双脚 第二段	左手 第三段	右手 第三段	双脚 第三段
Trigger 号	241	11	21	31	12	22	32	13	23	33

表 3 测试数据 Trigger 定义

定义	运动想象开始	已想象 2s	已想象 3s	成绩提交点
Trigger 号	200	202	203	241

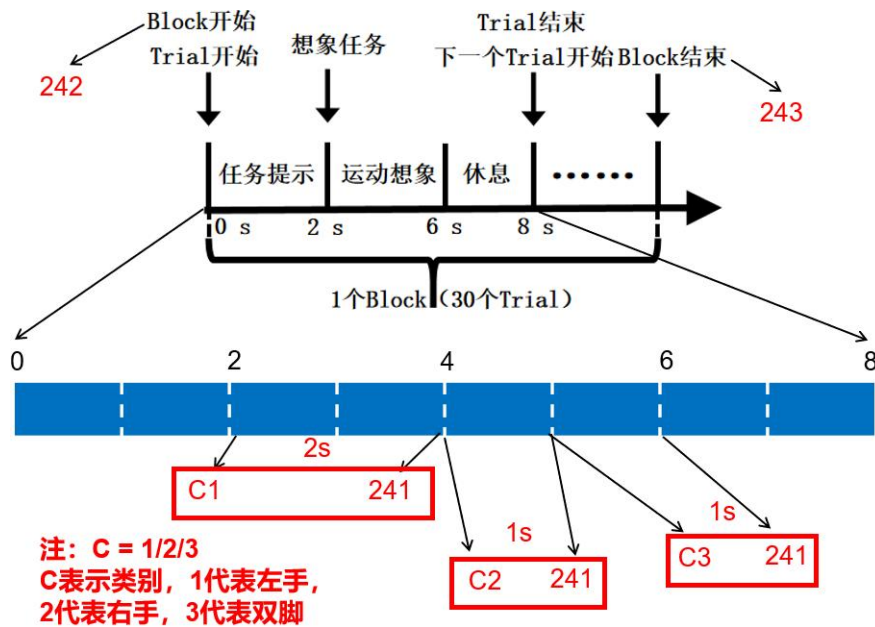


图 2 训练数据 Trigger 用例

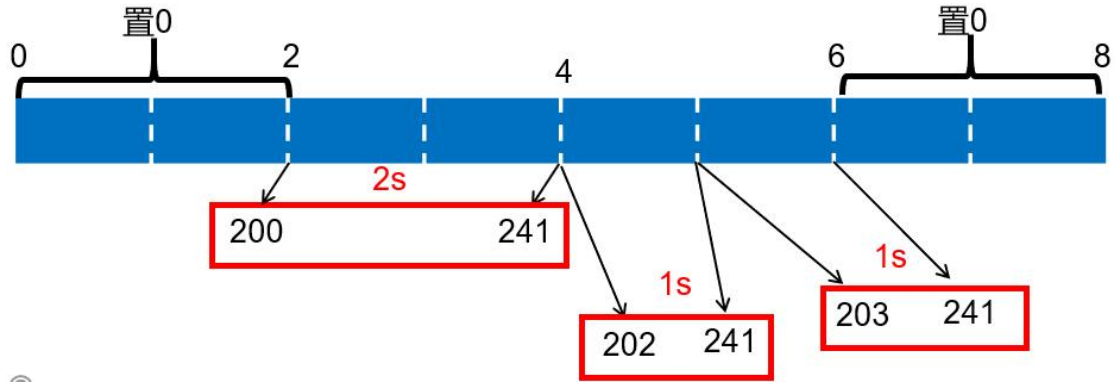


图 3 测试数据 Trigger 用例

表 4 导联序号-导联名称

导联序号	1	2	3	4	5	6	7	8
导联名称	Fpz	Fp1	Fp2	AF3	AF4	AF7	AF8	FZ
导联序号	9	10	11	12	13	14	15	16
导联名称	F1	F2	F3	F4	F5	F6	F7	F8
导联序号	17	18	19	20	21	22	23	24
导联名称	FCz	FC1	FC2	FC3	FC4	FC5	FC6	FT7
导联序号	25	26	27	28	29	30	31	32
导联名称	FT8	Cz	C1	C2	C3	C4	C5	C6
导联序号	33	34	35	36	37	38	39	40
导联名称	T7	T8	CP1	CP2	CP3	CP4	CP5	CP6
导联序号	41	42	43	44	45	46	47	48
导联名称	TP7	TP8	Pz	P3	P4	P5	P6	P7
导联序号	49	50	51	52	53	54	55	56
导联名称	P8	POz	PO3	PO4	PO5	PO6	PO7	PO8
导联序号	57	58	59	60	61	62	63	64
导联名称	Oz	O1	O2	ECG	HEOR	HEOL	VEOU	VEOL

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 finishedFlag 将被置为 1。参赛算法检测到 finishedFlag 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。比赛系统根据数据读取方法的调用次数计算出算法使用有效数据长度，并结合反馈正确率，综合计算出平均模拟信息传输速率。

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 0。

3、 算法终止约束：

当接收到数据包中 Endflag = 1 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

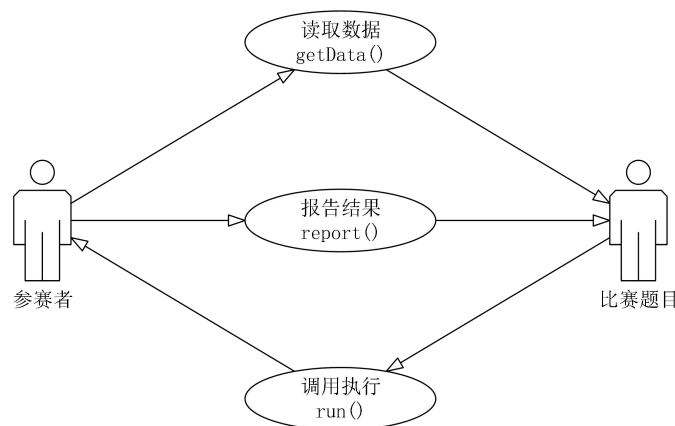


图 4 参赛者用例

2. 系统主体框架

(1) ProxyInterface 题目接口

该接口主要负责赛题程序与外部执行系统的连接。该框架接口的实现类为 `AlgorithmSystemManager`，实现接口定义的所有函数。

(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 `ProblemInterface` 接口获取数据，并且通过该接口报告结果。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。

算法接口及题目接口可以由出题单位自行设计修改。本框架的比赛题目仅为参考范本，比赛题目可以由出题单位具体实现。

3. 数据模型

(1) DataModel 参赛者数据模型

1) `data`: `float` 类型矩阵，分段数据。例如包含有 64 导 EEG 数据+1 导 trigger 信号，在 1000Hz 采样率下，以 40ms 对数据分段，则单次获取的 `data` 为 64*40 个点。

2) `start_pos`: `int` 类型标量，当前分段数据起始时刻相对于该 `block` 数据起始时刻的索引位置。

3) `subject_id`: `int` 类型标量，当前数据来源受试者序号。

4) `finish_flag`: `bool` 类型标量，测试结束标志。当参赛算法通过 `data_model.finish_flag` 获取数据包中该字段为 `True` 时，需要自行退出程序运行。

4. 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。**注意，出题方可规定好算法的配置文件的信息，并给出示例文件，让参赛者基于已有文件和类开发，不得更改文件名称和类名。**

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。

5. 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)。

6. 评分方式

本系统以平均信息传输速率作为评分标准:

$$ITR = \frac{1}{3} \left(\sum_{i=1}^3 60 \cdot \frac{\log_2 M + P_i \log_2 P_i + (1 - P_i) \log_2 \frac{1 - P_i}{M - 1}}{T_i} \right)$$

其中, T_i 表示时间窗长度($T_1 = 2, T_2 = 3, T_3 = 4$), M 表示目标个数, P_i 表示识别正确率。ITR 的单位是 bits/min。每个 trial 需提交三次结果, 分别计算三次提交的 ITR 后取平均作为一个 Session 的成绩。

特别需要指出:

① 有效提交:

- a. 第一次提交结果时在框架中获取的数据窗长在 2s 以内 (包含 2s);
- b. 第二次提交结果时在框架中获取的数据窗长在 3s 以内 (包含 3s);
- c. 第三次提交结果时在框架中获取的数据窗长在 4s 以内 (包含 4s)。

不满足条件的判定为无效提交。

有效提交的结果会进行正确率统计, 无效提交的结果强制判定为错误。

② 计算结果:

- a. 所有第一次提交的结果汇总后统计正确率, 按照 $T_1 = 2$ 计算 ITR;
- b. 所有第二次提交的结果汇总后统计正确率, 按照 $T_2 = 3$ 计算 ITR;
- c. 所有第三次提交的结果汇总后统计正确率, 按照 $T_3 = 4$ 计算 ITR。

最终成绩由 3 个 ITR 取平均后给出。

③ 当正确率 P_i 小于 $1/M$ 时, 当前次提交对应的 ITR 强制为 0。

7. 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。

8. 结果反馈异常处理

1)重复多次报告

在一个试次时间内，参赛算法需要反馈三次结果(2s,3s,4s)，超出则仅按照前三次反馈的时间及结果进行记录。

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。

四、 奖项设置

本赛项设特等奖、一等奖、二等奖、三等奖，四个奖项获奖总赛队数量为 20 支队，选取初赛前 10 支队伍在北京“2024 世界机器人大赛锦标赛（北京）”现场参加决赛。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。

其中特等奖、一等奖、二等奖设奖金奖励。

特等奖 1 名-- 奖金 30000 元、比赛证书

一等奖 2 名 -- 奖金 20000 元、比赛证书

二等奖 3 名 -- 奖金 10000 元、比赛证书

三等奖 14 名 -- 比赛证书。

奖金金额为税前金额。

五、 时间安排

预计在 6 月 17 日开始报名，初赛预计 6 月 19 日开始比赛，决赛在八月份北京线下开展。