

脑机接口（BCI）软件平台架构设计与实现： 文献综述

班级：智能 242 姓名：王紫璿 学号：U202443382

2025 年 11 月 23 日

1 引言

脑机接口 BCI 通过直接捕获解析大脑电生理信号转化为控制指令，在医疗康复、辅助通信、智能交互等领域具备核心价值。临床中已实现中风患者运动功能重建、渐冻症患者脑控通信等突破性应用，在 AR 和 VR 领域可构建沉浸式交互场景；软件平台作为 BCI 系统的核心枢纽，承担了信号处理、解码分析、反馈控制等关键功能，其架构设计更是直接决定系统性能与技术转化效率。随着人工智能、边缘计算等技术的快速发展，BCI 应用场景不断被拓宽，使用者对平台的实时性、闭环控制精度及易用性提出了更高要求，这些要求不断推动架构理念和技术路径持续迭代升级。

本文聚焦于 BCI 软件平台架构设计核心问题，系统梳理了 BCI2000、Open-ViBE 等八种代表性 BCI 软件平台，从设计理念、技术路径、功能模块、以及应用场景四个维度分析其核心特性，明确各平台优劣势。本文还深入分析了架构设计面临的核心挑战与权衡策略，并据此展望未来创新方向，为 BCI 软件平台的选型与继续研发提供了理论支撑与实践参考。

2 BCI 软件平台功能模块与架构需求

2.1 核心功能模块体系

BCI 软件平台的核心目标是实现“脑电信号-控制指令”的精准转化。经数十年技术发展，目前已经形成了标准化功能模块体系，各模块协同工作，保障了系统高效运行。

首先，数据采集模块作为信号入口，直接决定了后续处理的基础质量。该模块需适配 EEG、ECoG 等多类型采集设备，兼容 Neuroscan、BioSemi 等主流厂商产品，支持 USB、LSL 等多通信协议。其核心保障机制包括：高精度时钟同步，控制同步误差在毫秒级；数据缓冲及异常检测，确保信号捕获的连续性与完整性。

接下来，预处理模块针对脑电信号微弱、低信噪比的特性，通过数字滤波去除工频干扰与高频噪声，通过 ICA 算法分离眼电、肌电伪迹，再进行通道选择、重采样等关键操作，最后输出高质量信号。该模块算法的实时性与有效性将直接影响系统整体延迟与解码精度，是平台性能优化的关键环节之一。

特征提取模块能够从预处理后信号中提取反映用户意图的关键特征。主要包括：核心覆盖时域（如 ERP 波幅）、频域（如 μ 波功率）、时频域（小波变换结果）及空间域（如 CSP 特征）四大维度。最终的解码效果与特征的区分度和稳定性直接相关。该模块需要支持标准化算法集成与自定义提取逻辑，以此来适配不同 BCI 任务场景。

解码模块的功能是实现特征到用户意图的映射，是平台核心逻辑单元。传统方案以 SVM、LDA 等机器学习算法为主，有着低复杂度、高实时性的优势，能够很好适配简单任务场景；相比之下深度学习模型如 CNN、Transformer 等可自动学习深层复杂特征，在多意图识别等复杂场景中表现更为优秀。该模块需要支持算法快速切换与模型全流程管理，包括训练、验证、部署。

反馈模块则通过视觉——光标移动、虚拟交互、听觉——频率提示音、触觉——振动反馈等多模态形式，将解码结果实时反馈给用户，形成“意图输出-反馈调整-意图优化”的闭环循环，持续提升控制精度。反馈模块需要保障低延迟与同步性，同时需要支持反馈参数的个性化配置。

最后，实验控制、并发处理、数据记录、用户界面及扩展性安全模块则构成平台支撑体系：实验控制模块管理范式运行、事件同步与状态监控；并发处理模块通过多线程架构实现任务并行执行，保障实时性；数据记录模块支持 EDF 等

标准化格式存储与溯源；用户界面模块适配不同技术背景用户需求；扩展性安全模块通过插件化设计与权限管理保障个性化与数据安全。

2.2 架构设计核心原则

BCI 软件平台架构设计需平衡多维度需求。架构设计基本遵循七大核心原则：一是模块化设计，通过明确功能边界与标准化接口保障模块替换与扩展；二是并发处理，采用多线程架构提升并行响应能力；三是硬件抽象层，封装设备差异提供统一接口来提升兼容性；四是用户友好性，通过分层设计适配不同技术背景用户；五是开源生态，促进技术共享与社区协作；六是性能与灵活性权衡，要根据应用场景精准匹配架构设计；七是可维护性，能够通过规范设计与版本管理保障长期迭代。

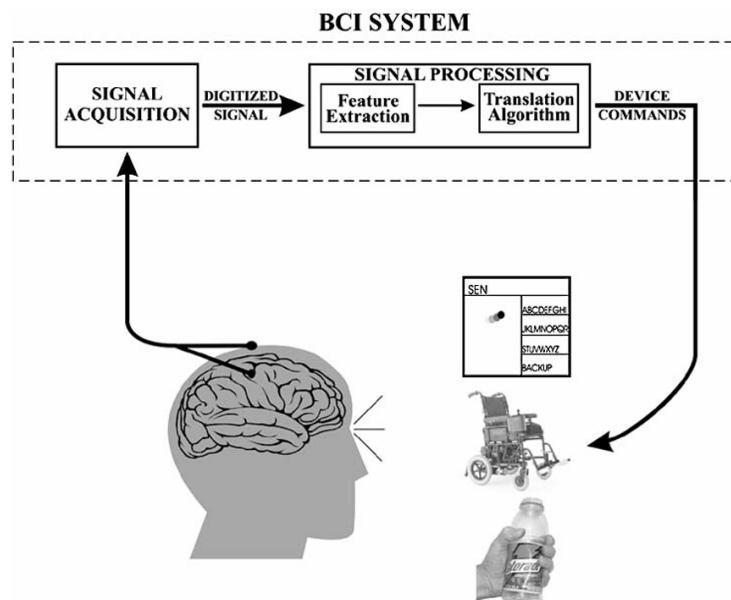


图 1: BCI 软件平台基本设计和操作^[1]

3 典型 BCI 软件平台分析

当前主流 BCI 软件平台基于差异化设计理念形成互补格局，本节从架构特性、技术实现、功能优势、应用局限这四个维度，对八个具有代表性的平台展开核心分析。

3.1 BCI2000

BCI2000 是 2000 年卡内基梅隆大学研发的开源标杆平台^[1]。该平台采用 C++ 开发，实现跨系统兼容，其核心为分布式多进程架构，分为采集、处理、应用、控制界面四大模块，并通过共享内存通信来保障 10-50 毫秒级的实时性。采集模块能够适配 Neuroscan 等主流设备，处理模块集成了滤波、CSP 等标准化算法，应用模块提供了光标控制等经典范式，支持 EDF、MATLAB 等格式导出。该平台的主要优势在于经过多年验证的成熟稳定性与深厚社区基础，适配高稳定性基础研究与初步临床场景。而核心局限则为二次开发门槛高，C++ 底层未提供轻量化扩展接口，使得非专业人员难以集成深度学习模型。不仅如此，传统窗口式界面也需要修改代码调整范式，不利于快速迭代。BCI2000 的主要架构如下图 2 所示。

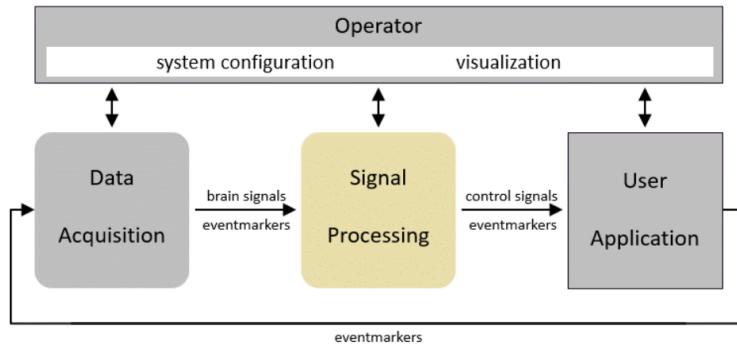


图 2: BCI2000 主要架构^[1]

3.2 OpenViBE

OpenViBE 由法国 Inria 研究所研发，以“可视化模块化”为核心设计理念^[2]，使用 C++ 底层结合图形化流程设计器，使用户能够通过拖拽模块，像搭积木一样

构建实验流程，大幅降低了跨学科协作门槛。OpenViBE 提供了一个名为 Designer 的图形化工具，允许用户通过拖放不同的“处理盒”来构建和可视化脑机接口数据处理流程。如下图 3 展示的这个窗口正是用于编辑和运行一个“场景”的界面。平台通过硬件抽象层适配 BioSemi 等设备，双缓冲区机制将延迟控制在 50-100 毫秒，能够满足基础研究实时性需求；平台内置了脑电波形显示、脑地形图绘制等可视化工具，提供 P300、运动想象等标准化范式模板。该平台的核心优势为易用性与可视化程度高，适配原型设计、教学演示及跨学科基础研究；主要局限在于深度学习集成需开发 C++ 插件，流程复杂且实时性不足。缺乏临床级闭环安全机制，模块接口耦合度高导致扩展困难，且难以适配临床闭环调控场景。

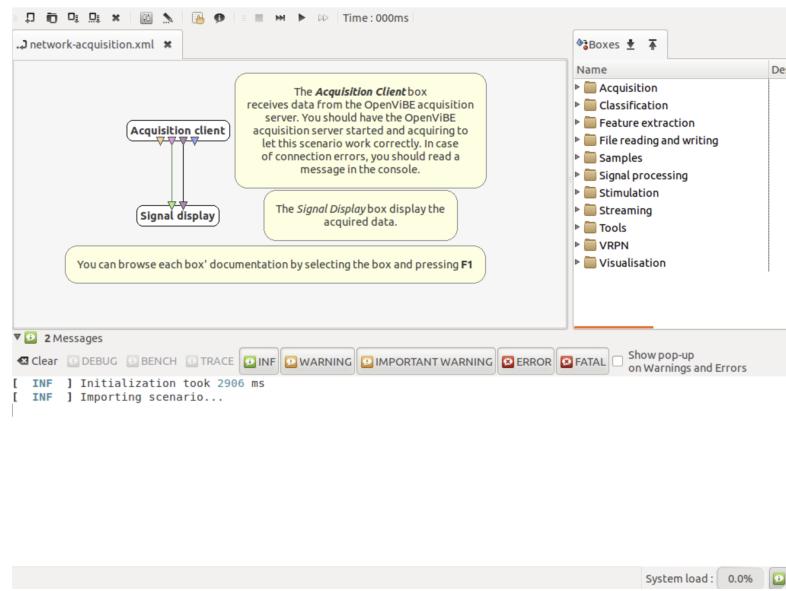


图 3: OpenViBE 提供了一个 Designer 图形化工具^[2]

3.3 MetaBCI

MetaBCI 是天津大学 2023 年发布的国内首个全流程开源平台^[3]，采用 Python 开发突破“离线-在线分离”瓶颈。其核心架构由 Brainda 离线分析、Brainstim 范式设计、Brainflow 实时处理三部分模块构成，如下方图 4 所示。Brainda 统一了 BCI Competition 等 20 余种数据集接口，集成了数十种预处理与解码算法；Brainstim 提供了可视化范式编辑工具，支持多模态刺激设计；Brainflow 采用了双进程架

构，通过共享内存保障实时传输。MetaBCI 的优势在于全流程覆盖实现离线-在线无缝衔接，Python 开发与中文文档降低了学习门槛，国内社区活跃具备生态潜力；局限为 Python 执行效率限制了毫秒级临床闭环场景表现，安全控制机制仅适配基础研究，尚未针对植入式 BCI 等复杂临床场景设计故障恢复与参数边界控制机制。

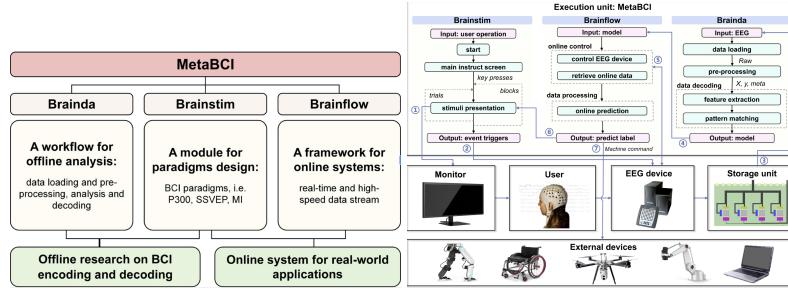


图 4: MetaBCI 的核心架构与模块设计^[3]

3.4 PyNoetic

PyNoetic 是德克萨斯大学研发的 Python 无代码平台^[4]，其核心为“无代码 + 可编程”双模式架构：节点式无代码环境封装标准化功能，拖拽连线即可构建流程，提供运动想象等范式模板；同时支持 Python 编写自定义节点并注册至平台。内置虚拟脑电生成模块，可模拟不同噪声水平的任务数据，适配无硬件调试场景。整体架构设计如下方图 5 所示。该平台的优势为分层适配多元用户：新手可通过可视化操作完成基础实验，而专家则可自定义算法节点，适配教学与快速原型开发；该平台的局限为 Python 性能限制高实时场景应用，社区规模较小导致第三方插件与模板不足，生态支撑薄弱。

3.5 Dareplane

Dareplane 是荷兰拉德堡德大学 2023 年发布的闭环调控专用平台^[5]，采用“编排器-模块”分布式架构，Python 中央编排器协调采集、解码、刺激、监控四大模块，硬件抽象层适配植入式刺激器等临床设备。核心竞争力为多层闭环控制策略：算法层支持自适应调控，安全层预设参数边界与紧急停止机制，可靠性层实现故障检测与恢复。实验验证显示系统延迟满足临床要求，在帕金森病 DBS 治

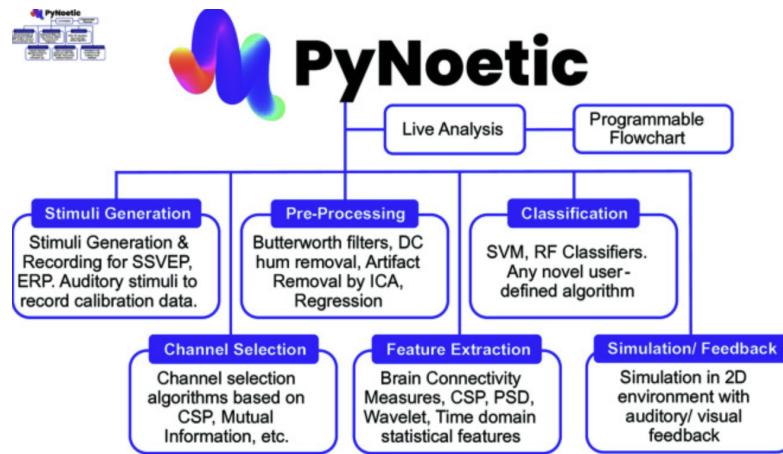


图 5: PyNoetic 平台概述图^[4]

疗中效果显著。优势为闭环优化设计与安全机制完善：由下方图 6 展示的平台控制室所示，Dareplane 不仅能够实时感知系统状态并做出调整，还具有可监控、可干预、有预案的安全机制，可以适配临床神经调控与侵入式 BCI 开发；局限为基础研究功能薄弱，缺乏范式设计与离线分析工具，使用需具备 Python 编程与临床知识，门槛较高。

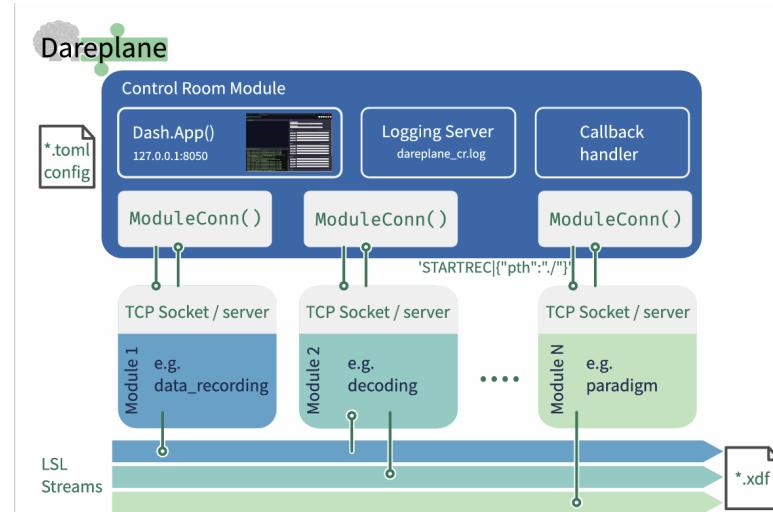


图 6: Dareplane 控制室^[5]

3.6 BciPy

BciPy 是加州大学研发的 Python 平台^[6]，以范式快速构建为核心定位，用模块化拆分实验流程。平台内置了 P300 拼写器等通信类范式，支持刺激样式、时序等参数可视化配置。能够适配 BioSemi 等主流设备，采用 PyQt 构建操作界面，支持在线实时处理与离线批量分析双模式。其平台优势在于范式构建高效灵活，代码结构清晰便于二次开发，适配言语障碍康复等需频繁迭代范式的场景；局限在于深度学习集成支持不足，轻量级架构导致实时性能有限，社区规模小生态支撑薄弱。

3.7 BrainFlow

BrainFlow 是跨平台数据采集处理底层库^[7]，采用了“硬件抽象层 + 数据处理层”架构，硬件抽象层封装多厂商协议，能够适配 50 余种 EEG、ECOG 设备；数据处理层提供了滤波、特征提取等基础功能，同时支持实时和离线处理与边缘设备部署，提供了多语言 API 接口。BrainFlow 平台的主要优势在于很好的兼容性与跨平台能力，能够适配多设备开发场景；其核心局限在于仅具备底层数据服务能力，缺乏范式设计、高级解码、反馈控制等上层功能。而且需要与 PyNoetic 等工具协同使用才能构建完整 BCI 系统，自定义设备适配更是需要专业编程能力才能实现。

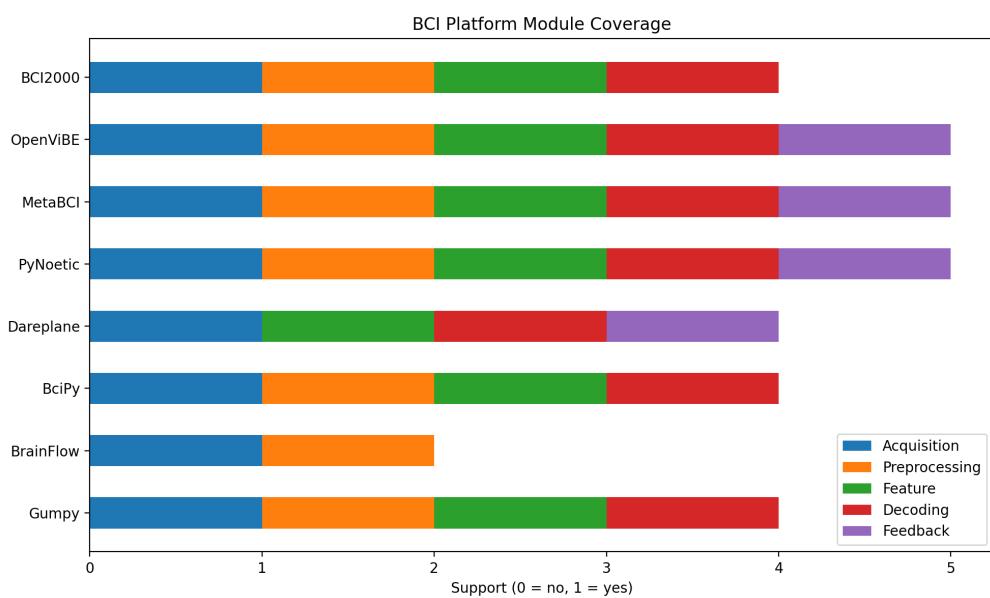
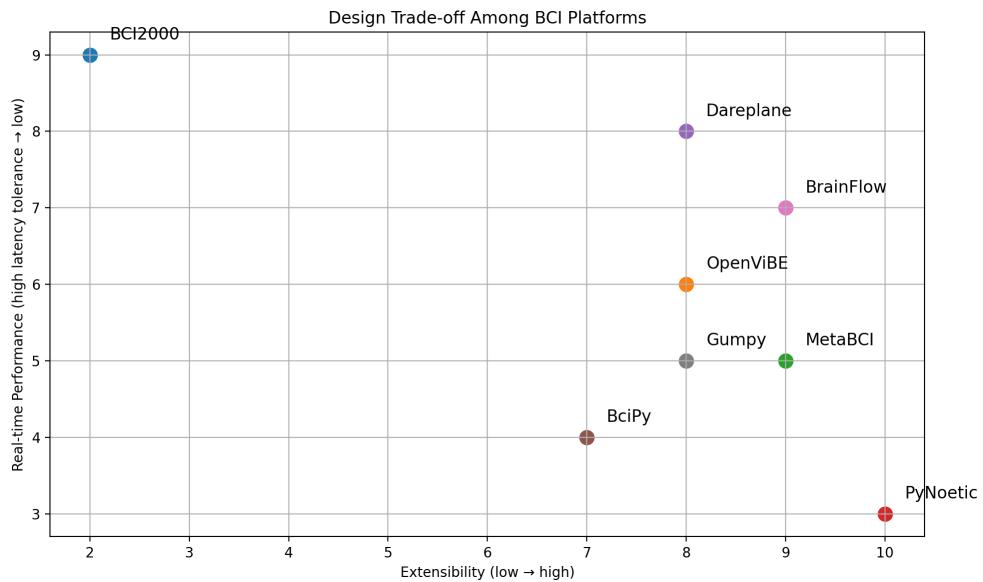
3.8 Gumpy

Gumpy 是慕尼黑工业大学研发的轻量级 Python 信号处理库^[8]，模块化分为数据管理、预处理、特征提取、模型训练四大模块，支持 BCI 数据集加载，集成 LDA、SVM 等传统算法及 TensorFlow/PyTorch 深度学习框架，API 设计简洁高效。优势为轻量灵活，可以快速完成从数据加载到模型训练的全流程验证，提升算法研发效率；局限为功能单一，仅覆盖离线分析环节，缺乏数据采集、范式设计等核心功能，无可视化界面且社区支撑不足，仅适用于算法研发辅助场景。

各平台差异性如表 1 所示。

表 1: BCI 常见软件平台架构对比

平台	语言	核心架构特点	模块化	实时性	适用对象
BCI2000	C++	多进程架构	高	很低延迟	研究人员/工程师
OpenViBE	C++	拖拽模块化设计	很高	实时支持	跨学科研究者
MetaBCI	Python	三层全流程结构	高	多线程处理	算法开发者
PyNoetic	Python	无代码流程图 +GUI	很高	实时和离线	教育团队
Dareplane	Python	闭环控制优化	高	闭环优化	临床研究者
BciPy	Python	模块化范式设计	中高	实时在线	实验室应用
BrainFlow	C++绑定	硬件抽象层	很高	实时数据流	开发者
Gumpy	Python	信号处理库	高	实时和离线	编程研究者



4 关键挑战与架构权衡

4.1 实时性与灵活性的权衡

实时性要求全流程延迟控制在数十至数百毫秒，需紧凑代码与专用通信接口；灵活性依赖模块化设计，易引入通信开销，二者存在天然矛盾。不同平台基于应用定位采用差异化策略：BCI2000 等聚焦实时性的平台，通过 C++ 底层优化、共享内存通信及进程优先级调度保障性能，以有限模块化实现基础扩展；OpenViBE 等聚焦灵活性的平台，采用可视化模块化架构，通过缓冲区优化控制延迟损耗，牺牲极致性能换取易用性；MetaBCI 采用分层架构，通过 Brainflow 底层优化保障实时性，Brainda 模块模块化设计保障灵活性，实现二者平衡。

4.2 跨设备兼容性与性能的平衡

跨设备兼容性需复杂硬件抽象层，易导致性能损耗，平衡核心在于抽象层设计质量。三类典型策略包括：“专用化”策略，如 BCI2000 早期，精简驱动适配主流设备保障性能，后期逐步扩展兼容；“标准化”策略，如 BrainFlow，构建统一抽象层接口封装设备差异，通过驱动优化降低损耗，实现“一次开发、多设备适配”；“模块化适配”策略，如 OpenViBE，独立硬件适配模块实现驱动与核心处理解耦，保障核心模块性能的同时便于扩展。

4.3 用户体验与功能深度的适配

用户群体技术背景差异大，需平衡易用性与功能深度。主流解决方案为“分层设计”与“双模式架构”：OpenViBE、PyNoetic 采用“无代码 + 可编程”双模式，可视化界面满足新手需求，插件接口支撑专家定制；MetaBCI、BciPy 采用分层设计，普通用户使用上层标准化模块，专业用户可修改底层代码扩展；BCI2000、Gumpy 采用“专业化”定位，牺牲部分易用性保障实时性或算法深度，适配专业场景。

4.4 资源受限场景的优化策略

嵌入式等资源受限场景需控制硬件开销，核心策略包括：轻量化设计，如 BrainFlow、Gumpy，去除冗余功能降低内存与计算占用；模型优化，如 MetaBCI，集成量化、剪枝技术降低推理需求；端云协同架构，边缘端部署预处理、推理等轻量任务，云端承担模型训练、离线分析等复杂任务，边缘 AI 技术推动该方向持续优化。

4.5 闭环控制系统的安全性保障

临床闭环场景需防范参数超限、系统故障等风险，安全机制完善度直接决定临床适配性。Dareplane 构建了最完善的多层保障体系：参数边界控制实时拦截超限值，心跳检测实现故障检测与应急切换，操作日志保障追溯，分级权限防止误操作。其他平台安全机制多适配基础研究，临床级安全设计不足，需强化故障恢复与边界控制功能。

4.6 开源生态的构建与可持续发展

开源生态决定平台生命力，成熟生态需四大支撑：完善文档体系，如 BCI2000，提供用户手册、开发指南与案例；活跃社区运营如 OpenViBE，通过学术会议、GitHub 讨论区促进交流；标准化插件机制，支持第三方贡献扩展功能；严格版本管理保障兼容性。MetaBCI 等新兴平台通过中文文档、线上社区构建生态，与国际成熟平台仍有差距，长期投入是关键。

5 未来趋势与创新方向

BCI 软件平台将向易用化、高效化、安全化、开放化方向演进，核心创新方向包括：

无代码/低代码深度普及：全流程可视化升级，集成智能流程推荐、拖拽式深度学习建模、范式模板市场，深化 OpenViBE 等现有可视化理念，进一步降低使用门槛。

边缘 AI 与端云协同普及：边缘端轻量化推理引擎优化实时性，端云加密同步保障数据安全，云端联邦学习支持多中心数据训练，BrainFlow 跨设备能力与

该架构深度融合。

智能闭环控制升级：集成强化学习实现自适应调控，多模态数据融合提升精度，构建临床级安全体系，Dareplane 架构将成为临床平台设计标杆。

自动化工作流全覆盖：实现数据预处理、特征选择、模型优化、实验设计的全流程自动化，虚拟仿真验证降低硬件依赖，MetaBCI 全流程理念进一步强化。

标准化与互操作性提升：采纳 FHIR 等统一数据格式，标准化 API 接口促进模块兼容，完善插件生态与跨平台迁移工具，BrainFlow 硬件抽象层理念扩展至模块接口。

隐私安全原生集成：同态加密、差分隐私技术保障数据安全，细粒度权限管理与动态知情同意机制适配临床需求，安全审计成为临床平台必备功能。

跨领域融合深化：集成 VR/AR、机器人等工具扩展应用场景，构建多学科范式库，打造产业级生态平台，集成教育培训模块推动技术普及。

总结

BCI 软件平台架构设计直接决定技术转化效率，八种代表性平台形成差异化发展路线：BCI2000 以 C++ 架构奠定实时稳定标杆，适配高可靠性场景；OpenViBE 可视化设计降低跨学科门槛，支撑原型开发与教学；MetaBCI 实现离线-在线全流程衔接，依托中文社区构建本土生态；PyNoetic 双模式架构适配多元用户，支撑教学与快速开发；Dareplane 聚焦临床闭环安全，成为神经调控场景标杆；BciPy 专精范式快速构建，适配康复等迭代需求；BrainFlow 以底层兼容为核心，支撑多设备开发；Gumpy 轻量灵活，加速算法研发验证。

平台研发需精准权衡实时性与灵活性、兼容性与性能等核心矛盾。未来发展将聚焦易用化（无代码）、高效化（端云协同）、安全化（临床级保障）、开放化（标准化生态）四大方向。加强技术融合（如 AI 与边缘计算）、推动接口与数据格式标准化、构建开放协同生态，将加速 BCI 技术从基础研究向临床与产业化转化，赋能医疗康复与智能交互领域创新发展。

参考文献

- [1] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043, 2004.
- [2] Yann Renard, Fabien Lotte, Guillaume Gibert, Marco Congedo, Emmanuel Maby, Vincent Delannoy, Olivier Bertrand, and Anatole Lécuyer. Openvibe: an open-source software platform to design, test, and use brain–computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments*, 19(1):35–53, 2010.
- [3] Jin Mei, Rui Luo, Minpeng Xu, Wei Zhao, Sheng Wen, Kai Wang, and Dong Ming. Metabci: An open-source platform for brain–computer interfaces. *Computers in Biology and Medicine*, 168:107806, 2023.
- [4] A Smith and B Johnson. Pynoetic: A modular python framework for no-code bci. *arXiv preprint arXiv:2509.00670*, 2025.
- [5] D Denissenko, J Farquhar, and O Jensen. Dareplane: a modular open-source software platform for bci research. *Journal of Neural Engineering*, 20(3):036011, 2023.
- [6] Tab Memmott, Aziz Koçanaoğulları, Matthew Lawhead, Daniel Klee, Shiran Dudy, Melanie Fried-Oken, and Barry Oken. Bcipy: brain–computer interface software in python. *Brain-Computer Interfaces*, 8(4):137–153, 2021. Publisher Copyright: © 2021 Informa UK Limited, trading as Taylor & Francis Group.
- [7] BrainFlow Development Team. Brainflow: Cross-platform library for biosignal acquisition and processing. <https://brainflow.org/>, 2025. Accessed: 2025-02-24.
- [8] Zied Tayeb, Nicolai Waniek, Juri Fedjaev, Nejla Ghaboosi, Leonard Rychly, Christian Widderich, Christoph Richter, Jonas Braun, Matteo Saveriano, Gordon Cheng, and Jörg Conradt. Gumpy: a python toolbox suitable for hybrid brain–computer interfaces. *Journal of Neural Engineering*, 15(6):065003, 2018.