面向元宇宙的脑电信号离体生成和检测技术 II 赛项参赛说明

一、项目介绍

元宇宙未启，身份认证先行。近年来，基于脑电信号的个体身份识别技术取得了长足进展，成为研究的前沿热点。然而，脑电信号模拟生成和检测技术仍是一片有待开垦的新领域。这不仅是实现元宇宙身份认证的必要一环，更是探索人工智能与生物智能边界的重要课题。随着深度学习的突破性进展，脑电信号模拟生成技术展现出了广阔的前景。但这样的生成技术能在多大程度上逼真地再现人类的脑电特征，实现脱离活体的数据生成，仍需要我们持续的探索和不断的验证。

本赛题设计能够判别静息态脑电数据真实性与所属被试身份的安防系统，要求参赛者在线生成仿照数据对赛题系统进行实时骇入，在确保被系统识别为准确被试的同时不被系统检测为仿照数据。比赛目标是探究是否能够模拟生成既高度拟真又具备个体特征的脑电数据，为这一前沿领域提供一个创新研究和技术交流的平台。比赛数据集为真实的静息态脑电数据，数据及判别系统皆由深圳大学提供。

作为一项前瞻性的探索，本赛题不仅将推动脑电信号离体生成和检测技术的进步，更有望引发人工智能与生物智能交叉领域的思考和讨论。我们诚邀来自全球相关领域的同学，科研人员、工程师加入这场探索未来的比赛，用你们的洞察力和创新能力在这一新兴领域开辟道路！一起探索科技的无限可能，为未来技术的发展共同铺就道路。

二、实验范式

赛题实验范式为静息态，每组实验包含睁眼和闭眼数据，如图 1 所示。本次比赛组织包含来自不同受试者共计 15 个 Block 的数据作为训练数据集，每个 Block 包含 92 个 Trial，每个 Trial 时间长度相同，Trial 之间用不同长度的零值间隔，如图 1 所示。

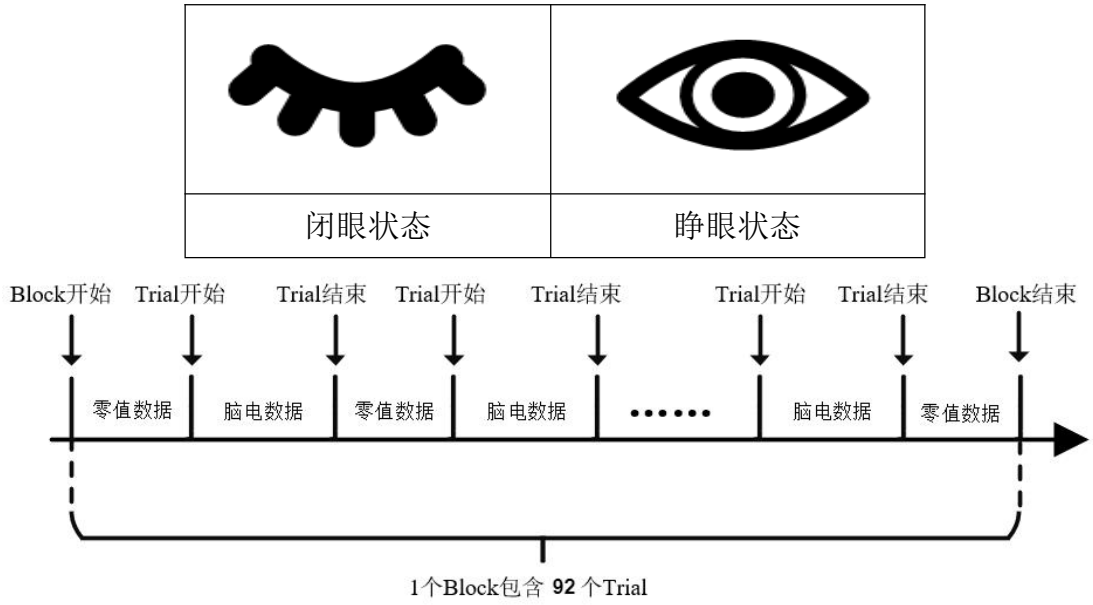

图 1 实验范式

三、实验数据

实验数据经由 64 通道 BrainAmp 设备采集后降采样至 250 Hz，重参考设置为共同平均参考，并选择通道 Fz，Cz，Oz，C3，C4 的数据，导联序号-名称对照表如表 1 所示。

赛题数据说明如表 2 所示，初赛开赛前赛题方提供基准数据集 Calibration_Test 供选手熟悉数据格式，随后根据初赛具体赛段在开赛前提供训练数据集 Calibration_A 与 Calibration_B，三者皆为真实 EEG 数据且**被试不重叠**。正式比赛时，参赛选手需设计生成式算法并根据接收的被试 ID 号（赛题框架所发出的 Trial 开始的 Trigger 号）生成指定大小的 EEG 数据并上传提交。

具体 Trigger 定义如表 3 所示。整体数据中共有 26 种 Trigger，分别为表示 Block 开始的标记 242、Block 结束的标记 243、提示选手开始生成并提交数据的 Trial 开始标记 1-23、生成与提交应结束的 Trial 结束标 241。

导联序号	1	2	3	4	5
导联名称	Fz	Cz	Oz	C3	C4

表 1 导联序号-名称对照表

数据集分发时间	初赛开赛前	A 赛前	B 赛前
数据集名称	Calibration_Test	Calibration_A	Calibration_B

表 2 分发数据集概况

定义	Trial 开始（被试 ID）	Trial 结束	Block 开始	Block 结束
Trigger 号	1-23	241	242	243

表 3 赛题 Trigger 定义

四、算法规范

参赛算法调用数据读取方法获取 Trigger 数据。数据读取方法被调用一次，比赛系统会返回一个新的 Trigger 数据包，数据包可能包含上述约定的 Trigger 号，需注意 Trigger 数据包也可能全为 0。当算法完成数据生成后，需要调用反馈方法向比赛系统报告生成结果。比赛系统根据对数据规范性进行验证后进行判别，最终结合 **生成数据对判别系统的“攻破”情况** 与 **被试判别的对应情况** 计算参赛队伍得分。

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 之后开始生成，并且在接收到 Trial 终止 Trigger 后停止生成，接收到终止 Trial 后直到下一个起始 Trial 前的报告结果均为无效结果。

2、单试次汇报次数约束：

本项目对于单一试次生成时间需等于当前试次开始 Trigger 标签出现到结束 Trigger 标签出现的间隔时间。从 Trigger 开始信号起，参赛算法应持续汇报 EEG 数据直到当前试次的 Trigger 结束信号出现。

3、算法终止约束：

当接收到数据包中 Endflag = 1 时，意味着所有 Trigger 信息均已发送完毕，

参赛算法需要停止处理并自行退出。

五、赛题框架

1、参赛者用例

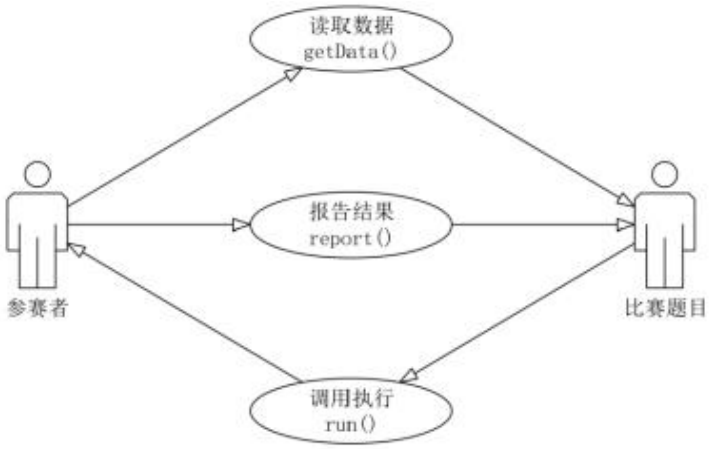

图 4 参赛者用例

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。

（2）ChallengeInterface 赛题接口

该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，系统将自动终止该计算进程，所获成绩无效。

3、数据模型

（1）DataModel 参赛者数据模型

1）data：int 类型矩阵，在 250Hz 采样率下，单次获取的 data 为 50 个点。

2）personID：int 类型标量，当前数据来源序号，注意该数据在本赛题中属于无效信号。

3）blockID：int 类型标量，当前数据来源 Block 序号。

4）srate：int 类型标量，数据采样率。

5）ch_names：list 类型，当前数据通道信息。

6）nchan：int 类型标量，当前数据通道数量。

7）dimensions：tuple 类型，当前数据大小。

4、参赛者相关接口函数

（1）ChallengeInterface

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 receive_message(self, receive_model: BaseMessageModel):

输入参数：BaseMessageModel

输出参数：DataModel

实现功能：处理框架发送的数据。

2）def report(self, reportModel):

输入参数：ReportModel

输出参数：ResultModel

实现功能：处理算法汇报的结果。

（2）AlgorithmInterface

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，参赛者需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。当通过 get_data 获取的数据中 finish_flag 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 get_source(self)

输入参数：无

输出参数：SourceInterface 对象

实现功能：用于算法从框架获取数据源对象。

2）def get_data(self):

输入参数：无

输出参数：AlgorithmDataObject 对象

实现功能：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）def get_device(self):

输入参数：无

输出参数：AlgorithmDeviceObject 对象

实现功能：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。

4）def run(self):

输入参数：无

输出参数：无

实现功能：算法分析过程。

5)def report(self, algorithm_result_object: AlgorithmResultObject):

输入参数：必须传入且仅传入一个 AlgorithmResultObject 对象

输出参数：无

实现功能：用于算法向框架报告结果。

提交格式

参赛者需根据比赛提供被试 ID 汇报 numpy 标准类型的生成脑电数据包，单个数据包的维度为 5*50。

本赛题程序使用 python 语言编写，需提交基于 python 3.10 版本的扩展名为.pyc 的文件。

5、提交样例

参考配套代码。

参赛者可通过修改“Algorithm\method\impl”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“Algorithm\method\impl”文件夹和“AlgorithmConfig.yml”）--> 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“Algorithm\method\impl”目录和“AlgorithmConfig.yml”） 在提交到比赛 --> 部署时 ，启动的实际为 评分程序 + 参赛者的“Algorithm\method\impl” 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

本赛题采用如下形式作为评分标准：

$$Result = \omega * A + (1 - \omega) * B$$

其中， A 为被检测为不被检测为虚假数据的成功率，B 为被系统判别为要求的被试的成功率。

7、结果反馈异常处理

1）结果未反馈

赛题系统会对选手的提交次数进行计数，确保选手的提交次数等同于接收到的 Trigger 数据包个数，当选手遗漏提交时系统会视选手放弃该试次，此时判别器记生成数据被检测为虚假数据并且非指定被试。

2）结果不符合预期

结果反馈时，系统会对提交的格式进行检测，如果选手的提交数据形式不是规定形式则判定为无效提交，此时该试次的处理情况同 1）。

六、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 21 支队，选取初赛前 9 支队伍在北京“2025 世界机器人大赛北京锦标赛”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 6 个，三等奖 12 个。

特等奖 1 名 ：奖金 20000 元，获奖证书；

一等奖 2 名 ：奖金各 10000 元，获奖证书；

二等奖 6 名 ：奖金各 5000 元，获奖证书；

三等奖 12 名 ：奖金各 2500 元，获奖证书。

奖金金额为税前金额，奖金由念通科技赞助，特别鸣谢。