SSVEP 赛题-窄带随机编码

本赛题基于窄带随机编码视觉诱发电位设计多目标键盘拼写实验，采集多人数据集，旨在测试在无训练或跨被试训练的条件下，不同算法识别窄带随机编码视觉诱发电位的性能。

实验范式：

实验范式为键盘拼写，刺激范式如图 1 所示。实验采用 on 型刺激，图 1 中每个目标的浅灰色部分作为不闪烁的背景（背景亮度约为 25 cd/m²，0 为黑，255 为白），白色网格部分为视觉刺激区域，在高于背景亮度（on）的条件下进行闪烁。刺激包含 40 个目标（包括数字 0-9、字母 A-Z、“, ”、“.”、空格和退格），所有目标刺激的编码能量均主要集中在 15~25Hz，显示器刷新率为 120 Hz，刺激序列长度为 120 帧，即完整刺激一次编码需要 1 秒。实验中，刺激通过对屏幕上每个目标亮度的随机编码调制实现（亮度范围：约为 25 - 200 cd/m²，on 型）。刺激序列如图 2 所示，图中给出的是每个目标的刺激亮度变化序列，具体数值从赛题网站下载，算法框架中的 Sequence.npy 文件给出了所有 40 个序列（采样率为 **120Hz**）。每个试次包含提示 1 秒，刺激 4 秒（即同一编码刺激 4 轮），反馈 1 秒。一个 block 共 40-45 个试次，每次实验共 3 个 block。被试视力或矫正视力正常，在实验过程中被要求集中注意力，刺激时注视提示的目标。实验中各个试次使用并口进行同步，trigger 标记在每个刺激开始时刻。

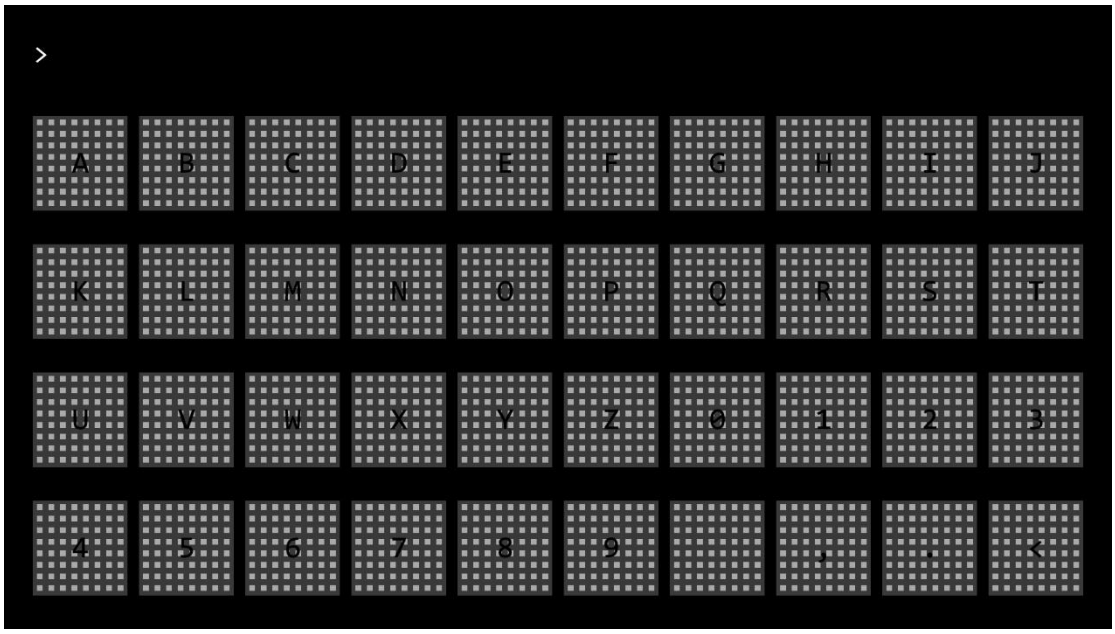

图 1 40 目标刺激界面

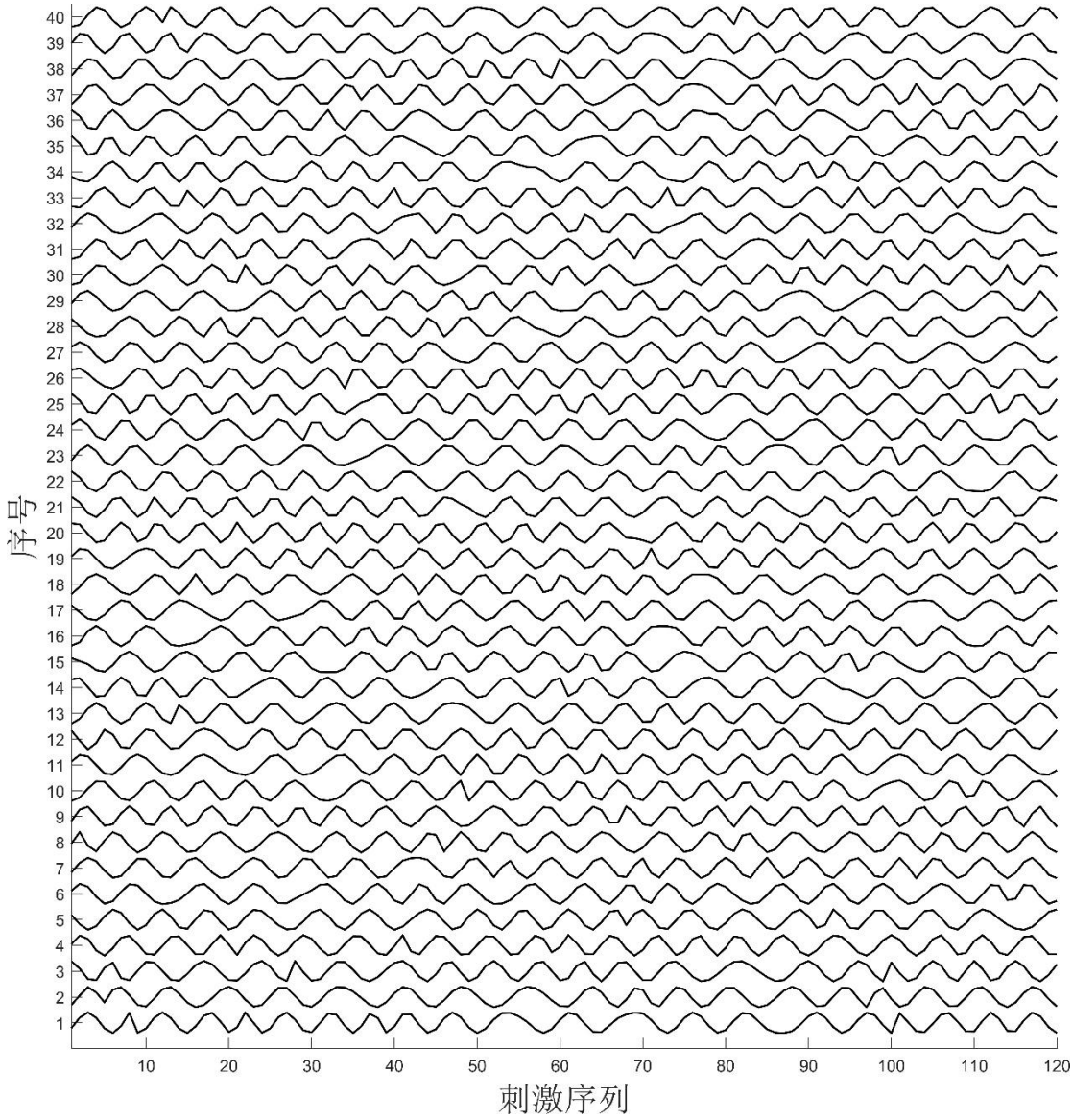

图 2 各个目标使用的刺激序列，横坐标为帧序号，纵坐标为刺激序列标签

实验数据:

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，如导联序号-名称对照表如表 1 所示。数据原始采样率为 1000Hz，降采样到 **250 Hz**，未做其他处理。**工频噪声频率为 50Hz。线上测试时赛题数据的试次开始 trigger 全部被替换为 240，**仅用作同步，不包含试次具体信息。除试次开始 trigger 之外数据还包含其他 trigger 用于系统控制，具体 trigger 定义如表 2 所示。参赛者可只关注 trial 开始 trigger 的时刻，系统会根据真实 trigger 计算结果。

表 1 导联序号-名称对照表

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

表 2 系统 trigger 定义

定义	Trial 开始	Trial 结束	Block 开始	Block 结束	数据采集开始	数据采集结束
Trigger 号	1-40	241	242	243	250	251

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 EEG。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。

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的有效数据长度，并结合反馈正确率，综合计算出模拟信息传输速率。

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 4 秒内进行反馈报告。否则，报告结果将被错判。

2、 单试次最大数据长度约束：

本项目对于单一试次最大获取数据量需小于 4 秒。从试次的 trigger 信号开始，参赛算法给出反馈结果使用的数据长度不应超过 4 秒，否则该试次识别结果将被视为无效，该试次结果为 0。不同被试，不同试次的检测时间可以不同。

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。

赛题框架

参赛者用例

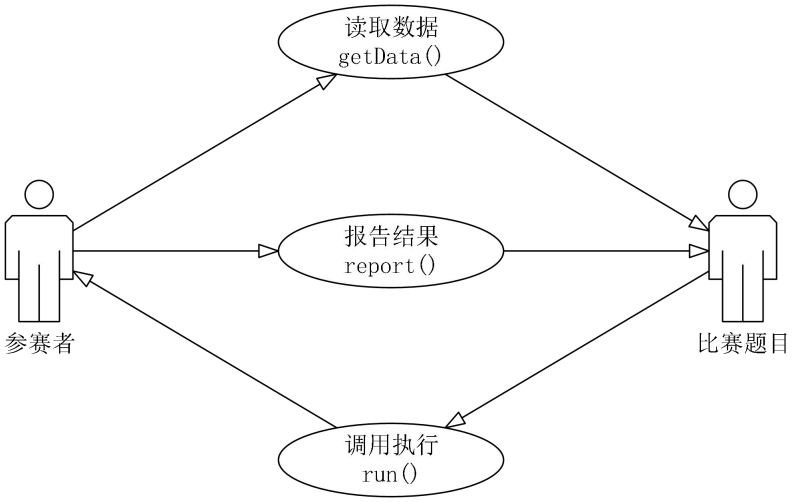

图 3 参赛者用例

系统主体框架如图 3 所示。

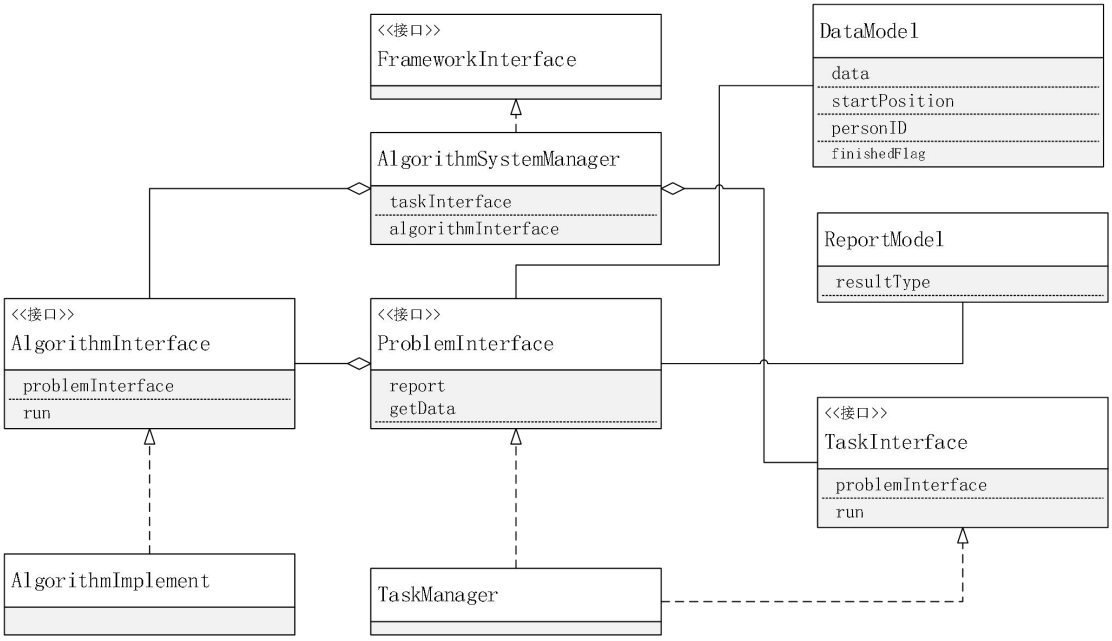

图 4 系统主体框架

(1) FrameworkInterface 框架接口

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) ProblemInterface 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，以及报告结果的正确率综合给出比赛评分。

(3) TaskInterface 赛题接口

该接口主要负责实现赛题的数据填充、获取得分、清除数据以及清除报告结果，框架通过该接口实现对赛题的调用。

(4) 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。**参赛者需要实现该接口**。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，系统将自动终止该计算进程，所获成绩无效。

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 65*40 个点。

2) startPosition: 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) personID：int 类型标量，当前数据来源受试者序号。

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

参赛者相关接口函数

(1) ProblemInterface

该接口由出题方负责实现，包括数据获取方法及结果反馈方法。在算法运行前，该接口的实现类会被注入参赛算法实现类中。算法执行过程中，可以调用该接口获取数据，并通过

结果反馈方法报告识别结果。出题方根据数据获取方法的被调用次数，及结果反馈的正确性进行综合评分。

1) def get_data(self):
输入参数：无
输出参数：DataModel
实现功能：获取下一分段实验数据。

2) def report(self, reportModel):
输入参数：ReportModel
输出参数：无
实现功能：反馈识别结果。

(2) AlgorithmInterface

参赛者需要将程序运行过程填入 run 函数中。在算法执行过程中，通过 ProblemInterface 接口 get_data 方法获取 DataModel 类型数据，并通过 report 方法返回 ReportModel 类型结果。当通过 get_data 获取的 DataModel 数据中 finishedFlag 为 true 时，意味着数据处理完毕，该函数需要自行退出运行。

1) def run(self):
输入参数：无
输出参数：无
实现功能：算法分析过程。

提交格式

本赛题程序使用 python 语言编写，需提交基于 python 3.8 版本的扩展名为.pyc 的文件。

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。

部署完成后在赛题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）。

评分方式

本系统以模拟信息传输速率作为评分标准:

$$ITR = 60 \cdot (\log_2 M + P\log_2 P + (1-P)\log_2 \frac{1-P}{M-1})/T$$

其中，T 表示平均试次时长，M 表示目标个数，P 表示识别正确率。ITR 的单位是 bits/min。特别需要指出，本系统中 ITR 是按照理想 ITR 进行计算，即平均试次时间不包含模拟休息时长。

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。

结果反馈异常处理

1)重复多次报告
在一个试次时间内，参赛算法多次反馈结果，则仅按照第一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，则该试次时长将被记录为 4 秒，同时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 4 秒的 EEG 数据，则判决结果将被记录为误判，同时该试次时长将按照从 trigger 开始时刻到结果反馈时刻的数据长度进行计算。

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。