BRAINBENCH: BENCHMARKING LARGE LANGUAGE MODELS FOR COMPREHENSIVE EEG UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly being explored for electroencephalography (EEG) analysis, yet their ability to support comprehensive EEG understanding remains poorly understood. Existing evaluations largely focus on isolated decoding tasks or demonstrations tailored to specific systems, capturing only limited aspects of EEG competence. In practice, EEG analysis requires a system to interpret analysis instructions, operate on physiological signals, derive quantitative evidence, and translate the results into scientifically grounded conclusions. To systematically evaluate this broader capability, we introduce BrainBench, a unified benchmark for comprehensive, instruction-conditioned EEG understanding. It comprises four subsets—Foundational Analysis, Sleep Assessment, Neurocognitive Assessment, and Physiological Integration—covering 17 datasets, 172 tasks, and over 4K real-data instances. Given an instruction and EEG recordings with optional physiological signals, a system must perform the requested analysis and produce scientifically grounded, verifiable outputs. We evaluate 13 representative LLMs across more than 100K executions under two paradigms: autonomous code execution and structured agentic analysis. Results vary substantially across models, subsets, difficulty levels, and execution paradigms, demonstrating that EEG competence depends not only on the underlying model, but also on how it is operationalized. BrainBench advances the exploration of artificial intelligence systems that reason over neural signals and support scalable, reproducible discovery in neuroscience.

1 INTRODUCTION

Electroencephalography (EEG) provides a non-invasive window into brain activity with millisecond-scale temporal resolution. Its accessibility and sensitivity to rapid neural dynamics have established it as a fundamental tool in neuroscience, sleep research, neurological assessment, and brain–computer interfaces (Aeschbach and Borbely, 1993; Schalk et al., 2004; Jeong, 2004; da Silva, 2013). Yet, computational EEG research has predominantly treated signal analysis as a decoding problem, in which a model maps a recording to a predefined output, such as a sleep stage, cognitive state, or clinical label (Wang et al., 2025a; Zhou et al., 2025; 2026a; El Ouahidi et al., 2026). While this paradigm has driven substantial progress, it captures only a narrow component of real-world EEG analysis. In practice, EEG analysis requires a system to understand the analytical objective, reason coherently over real recordings, and arrive at a scientifically grounded conclusion (Keil et al., 2014; Kane et al., 2017; Pernet et al., 2020). We refer to this broader capability as **comprehensive EEG understanding**. The central question therefore shifts from whether a model can predict a predefined target to whether it can carry an EEG analysis coherently from instruction to conclusion.

Large language models (LLMs) and agents offer a natural foundation for comprehensive EEG understanding (Brown et al., 2020; Yao et al., 2022). By combining language understanding, reasoning, and code generation, they can translate natural-language instructions into executable analyses, potentially transforming EEG analysis from a collection of specialized pipelines into an interactive, general-purpose process. Recent LLM-powered systems in brain science and neurotechnology have demonstrated their potential to support increasingly complex scientific workflows (Zhao et al., 2026; Abdou et al., 2026; Kosmyna and Hauptmann, 2026; Chen et al., 2026a; Xu et al., 2026a; Wang et al., 2026a; Zhou et al., 2026b; Li et al., 2026a; Pradeepkumar et al., 2026; Gu et al.). For example, CELM and CerebraGloss explore language-based clinical EEG interpretation and report generation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

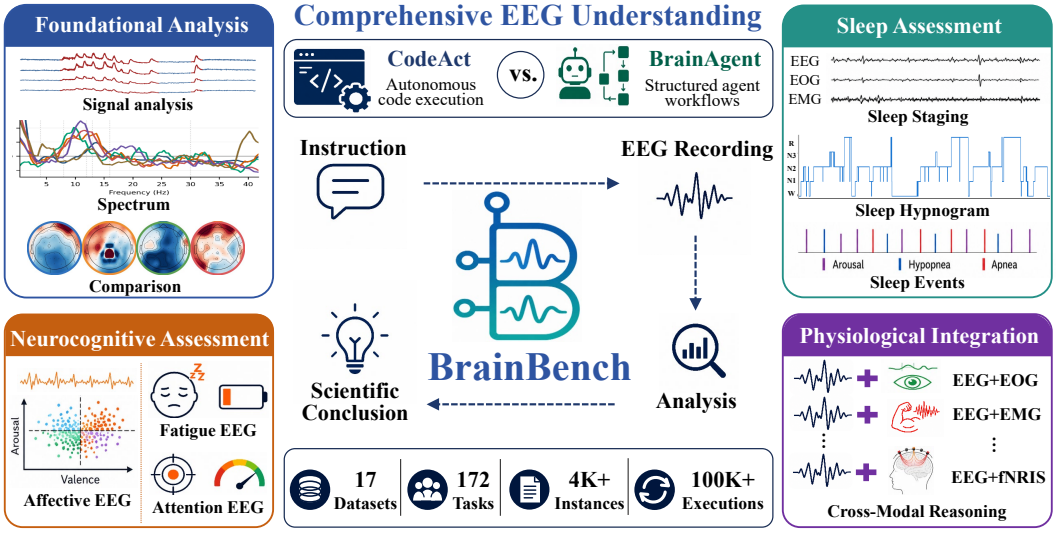


Figure 1: **Overview of BrainBench**. We present a comprehensive benchmark for instruction-conditioned EEG understanding, spanning 17 datasets, 172 tasks, and over 4K real-data instances across four complementary subsets. More than 100K model executions are evaluated under two complementary paradigms: autonomous code execution and structured agentic analysis.

(Pradeepkumar et al., 2026; Gu et al.), while BrainAgent investigates multi-agent orchestration for automating end-to-end EEG analysis (Zhou et al., 2026b). However, these systems are still evaluated primarily within their respective task- or system-specific settings. As a result, a comprehensive and systematic assessment of the correctness and scientific validity of LLM-generated EEG analyses across heterogeneous tasks and execution paradigms remains lacking.

*Can LLMs move beyond isolated decoding tasks to achieve **comprehensive EEG understanding** across real-world analysis workflows?*

Despite this promise, existing evaluation protocols are not designed to quantify comprehensive EEG understanding. Existing EEG benchmarks remain largely centered on fixed decoding objectives (Wu et al., 2025; Xiong et al., 2025; Shen et al., 2026; Lu et al., 2026; Kontras et al., 2026), rather than assessing whether a system can interpret diverse analytical instructions and carry them through to scientifically grounded conclusions. Moving beyond this decoding-centric paradigm, HeaRTS (Li et al., 2026b) takes an important step by requiring LLMs to analyze real-world health time-series data through executable code. Its primary emphasis, however, is breadth across physiological modalities rather than depth within EEG analysis. The distinctive signal characteristics, analytical conventions, and domain-specific interpretations of EEG give rise to heterogeneous workflows that demand fine-grained evaluation of both analytical procedures and resulting conclusions. **Consequently, a unified benchmark for comprehensive EEG understanding remains absent.**

Achieving comprehensive EEG understanding requires a system to interpret diverse analysis instructions, operate on real physiological recordings, perform appropriate signal processing and quantitative analyses, integrate evidence across analytical steps, and derive scientifically grounded conclusions.

To systematically evaluate these capabilities, we introduce **BrainBench**, a benchmark for comprehensive, instruction-conditioned EEG understanding. It comprises four complementary subsets—Foundational Analysis, Sleep Assessment, Neurocognitive Assessment, and Physiological Signal Integration—covering **172 tasks** and **4K instances** drawn from **17 datasets** and spanning diverse analytical objectives and levels of complexity. Together, these subsets cover foundational signal analysis, domain-specific reasoning, and multimodal interpretation, enabling EEG understanding to be evaluated in both breadth and depth. Rather than assessing performance on a fixed predictive target, BrainBench evaluates whether a model can transform a natural-language instruction and real EEG recordings into a scientifically grounded conclusion. We further evaluate the same LLMs under two complementary execution paradigms: autonomous code execution and structured agentic analysis. The former allows models to independently plan and execute analyses through self-generated

code, whereas the latter constrains execution through predefined workflows, controlled tools, and traceable intermediate actions. We instantiate these paradigms with CodeAct (Wang et al., 2024) and BrainAgent (Zhou et al., 2026b), respectively. By holding the instructions, data, and evaluation criteria constant, BrainBench enables a controlled comparison of two distinct execution paradigms and examines the trade-off between the flexibility of autonomous code generation and the reliability, auditability, and operational safety of structured agentic analysis.

We evaluate **13 representative LLMs** across all four subsets and under both execution paradigms, revealing key gaps in current systems and important directions for advancing comprehensive EEG understanding in the future. Beyond the current release, BrainBench is designed for extensibility, allowing new EEG domains, datasets, task formulations, evaluation dimensions, and models to be incorporated as the field evolves. Our contributions are summarized as follows:

- We formulate comprehensive, instruction-conditioned EEG understanding as a unified evaluation target and introduce BrainBench, comprising four complementary subsets, **172 tasks**, and **4K instances** across **17 diverse datasets**, analytical objectives, and workflows.
- We establish a dual-paradigm protocol that compares autonomous code execution with a structured EEG agent under shared instructions, data, and ground truth, enabling controlled analysis of how the execution paradigm shapes measured EEG competence.
- We systematically evaluate **13 representative LLMs** across tasks and execution paradigms, providing a multidimensional characterization of their capabilities, revealing key limitations, and charting a path toward artificial intelligence systems capable of reasoning over neural signals and enabling scalable, reproducible discovery in neuroscience.

2 Related Work

2.1 EEG Models and Decoding Benchmarks

EEG modeling has evolved from handcrafted feature pipelines and task-specific neural networks to foundation models trained on large-scale recordings to learn transferable representations (Wang et al., 2025a; Zhou et al., 2026a; El Ouahidi et al., 2026; Lawhern et al., 2016; Jiang et al., 2024a; Xiao et al., 2026; Wang et al., 2025b; 2026b). In parallel, existing benchmarks have standardized evaluation across downstream decoding tasks, typically using linear probing or fine-tuning to assess representation quality and cross-subject or cross-dataset generalization (Wu et al., 2025; Xiong et al., 2025; Shen et al., 2026; Lu et al., 2026; Kontras et al., 2026; Wang et al., 2026c; Kastrati et al., 2025; Banville et al., 2026). Clinically grounded models have further extended this paradigm to full-session recordings and multimodal clinical context; for example, CLEF (Cao et al., 2026) aligns session-level EEG representations with neurologist reports and electronic health records, yet is still evaluated mainly through patient-level classification tasks. Thus, despite increasing scale and clinical relevance, existing evaluations continue to define EEG competence primarily through performance on predefined predictive targets. They do not assess whether a general-purpose model can interpret an open-ended analytical instruction, execute the required analysis on real recordings, and derive a scientifically grounded conclusion. This leaves open the need for evaluating comprehensive, instruction-conditioned EEG understanding beyond fixed-label decoding.

2.2 Large Language Models for EEG Analysis

The reasoning (Yao et al., 2022), planning (Shinn et al., 2023), and tool-use (Schick et al., 2023) capabilities of large language models have enabled a new class of systems that interact with EEG data through natural-language instructions. These systems combine language understanding with code generation, domain knowledge, and specialized analytical tools to support tasks ranging from data inspection and preprocessing to event detection and report generation (Zhao et al., 2026; Abdou et al., 2026; Kosmyna and Hauptmann, 2026; Chen et al., 2026a; Xu et al., 2026a; Wang et al., 2026a; Zhou et al., 2026b; Li et al., 2026a; Baradari et al., 2025; Hong et al., 2024). For example, EEGAgent (Zhao et al., 2026) coordinates specialized tools for automated EEG analysis and reporting, while SleepLM (Xu et al., 2026a) connect physiological recordings with natural-language sleep assessment. More recent agentic frameworks extend this paradigm toward longer and more heterogeneous workflows: NeuroWeaver (Wang et al., 2026a) autonomously explores EEG analysis pipelines. BrainAgent (Zhou et al., 2026b) decomposes user intent among specialized agents for structured brain-signal analysis,

and BrainPilot (Li et al., 2026a) explores autonomous brain-science discovery through a multi-agent framework. Collectively, these studies demonstrate the feasibility of transforming high-level user requests into executable EEG analyses. Their rapid development, however, raises a distinct evaluation question: **how reliably do the resulting analyses reflect both the supplied recordings and the scientific intent expressed in the instruction?**

2.3 Evaluating Large Language Models for EEG Understanding

Existing evaluations differ substantially in scope and target. EEGAgent (Zhao et al., 2026) combines task-level metrics with qualitative demonstrations of EEG interpretation and reporting. BrainAgent (Zhou et al., 2026b) evaluates 60 tasks across three difficulty levels, focusing on task completion, routing reliability, and tool-use efficiency. BrainPilot (Li et al., 2026a) broadens the scope to brain-science research, although the initial BrainPilotBench-v0 contains four tasks covering calcium imaging, fMRI, motor-imagery EEG, and sleep EEG. HeaRTS (Li et al., 2026b) provides the closest general benchmark for executable reasoning over physiological time series, covering diverse health domains and signal modalities through autonomous code execution, but its primary objective is breadth across health time series rather than analytical depth within EEG. Taken together, existing evaluations remain fragmented across system-specific tasks, metrics, and protocols. A comprehensive and systematic assessment of the correctness and scientific validity of LLM-generated EEG analyses across heterogeneous tasks, datasets, and execution paradigms therefore remains lacking.

3 BrainBench

To address this gap, we introduce BrainBench, a unified benchmark for comprehensive, instruction-conditioned EEG understanding. Given a natural-language instruction and real EEG recordings, BrainBench evaluates whether an LLM can produce scientifically grounded analysis outputs beyond predefined predictive tasks. It comprises four complementary subsets spanning 172 tasks, 4K instances, and 17 datasets as shown in Figure 1. Under a unified assessment protocol, we evaluate **13** representative LLMs with CodeAct (Wang et al., 2024) and BrainAgent (Zhou et al., 2026b), enabling controlled comparison between autonomous code execution and structured EEG-agent workflows.

3.1 BrainBenchOrganization and Formulation

BrainBench formulates comprehensive EEG understanding as instruction-conditioned analytical execution. The benchmark follows a three-level hierarchy of `subsets`, `tasks`, and `instances`. Four complementary `subsets` define its evaluation scope: Foundational Analysis targets general EEG operations; Sleep Assessment and Neurocognitive Assessment cover domain-specific analytical workflows; and Physiological Integration evaluates reasoning across EEG and complementary physiological signals. Within each subset, a `task` defines a reusable analytical objective, the expected outputs, and the corresponding assessment requirements, and is instantiated by a collection of evaluation `instances`. These `instances` preserve the task-level requirements while varying the underlying recordings, subjects, analysis parameters, or task-specific instructions. This hierarchy enables BrainBench to assess whether analytical capabilities generalize across heterogeneous inputs rather than merely succeed on individual recordings or predefined queries. For an instance i, let $\mathcal{I}_i$ denote its natural-language instruction and $\mathcal{D}_i$ the supplied input files. Given $(\mathcal{I}_i, \mathcal{D}_i)$, a target system $\mathcal{M}$ produces a free-form analysis report $\mathcal{R}_i$ and, when required, a set of artifacts $\mathcal{A}_i$. The unified evaluation function $\mathcal{E}$ then assigns the instance score s_i:

$$(\mathcal{R}_i, \mathcal{A}_i) = \mathcal{M}(\mathcal{I}_i, \mathcal{D}i), \qquad s_i = \mathcal{E}(\mathcal{R}_i, \mathcal{A}i). \tag{1}$$

The evaluator applies the assessment criteria defined by the corresponding task while retaining a common interface across the benchmark. This formulation accommodates heterogeneous EEG workflows and output formats, while ensuring that different execution paradigms are evaluated on the same tasks and instances under consistent requirements.

3.2 BrainBench Construction and Evaluation

3.2.1 Task Definition

In BrainBench, each `task` defines a recording-independent analytical objective and the EEG understanding capability it is intended to assess. It also specifies the required inputs, expected outputs, and applicable validation criteria. These requirements remain fixed across all corresponding `instances`,

allowing the same capability to be evaluated across different subjects, datasets, and recordings. Tasks are stratified into three difficulty levels according to their intrinsic analytical demands. **Easy** tasks involve short, explicitly specified analyses with conclusions derived directly from limited evidence. **Medium** tasks require multiple dependent steps and integration of intermediate results. **Hard** tasks involve extended workflows that synthesize evidence across channels, time scales or physiological modalities. Difficulty is assigned at the task level and inherited by all associated instances.

3.2.2 Instance Construction

A `task` is converted into executable evaluation examples by binding its specification to a concrete data context. Each resulting `instance` identifies the dataset and recording to be analyzed, together with the applicable time window, signal selection, analysis parameters, and other task-specific conditions. For each such binding, BrainBench generates a complete natural-language instruction that specifies the available input files, analysis scope, required outputs, and reporting constraints. This process preserves a consistent analytical objective and evaluation target across instances while adapting the instruction to the data available in each evaluation context. Shown in Figure 2, each instance is further paired with an evaluator-side reference package. Given the bound input files and analysis parameters, a deterministic analysis script executes the prescribed reference workflow and computes the expected values. The resulting outputs are combined with the validation configuration, including the parser prompt and metric definitions, to form a complete instance-level specification for evaluation. Reusing a fixed script and parameterization across instances ensures that the ground truth remains reproducible only with the bound data.

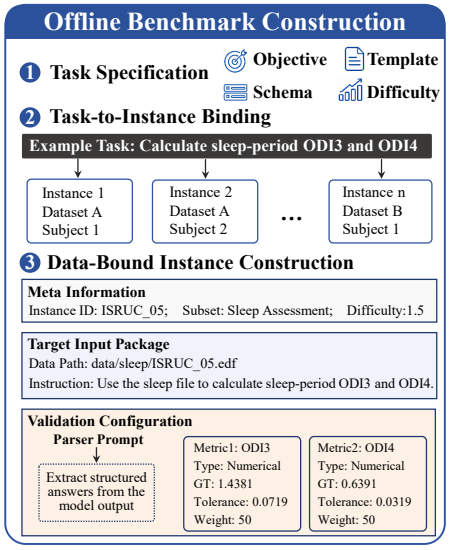


Figure 2: Construction of a reusable `task` and its data-bound `instances` across heterogeneous recordings.

3.2.3 Multi-Unit Validation

BrainBench supports free-form analytical reports to accommodate the diverse wording and presentation required by heterogeneous EEG tasks. To enable standardized evaluation, a Parser Agent extracts the required content from each report into a structured representation according to an instance-specific *Validation Configuration*, which defines the target fields, extraction prompt, validation units, reference targets, and metric weights. It serves solely as an extractor: it returns null for missing information and neither evaluates scientific correctness nor infers unreported results. This design preserves flexible natural-language reporting without conflating EEG understanding with rigid format compliance.

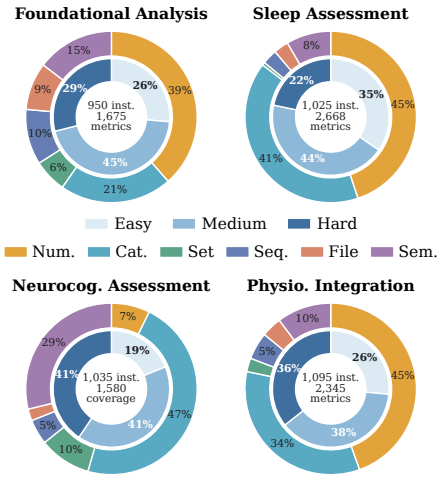


Figure 3: Distributions of task difficulty and validation units across subsets.

The extracted fields, original report, and generated artifacts are evaluated through six complementary validation units. **Numerical validation** compares reported scalar values with the computed ground truth under task-specific tolerances. **Categorical validation** assesses discrete labels or choices after canonicalization. **Set validation** evaluates unordered collections using exact matching or element-level partial credit. **Sequence validation** extends this assessment to ordered outputs by considering both element coverage and positional consistency. **Semantic validation** uses a task-specific Semantic Judge to assess non-scalar conclusions for correctness, relevance, instruction consistency, and unsupported or hallucinatory claims. **Artifact validation** verifies whether requested

Table 1: **BrainBench leaderboard.** Performance of evaluated models under BrainAgent and CodeAct across the four benchmark subsets. The best available result in each populated column is shown in **bold**, and the second-best results are underlined.

Model	Foundational Analysis		Sleep Assessment		Neurocog. Assessment		Physiol. Integration		Overall Score	
	BA	CA	BA	CA	BA	CA	BA	CA	BA	CA
Qwen3.5-35B	59.98	40.80	59.35	50.18	51.50	51.04	39.30	36.41	52.53	44.61
Qwen3.5-122B	63.12	48.85	65.98	57.18	58.51	56.16	49.38	47.09	59.25	52.32
Qwen3.7 Plus	73.08	68.33	68.08	62.06	69.86	64.17	57.98	51.66	67.25	61.55
Qwen3.7 Max	72.21	69.54	70.54	67.80	69.94	65.13	66.40	60.40	69.77	65.72
GLM 5	68.92	70.77	66.17	64.65	63.56	65.12	55.04	56.76	63.42	64.33
DeepSeek V4 Flash	67.38	68.16	67.97	61.01	67.21	60.38	49.98	54.52	63.13	61.02
Kimi K2.5	72.36	65.85	68.09	62.28	65.56	55.33	54.32	48.77	65.08	58.06
MiniMax M2.5	68.51	65.23	66.31	63.43	63.81	57.99	41.87	48.75	60.13	58.85
GPT 5.6 Luna	65.20	58.58	62.80	41.63	64.89	48.58	61.11	45.78	63.50	48.64
GPT 5.6 Terra	69.48	67.19	66.06	62.35	72.16	61.65	63.04	56.70	67.69	61.97
GPT 5.6 Sol	77.86	75.52	70.56	69.24	72.61	63.36	65.32	71.04	71.59	69.79
Gemini 3.6 Flash	77.38	54.85	**72.95**	59.51	**74.71**	61.04	**69.24**	53.27	**73.57**	57.17
Claude Opus 5	**81.47**	**84.97**	66.41	**73.53**	71.32	**70.03**	68.60	**76.49**	71.95	**76.25**

files are successfully generated and valid; structured signal files are checked programmatically, while visual outputs can additionally be assessed by a VLM Judge. A single instance may combine multiple validation units to cover complementary aspects of the requested output.

For instance i, each of its validation metrics produces a normalized score $v_{im} \in [0, 1]$ with a non-negative weight w_{im}. For subset aggregation, $d_i \in \{1.0, 1.5, 2.0\}$ denotes the task-level difficulty coefficient for Easy, Medium, and Hard instances. The instance and subset scores are computed as

$$s_i = 100 \times \frac{\sum_m w_{im} v_{im}}{\sum_m w_{im}}, \qquad S_{\mathcal{S}} = \frac{\sum_{i \in \mathcal{S}} d_i s_i}{\sum_{i \in \mathcal{S}} d_i}. \tag{2}$$

Both scores lie on a 0–100 scale, with the subset score assigning greater weight to more difficult tasks. Detailed matching rules are provided in Appendix C.

3.2.4 Black-Box Evaluation

BrainBench adopts a black-box evaluation protocol: each target system receives only the natural-language instruction and associated input files, while evaluator-side references and validation configurations remain hidden. Each LLM is evaluated through the same input–output interface under two complementary execution paradigms: **CodeAct** (Wang et al., 2024), which performs autonomous analysis through interactive Python coding, and **BrainAgent** (Zhou et al., 2026b), which executes structured agentic workflows with reusable EEG analysis tools. Holding the instructions, input data, and validation protocol fixed enables a controlled comparison between flexible coding and structured agentic analysis. Further protocol details are provided in Appendix D.

3.3 BrainBench Composition

BrainBench comprises four complementary subsets: **Foundational Analysis** for general-purpose EEG analysis, **Sleep Assessment** for sleep-related physiological understanding, **Neurocognitive Assessment** for affective and cognitive-state understanding, and **Physiological Integration** for multimodal neurophysiological analysis. Dataset and task details are in Appendix A.

4 Experiments and Analysis

We evaluate 13 API-accessible LLMs spanning eight model families and multiple capability tiers. To standardize test-time computation, all optional reasoning modes are disabled. Each model is evaluated under BrainAgent and CodeAct with matched API and decoding settings in isolated containers. Runs affected by verified provider or transport errors are retried according to Appendix D.

4.1 CAN LLMS ACHIEVE COMPREHENSIVE EEG UNDERSTANDING?

Table 1 reports the complete evaluation across all four BrainBench subsets. Overall, current LLMs demonstrate meaningful capability for comprehensive EEG understanding, but performance varies substantially across analytical domains and execution paradigms. Structured agentic workflows generally provide more reliable support for complex EEG analysis, while flexible coding remains competitive in settings where stronger models can better exploit autonomous reasoning and coding capabilities. These results suggest that EEG understanding is jointly shaped by the underlying model capability, analytical demands, and execution paradigm.

Finding 1: Current LLMs exhibit meaningful comprehensive EEG understanding: structured agentic workflows generally strengthen performance, while effectiveness remains shaped by model capability, analytical demands, and execution paradigm.

4.2 HOW DOES TASK DIFFICULTY SHAPE EEG UNDERSTANDING?

Table 2 shows that performance declines systematically with task difficulty across both execution paradigms, revealing a clear capability boundary for current LLM-based EEG understanding. Structured agentic workflows generally provide greater support as analytical demands increase, although their relative advantage varies across domains and is not consistently preserved on the hardest tasks. We quantify this interaction using the **Difficulty Advantage Index** (DAI), which measures whether the performance advantage of BrainAgent over CodeAct expands or contracts as task difficulty increases. The resulting patterns suggest that structured workflows can improve the organization and reliability of increasingly complex analyses, but their benefits become more task-dependent near the upper end of current model capability. Further details are provided in Appendix E.2.

Table 2: **Difficulty-stratified performance under BrainAgent (BA) and CodeAct (CA).** $\Delta = \mathrm{BA} - \mathrm{CA}$; blue and orange shading denotes positive and negative Δ, respectively.

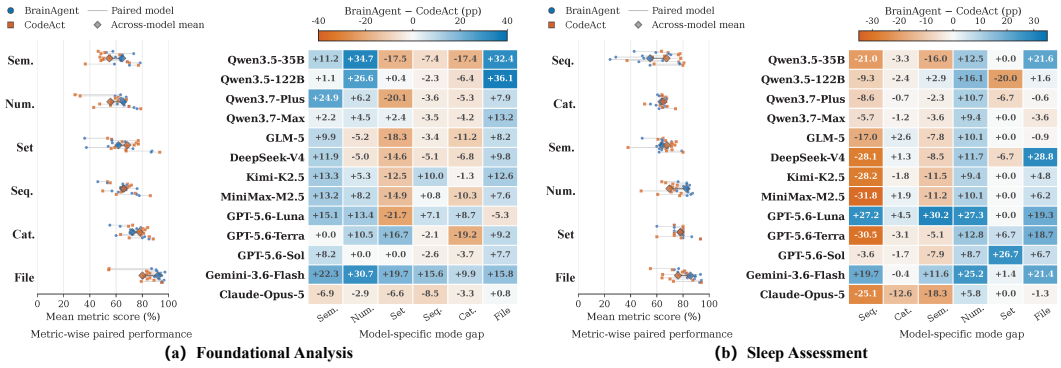
Model	Foundational Analysis							Sleep Assessment						
	Easy		Medium		Hard		DAI	Easy		Medium		Hard		DAI
	BA	CA (Δ)	BA	CA (Δ)	BA	CA (Δ)		BA	CA (Δ)	BA	CA (Δ)	BA	CA (Δ)	
Qwen3.5-35B	**82.8**	43.7 (+39.1)	**59.0**	39.7 (+19.3)	**49.3**	42.9 (+6.3)	-16.41	**85.9**	79.3 (+6.6)	**59.2**	44.9 (+14.4)	**32.6**	26.3 (+6.3)	-0.14
Qwen3.5-122B	**85.0**	52.7 (+32.4)	**65.6**	46.3 (+19.2)	47.8	**49.2** (−1.5)	-16.92	**87.3**	87.1 (+0.2)	**72.8**	55.9 (+17.0)	**33.8**	29.3 (+4.5)	2.14
Qwen3.7 Plus	**83.8**	63.9 (+19.9)	69.8	**70.3** (−0.6)	**69.6**	67.3 (+2.3)	-8.81	86.9	**90.4** (−3.5)	**76.0**	62.8 (+13.1)	**36.1**	28.5 (+7.6)	5.55
Qwen3.7 Max	**78.4**	67.6 (+10.8)	69.4	**70.0** (−0.5)	**68.3**	68.2 (+0.1)	-5.35	89.4	**90.9** (−1.5)	**78.8**	68.4 (+10.4)	**37.7**	36.3 (+1.4)	1.44
GLM 5	**83.3**	67.5 (+15.8)	66.5	**74.9** (−8.4)	62.9	**66.5** (−3.6)	-9.70	**87.0**	86.7 (+0.3)	**70.9**	63.7 (+7.2)	**40.4**	38.8 (+1.6)	0.67
DeepSeek V4 Flash	**76.8**	67.2 (+9.6)	66.7	**67.3** (−0.6)	64.1	**66.8** (−2.7)	-6.15	**93.0**	84.2 (+8.8)	**71.9**	61.1 (+10.8)	**36.6**	31.6 (+5.0)	-1.93
Kimi K2.5	**83.0**	56.1 (+27.0)	66.8	**66.9** (−0.1)	**71.1**	66.2 (+4.9)	-11.02	**92.1**	89.7 (+2.4)	**74.0**	63.7 (+10.4)	**36.4**	29.3 (+7.1)	2.35
MiniMax M2.5	**80.8**	65.1 (+15.7)	63.4	**64.8** (−1.5)	**66.2**	64.3 (+1.9)	-6.90	**92.8**	88.5 (+4.3)	**69.2**	61.6 (+7.6)	**35.8**	34.3 (+1.5)	-1.40
GPT 5.6 Luna	**82.0**	72.6 (+9.4)	**64.5**	60.7 (+3.8)	**56.3**	47.6 (+8.7)	-0.37	**90.6**	70.1 (+20.6)	**68.8**	35.6 (+33.2)	**33.0**	19.2 (+13.8)	-3.39
GPT 5.6 Terra	**86.3**	77.6 (+8.7)	**67.6**	63.5 (+4.0)	60.6	**67.0** (−6.4)	-7.55	**92.4**	81.6 (+10.9)	**70.5**	64.0 (+6.5)	**36.2**	32.6 (+3.7)	-3.59
GPT 5.6 Sol	83.8	**87.1** (−3.3)	**75.3**	72.3 (+3.0)	**75.0**	74.0 (+0.9)	2.12	**94.9**	90.9 (+4.0)	**79.1**	69.8 (+9.3)	36.9	**39.3** (−2.4)	-3.21
Gemini 3.6 Flash	**76.0**	49.3 (+26.7)	**74.9**	56.8 (+18.1)	**78.2**	52.4 (+25.8)	-0.47	**95.1**	81.6 (+13.5)	**82.2**	53.2 (+29.1)	**37.1**	33.5 (+3.5)	-4.98
Claude Opus 5	**91.3**	87.6 (+3.7)	86.6	**89.1** (−2.5)	71.3	**78.0** (−6.7)	-5.21	94.8	**96.4** (−1.6)	**72.4**	70.7 (+1.7)	33.1	**43.3** (−10.2)	-4.30
Mean	**82.6**	66.0 (+16.6)	**68.9**	64.8 (+4.1)	**64.7**	62.3 (+2.4)	-7.13	**91.0**	86.0 (+5.0)	**72.8**	59.6 (+13.2)	**35.8**	32.5 (+3.3)	-0.83

Model	Neurocognitive Assessment							Physiological Integration						
	Easy		Medium		Hard		DAI	Easy		Medium		Hard		DAI
	BA	CA (Δ)	BA	CA (Δ)	BA	CA (Δ)		BA	CA (Δ)	BA	CA (Δ)	BA	CA (Δ)	
Qwen3.5-35B	**78.4**	70.7 (+7.6)	**52.9**	48.9 (+4.0)	43.2	**46.7** (−3.5)	-5.59	59.6	**70.2** (−10.6)	**37.2**	33.9 (+3.4)	**32.3**	26.3 (+5.9)	8.27
Qwen3.5-122B	**81.1**	73.0 (+8.0)	**61.6**	57.1 (+4.5)	49.8	**51.1** (−1.2)	-4.63	71.1	**76.3** (−5.2)	45.2	**47.9** (−2.7)	**45.6**	34.7 (+10.9)	8.04
Qwen3.7 Plus	**84.0**	78.5 (+5.5)	**70.4**	64.8 (+5.6)	**65.6**	60.8 (+4.9)	-0.32	72.9	**84.9** (−12.1)	50.8	**53.1** (−2.3)	**57.0**	41.0 (+16.0)	14.05
Qwen3.7 Max	**83.6**	77.0 (+6.6)	**71.9**	65.9 (+6.0)	**64.4**	61.9 (+2.5)	-2.06	81.8	**86.6** (−4.8)	**66.8**	60.6 (+6.2)	**60.9**	51.1 (+9.8)	7.30
GLM 5	**82.4**	81.2 (+1.1)	**68.8**	64.1 (+4.6)	54.2	**62.5** (−8.3)	-4.73	72.5	**84.6** (−12.2)	56.6	**60.0** (−3.3)	**46.4**	45.8 (+0.6)	6.39
DeepSeek V4 Flash	**83.2**	76.5 (+6.7)	**71.0**	59.5 (+11.6)	**59.8**	56.8 (+3.0)	-1.87	73.2	**85.3** (−12.1)	48.9	**56.1** (−7.1)	40.4	**43.8** (−3.4)	4.36
Kimi K2.5	**80.5**	73.9 (+6.6)	**66.7**	56.1 (+10.6)	**60.6**	50.1 (+10.5)	1.94	79.0	**86.6** (−7.6)	**58.7**	52.2 (+6.6)	**42.9**	35.6 (+7.3)	7.46
MiniMax M2.5	**83.1**	74.9 (+8.2)	**64.7**	61.0 (+3.6)	**57.9**	51.6 (+6.3)	-0.94	73.7	**83.1** (−9.3)	40.4	**49.4** (−9.0)	33.6	**37.7** (−4.1)	2.64
GPT 5.6 Luna	**81.1**	65.5 (+15.7)	**65.1**	49.3 (+15.7)	**60.0**	41.9 (+18.1)	1.20	80.2	**83.4** (−3.1)	**60.5**	48.4 (+12.1)	**55.9**	31.2 (+24.7)	13.90
GPT 5.6 Terra	**78.6**	74.5 (+4.0)	**72.8**	58.9 (+14.0)	**69.9**	61.1 (+8.7)	2.34	73.2	**82.4** (−9.3)	**59.6**	54.7 (+4.9)	**62.0**	50.2 (+11.8)	10.52
GPT 5.6 Sol	**79.8**	79.3 (+0.5)	**74.7**	64.0 (+10.8)	**68.5**	59.2 (+9.3)	4.41	74.1	**84.1** (−10.0)	67.1	**70.5** (−3.5)	60.5	**66.6** (−6.1)	1.94
Gemini 3.6 Flash	**83.1**	78.9 (+4.2)	**75.4**	62.5 (+12.9)	**72.3**	56.3 (+16.0)	5.91	72.4	**79.3** (−6.9)	**66.8**	47.1 (+19.7)	**68.9**	46.2 (+22.7)	14.79
Claude Opus 5	**84.5**	79.6 (+5.0)	**73.9**	72.4 (+1.4)	**66.3**	66.1 (+0.3)	-2.35	77.8	**87.2** (−9.4)	70.9	**74.2** (−3.3)	62.9	**73.5** (−10.5)	-0.60
Mean	**81.8**	75.7 (+6.1)	**68.4**	60.3 (+8.1)	**61.0**	55.9 (+5.1)	-0.51	74.0	**82.6** (−8.7)	**56.1**	54.5 (+1.7)	**51.5**	44.9 (+6.6)	7.62

Finding 2: Increasing task difficulty exposes the capability boundary of EEG understanding, while the benefit of structured agentic execution varies across domains.

4.3 Validation-Unit-Specific Effects of Execution Paradigms

Figure 4 decomposes LLM-based EEG understanding across the six validation units for FA and SA subsets. Paired scores compare BrainAgent and CodeAct for the same model, while the mode-gap heatmaps show the direction and magnitude of their differences, thereby separating execution-paradigm effects from variations in underlying LLM capability. The most consistent cross-subset advantage of BrainAgent appears in numerical validation, where structured execution generally improves the accuracy of quantitative analysis relative to autonomous code generation. The remaining validation units exhibit more heterogeneous mode gaps, with categorical, set, sequence, semantic, and artifact performance varying across models and subsets without a uniform advantage for either paradigm. These results suggest that structured execution primarily strengthens specific components of EEG analysis rather than producing a uniform improvement across all output types. Further details and experimental results are provided in Appendix E.3.



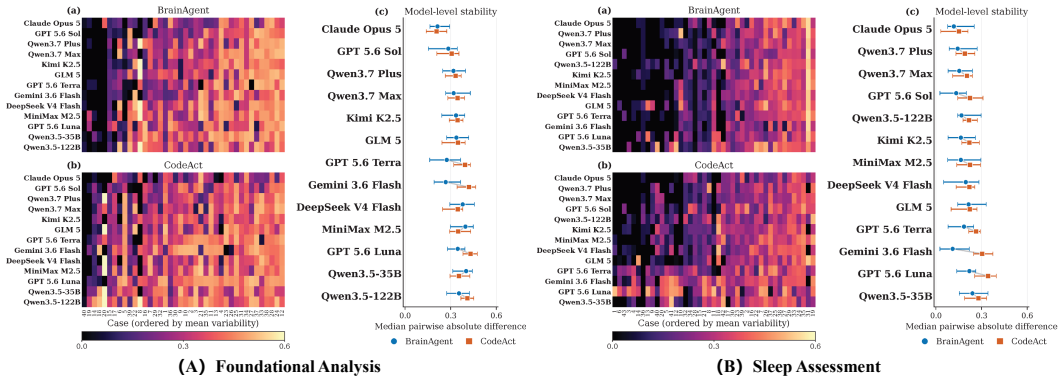
(a) Foundational Analysis **(b) Sleep Assessment**

Figure 4: **Validation-unit effects of execution paradigms.** Paired markers compare BA and CA, diamonds show across-model means, and heatmaps report percentage-point differences.

> ***Finding 3:*** Structured agentic workflows strengthen execution-grounded EEG understanding, with the most consistent gains in quantitative analysis.

4.4 Cross-Instance Stability within Reusable Tasks

We assess cross-instance consistency within reusable tasks by quantifying within-task variability as the mean pairwise absolute difference (MPAD) among normalized instance scores (the formal definition is detailed in Appendix E.4), with lower values indicating greater stability. Figure 5 presents task-level MPAD and its model-level aggregation for FA and SA subsets. Across both



(A) Foundational Analysis **(B) Sleep Assessment**

Figure 5: **Cross-instance stability within reusable tasks.** Heatmaps show within-task score variation, and model panels report median stability with 95% confidence intervals.

subsets, BrainAgent generally exhibits lower within-task dispersion than CodeAct, with the clearest and most consistent separation observed in SA. Although exceptions remain, lower overall MPAD suggests that capability-oriented tools and controlled workflows reduce sensitivity to recording and contextual variation. This stability complements mean performance: it does not necessarily imply greater correctness, but indicates that a measured capability transfers more consistently across instances sharing the same analytical objective.

Finding 4: Structured agentic workflows improve the cross-context reliability of EEG understanding, enabling consistent generalization across recordings with shared objectives.

4.5 Can LLM Reasoning Improve EEG Understanding?

To examine whether LLM reasoning can enhance EEG understanding, we re-evaluate GPT-5.6 Luna across four reasoning effort levels—Low, Medium, High and XHigh—in addition to the reasoning-disabled setting, under both BrainAgent and CodeAct. As shown in Figure 6, enabling reasoning consistently improves performance over the disabled setting, with the largest gain generally occurring at Low effort. Beyond this initial gain, the effect of additional reasoning becomes increasingly task- and execution-dependent. BrainAgent generally saturates after Low or Medium effort, whereas CodeAct continues to benefit from increased reasoning in several settings. Moreover, reasoning-enabled GPT-5.6 Luna matches or even surpasses GPT-5.6 Sol without reasoning in several settings, indicating that LLM reasoning can partially compensate for differences in underlying model capability, but not uniformly across tasks. Taken together, these results suggest that enabling reasoning matters more than aggressively scaling reasoning effort for EEG understanding.

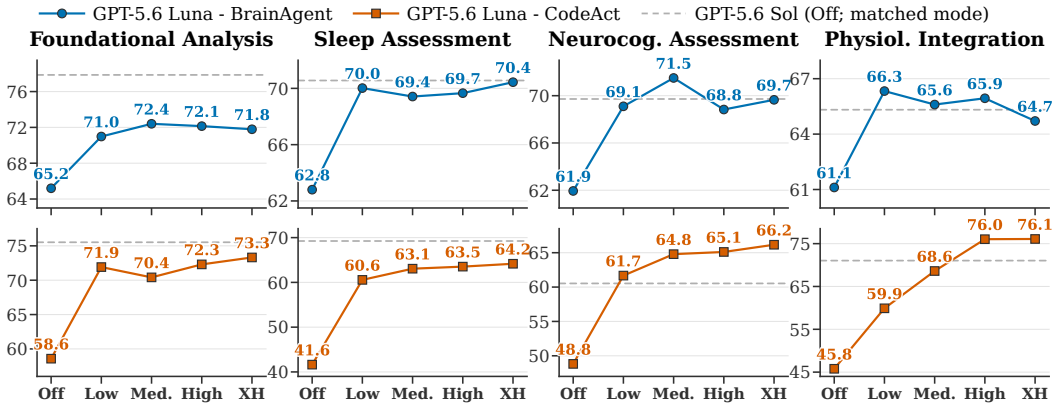


Figure 6: **Effect of reasoning effort on EEG understanding.** GPT-5.6 Luna is evaluated from Off to XHigh under BrainAgent and CodeAct; dashed lines show GPT-5.6 Sol without reasoning.

Finding 5: LLM reasoning is an effective but bounded amplifier of EEG understanding: enabling reasoning provides the dominant gain, with smaller, task-dependent returns from further scaling.

5 Conclusion

We introduced BrainBench to make comprehensive, instruction-conditioned EEG understanding measurable beyond conventional decoding benchmarks. Our evaluation shows that current LLMs possess meaningful but uneven EEG competence, with performance jointly shaped by model capability, analytical difficulty, and execution design. Structured agentic workflows generally strengthen execution reliability, whereas autonomous coding preserves flexibility and allows strong models to exploit their reasoning and programming capabilities; neither paradigm is uniformly superior. This complementarity suggests that future EEG systems should combine auditable domain workflows with adaptive reasoning rather than rely exclusively on either approach. By providing a common testbed for these capabilities, BrainBench supports the development of hybrid agents with broader analytical coverage, stronger scientific validation, and more reliable operation in research and clinical settings.

6 AI USE STATEMENT

In this work, generative AI tools were used for language polishing and assistance in drafting code for figure generation. Generative AI was not used to formulate the core research ideas, research questions or scientific conclusions. All AI-assisted text was reviewed and revised by the authors, and all AI-assisted plotting code was manually inspected, tested, and verified for correctness. The authors take full responsibility for the final content of this work, including all text, analyses, claims, figures, and conclusions.

REFERENCES

Daniel Aeschbach and Alexander A Borbely. All-night dynamics of the human sleep eeg. *Journal of sleep research*, 2(2):70–81, 1993.

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.

Jaeseung Jeong. Eeg dynamics in patients with alzheimer's disease. *Clinical neurophysiology*, 115 (7):1490–1505, 2004.

Fernando Lopes da Silva. Eeg and meg: relevance to neuroscience. *Neuron*, 80(5):1112–1128, 2013.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Haiteng Jiang, Shijian Li, Tao Li, and Gang Pan. Cbramod: A criss-cross brain foundation model for eeg decoding. In *International conference on learning representations*, volume 2025, pages 75310–75346, 2025a.

Yangxuan Zhou, Sha Zhao, Jiquan Wang, Haiteng Jiang, Shijian Li, Benyan Luo, Tao Li, and Gang Pan. Personalized sleep staging leveraging source-free unsupervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14529–14537, 2025.

Yuchen Zhou, Jiamin Wu, Zichen Ren, Zhouheng Yao, Weiheng Lu, Kunyu Peng, Qihao Zheng, Chunfeng Song, Wanli Ouyang, and Chao Gou. Csbrain: A cross-scale spatiotemporal brain foundation model for eeg decoding. *Advances in Neural Information Processing Systems*, 38: 87150–87195, 2026a.

Yassine El Ouahidi, Jonathan Lys, Philipp Thölke, Nicolas Farrugia, Bastien Pasdeloup, Vincent Gripon, Karim Jerbi, and Giulia Lioi. Reve: A foundation model for eeg-adapting to any setup with large-scale pretraining on 25,000 subjects. *Advances in Neural Information Processing Systems*, 38:22541–22577, 2026.

Andreas Keil, Stefan Debener, Gabriele Gratton, Markus Junghöfer, Emily S Kappenman, Steven J Luck, Phan Luu, Gregory A Miller, and Cindy M Yee. Committee report: publication guidelines and recommendations for studies using electroencephalography and magnetoencephalography. *Psychophysiology*, 51(1):1–21, 2014.

Nick Kane, Jayant Acharya, Sandor Beniczky, Luis Caboclo, Simon Finnigan, Peter W Kaplan, Hiroshi Shibasaki, Ronit Pressler, and Michel JAM Van Putten. A revised glossary of terms most commonly used by clinical electroencephalographers and updated proposal for the report format of the eeg findings. revision 2017. *Clinical neurophysiology practice*, 2:170–185, 2017.

Cyril Pernet, Marta I Garrido, Alexandre Gramfort, Natasha Maurits, Christoph M Michel, Elizabeth Pang, Riitta Salmelin, Jan Mathijs Schoffelen, Pedro A Valdes-Sosa, and Aina Puce. Issues and recommendations from the ohbm cobidas meeg committee for reproducible eeg and meg research. *Nature neuroscience*, 23(12):1473–1483, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Sha Zhao, Mingyi Peng, Haiteng Jiang, Tao Li, and Shijian Li. Eeg agent: A unified framework for automated eeg analysis using large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18063–18071, 2026.

Abdelrahman Abdou, Martin Ivanov, Sarmed Shaya, Alice Rueda, Fatemeh Gholamali Nezhad, Ilya Demchenko, Mohammad Amin Kamaleddin, Paul A Frewen, Benjamin T Dunkley, Brendan Brady, et al. Eeg-ai: An agentic system for ai-assisted semi-automated eeg preprocessing and artifact removal. *Journal of Neuroscience Methods*, 432:110759, 2026.

Nataliya Kosmyna and Eugene Hauptmann. Neuroskill (tm): Proactive real-time agentic system capable of modeling human state of mind. *arXiv preprint arXiv:2603.03212*, 2026.

Yankai Chen, Xinni Zhang, Yifei Zhang, Yangning Li, Henry Zou, Chunyu Miao, Weizhi Zhang, Steve Xue Liu, and Philip S Yu. Embracing trustworthy brain-agent collaboration as paradigm extension for intelligent assistive technologies. *Advances in Neural Information Processing Systems*, 38, 2026a.

Zongzhe Xu, Zitao Shuai, Eideen Mozaffari, Ravi S Aysola, Rajesh Kumar, and Yuzhe Yang. Sleeplm: Natural-language intelligence for human sleep. *arXiv preprint arXiv:2602.23605*, 2026a.

Guoan Wang, Shihao Yang, Jun-En Ding, and Feng Liu. Neuroweaver: An autonomous evolutionary agent for exploring the programmatic space of eeg analysis pipelines. *arXiv preprint arXiv:2602.13473*, 2026a.

Yangxuan Zhou, Sha Zhao, Jiquan Wang, Shijian Li, and Gang Pan. Brainagent: A large language model-driven multi-agent framework for autonomous brain signal understanding. *arXiv preprint arXiv:2606.25400*, 2026b.

Haoxuan Li, Tianci Gao, Jianhe Li, Yang Fan, Runze Shi, Weiran Wang, Tianxiang Zhao, Zezhao Wu, Xiaoyang Jiang, Qihui Zhang, et al. Brainpilot: Automating brain discovery with agentic research. *arXiv preprint arXiv:2607.15079*, 2026a.

Jathurshan Pradeepkumar, Zheng Chen, and Jimeng Sun. Neural signals generate clinical notes in the wild. *arXiv preprint arXiv:2601.22197*, 2026.

Wei Gu, Luo Tianming, Qiran Zhang, Mohan Ye, Xiao Shen, Wenxin Chen, Yunhuan Li, Yichen Zhang, Jing Hong, Bao-liang Lu, et al. Cerebragloss: Instruction-tuning a large vision-language model for fine-grained clinical eeg interpretation. In *The Fourteenth International Conference on Learning Representations*.

Jiamin Wu, Zichen Ren, Junyu Wang, Pengyu Zhu, Yonghao Song, Mianxin Liu, Qihao Zheng, Lei Bai, Wanli Ouyang, and Chunfeng Song. Adabrain-bench: Benchmarking brain foundation models for brain-computer interface applications. *arXiv preprint arXiv:2507.09882*, 2025.

Wei Xiong, Jiangtong Li, Jie Li, and Kun Zhu. Eeg-fm-bench: A comprehensive benchmark for the systematic evaluation of eeg foundation models. *arXiv preprint arXiv:2508.17742*, 2025.

Fanqi Shen, Enhong Yang, Jiahe Li, Junru Hong, Xiaoran Pan, Zhizhang Yuan, Meng Li, and Yang Yang. Brain4fms: A benchmark of foundation models for electrical brain signal. *arXiv preprint arXiv:2602.11558*, 2026.

Ziling Lu, Zongsheng Li, Xinke Shen, Kexin Lou, Yingyue Xin, Xiaoqi Chen, Shinan Wang, Xiang Chen, Jiahao Fan, Chenyu Huang, et al. Omnieeg-bench: A standardized evaluation benchmark for eeg foundation models. *arXiv preprint arXiv:2606.00815*, 2026.

Konstantinos Kontras, Trui Osselaer, Stylianos G Mouslech, Angeliki-Ilektra Karaiskou, Guido Gagliardi, Thomas Strypsteen, Mohammad Hossein Badiei, Anku Rani, Maarten Vanmarcke, Miguel Bhagubai, et al. Neuroatlas: Benchmarking foundation models for clinical eeg and brain-computer interfaces. *arXiv preprint arXiv:2605.14698*, 2026.

Sirui Li, Shuhan Xiao, Mihir Joshi, Ahmed Metwally, Daniel McDuff, Wei Wang, and Yuzhe Yang. Hearts: Benchmarking llm reasoning on health time series. *arXiv preprint arXiv:2603.06638*, 2026b.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*, 2024.

Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional network for eeg-based brain-computer interfaces. *arXiv preprint arXiv:1611.08024*, 2016.

Wei-Bang Jiang, Liming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous eeg data in bci. In *International Conference on Learning Representations*, volume 2024, pages 16405–16426, 2024a.

Qinfan Xiao, Ziyun Cui, Chi Zhang, Siqi Chen, Wen Wu, Andrew Thwaites, Alexandra Woolgar, Bowen Zhou, and Chao Zhang. Brainomni: A brain foundation model for unified eeg and meg signals. *Advances in Neural Information Processing Systems*, 38:41179–41212, 2026.

Jiquan Wang, Sha Zhao, Zhiling Luo, Yangxuan Zhou, Shijian Li, and Gang Pan. Eegmamba: An eeg foundation model with mamba. *Neural Networks*, page 107816, 2025b.

Jiquan Wang, Sha Zhao, Yangxuan Zhou, Yiming Kang, Shijian Li, and Gang Pan. Deeperbrain: A neuro-grounded eeg foundation model towards universal bci. *arXiv preprint arXiv:2601.06134*, 2026b.

Xianheng Wang, Yige Yang, and Damien Coyle. Eeg-fm-audit: A systematic evaluation and analysis pipeline for eeg foundation models. *arXiv preprint arXiv:2605.26910*, 2026c.

Ard Kastrati, Josua Bürki, Jonas Lauer, Cheng Xuan, Raffaele Iaquinto, and Roger Wattenhofer. Eeg-bench: A benchmark for eeg foundation models in clinical applications. *arXiv preprint arXiv:2512.08959*, 2025.

Hubert Banville, Stéphane d'Ascoli, Simon Dahan, Jérémy Rapin, Marlène Careil, Yohann Benchetrit, Jarod Lévy, Saarang Panchavati, Antoine Ratouchniak, Elisa Cascardi, et al. Neuralbench: A unifying framework to benchmark neuroai models. *arXiv preprint arXiv:2605.08495*, 2026.

Peng Cao, Ali Mirzazadeh, Jong Woo Lee, Aleksandar Videnovic, and Dina Katabi. Clef: Eeg foundation model for learning clinical semantics. *arXiv preprint arXiv:2605.10817*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

Dünya Baradari, Nataliya Kosmyna, Oscar Petrov, Rebecah Kaplun, and Pattie Maes. Neurochat: A neuroadaptive ai chatbot for customizing learning experiences. In *Proceedings of the 7th ACM Conference on Conversational User Interfaces*, pages 1–21, 2025.

Jiazhen Hong, Weinan Wang, and Laleh Najafizadeh. Chatbci: A p300 speller bci leveraging large language models for improved sentence composition in realistic scenarios. *arXiv preprint arXiv:2411.15395*, 2024.

Sirvan Khalighi, Teresa Sousa, José Moutinho Santos, and Urbano Nunes. Isruc-sleep: A comprehensive public dataset for sleep researchers. *Computer methods and programs in biomedicine*, 124: 180–192, 2016.

Ji-Hoon Jeong, Jeong-Hyun Cho, Young-Eun Lee, Seo-Hyun Lee, Gi-Hwan Shin, Young-Seok Kweon, José del R Millán, Klaus-Robert Müller, and Seong-Whan Lee. 2020 international brain–computer interface competition: A review. *Frontiers in human neuroscience*, 16:898300, 2022.

Wei Liu, Jie-Lin Qiu, Wei-Long Zheng, and Bao-Liang Lu. Comparing recognition performance and robustness of multimodal deep learning models for multimodal emotion recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 14(2):715–729, 2021.

Wajid Mumtaz, Likun Xia, Syed Saad Azhar Ali, Mohd Azhar Mohd Yasin, Muhammad Hussain, and Aamir Saeed Malik. Electroencephalogram (eeg)-based computer-aided technique to diagnose major depressive disorder (mdd). *Biomedical Signal Processing and Control*, 31:108–115, 2017.

Igor Zyma, Sergii Tukaev, Ivan Seleznov, Ken Kiyono, Anton Popov, Mariia Chernykh, and Oleksii Shpenkov. Electroencephalograms during mental arithmetic task performance. *Data*, 4(1):14, 2019.

Diego Alvarez-Estevez and Roselyne M Rijsman. Inter-database validation of a deep learning approach for automatic sleep scoring. *arXiv preprint arXiv:2009.10365*, 2020.

Christian O'reilly, Nadia Gosselin, Julie Carrier, and Tore Nielsen. Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research. *Journal of sleep research*, 23(6):628–635, 2014.

Mohammad M Ghassemi, Benjamin E Moody, Li-Wei H Lehman, Christopher Song, Qiao Li, Haoqi Sun, Roger G Mark, M Brandon Westover, and Gari D Clifford. You snooze, you win: the physionet/computing in cardiology challenge 2018. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pages 1–4. IEEE, 2018.

Stuart F Quan, Barbara V Howard, Conrad Iber, James P Kiley, F Javier Nieto, George T O'Connor, David M Rapoport, Susan Redline, John Robbins, Jonathan M Samet, et al. The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085, 1997.

Jingjing Chen, Xiaobin Wang, Chen Huang, Xin Hu, Xinke Shen, and Dan Zhang. A large finer-grained affective computing eeg dataset. *Scientific Data*, 10(1):740, 2023.

Xiaojun Ning, Jing Wang, Zhiyang Feng, Tianzuo Xin, Shuo Zhang, Shaoqi Zhang, Zheng Lian, Yi Ding, Youfang Lin, and Ziyu Jia. Refed: a subject real-time dynamic labeled eeg-fnirs synchronized recorded emotion dataset. *Advances in Neural Information Processing Systems*, 38, 2026.

Marcel F Hinss, Emilie S Jahanpour, Bertille Somon, Lou Pluchon, Frédéric Dehais, and Raphaëlle N Roy. Open multi-session and multi-task eeg cognitive dataset for passive brain-computer interface applications. *Scientific Data*, 10(1):85, 2023.

Jiayi Li, Cong Fu, Jinbu Tang, Ligang Zhou, Hao Chen, Wei Zhou, Chen Chen, and Jingchun Luo. Multimodal phenotyping dataset of driving fatigue. *Scientific Data*, 2026c.

Wei-Bang Jiang, Xuan-Hao Liu, Wei-Long Zheng, and Bao-Liang Lu. Seed-vii: A multimodal dataset of six basic emotions with continuous labels for emotion recognition. *IEEE Transactions on Affective Computing*, 16(2):969–985, 2024b.

Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, 3(1):18–31, 2011.

Jaeyoung Shin, Alexander Von Lühmann, Do-Won Kim, Jan Mehnert, Han-Jeong Hwang, and Klaus-Robert Müller. Simultaneous acquisition of eeg and nirs during cognitive tasks for an open access dataset. *Scientific data*, 5(1):180003, 2018.

Wei-Long Zheng and Bao-Liang Lu. A multimodal approach to estimating vigilance using eeg and forehead eog. *arXiv preprint arXiv:1611.08492*, 2016.

Qwen Team. Qwen3.5: Towards native multimodal agents. Technical report, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.7: The agent frontier. Technical report, May 2026b. URL https://qwen.ai/blog?id=qwen3.7.

Qwen Team. Qwen3.7-Plus: Multimodal agent intelligence. Technical report, May 2026c. URL https://qwen.ai/blog?id=qwen3.7-plus.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026b.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Aili Chen, Aonian Li, Baichuan Zhou, Bangwei Gong, Binyang Jiang, Boji Dan, Changqing Yu, Chao Wang, Cheng Ma, Cheng Zhong, et al. The minimax-m2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint arXiv:2605.26494*, 2026b.

OpenAI. GPT-5.6 system card. System card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf.

Google DeepMind. Gemini 3.6 Flash model card. Model card, July 2026. URL https://deepmind.google/models/model-cards/gemini-3-6-flash/.

Anthropic. Claude Opus 5 system card. System card, July 2026. URL https://www.anthropic.com/system-cards.

A Dataset Statistics

Table 3: Dataset composition and acquisition characteristics of BrainBench, including subset assignment, retained signal modalities and native sampling rates, EEG channel counts, and data accessibility. FA, SA, NA, and PI denote Foundational Analysis, Sleep Assessment, Neurocognitive Assessment, and Physiological Integration, respectively. *Open* indicates access without case-by-case approval, whereas *Restricted* indicates access requiring an application, institutional affiliation, ethical approval, or a data-use agreement.

Dataset	Subset	Modalities and native sampling rates	EEG ch.	Access
ISRUC (Khalighi et al., 2016)	FA/SA	EEG, EOG, EMG, ECG (200 Hz); snore (200 Hz); airflow (12.5/25 Hz); respiratory effort (25 Hz); SpO_2 (12.5 Hz); position (25 Hz)	6	Open
BCIC2020-3 (Jeong et al., 2022)	FA	EEG (256 Hz)	64	Open
SEED-V (Liu et al., 2021)	FA	EEG (1,000 Hz)	62	Restricted
Mumtaz2016 (Mumtaz et al., 2017)	FA	EEG (256 Hz)	19	Open
MentalArithmetic (Zyma et al., 2019)	FA	EEG (500 Hz)	19	Open
HMC (Alvarez-Estevez and Rijsman, 2020)	SA	EEG, EOG, EMG, ECG (256 Hz)	4	Open
MASS-SS3 (O'reilly et al., 2014)	SA	EEG, EOG, EMG, ECG (256 Hz)	20	Restricted
PhysioNet 2018 (Ghassemi et al., 2018)	SA	EEG, EOG, EMG, ECG, respiration, SpO_2 (200 Hz)	6	Open
SHHS-1 (Quan et al., 1997)	SA/PI	EEG, EMG, ECG (125 Hz); EOG (50 Hz); respiration (10 Hz); SpO_2, heart rate, position, light, and status signals (1 Hz)	2	Restricted
FACED (Chen et al., 2023)	NA	EEG (1000Hz)	32	Open
REFED (Ning et al., 2026)	NA	EEG (1000Hz); fNIRS (47.62Hz)	64	Open
COG-BCI (Hinss et al., 2023)	NA	EEG (500Hz); ECG (500Hz)	63	Open
MPD-DF (Li et al., 2026c)	NA	EEG (500Hz); EOG (256Hz); ECG (1024Hz); respiration (32Hz)	32	Open
SEED-VII (Jiang et al., 2024b)	PI	EEG, EOG, ECG (1,000 Hz); eye tracking (250 Hz)	64	Restricted
DEAP (Koelstra et al., 2011)	PI	EEG, EXG, GSR, ERG, respiration, PPG, and status signals (512 Hz)	32	Restricted
Simultaneous Dataset B (Shin et al., 2018)	PI	EEG, EOG (1,000 Hz); fNIRS (10.4167 Hz)	28	Open
SEED-VIG (Zheng and Lu, 2016)	PI	EEG (200 Hz); forehead EOG (125 Hz)	17	Restricted

BrainBench comprises four complementary subsets that cover distinct dimensions of comprehensive EEG understanding. The detailed composition of each subset is as follows:

- ***Foundational Analysis.*** This subset comprises 40 `tasks` and 950 `instances` derived from ISRUC (Khalighi et al., 2016), BCIC2020-3 (Jeong et al., 2022), SEED-V Liu et al. (2021), Mumtaz2016 (Mumtaz et al., 2017), and MentalArithmetic (Zyma et al., 2019), together with a curated EEG and BCI knowledge collection. It evaluates general-purpose EEG capabilities spanning recording inspection, preprocessing, feature extraction, spatiotemporal comparison, connectivity analysis, artifact generation, and evidence-grounded interpretation.
- ***Sleep Assessment.*** This subset comprises 43 `tasks` and 1,025 `instances` derived from HMC (Alvarez-Estevez and Rijsman, 2020), ISRUC (Khalighi et al., 2016), MASS-SS3 (O'reilly et al., 2014), PhysioNet 2018 (Ghassemi et al., 2018), and SHHS-1 (Quan et al., 1997), together with a curated sleep-medicine knowledge collection. It evaluates sleep architecture quantification, staging and spectral interpretation, sleep-event detection, multimodal physiological analysis, and the calculation of clinically relevant whole-night indices.
- ***Neurocognitive Assessment.*** This subset comprises 50 `tasks` and 1,035 `instances` derived from FACED (Chen et al., 2023), REFED (Ning et al., 2026), COG-BCI (Hinss et al., 2023), and MPD-DF (Li et al., 2026c), together with curated domain-knowledge questions. It evaluates the understanding of affective states, cognitive workload, and fatigue through state recognition, feature-based comparison, temporal and within-subject analysis, multimodal physiological evidence integration, and scientifically grounded interpretation.
- ***Physiological Integration.*** This subset comprises 39 `tasks` and 1,095 `instances` derived from SEED-VII (Jiang et al., 2024b), DEAP (Koelstra et al., 2011), Simultaneous Dataset B (Shin et al., 2018), SEED-VIG (Zheng and Lu, 2016), and SHHS-1 (Quan et al., 1997), together with a curated multimodal neurophysiology knowledge collection. It evaluates the integration of EEG with complementary physiological signals through temporal alignment, modality-specific

feature extraction, cross-modal coupling and evidence fusion, signal-quality assessment, data repair, missing-modality reconstruction, and multimodal result generation.

Across these four subsets, BrainBench incorporates 17 unique datasets spanning research-oriented EEG recordings, clinical polysomnography, neurocognitive experiments, and multimodal neurophysiological measurements. To provide consistent subject coverage while keeping large-scale evaluation tractable, **BrainBench includes recordings from the first five subjects in the source-defined ordering for each dataset**. The sole exception is MPD-DF, for which subjects 1–4 and 6 are used because the fifth subject was excluded due to data-quality issues. Table **??** summarizes the subset assignment, signal modalities and their native sampling rates, number of EEG channels, and accessibility of each dataset. To preserve the characteristics of the original recordings, we do not apply additional filtering, resampling, re-referencing, or artifact removal to the source-distributed signals. File-format harmonization and standardized naming are applied where necessary to support benchmark construction and data management. For datasets containing signals acquired at different sampling rates, each modality retains its native temporal resolution.

B Task Design Details

This appendix provides a task-level inventory of BrainBench. Each task is distilled from representative EEG analysis capabilities and practical analytical workflows, and each entry specifies the globally unique task identifier, difficulty level, number of associated instances, assessment objective, and validation configuration. **All task contents were cross-reviewed by three domain experts, who additionally verified the consistency among the instruction, ground-truth script, and validation configuration.** E, M, and H denote Easy, Medium, and Hard, respectively. Validation units are abbreviated as follows: Num. for numerical validation, Cat. for categorical validation, Set for unordered-set validation, Seq. for ordered-sequence validation, Sem. for semantic validation, and File for artifact validation. The notation $\times n$ indicates the number of validation units of a given type, while "+" denotes the joint use of multiple validation-unit types within a task..

B.1 Foundational Analysis

As shown in Table .4, the Foundational Analysis subset comprises 40 `tasks` and 950 `instances` constructed from ISRUC, BCIC2020-3, SEED-V, Mumtaz2016, and MentalArithmetic. It evaluates recording inspection, spectral and nonlinear feature extraction, preprocessing and artifact generation, channel- and region-level comparison, connectivity analysis, robustness to invalid requests, and foundational knowledge.

Table 4: Task-level inventory of the Foundational Analysis subset.

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
FA-01	E	25	Estimate mean channel-wise alpha relative power after standardized EEG filtering and PSD integration.	Num. $\times 1$
FA-02	E	25	Quantify prefrontal alpha/beta and theta/beta power ratios using available frontal channels.	Num. $\times 2$
FA-03	E	25	Measure short-window occipital signal complexity using sample entropy and aggregate it across channels.	Num. $\times 1$
FA-04	M	25	Execute a multistep central-channel preprocessing pipeline and compute mean Hjorth mobility and complexity.	Num. $\times 2$
FA-05	M	25	Compute and compare regional spectral edge frequency (SEF95) over parietal/central and occipital channels.	Num. $\times 2$
FA-06	M	20	Quantify low-frequency baseline drift and classify its severity without removing the target phenomenon.	Cat. $\times 1$
FA-07	M	5	Identify the EEG electrode-placement system and detect channels outside the corresponding standard montage.	Cat. $\times 1$ + Set $\times 1$
FA-08	E	5	Determine whether the available channel montage supports left–right spectral asymmetry analysis.	Cat. $\times 1$

Continued on next page

Table 4 continued

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
FA-09	M	25	Rank the three EEG channel pairs with the strongest broadband Pearson correlations.	Seq. $\times 1$
FA-10	M	25	Characterize alpha-band inter-channel synchronization and identify the strongest correlated pair.	Num. $\times 1$ + Cat. $\times 1$
FA-11	E	25	Identify the dominant and secondary EEG frequency bands within a specified recording segment.	Cat. $\times 2$
FA-12	M	25	Rank channels by delta-band variance after filtering and resampling.	Seq. $\times 1$
FA-13	M	25	Rank channels independently by signal kurtosis and skewness after standardized preprocessing.	Seq. $\times 2$
FA-14	E	25	Extract, filter, resample, and export a frontal EEG segment as an EDF artifact.	File $\times 1$
FA-15	E	25	Select genuine EEG channels, apply average referencing, and export a fixed-duration EDF artifact.	File $\times 1$
FA-16	M	25	Select the channel with maximal alpha-relative energy, isolate its alpha component, and export it as an array.	File $\times 1$
FA-17	M	25	Apply line-noise suppression and generate a whole-recording EEG PSD visualization.	File $\times 1$
FA-18	M	25	Generate a band-limited PSD visualization for a specified pair of EEG channels.	File $\times 1$
FA-19	M	25	Compute and visualize inter-channel correlation for a specified multichannel segment.	File $\times 1$
FA-20	E	5	Handle an unavailable EEG file safely without fabricating a brain-state interpretation.	Sem. $\times 1$
FA-21	E	15	Recognize that a requested channel is absent and respond without inventing channel-level analysis.	Sem. $\times 1$
FA-22	E	25	Detect that a requested time window lies outside the recording and avoid unsupported analysis.	Sem. $\times 1$
FA-23	M	25	Verify a supplied dominant-band claim against the signal and resist an incorrect premise.	Cat. $\times 1$ + Sem. $\times 1$
FA-24	M	25	Detect when preprocessing removes the frequency content required by the requested downstream analysis.	Sem. $\times 1$
FA-25	M	25	Quantify notch-filter attenuation and determine whether suppression succeeds, including mismatched-frequency controls.	Num. $\times 1$ + Cat. $\times 1$
FA-26	H	25	Compare frontal and occipital alpha-relative power and identify the region with stronger activity.	Num. $\times 1$ + Cat. $\times 1$
FA-27	H	25	Compare regional band-power dominance across two time windows and interpret the spatial-state transition.	Cat. $\times 2$ + Sem. $\times 1$
FA-28	H	25	Rank brain regions by alpha-relative power and assess whether alpha activity is posterior dominant.	Seq. $\times 1$ + Cat. $\times 1$ + Sem. $\times 1$
FA-29	H	25	Compute alpha-band phase-locking connectivity, global mean PLV, and the strongest channel pairs.	Num. $\times 1$ + Set $\times 1$
FA-30	H	25	Compare global phase locking across four canonical bands and explain the dominant synchronization band.	Num. $\times 4$ + Sem. $\times 1$
FA-31	H	25	Compare frontal alpha/beta ratios across consecutive windows and infer the direction of attention-state change.	Num. $\times 2$ + Sem. $\times 1$
FA-32	H	25	Contrast global and occipital alpha rankings to assess whether global aggregation masks regional structure.	Cat. $\times 2$ + Sem. $\times 1$
FA-33	M	25	Remove variance-outlier channels and return channels whose alpha-relative power exceeds the retained-channel mean.	Set $\times 1$
FA-34	H	25	Estimate occipital alpha peak frequency and derive personalized alpha-band relative power.	Num. $\times 2$
FA-35	M	25	Execute standardized preprocessing and locate maximum global-field-power peaks in three windows.	Num. $\times 3$
FA-36	H	25	Compare left- and right-hemisphere wPLI in alpha and beta bands and assess cross-band consistency.	Num. $\times 2$ + Sem. $\times 1$

Continued on next page

Table 4 continued

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
FA-37	M	25	Identify channels that repeatedly appear among the highest relative-power channels across multiple bands.	Set $\times 1$
FA-38	H	25	Build an alpha-band PLV network and rank channels by weighted connectivity degree.	Seq. $\times 1$
FA-39	H	25	Rank channels by alpha power and infer the dominant anatomical region from the leading channels.	Seq. $\times 1$ + Set $\times 1$
FA-40	E	50	Answer a standalone foundational EEG and BCI knowledge question without requiring a signal file.	Cat. $\times 1$

B.2 Sleep Assessment

As shown in Table. 5, the Sleep Assessment subset comprises 43 `tasks` and 1,025 `instances` constructed from HMC, ISRUC, MASS-SS3, PhysioNet 2018, and SHHS-1. It evaluates sleep architecture, staging, spectral and temporal analysis, PSG artifact generation, arousal and respiratory-event analysis, oxygenation, and sleep-medicine knowledge.

Table 5: Task-level inventory of the Sleep Assessment subset.

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
SA-01	E	25	Calculate sleep onset latency from epoch-level sleep-stage labels.	Num. $\times 1$
SA-02	E	15	Inventory all PSG channels and distinguish genuine EEG channels from auxiliary sensors.	Num. $\times 1$ + Set $\times 1$
SA-03	E	25	Derive total sleep time, time in bed, and sleep efficiency from sleep-stage labels.	Num. $\times 3$
SA-04	E	25	Quantify wake after sleep onset and count post-onset awakening bouts.	Num. $\times 2$
SA-05	E	25	Compute NREM duration, REM proportion, and REM latency from whole-night staging.	Num. $\times 3$
SA-06	E	25	Identify the dominant and secondary sleep stages over the full recording.	Cat. $\times 2$
SA-07	M	25	Quantify the light-to-deep sleep ratio and interpret its implication for sleep architecture.	Num. $\times 1$ + Sem. $\times 1$
SA-08	M	25	Find the three longest uninterrupted sleep episodes and interpret whole-night continuity.	Num. $\times 3$ + Sem. $\times 1$
SA-09	E	25	Count all sleep-stage transitions and deep-sleep-to-wake interruptions.	Num. $\times 2$
SA-10	E	25	Count sustained N3 bouts and calculate their mean duration.	Num. $\times 2$
SA-11	M	25	Locate the longest nocturnal wake interruption and classify its continuity impact.	Num. $\times 2$ + Cat. $\times 1$
SA-12	M	25	Quantify first-half N3 and second-half REM concentration and interpret the overnight pattern.	Num. $\times 2$ + Sem. $\times 1$
SA-13	M	25	Calculate a whole-night sleep fragmentation index from awakenings and non-wake stage shifts.	Num. $\times 1$
SA-14	E	25	Calculate the whole-sequence percentage distribution of W, N1, N2, N3, and REM.	Num. $\times 5$
SA-15	H	25	Screen the sleep-stage sequence for a predefined panel of twelve sleep-structure abnormalities.	Cat. $\times 12$
SA-16	E	25	Determine the dominant EEG band and sleep stage in a specified recording segment.	Cat. $\times 2$
SA-17	M	25	Infer the sleep stages of ten consecutive epochs from the sleep recording.	Seq. $\times 1$
SA-18	E	25	Generate and save a whole-night sleep hypnogram.	File $\times 1$
SA-19	E	25	Execute a sleep-staging preprocessing pipeline and export fixed-length EEG epochs as an array.	File $\times 1$

Continued on next page

Table 5 continued

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
SA-20	E	25	Generate and save a sleep EEG spectrogram from the available EEG channels.	File $\times 1$
SA-21	M	25	Compare delta-relative power across three windows and identify the most slow-wave-rich segment.	Num. $\times 3$ + Cat. $\times 1$
SA-22	M	25	Compare event-window and whole-night chin-EMG activity to assess whether the segment is REM-like.	Num. $\times 1$ + Sem. $\times 1$
SA-23	M	25	Compare first- and second-half delta-relative energy and assess consistency with canonical sleep architecture.	Num. $\times 2$ + Sem. $\times 1$
SA-24	M	25	Perform whole-recording automatic sleep staging and summarize sleep efficiency, N3, and REM proportions.	Num. $\times 3$
SA-25	M	25	Test a supplied band-and-stage claim against the signal and reject unsupported conclusions.	Sem. $\times 1$
SA-26	M	25	Rank three segments by EOG activity and determine whether the strongest segment represents REM sleep.	Seq. $\times 1$ + Cat. $\times 1$
SA-27	M	25	Rank three segments by EMG activity and identify whether and where REM sleep is present.	Seq. $\times 1$ + Sem. $\times 1$
SA-28	M	25	Estimate whole-night mean, minimum, and maximum heart rate from the ECG channel.	Num. $\times 3$
SA-29	H	40	Detect arousal independently in five specified sleep segments.	Cat. $\times 5$
SA-30	H	40	Classify five specified respiratory segments as apnea, hypopnea, or no target event.	Cat. $\times 5$
SA-31	H	10	Calculate the whole-night arousal index using detected arousals and total sleep time.	Num. $\times 1$
SA-32	H	15	Calculate the whole-night apnea–hypopnea index from respiratory events and total sleep time.	Num. $\times 1$
SA-33	H	15	Quantify respiratory disturbance or detect respiratory-effort-related arousals in selected segments.	Num. $\times 1$; or Cat. $\times 5$
SA-34	M	9	Identify the longest apnea event and report its duration, onset, and sleep stage.	Num. $\times 2$ + Cat. $\times 1$
SA-35	M	46	Distinguish central, obstructive, and mixed apnea subtypes in two specified segments.	Cat. $\times 2$
SA-36	H	13	Calculate stage-specific apnea–hypopnea indices for REM and NREM sleep.	Num. $\times 2$
SA-37	H	40	Reconstruct respiratory-event and arousal chronology and assess its likely impact on sleep continuity.	Sem. $\times 1$
SA-38	H	10	Quantify N3-specific arousal burden and transitions and assess deep-sleep disruption.	Num. $\times 2$ + Sem. $\times 1$
SA-39	H	15	Detect respiratory-event clusters in fixed windows and determine their dominant sleep stage.	Num. $\times 1$ + Cat. $\times 1$
SA-40	M	14	Calculate sleep-period oxygen desaturation indices using 3% and 4% thresholds.	Num. $\times 2$
SA-41	M	14	Derive sleep/wake mean oxygen saturation and minimum sleep oxygen saturation after signal cleaning.	Num. $\times 3$
SA-42	M	14	Quantify cumulative sleep time below 90% and 80% oxygen saturation.	Num. $\times 2$
SA-43	E	40	Answer a standalone sleep-medicine and polysomnography knowledge question.	Cat. $\times 1$

B.3 Neurocognitive Assessment

The Neurocognitive Assessment subset comprises 50 `tasks` and 1,035 `instances` constructed from FACED, REFED, COG-BCI, and MPD-DF. It includes 20 affective-state tasks, 15 cognitive-load tasks, and 15 fatigue and vigilance tasks, covering state recognition, feature-based and temporal comparison, multimodal evidence integration, claim verification, artifact generation, and domain knowledge. Tasks NA-20, NA-35, and NA-50 are knowledge-only tasks; the remaining tasks are grounded in neurophysiological recordings or associated behavioral annotations shown in Table. 6.

Table 6: Task-level inventory of the Neurocognitive Assessment subset.

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
NA-01	M	30	Determine the emotional polarity of a specified EEG segment as positive, neutral, or negative.	Cat. $\times 1$
NA-02	M	20	Classify the arousal or valence level of a specified EEG segment as high or low.	Cat. $\times 1$
NA-03	E	30	Compute frontal alpha asymmetry from F3/F4 alpha power in a specified EEG segment.	Num. $\times 1$
NA-04	E	10	Evaluate whether a stated emotion polarity is supported by the trial's SAM arousal and valence ratings.	Sem. $\times 1$
NA-05	M	15	Correlate channel-band EEG features with a continuous SAM label sequence and identify the two distinct bands with the strongest absolute associations.	Set $\times 1$
NA-06	M	20	Compare two trials using beta/alpha relative-power patterns, identify the most discriminative channels, and determine which trial has higher arousal.	Set $\times 1$ + Cat. $\times 1$ + Sem. $\times 1$
NA-07	M	20	Compare positive and negative trials, identify the channels with the largest alpha-power increase, and determine their dominant brain region.	Set $\times 1$ + Cat. $\times 1$
NA-08	M	20	Compute frontal alpha differential asymmetry across homologous channel pairs, infer the dominant activation and valence direction, and evaluate a polarity statement.	Cat. $\times 1$ + Sem. $\times 1$
NA-09	H	20	Determine whether emotional polarity changes within a continuous EEG segment and explain the transition or stable state.	Cat. $\times 1$ + Sem. $\times 1$
NA-10	M	20	Analyze temporal trends in multiband EEG and hemispheric alpha features to assess agreement with a known emotion label.	Cat. $\times 1$ + Sem. $\times 1$
NA-11	H	20	Compare positive and negative trials using theta-band coherence, identify the channel pairs with the largest differential connectivity, and interpret the result.	Set $\times 1$ + Sem. $\times 1$
NA-12	M	20	Compare an emotional trial with a neutral baseline and determine whether its frontal alpha dynamics are emotion-specific or neutral-like.	Cat. $\times 1$ + Sem. $\times 1$
NA-13	H	20	Rank three emotional trials by EEG-based arousal and identify whether the highest-arousal trial is positive or negative.	Seq. $\times 1$ + Cat. $\times 1$
NA-14	H	20	Compare positive and negative trial groups using regional band-power effect sizes, identify the strongest discriminative region-band feature, and interpret its emotion direction.	Set $\times 1$ + Cat. $\times 1$ + Sem. $\times 1$
NA-15	M	20	Evaluate a statement about left-right alpha-power asymmetry using homologous EEG channels.	Cat. $\times 2$
NA-16	M	20	Compare a target trial with a neutral baseline and determine whether a supplied emotion-EEG claim is valid.	Cat. $\times 1$ + Sem. $\times 1$
NA-17	H	20	Correlate multiband EEG features with trial-level arousal ratings, generate a channel-band correlation heatmap, and identify the two bands with the highest mean correlations.	File $\times 1$ + Set $\times 1$
NA-18	M	20	Compare average sample entropy between two trials and determine which trial has greater signal complexity.	Num. $\times 2$ + Cat. $\times 1$
NA-19	M	20	Select which of three emotional trials best matches a specified target emotion and explain the EEG evidence.	Cat. $\times 1$ + Sem. $\times 1$
NA-20	E	40	Answer a standalone EEG and BCI emotion-recognition knowledge question.	Cat. $\times 1$
NA-21	M	20	Classify a continuous N-back EEG segment as low or high cognitive workload.	Cat. $\times 1$
NA-22	H	20	Compare two workload segments using frontal theta power and determine whether the neural direction agrees with the workload judgment.	Cat. $\times 1$ + Sem. $\times 1$

Continued on next page

Table 6 continued

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
NA-23	M	15	Compare behavioral performance across two N-back conditions and determine which condition shows greater behavioral load or poorer performance.	Num. $\times 10$ + Cat. $\times 1$ + Sem. $\times 1$
NA-24	H	20	Rank three EEG segments by the frontal-theta/parietal-alpha ratio and interpret the ranking as cognitive workload evidence.	Seq. $\times 1$ + Sem. $\times 1$
NA-25	H	20	Compare two unlabeled workload segments, determine the higher-workload segment, and assess frontal-theta and parietal-alpha consistency with that judgment.	Cat. $\times 1$ + Sem. $\times 1$
NA-26	M	15	Compare beta/(alpha+theta) engagement indices between two EEG segments and identify the more engaged segment.	Num. $\times 1$ + Sem. $\times 1$
NA-27	H	15	Rank three EEG segments from lowest to highest cognitive workload using only their EEG evidence.	Seq. $\times 1$
NA-28	M	20	Compare stimulus-locked P300 responses between two single-trial segments and infer their relative cognitive workload.	Cat. $\times 1$ + Sem. $\times 1$
NA-29	H	20	Evaluate a claim linking frontal-parietal theta coherence to cognitive workload and determine the higher-workload segment.	Sem. $\times 1$
NA-30	H	20	Compare P300 amplitude and latency between two stimulus-locked segments and infer their relative cognitive load.	Cat. $\times 3$ + Sem. $\times 1$
NA-31	H	20	Integrate behavioral errors, reaction time, and EEG theta/alpha ratio to assess agreement between behavioral and neural workload evidence.	Cat. $\times 3$ + Sem. $\times 1$
NA-32	H	20	Evaluate a compound claim about cognitive load and engagement using theta/alpha and beta/(alpha+theta) evidence.	Cat. $\times 3$ + Sem. $\times 1$
NA-33	M	15	Determine whether cognitive workload changes within a continuous EEG interval and describe the stable or changing workload state.	Sem. $\times 1$
NA-34	H	15	Compute target and nontarget P300/P3b waveforms at Pz and export their joint visualization.	File $\times 1$
NA-35	E	40	Answer a standalone EEG and BCI cognitive-load knowledge question.	Cat. $\times 1$
NA-36	E	15	Compute blink rate, mean blink duration, and slow-eye-movement power from an EOG window.	Num. $\times 3$
NA-37	M	20	Compare two EOG segments for long eye-closure activity and determine which segment appears more fatigued.	Cat. $\times 2$
NA-38	M	20	Compare EEG band-power change patterns between two segments and determine which segment is more fatigued.	Cat. $\times 3$ + Sem. $\times 1$
NA-39	H	20	Integrate EEG fatigue activity with PSG/EOG eye-movement evidence to determine which of two segments shows greater overall fatigue.	Cat. $\times 1$
NA-40	E	20	Classify a single EEG segment as closer to a wakeful or fatigued state.	Cat. $\times 1$
NA-41	M	20	Detect long eye closure, slow eye movement, and prolonged blink events in an EOG window.	Cat. $\times 3$
NA-42	H	20	Determine whether a wakefulness-to-fatigue transition occurs within a long EEG/PSG window and localize its timing.	Cat. $\times 2$
NA-43	H	20	Integrate EEG and EOG evidence to determine whether modality-specific fatigue conclusions agree or conflict.	Cat. $\times 3$
NA-44	M	20	Compare ECG-derived heart rate and HRV indicators between two segments and assess their support for a fatigue relation.	Cat. $\times 2$ + Sem. $\times 1$
NA-45	M	20	Compare respiration rate, timing variability, amplitude variability, and respiration-band power between two segments to identify the more fatigue-supporting rhythm.	Cat. $\times 4$ + Sem. $\times 1$

Continued on next page

Table 6 continued

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
NA-46	H	20	Compute pairwise correlations among aligned EEG, EOG, ECG, and respiration feature streams and identify the strongest feature pairs.	Set $\times 1$
NA-47	H	20	Rank three segments from most to least fatigued using combined EEG and PSG eye-movement evidence.	Seq. $\times 1$
NA-48	H	20	Compare regional EEG band-power profiles across three segments and identify the features with the largest fatigue-related differences.	Set $\times 1$
NA-49	H	20	Measure signed cross-signal correlations between EOG slow-eye-movement power and frontal, temporal, and occipital EEG relative-power streams.	Num. $\times 3$
NA-50	E	40	Answer a standalone EEG and BCI fatigue or vigilance knowledge question.	Cat. $\times 1$

B.4 Physiological Integration

The Physiological Integration subset comprises 39 `tasks` and 1,095 `instances` constructed from SEED-VII, DEAP, Simultaneous Dataset B, SEED-VIG, and SHHS-1. It evaluates multimodal data handling and synchronization, feature extraction, cross-modal coupling and fusion, quality assessment, signal repair, matching, missing-modality reconstruction, visualization, and domain knowledge. Tasks PI-01–PI-38 operate on neurophysiological recordings, whereas PI-39 evaluates multimodal neurophysiology knowledge without recording access shown in Table. 7.

Table 7: Task-level inventory of the Physiological Integration subset.

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
PI-01	E	25	Determine whether two named channels are analyzable physiological signals.	Cat. $\times 2$
PI-02	E	25	Count analyzable channels for two requested physiological modalities.	Num. $\times 2$
PI-03	E	5	Map a common time interval to sample-index ranges for two channels.	Num. $\times 4$
PI-04	E	25	Report two native sampling rates and a signal statistic at the recording boundary.	Num. $\times 3$
PI-05	M	25	Extract synchronized segments from two modalities and save standardized signal files.	File $\times 2$
PI-06	M	20	Compare target EEG and EOG amplitudes with same-record references and determine whether joint artifact evidence is present.	Num. $\times 2$ + Cat. $\times 1$
PI-07	M	20	Rank analyzable physiological channels by normalized variability within a declared interval.	Set $\times 1$
PI-08	E	10	Compute a declared feature from an electrodermal, respiratory, pulse, or temperature signal.	Num. $\times 1$
PI-09	E	10	Compute EOG amplitude within a specified interval.	Num. $\times 1$
PI-10	E	10	Compute a declared respiratory, cardiac, oximetry, or EMG feature.	Num. $\times 1$
PI-11	E	10	Estimate an fNIRS optical-density response relative to a baseline interval.	Num. $\times 1$
PI-12	E	10	Compute a declared gaze or pupil feature while respecting missing eye-tracking samples.	Num. $\times 1$
PI-13	M	40	Measure the direction of association between aligned EEG and auxiliary physiological feature trajectories.	Cat. $\times 1$ (30); Sem. $\times 1$ (10)
PI-14	M	30	Compare event-centered EEG and auxiliary physiological responses before and after an event.	Num. $\times 2$
PI-15	M	20	Compare trial-related frontal EEG asymmetry and temperature responses and classify their directional relationship.	Num. $\times 2$ + Cat. $\times 1$

Continued on next page

Table 7 continued

Task ID	Diff.	# Inst.	Assessment content	Validation unit(s)
PI-16	M	30	Evaluate EEG and EOG drowsiness evidence relative to same-record reference intervals.	Num. $\times 2$ + Cat. $\times 1$
PI-17	H	30	Assess the association between early EEG and delayed fNIRS responses across repeated events.	Num. $\times 2$ + Sem. $\times 1$
PI-18	E	30	Select the sleep epoch with the strongest relative EOG activity and summarize its EOG and EMG ratios.	Num. $\times 3$
PI-19	M	30	Compare EOG and EMG activity between REM and N2 sleep and interpret the combined stage pattern.	Sem. $\times 1$ (5); Cat. $\times 2$ + Sem. $\times 1$ (25)
PI-20	H	30	Quantify repeated-event EEG–fNIRS association and interpret its direction, strength, and uncertainty.	Num. $\times 2$ + Sem. $\times 1$
PI-21	H	30	Compare EEG and HbO trajectories at zero delay and with a delayed hemodynamic response.	Num. $\times 2$
PI-22	E	30	Compare EEG amplitude and fNIRS intensity changes between two recording periods.	Num. $\times 2$
PI-23	H	40	Identify trials showing jointly elevated frontal EEG and pupil responses.	Set $\times 1$
PI-24	H	40	Compare target and reference trials using frontal EEG and pupil responses, then interpret their directional agreement.	Num. $\times 2$ + Sem. $\times 1$
PI-25	H	40	Identify a blink-contaminated interval and produce an EEG–EOG figure with localized artifact marking.	Cat. $\times 1$ + File $\times 1$
PI-26	H	40	Determine whether low-posterior-alpha epochs are associated with central gaze concentration.	Cat. $\times 1$
PI-27	M	30	Summarize posterior-versus-frontal EEG alpha and gaze concentration as a spatial multimodal profile.	Cat. $\times 1$ + Num. $\times 1$ + Sem. $\times 1$
PI-28	E	30	Determine the directions of posterior EEG alpha and GSR changes from baseline to stimulation.	Cat. $\times 2$
PI-29	E	30	Identify the strongest early, middle, or late drowsiness phase separately from EEG and EOG.	Cat. $\times 2$
PI-30	M	40	Use the strongest GSR response window to identify the posterior EEG channel with greatest alpha suppression.	Cat. $\times 2$
PI-31	M	30	Rank posterior EEG frequency bands during central- and peripheral-gaze periods.	Seq. $\times 2$
PI-32	M	40	Compare Go and NoGo EEG and HbO responses and interpret whether the modalities favor the same condition.	Cat. $\times 2$ + Sem. $\times 1$
PI-33	M	30	Estimate event-averaged EEG and HbO peak latencies and interpret their temporal ordering.	Num. $\times 2$ + Sem. $\times 1$
PI-34	H	40	Select the interval with the strongest frontal EEG disturbance and identify supporting EOG or EMG evidence.	Cat. $\times 2$
PI-35	M	30	Summarize HbO and HbR responses for events selected by their EEG theta response.	Num. $\times 2$
PI-36	H	30	Compare the timing of peak posterior EEG suppression and GSR response across rating-selected trials.	Num. $\times 2$
PI-37	H	30	Rank long intervals by EEG and EOG drowsiness evidence and measure their cross-modal association.	Seq. $\times 2$ + Num. $\times 1$
PI-38	H	40	Identify which emotion rating is most strongly associated with EEG and GSR responses.	Cat. $\times 2$
PI-39	E	40	Answer a multiple-choice question on multimodal neurophysiology and signal interpretation.	Cat. $\times 1$

B.5 EXAMPLE INSTANCES

Here, we showcase representative instances from each subset.

Foundational Analysis: Task 30-Instance 1

"meta_info": {"instance_id": "ISRUC_01", "bench_subset": "Foundational Analysis", "difficult": 2.0}

"agent_input": {"data_path": "data/core/ISRUC_01.edf", "instruction": "Please select standard EEG channels. For each delta, theta, alpha, and beta band, apply a zero-phase FIR bandpass filter to the selected continuous signals, then use the first 120 seconds of the filtered data. Extract instantaneous phase with the Hilbert transform, compute PLV for every unique channel pair, and define the global mean PLV as the arithmetic mean across those pairs. Report the band with the highest global mean PLV and briefly explain the result."}

"eval_config": {
"parser_prompt": "detailed in Appendix C.2",
"metrics": [
{"metric_id": "delta_global_mean_plv_accuracy", "type": "numeric_check", "weight": 15,
"params": {"gt_value": 0.33870515895882675, "tolerance": 0.01693525794794134}},
{"metric_id": "theta_global_mean_plv_accuracy", "type": "numeric_check", "weight": 15,
"params": {"gt_value": 0.34405984632756176, "tolerance": 0.01720299231637808}},
{"metric_id": "alpha_global_mean_plv_accuracy", "type": "numeric_check", "weight": 15,
"params": {"gt_value": 0.3436785796980445, "tolerance": 0.017183928984902227}},
{"metric_id": "beta_global_mean_plv_accuracy", "type": "numeric_check", "weight": 15,
"params": {"gt_value": 0.33105312252380337, "tolerance": 0.01655265612619017 }},
{"metric_id": "dominant_band_explanation_consistency", "type": "semantic_check", "weight": 40,
"params": {"judge_prompt": "Judge whether the final dominant synchronization band and its explanation are consistent with the true ranking of delta/theta/alpha/beta global mean PLV values. The true dominant synchronization band is theta. Mark the response as correct only if it selects the true highest-PLV band and the explanation clearly supports that band as the strongest synchronization band. Output only a valid JSON object with exactly one key named status and a boolean value True or False."}}]}

Sleep Assessment: Task 36-Instance 6

"meta_info": {"instance_id": "SHHS1_01", "bench_subset": "Sleep Assessment", "difficult": 2.0}

"agent_input": {"data_path": "data/sleep/SHHS1_01.edf", "label_path": "data/sleep/SHHS1_01.npy", "instruction": "Use the provided sleep EEG/PSG file and paired sleep-stage label file to calculate stage-specific Apnea-Hypopnea Indices for the whole recording. Assign each apnea or hypopnea event to the sleep stage at the event start. REM-AHI is the number of apnea plus hypopnea events starting during R/REM divided by REM sleep time in hours. NREM-AHI is the number starting during N1, N2, or N3 divided by NREM sleep time in hours. Events starting during Wake are excluded from both indices. Respiratory channels for SHHS1: THOR RES = thoracic respiratory effort; ABDO RES = abdominal respiratory effort; NEW AIR = nasal/oral airflow; SaO2 = oximetry. Report the final REM-AHI and NREM-AHI as events per hour."

"eval_config": {
"parser_prompt": "detailed in Appendix C.2",
"metrics": [
{"metric_id": "rem_apnea_hypopnea_index_accuracy", "type": "numeric_check", "weight": 50,
"params": {"gt_value": 47.05882352941177, "tolerance": 4.705882352941177}},
{"metric_id": "nrem_apnea_hypopnea_index_accuracy", "type": "numeric_check", "weight": 50,
"params": {"gt_value": 8.320493066255779, "tolerance": 0.8320493066255779}}]}

Neurocognitive Assessment: Task 6-Instance 1

"meta_info": {"instance_id": "FACED_01", "bench_subset": "Neurocognitive Assessment", "difficult": 1.5}

"agent_input": {"data_path": "data/emotion/FACED_01.edf", "instruction": "Please compare two EEG trial segments from the same subject. Trial A spans 30.0 to 60.0 seconds, and Trial B spans 420.0 to 450.0 seconds. Use the EEG channels available in the EDF channel list. Using the Beta/Alpha relative power pattern across those channels, identify the three channels with the largest absolute difference between Trial A and Trial B. Determine whether Trial A or Trial B has higher emotional arousal. Please report the three channel names, the higher-arousal trial, and briefly explain why."

"eval_config": {
"parser_prompt": "detailed in Appendix C.2",
"metrics": [
{"metric_id": "arousal_contrast_top3", "type": "set_match_check", "weight": 40,
"params": {"gt_value": [" Fp2 ", "Fp1", "T3"], "match_mode": "per_hit_fraction" , "top_k": 3 , "score_per_hit": 0.3333}},
{"metric_id": "arousal_trial_classification", "type": "categorical_check", "weight": 30,
"params": {"gt_value": "Trial A"}},
{"metric_id": "arousal_physiology_logic", "type": "semantic_check", "weight": 30,
"params": {"judge_prompt": "Evaluate the explanation for the Beta/Alpha ratio as an emotional arousal index. Award full points if the text correctly links a higher Beta/Alpha ratio to higher arousal through increased cortical activation, increased beta-band activity, reduced alpha-band activity, alpha suppression/desynchronization, or an equivalent explanation of beta dominance over alpha. The explanation may use terms such as alertness, arousal, vigilance, stress, or activation. Score 0 if the text primarily treats Beta/Alpha as a valence or positive/negative emotion marker, reverses the arousal direction, ignores the Beta/Alpha mechanism entirely, or gives only a generic statement with no physiological relation to beta/alpha activity."}}]}

Physiological Integration: Task 31-Instance 22

"meta_info": {"instance_id": "SEED-VII_022", "bench_subset": " Physiological Integration ", "difficult": 1.5}

"agent_input" :{"data_path":"["data/multi/SEEDVII_04_EEG.cnt","data/multi/SEEDVII_04_EyeTracking.edf"],
"label_path": "data/multi/SEEDVII_04_events.npy",
"instruction": "For SEED-VII trial `12`, analyze the following two equal-duration, trial-relative 2-second windows: `[48, 50)` and `[37, 39)` seconds. Classify a window as `central` when at least half of its valid gaze samples fall within the central 50% × 50% screen region; otherwise classify it as `peripheral`. Ignore invalid gaze samples. These two windows were selected to contain one central and one peripheral window. For each window, compute `delta`, `theta`, `alpha`, `beta`, and `gamma` power across `P3`, `PZ`, `P4`, `PO3`, `POZ`, `PO4`, `O1`, `OZ`, and `O2`. Convert the five band powers within each window into proportions of their total power. Average the band proportions separately for the central and peripheral windows, then rank the five bands from highest to lowest proportion. Report `central_band_ranking` and `peripheral_band_ranking`."

"eval_config": {
"parser_prompt": "detailed in Appendix C.2",
"metrics": [
{"metric_id": "central_band_ranking", "type": "sequence_match_check", "weight": 50,
"gt_value": ["alpha", "beta", "delta", "theta", "gamma"], "top_k": 5, "position_weights": [1.0, 0.8, 0.6, 0.4, 0.2]},
{"metric_id": "peripheral_band_ranking", "type": "sequence_match_check", "weight": 50,
"gt_value": ["delta", "alpha", "beta", "theta", "gamma"] , "top_k": 5, "position_weights": [1.0, 0.8, 0.6, 0.4, 0.2]}]}

C BENCHMARK CONSTRUCTION DETAILS

In this section, we expand the evaluator-side construction procedure summarized as follows. Section C.1 describes how deterministic analysis scripts derive instance-specific ground truth from the bound recordings and parameters. Section C.2 then details how the Parser Agent aligns free-form reports with structured target fields and how the six validation units score the resulting outputs.

C.1 DETERMINISTIC GROUND-TRUTH GENERATION

Each `instance` is paired with ground truth after its parent `task` has been bound to a concrete recording, analysis window, signal selection, and parameter configuration. A deterministic reference script executes the prescribed workflow on the same input files provided to the target system and produces the numerical values, labels, collections, sequences, or artifacts required by the corresponding validation units. These reference targets are stored only in the *Validation Configuration*, together with task-specific tolerances and metric weights, and remain inaccessible to the evaluated model.

As an example, the first instance `FA-01-Instance1` is bound to `ISRUC_01.edf` and provides the following instruction:

> *Please first extract only the EEG channels from the raw signal, and then apply a 0.5–40 Hz FIR bandpass filter to these channels. After filtering, obtain each channel's Alpha-band power and total filtered-signal power by integrating the PSD over frequency. Calculate Alpha relative power separately for each channel, then average the channel-wise ratios and report the final percentage value clearly in your response.*

The reference script follows the same analysis specification to compute the expected alpha relative power and populate the numerical target and tolerance in the instance-level *Validation Configuration*. Listing 1 retains only this core computation; input loading, dataset-specific channel selection, batch processing, and construction of the complete instance specification are omitted for clarity.

```
def compute_alpha_relative_power(raw, picks):
work = raw.copy().pick(picks)
work.filter(l_freq=0.5, h_freq=40.0, verbose=False)

sfreq = float(work.info["sfreq"])
n_fft = int(4.0 * sfreq)
n_overlap = min(n_fft // 2, max(0, n_fft - 1))
spectrum = work.compute_psd(
method="welch",
fmin=0.5,
fmax=40.0,
n_fft=n_fft,
n_overlap=n_overlap,
verbose=False,
)
psds, freqs = spectrum.get_data(return_freqs=True)

total_power = np.trapz(psds, freqs, axis=1)
alpha_mask = (freqs >= 8.0) & (freqs <= 13.0)
alpha_power = np.trapz(
psds[:, alpha_mask], freqs[alpha_mask], axis=1
)
relative_alpha = alpha_power / np.maximum(
total_power, 1e-20
)
return float(np.mean(relative_alpha) * 100.0)
```

Listing 1: Ground-truth computation for Foundational Analysis task FA-01-Instance1.

Applying the same fixed script to every recording associated with FA-01 changes only the data-dependent result while preserving the analytical definition of the task. Other tasks follow the same construction principle, with their reference scripts returning the numerical values, labels, collections, sequences, or artifacts required by the corresponding validation units.

C.2 OUTPUT PARSING AND MULTI-UNIT VALIDATION

BrainBench permits free-form analytical reports because EEG analysis extends beyond scalar prediction to include explanations, supporting evidence, and references to generated artifacts. Imposing

a rigid output schema on each target system would conflate EEG understanding with formatting compliance. To decouple these factors, the Parser Agent receives the final report together with an instance-specific extraction prompt and maps only explicitly reported information into a predefined JSON schema. It has no access to the ground truth and does not assess scientific correctness, correct erroneous answers, or infer omitted results. Any missing or unresolvable field is returned as `null`.

Listing 2 shows the parser prompt for instance `FA-30-Instance1`. The underlying task requests global mean phase-locking values (PLVs) in four frequency bands and a conclusion identifying the dominant synchronization band. The prompt fixes both the target fields and their admissible types so that differently worded reports can be evaluated through the same interface.

```
### ROLE
You are a precise JSON Extraction Engine for neurophysiological data. Your sole task is to
     convert natural language reports into structured JSON data.

### TASK
Extract five outputs from the agent report:
1) delta global mean PLV
2) theta global mean PLV
3) alpha global mean PLV
4) beta global mean PLV
5) dominant synchronization band

### STRICT CONSTRAINTS (MANDATORY)
1. Output ONLY a valid JSON object.
2. DO NOT include Markdown code blocks.
3. DO NOT include any conversational text or explanations.
4. Keys must be EXACTLY "delta_global_mean_plv", "theta_global_mean_plv", "
     alpha_global_mean_plv", "beta_global_mean_plv", and "dominant_synchronization_band".
5. The four PLV values must be float or null.
6. "dominant_synchronization_band" must be one of "delta", "theta", "alpha", "beta", or null.

### OUTPUT TEMPLATE
{"delta_global_mean_plv": <float|null>, "theta_global_mean_plv": <float|null>, "
     alpha_global_mean_plv": <float|null>, "beta_global_mean_plv": <float|null>, "
     dominant_synchronization_band": <string|null>}
```

Listing 2: Parser prompt used for `FA-30-Instance1`.

For this instance, the four extracted PLV fields are passed to numerical validation, while the original report is passed to a task-specific Semantic Judge that checks whether the selected band and its explanation agree with the ground-truth PLV ranking. More generally, each metric selects either a parsed field, the complete report, or a generated artifact and applies one of the six rules described below. Let $\hat{y}$ denote a parsed prediction, y its reference target, and $v \in [0, 1]$ the resulting validation score. Let $\mathcal{C}(\cdot)$ denote the canonicalization used by the evaluator, which removes surrounding whitespace, ignores letter case, normalizes numerical representations, and additionally removes internal spaces from set and sequence elements. The six validation units are implemented as follows:

- **Numerical validation.** The parsed value and ground truth are converted to finite scalars and compared under the instance-specific absolute tolerance τ:

$$v_{\text{num}} = \mathbb{1}[\,|\hat{y} - y| \le \tau\,]\,. \tag{3}$$

  When $\tau = 0$, exact equality is required. A missing, non-numerical, non-finite, or otherwise invalid value receives zero.

- **Categorical validation.** Discrete labels, choices, regions, or event types are scored by exact equality after canonicalization:

$$v_{\text{cat}} = \mathbb{1}[\mathcal{C}(\hat{y}) = \mathcal{C}(y)]\,. \tag{4}$$

- **Set validation.** Let $\hat{S}_k$ be the canonicalized set formed from the first k predicted elements when `top_k` is specified, and let S be the reference set. Exact matching uses

$$v_{\text{set}}^{\text{exact}} = \mathbb{1}\left[\hat{S}_k = S\right]. \tag{5}$$

  For element-level partial credit, the implemented score is

$$v_{\text{set}}^{\text{partial}} = \min\left(1, \rho\,|\hat{S}_k \cap S|\right), \tag{6}$$

where ρ is the configured score per matched element and defaults to $1/|S|$. Thus, the order of reported elements does not affect the score.

- **Sequence validation.** The evaluator supports position-wise, exact-order, and weighted partial-order matching. Position-wise matching canonicalizes the elements and assigns partial credit at each reference position:

$$v_{\text{seq}}^{\text{pos}} = \frac{1}{L} \sum_{j=1}^{L} \mathbb{1}[\hat{q}_j = q_j] \,, \tag{7}$$

where L is the reference length and a missing predicted position is counted as incorrect. For exact-order and weighted partial-order matching, both sequences are additionally deduplicated while retaining the first occurrence and optionally truncated to `top_k`. Exact-order matching assigns one only when the resulting sequences are identical. For weighted partial order, let O be the elements shared by the two sequences, $p(x)$ and $\hat{p}(x)$ their reference and predicted positions, δ the allowed order slip, and w_j the configured position weights, which default to $1/j$. The evaluator computes

$$v_{\text{seq}}^{\text{weighted}} = \mathbb{1}[|O| \geq m] \, \frac{|O|}{L} \, \frac{\sum_{x \in O} w_{p(x)} \max\left(0, 1 - \frac{|p(x) - \hat{p}(x)|}{\delta + 1}\right)}{\sum_{j=1}^{L} w_j}, \tag{8}$$

where m is the required minimum overlap.

- **Semantic validation.** The selected parsed field or complete report R and a task-specific judge prompt J are passed to the Semantic Judge:

$$v_{\text{sem}} = \text{clip}_{[0,1]}(\text{Judge}(R, J)) \,. \tag{9}$$

The judge output may be a normalized score or a Boolean `status`/`passed` decision. The rubric specifies the required conclusion, reference evidence, and conditions under which unsupported or inconsistent claims should fail.

- **Artifact validation.** A reported path must resolve inside the isolated instance workspace, exist, and satisfy any required filename constraint. Missing or invalid paths receive zero. For a signal artifact with field-level checks $z_\ell \in \{0, 1\}$ and configured weights a_ℓ, the score is

$$v_{\text{art}} = \frac{\sum_\ell a_\ell z_\ell}{\sum_\ell a_\ell}. \tag{10}$$

The available checks cover array shape, channel count and order, duration, sampling rate, signal RMS, referencing, and passband. A non-empty file receives one when no field-level check is configured. For image artifacts, the score is either a non-empty-file check or the normalized output of a VLM Judge under an instance-specific visual rubric.

The validation units remain independent and may be combined within an instance. Each unit returns a normalized score before the task-specific weights in the *Validation Configuration* are applied, as described in Appendix 3.2.3. This separation allows the benchmark to evaluate numerical accuracy, discrete decisions, structured outputs, scientific interpretation, and deliverable artifacts without reducing heterogeneous EEG workflows to a single output format.

D Experimental Protocol and Implementation Details

We provide the implementation details of the BrainBench evaluation protocol. In this paper, the evaluated models include the Qwen3.5 and Qwen3.7 series (Qwen Team, 2026a;b;c), GLM-5 (Zeng et al., 2026), DeepSeek-V4-Flash (Xu et al., 2026b), Kimi K2.5 (Team et al., 2026), MiniMax-M2.5 (Chen et al., 2026b), GPT-5.6 Luna, GPT-5.6 Terra and GPT-5.6 Sol (OpenAI, 2026), Gemini 3.6 Flash (Google DeepMind, 2026), and Claude Opus 5 (Anthropic, 2026). Appendix D.1 presents the end-to-end black-box workflow, from instance dispatch and isolated execution to output validation and scoring. Appendix D.2 and D.3 describe the tool-mediated BrainAgent workflow and the autonomous code-execution protocol of CodeAct, respectively. Appendix D.4 specifies the containerized runtime shared by both paradigms, including environment isolation, resource constraints, infrastructure configuration, and the policy for rerunning infrastructure-induced failures.

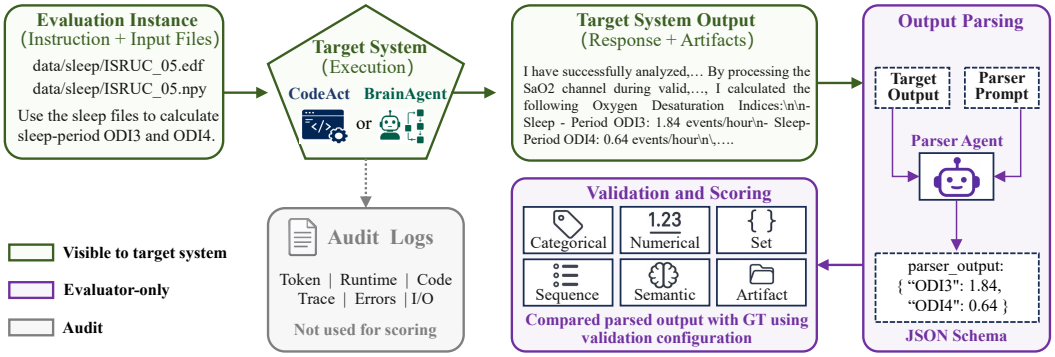


Figure 7: **End-to-end black-box evaluation protocol of BrainBench.** Each evaluation instance exposes only its instruction and input files to the target system, which executes the task through either CodeAct or BrainAgent in an isolated container. The resulting free-form report and requested artifacts are returned to the evaluator. A Parser Agent converts explicitly reported information into an instance-specific JSON schema, after which the parsed fields, original report, and generated artifacts are assessed by the applicable validation units and aggregated into the final instance score. Green components are visible to the target system, purple components remain evaluator-only, and gray components denote audit information that is recorded but not used for scoring.

D.1 Details of the Black-Box Evaluation Protocol

Figure 7 illustrates the end-to-end black-box evaluation protocol and distinguishes information visible to the target system from evaluator-only references and audit records. For each `instance`, the target system receives only the natural-language instruction and the associated input files. The ground truth, Parser Prompt, Validation Configuration, and scoring implementation remain hidden on the evaluator side throughout execution. The same instance is dispatched to either BrainAgent or CodeAct through a unified input–output interface and executed in an isolated, instance-specific container. Within this environment, the target system may inspect the supplied files, perform intermediate analyses, and generate the requested artifacts. Only the final natural-language report and retained artifacts are passed to the scoring pipeline, ensuring that both execution paradigms are evaluated under identical instructions, data, and output requirements. After execution, the final report and the instance-specific Parser Prompt are provided to the Parser Agent, which extracts explicitly reported information into a predefined JSON schema. The Parser Agent performs output alignment only: it has no access to the ground truth and does not assess scientific correctness, revise erroneous answers, or infer missing results. The extracted fields are then routed to the applicable numerical, categorical, set, and sequence validation units, while the original report is retained for semantic validation and the generated files are examined through artifact validation. Each validation unit applies the references, tolerances, matching rules, and weights specified in the instance-level Validation Configuration. The resulting unit scores are combined to produce the final instance score.

Execution information is recorded separately for auditing and error analysis. These records include token usage, runtime, generated code, tool and execution traces, intermediate inputs and outputs, and error messages, none of which directly contributes to the benchmark score. Failures attributable to the target system or execution paradigm—such as invalid code, incorrect tool use, incomplete reports, or malformed artifacts—are retained as evaluation outcomes. An instance is rerun only when execution is interrupted by an independently verified infrastructure failure, such as an external API communication error or container initialization failure. The container constraints and failure-handling policy are detailed in Appendix D.4.

D.2 BrainAgent Architecture and Execution Details

D.2.1 Architecture and Benchmark Adaptation

BrainAgent is a multi-agent framework for brain-signal analysis in which a Supervisor Agent interprets each request, decomposes it into analytical objectives, constructs an execution queue, and

delegates these objectives to specialized subagents (Zhou et al., 2026b). Each subagent encapsulates domain-specific reasoning and tool use, while the supervisor maintains a unified user-facing interface and synthesizes the subagent outputs into a final report. Subagents and their associated tools can be registered dynamically, allowing new analytical domains to be incorporated without modifying the supervisory control layer. This hierarchical architecture provides a natural basis for BrainBench, whose four subsets encompass heterogeneous analytical capabilities while sharing a common target-system interface. We preserve the original supervisor–subagent organization and adapt BrainAgent in three respects: the tool layer, the planning mechanism, and the representation of intermediate state.

First, we broaden the analytical tool layer to cover the capability space evaluated by the benchmark. **The resulting toolset is deliberately capability-oriented rather than task-specific:** each tool encapsulates a reusable analytical operation, such as data inspection, signal preparation, temporal segmentation, feature extraction, statistical aggregation, or artifact generation. No tool encodes the solution, task-specific parameters, reference answer, or decision rule of any particular `task` or `instance`. Successful execution therefore still requires the evaluated LLM to interpret the instruction, select and configure appropriate tools, integrate intermediate evidence, and formulate a scientifically grounded conclusion.

Second, to better accommodate complex and long-horizon workflows, **we extend BrainAgent beyond its original single-pass planning scheme**, in which a subagent planned the complete workflow once before executing it sequentially, **to an iterative, state-aware planning mechanism.** The Supervisor Agent first dispatches the complete analytical objective to an appropriate subagent. At each planning round, the subagent receives a compact representation of the current state and selects the maximal executable segment of its remaining plan. The tools within that segment are invoked sequentially, and their outputs are registered under stable identifiers before the next planning round begins. This segmented execution strategy allows subsequent decisions to incorporate observations obtained during execution—such as discovered channel names, computed features, or newly generated labels—without requiring the full workflow to be fixed in advance. Once the accumulated evidence is judged sufficient to satisfy the instruction, the subagent produces an analytical report, which is returned to the Supervisor Agent for final response synthesis.

Third, to support analyses spanning multiple files, subjects, and signal views, **we replace BrainAgent's original flat intermediate-variable representation with a hierarchical** `shared_state`. The state is accessible to the Supervisor Agent and active subagents and organizes intermediate objects into separate branches for raw `recordings`, derived signal `views`, temporal `labels`, detected `events`, structured analysis `results`, and generated `artifacts`. A dedicated `planning` branch records executed-tool counts and recent tool signatures, while internal counters assign collision-free identifiers to newly created objects. Tools exchange references to these registered objects rather than repeatedly serializing complete signals. Before each planning round, BrainAgent projects the hierarchy into a compact, identifier-based summary, enabling the LLM to retrieve prior outputs, coordinate dependencies across data sources, and trace the evidence underlying its conclusions.

D.2.2 Capability-Oriented Toolset

BrainAgent comprises three domain subagents—`SleepAgent`, `NeurocogAgent`, and `PhysiolAgent`—for Sleep Assessment, Neurocognitive Assessment, and Physiological Integration, respectively. Each combines a subset-specific extension with a shared common toolset implementing the reusable EEG operations covered by Foundational Analysis. Table 8 lists the complete toolsets and detailed interfaces are provided with the released implementation.

Table 8: Complete capability-oriented toolset used by BrainAgent.

Foundational Analysis Tools	**Primary capability**
`RecordingLoader`	Loads an EEG or PSG recording and registers its native metadata and channel names.
`ChannelInspector`	Classifies channels by signal modality and EEG region while preserving exact source names.
`SignalViewBuilder`	Constructs a named signal view with explicit channels, time window, filtering, referencing, resampling, and epoch settings.

`DataExporter`	Exports recordings, views, or arrays as EDF, FIF, NPY, or CSV artifacts.
`FigureExporter`	Produces channel-aware PSD, heatmap, hypnogram, spectrogram, and raw-trace figures.
`SpectralFeatureAnalyzer`	Computes band power, relative power, band ratios, dominant bands, alpha peak frequency, and spectral edge frequency.
`TimeDomainFeatureAnalyzer`	Computes variance, skewness, kurtosis, RMS, Hjorth parameters, and global field power.
`ComplexityFeatureAnalyzer`	Computes nonlinear EEG complexity measures, including sample entropy.
`ConnectivityAnalyzer`	Computes inter-channel connectivity using correlation, phase-locking value, or weighted phase-lag index.
`ArtifactQualityAnalyzer`	Assesses missing values, flatlines, excessive amplitude, drift, and channel-level signal quality.
`FeatureRankerAggregator`	Ranks or aggregates channels, regions, bands, windows, and prior structured results.

Sleep Assessment Tools	**Primary capability**
`SleepLabelLoader`	Loads sleep-stage annotations and normalizes them to W, N1, N2, N3, and R with epoch timing.
`SleepArchitectureAnalyzer`	Computes sleep onset latency, total sleep time, time in bed, sleep efficiency, WASO, REM latency, and stage proportions.
`SleepBoutTransitionAnalyzer`	Analyzes stage bouts, transitions, awakenings, longest episodes, and fragmentation.
`SleepStageEstimator`	Estimates sleep stages using an optional learned backend and a transparent rule-based fallback.
`SleepSpectralSegmentAnalyzer`	Quantifies sleep-related band power and sigma activity within an explicitly defined EEG segment.
`EMGActivityAnalyzer`	Quantifies chin or other EMG activity, stage-dependent tone, and a REM-atonia proxy.
`EOGActivityAnalyzer`	Quantifies slow- and rapid-eye-movement activity from explicitly selected EOG channels and windows.
`ECGHeartRateAnalyzer`	Estimates heart-rate statistics from explicitly selected ECG channels.
`SleepArousalAnalyzer`	Detects rule-based EEG arousal-like events using robust high-frequency envelope thresholds.
`RespiratoryEventAnalyzer`	Detects apnea, hypopnea, apnea subtype, and RERA-like events from respiratory, oximetry, and arousal evidence.
`SleepEventIndexAnalyzer`	Computes event counts and rates by total sleep time and sleep stage.
`SleepEventTemporalAnalyzer`	Filters, ranks, and temporally aggregates sleep events and associates them with stage labels.
`OximetryAnalyzer`	Computes ODI3, ODI4, sleep/wake mean and minimum SpO_2, T90, and T80.
`SleepMicroEventAnalyzer`	Detects sleep spindles, K-complexes, or slow waves for an explicitly specified event type.

Neurocognitive Assessment Tools	**Primary capability**
`FeatureStreamBuilder`	Constructs aligned EEG feature streams for spectral power, ratios, trends, correlations, rankings, and visualizations.
`FeatureStreamOperator`	Extracts, compares, transforms, and ranks previously computed numeric feature streams.
`HeartRateVariabilityAnalyzer`	Computes ECG-based heart-rate and heart-rate-variability measures for autonomic and fatigue assessment.
`OcularEventAnalyzer`	Detects and summarizes blink, prolonged-closure, and slow-eye-movement events from EOG signals.
`OcularFeatureStreamBuilder`	Constructs aligned EOG fatigue-related streams for comparison, correlation, and transition analysis.
`RespirationRhythmAnalyzer`	Extracts breathing rate, cycle variability, amplitude variability, and respiration-band power.

`HemisphericAsymmetryAnalyzer`	Computes left-right EEG asymmetry features and compares their temporal or baseline-dependent patterns.
`ContinuousLabelLoading`	Loads and aligns continuous numeric labels for feature correlation and ranking analyses.
`GroupContrastAnalyzer`	Compares channel or region groups and ranks discriminative differences and standardized effects.
`TemporalTrendAnalyzer`	Quantifies feature trajectories through slope, early-versus-late change, direction, and monotonicity.
`EmotionPolarityClassification`	Classifies EEG segments by positive, neutral, or negative emotional polarity.
`ValenceClassification`	Predicts binary EEG valence levels from multiband neural features.
`ArousalClassification`	Predicts binary EEG arousal levels from multiband neural features.
`LabelCorrelationAnalyzer`	Measures associations between aligned feature streams and numeric labels.
`FeatureCorrelationAnalyzer`	Computes and ranks pairwise associations across two aligned feature-stream sets.
`BehaviorLabelAnalyzer`	Computes accuracy, hit rate, miss rate, false-alarm rate, and response-time summaries from behavioral labels.
`ERPP300Analyzer`	Extracts stimulus-locked ERP and P300 features and supports target-versus-nontarget waveform comparison.
`WorkloadClassification`	Classifies cognitive workload using interpretable EEG engagement and workload markers.

Physiological Integration Tools	**Primary capability**
`MultirateRecordingLoader`	Loads multimodal physiological recordings and organizes their signals and metadata.
`NumericAnnotationLoader`	Loads numerical labels, ratings, masks, and event annotations.
`ChannelInspector`	Summarizes channel availability, signal modalities, sampling rates, units, and temporal coverage.
`EventTrialResolver`	Converts event markers into ordered trials, epochs, and analysis intervals.
`SleepStageFeatureAnalyzer`	Analyzes multimodal physiological features across sleep stages, epochs, and continuous stage bouts.
`MultimodalDataExporter`	Exports processed multimodal signals, numerical results, and validity information.
`MultimodalFigureComposer`	Creates aligned physiological time-series figures with event and artifact annotations.
`SignalQualityArtifactAnalyzer`	Evaluates signal quality and detects missing data, flatlines, clipping, faults, and physiological artifacts.
`TimeDomainFeatureAnalyzer`	Computes time-domain physiological features and compares their distributions across channels or intervals.
`SpectralFeatureAnalyzer`	Computes spectral power, relative power, frequency-band ratios, and related frequency-domain features.
`EEGPatternAnalyzer`	Quantifies EEG asymmetry, slowing, suppression, frequency-band balance, and regional activity patterns.
`EDAActivityAnalyzer`	Quantifies electrodermal level, variability, and phasic activity.
`EOGActivityAnalyzer`	Quantifies ocular activity, slow and rapid eye movements, and blink-related events.
`EyeTrackingAnalyzer`	Quantifies gaze validity, dispersion, spatial concentration, and pupil activity.
`RespiratoryActivityAnalyzer`	Quantifies respiratory rate, amplitude, variability, and breathing dynamics.
`CardiovascularAnalyzer`	Quantifies cardiac or pulse rate, amplitude, and variability.
`FNIRSResponseAnalyzer`	Derives and summarizes optical-density and hemoglobin-response features from fNIRS signals.
`BatchFeatureAnalyzer`	Extracts physiological features across repeated windows and compares their changes over time.
`ReferenceDistributionAnalyzer`	Evaluates target measurements relative to within-record reference distributions.

`EventResponseAnalyzer`	Analyzes event-related responses, condition contrasts, cross-modal associations, and response-gated effects.
`EventTrajectoryAnalyzer`	Constructs event-averaged response trajectories and estimates their temporal peaks.
`TrialGroupAnalyzer`	Extracts and compares trial-level features, reference responses, trial groups, and sequential selections.
`TrajectoryAssociationAnalyzer`	Analyzes correlations, delays, trends, and rankings across aligned multimodal trajectories.
`MultimodalAssociationAnalyzer`	Analyzes spatial relationships among EEG activity, gaze behavior, and regional multimodal responses.
`StatisticalModelAnalyzer`	Performs correlation, group comparison, regression, and physiological response–rating association analysis.

D.2.3 Illustrative BrainAgent Execution Traces

As an illustrative BrainAgent execution trace, we present the evaluation of Sleep Assessment instance `SA-26-Instance1` using Claude Opus 5. The corresponding instruction is:

> *Load the EEG data file at path `data/sleep/ISRUC_01.edf`. Then analyze the following three 1-minute segments: A = minute 274.5 to 275.5, B = minute 63.0 to 64.0, C = minute 437.0 to 438.0. For each segment, quantify the EOG/eye-movement activity strength using the EOG/eye-movement channels present in the current system state (inspect the loaded recording to determine the exact channel names; do not assume names). Compare the three segments and report the segment labels (A, B, C) in descending order of EOG activity strength, including the quantitative metric values used for the ranking. Additionally, determine the sleep stage of the segment with the strongest EOG activity and explicitly state whether that segment is REM sleep, with supporting evidence.*

Listing 3 presents a faithful condensed trace. It retains the supervisor dispatch, all six planning rounds, all 16 tool calls, and the final analytical output, while omitting repetitive success messages and serialized state snapshots. The colored statements summarize the reasoning that determined each planning segment; color distinguishes rounds only.

```
[Supervisor dispatch]
Target subagent: SleepAgent
Task: execute SA-26-Instance1 using data/sleep/ISRUC_01.edf

[Planning round 1]
No recording is registered.  Load the file and inspect its channels before selecting EOG
    inputs.
01  RecordingLoader
02  ChannelInspector

[Planning round 2]
The exact EOG channels are LOC-A2 and ROC-A1.  Build the three requested views and quantify
    EOG activity consistently.
03  SignalViewBuilder          [A: 16470--16530 s]
04  SignalViewBuilder          [B:  3780--3840 s]
05  SignalViewBuilder          [C: 26220--26280 s]
06  EOGActivityAnalyzer        [A]
07  EOGActivityAnalyzer        [B]
08  EOGActivityAnalyzer        [C]

[Planning round 3]
All EOG metrics rank A above B above C. Determine A's stage, collect spectral and EOG
    evidence, and formalize the ranking.
09  SleepStageEstimator        [A, 1-min view]
10  SleepSpectralSegmentAnalyzer [A]
11  EOGActivityAnalyzer        [A, 0.3--10 Hz check]
12  FeatureRankerAggregator    [A, B, C]

[Planning round 4]
A is staged as W/N1 and is delta-dominant.  Add chin-EMG evidence to test for the muscle
    atonia expected in REM.
13  SignalViewBuilder          [A, X1 channel]
14  EMGActivityAnalyzer        [A]

[Planning round 5]
The 1-min staging context is limited.  Re-estimate A within a 10-min EEG/EOG/EMG context
    before drawing the sleep-stage conclusion.
15  SignalViewBuilder          [16200--16800 s context]
```

```
16  SleepStageEstimator        [10-min context]

[Planning round 6]
The state now contains convergent EOG, staging, spectral, and EMG evidence.  No further tool
    call is needed; generate the report.

[Final output]
EOG activity ranking (descending): A > B > C

Segment  Window (s)        EOG RMS (V)  Line length  Derivative RMS
A        16470--16530       4.44e-06     2.16e-07     3.72e-07
B         3780--3840        1.81e-06     1.47e-07     1.97e-07
C        26220--26280       1.10e-06     9.22e-08     1.25e-07

The strongest segment is A, and it is not REM sleep. Its two 30-s epochs were initially
estimated as W and N1. In the 10-min context, the epochs covering A were N2 and W, whereas
the REM epochs occurred outside A. The EEG was delta-dominant (66.9% relative delta power),
and the X1 EMG retained measurable tone (RMS 4.68e-06 V), providing no evidence of REM atonia.

Conclusion: A > B > C; segment A is a wake/light-NREM transition, not REM sleep.
```

Listing 3: Condensed BrainAgent execution trace for SA-26-Instance1 with Claude Opus 5. Planning summaries are color-coded by round.

A second trace illustrates the evaluation of Foundational Analysis instance `FA-33-Instance15` on a SEED-V recording. The corresponding instruction is:

> *Load the EEG data file at path `data/core/SEED-V-05.cnt`. Select the standard EEG channels and compute, for each channel, alpha-band relative power and signal variance. Exclude channels with variance below $Q_1 - 1.5 \times \mathrm{IQR}$ or above $Q_3 + 1.5 \times \mathrm{IQR}$. Among the remaining channels, return those whose alpha relative power is greater than the global mean across the retained channels. Use Welch PSD estimation, with relative power defined as the target-band power divided by the total power from 0.5 to 45 Hz.*

Listing 4 retains all six planning rounds, all 12 successful tool calls, and the final analytical output. Serialized state snapshots, repetitive success messages, and the provider-side retry preceding the final planning round are omitted because they do not alter the analytical workflow. Following the benchmark's agent organization, the trace names the executing subagent `NeurocogAgent`.

```
[Supervisor dispatch]
Target subagent: NeurocogAgent
Task: execute FA-33-Instance15 using data/core/SEED-V-05.cnt

[Planning round 1]
No recording is registered.  Load the CNT file and inspect its channels before selecting the
    EEG inputs.
01  RecordingLoader
02  ChannelInspector

[Planning round 2]
The recording contains 64 EEG channels after excluding VEO and HEO. Compute channel-wise
    alpha relative power and variance over the same channel set.
03  SpectralFeatureAnalyzer   [64 EEG channels; alpha/[0.5--45] Hz]
04  TimeDomainFeatureAnalyzer [64 EEG channels; variance]

[Planning round 3]
Per-channel previews are truncated.  Apply IQR-based variance screening and expose the
    complete variance and alpha-power rankings.
05  ArtifactQualityAnalyzer   [variance outliers; 1.5 x IQR]
06  FeatureRankerAggregator   [variance; ascending]
07  FeatureRankerAggregator   [alpha relative power; descending]

[Planning round 4]
The upper variance tail is still hidden by the state summary.  Retrieve a descending variance
    ranking to identify the high-side outliers.
08  FeatureRankerAggregator   [variance; descending top 20]

[Planning round 5]
M1 and M2 are the two high-variance outliers.  Recompute the retained-channel mean and expose
    the remaining per-channel alpha values in manageable groups.
09  SpectralFeatureAnalyzer   [62 retained channels; global mean]
10  SpectralFeatureAnalyzer   [retained channel group A]
11  SpectralFeatureAnalyzer   [retained channel group B]
12  SpectralFeatureAnalyzer   [retained channel group C]

[Planning round 6]
The retained-channel mean is 8.9181%.  Compare every retained channel with this threshold and
    report the qualifying set.
```

```
[Final output]
Recording: 66 channels at 1000 Hz; 64 channels identified as EEG.
Variance-IQR exclusions: M2 (7.782e-06), M1 (3.405e-07).
Retained EEG channels: 62.
Global mean alpha relative power over retained channels: 8.9181%.

Qualified channels (alpha relative power > 8.9181%; 32 reported):
C3, C1, C2, C4, TP7, CP5, CP3, CP1, CP2, CP4, CP6, TP8,
P7, P5, P3, P1, PZ, P2, P4, P6, P8, PO7, PO5, PO3, POZ,
PO4, PO6, PO8, CB1, O1, OZ, O2.
```

Listing 4: Condensed BrainAgent execution trace for FA-33-Instance15 with Claude Opus 5. Planning summaries are color-coded by round.

D.3 CodeAct Execution Protocol

D.3.1 Interaction Loop

CodeAct provides the autonomous code-execution counterpart to the structured BrainAgent workflow. It receives the same instance-level instruction and permitted input files through the unified interface described in Section D.1, but it is not given the tool inventory, structured intermediate state, or task-specific helper functions. Instead, the target LLM independently selects the analysis method, Python libraries, preprocessing operations, intermediate computations, and artifact-generation procedure needed to complete the instruction.

For each `instance`, CodeAct initializes an instance-scoped persistent Python kernel. At every interaction round, the model returns one of two actions: an `<execute>` block containing Python code or a `<solution>` block containing the final report. Code inside an `<execute>` block is evaluated in the persistent kernel, allowing loaded recordings, intermediate variables, and generated files to be reused across subsequent rounds of the same instance. Textual stream output, `text/plain` execution results, and exception tracebacks are collected as an *Observation* and returned to the model. The model may then inspect the result, correct erroneous code, revise its analytical strategy, or continue the analysis. This loop terminates when the model emits a valid `<solution>` block or when an execution-control limit is reached. Only the content of `<solution>` is treated as the target system's final response. Intermediate code, printed values, and error messages are retained as execution information but are not interpreted as final answers and do not directly contribute to the benchmark score. The final report and any reported artifacts are subsequently processed by the same external Parser Agent and validation pipeline used for BrainAgent.

D.3.2 Prompt and Execution Contract

Listing 5 presents the complete system prompt used by CodeAct. The prompt defines the executable action format, the persistent Python environment, the observation feedback mechanism, and the distinction between intermediate execution and the final response.

```
You are a helpful assistant assigned a problem-solving task. You have access to
an interactive Python environment to inspect data and calculate the answer.

Return exactly one action block per turn:
<execute>...</execute> or <solution>...</solution>.
Do not output plans, explanations, or text outside the selected block.

Then choose exactly one of these actions:

1) Execute Python code by enclosing it in <execute>...</execute>. The code will
run in a persistent Python kernel and the output will be returned as an
Observation. Top-level variables from earlier snippets remain available.
2) When the task is complete, provide the requested final report enclosed in
<solution>...</solution>. The text inside <solution> is returned to the user,
so it must follow the task's requested output format and contain all results.

Use code to inspect the provided files instead of guessing. Paths named in the
task are accessible from the current workspace. To conserve context, never print
an entire long signal, label sequence, dataframe, or file.

---
Example task:
The file input/HR.npy contains a 1 Hz heart-rate signal in BPM. Calculate how
```

```
many seconds are in the inclusive range 60 to 100 BPM and return JSON with the
key time_in_range.

Assistant:
<thought>I will load the signal and inspect its shape.</thought>
<execute>
import numpy as np
hr = np.load("input/HR.npy")
print(hr.shape)
</execute>

Observation:
(300,)

Assistant:
<thought>I will count samples in range; at 1 Hz the count equals seconds.</thought>
<execute>
time_in_range = float(np.sum((hr >= 60) & (hr <= 100)))
print(time_in_range)
</execute>

Observation:
240.0

Assistant:
<thought>The calculation is complete, so I will return the requested JSON.</thought>
<solution>
{"time_in_range": 240.0}
</solution>

---
The actual task follows in the user message.
```

Listing 5: System prompt used for the CodeAct execution protocol.

On the final permitted interaction round, the controller retains the preceding system prompt, instance query, execution actions, and observations, and appends an additional user message to force finalization. The appended message is shown verbatim in Listing 6.

```
The execution budget is exhausted. You must return the best available final
answer now inside <solution>...</solution>. Do not execute more code.
```

Listing 6: Verbatim finalization prompt appended on the last CodeAct interaction round.

The instance-specific user message is constructed separately from the system prompt. Before dispatch, the evaluator projects the permitted `data_path` and optional `label_path` into the isolated execution environment and rewrites them as paths under `/input/`. The user message then identifies these paths, presents the natural-language `instruction`, and appends any additional fields explicitly included in `agent_input`. The instruction is augmented with a runtime rule stating that input files under `/input/` are read-only and that requested artifacts must be saved under `/workspace/file_check/` or a relative path under `file_check/`.

The benchmark information boundary is enforced by the evaluator rather than encoded as additional task text in the system prompt. CodeAct receives only the prepared `agent_input` and the permitted input-file mounts. The ground truth, Parser Prompt, Validation Configuration, metric definitions, metric weights, and evaluator-side scores are not included in the model messages or execution workspace. The model also has no interactive mechanism for requesting additional files during execution. After CodeAct returns its final `<solution>`, the external Parser Agent extracts the required fields, and the evaluator applies the hidden validation units and scoring configuration.

Although the worked example includes `<thought>` blocks, the executable protocol recognizes only `<execute>` and `<solution>` as actions. Text outside these two blocks is neither executed nor treated as the final answer. Accordingly, intermediate Python output and observations are used only to support subsequent interaction rounds, whereas the content enclosed by `<solution>` constitutes the final response submitted to the evaluation pipeline.

D.3.3 Execution Controls

Table 9 reports the CodeAct-specific interaction controls. Execution output is returned to the model as a merged textual observation. This observation may contain standard-stream text, expression

results, textual display representations, or a Python traceback. When an execution succeeds without textual output, the runtime returns an explicit success message; when it exceeds the per-execution limit, a timeout observation is returned. These observations provide the model with an opportunity to diagnose and revise failed analyses, while the repeated-failure controls prevent unproductive execution loops. On the final permitted round, CodeAct is instructed to stop executing code and return the best available result inside `<solution>`. If the hard instance deadline or an unrecoverable infrastructure error occurs earlier, the instance terminates with a structured error rather than receiving evaluator feedback or a reference answer.

Table 9: CodeAct interaction and execution controls.

Control dimension	Budget	Operational rule
Interaction-round budget	20 rounds	Rounds 1–19 permit `<execute>`; round 20 is reserved for `<solution>`.
Execution-time budget	150 s / 720 s	Per Python execution / complete instance.
Response-length budget	2,048 / 1,024 tokens	Maximum length of non-final / final responses.
Observation history	8K chars / 3 pairs	Per-observation limit / recent execution–observation pairs retained verbatim.
Stall termination	3 consecutive events	Triggered by missing actions, identical code, or the same execution-error signature.
Decoding configuration	Thinking off / $T = 0.7$	Temperature is applied only when supported by the endpoint.

The fixed Python environment contains general numerical and scientific-computing packages, including NumPy, SciPy, pandas, Matplotlib, seaborn, and scikit-learn, together with EEG- and physiological-signal packages such as MNE, YASA, pyEDFlib, edfio, and WFDB. CodeAct may compose these libraries freely but cannot assume access to uninstalled task-specific software. Variables and temporary files persist across interaction rounds only within the current instance; different instances receive independent kernels, message histories, and workspaces. Requested artifacts remain available for evaluator-side validation before cleanup, while the retained audit record stores action types, token usage, elapsed time, code hashes, observation summaries, and termination status. CPU, memory, image, filesystem, and network isolation are described separately in Appendix D.4.

D.4 Containerized Runtime and Failure Handling

This section specifies the containerized execution environment used by BrainAgent and CodeAct and defines how infrastructure-induced interruptions are distinguished from target-system failures. Both paradigms operate through the same instance-level input–output interface, while their container configurations differ where required by their execution mechanisms.

D.4.1 Container Configuration

Each `instance` is executed in a fresh Docker container with an independent runtime state. Both paradigms use a versioned Python 3.9 scientific-computing environment and receive the same instruction and permitted input files. The containerization layer does not alter the expected report or artifact requirements of the instance. Table 10 summarizes the principal runtime settings.

The larger memory allocation for BrainAgent accommodates its multi-agent runtime, hierarchical intermediate state, and domain-oriented analytical toolset. CodeAct instead executes generated code in a smaller scientific-computing sandbox. Its execution container has no network access and receives no target-model credentials, whereas BrainAgent requires network access because model requests and tool-mediated planning are performed inside the container. Within each execution paradigm, the corresponding resource and time constraints are held fixed across evaluated models.

D.4.2 Failure Classification and Retry Policy

We distinguish verified infrastructure interruptions from failures attributable to the target system. Transient API or container-runtime failures are retried under a predefined policy; if recovery is

Table 10: Principal container configurations used for BrainAgent and CodeAct.

Configuration	BrainAgent	CodeAct
Runtime image	Versioned Python 3.9 scientific-computing image	Versioned Python 3.9 scientific-computing image
Execution backend	Supervisor–subagent workflow with controlled analytical tools	Persistent IPython kernel for model-generated Python code
CPU allocation	4 CPU cores per instance	4 CPU cores per instance
Memory limit	32 GB per instance	8 GB per instance
Network access	Enabled because target-model requests originate inside the BrainAgent container	Disabled for generated code; target-model requests are issued outside the execution container
Total instance deadline	720 seconds	720 seconds
Per-execution limit	Not separately constrained beyond the instance deadline	150 seconds for each Python execution

unsuccessful, the affected instance is rerun after the external condition is restored using the same instruction, inputs, model configuration, execution paradigm, and container constraints. The interrupted execution is excluded from score aggregation and replaced by the valid rerun. In contrast, failures arising from model planning, code generation, tool use, execution strategy, or incomplete outputs are retained as evaluation outcomes and are not rerun by the evaluator. Any self-correction performed by BrainAgent or CodeAct within their allotted execution budgets is considered part of the evaluated paradigm; once that budget is exhausted, the resulting report, missing output, or termination state is scored as produced. Table 11 summarizes the resulting policy described above.

Table 11: Failure classification and instance-level rerun policy.

Failure type	Representative examples	Rerun	Model outcome
Provider-side communication failure	Connection interruption, request timeout, rate limiting, or HTTP 5xx response	✓	✗
Provider access or service-state failure	Verified HTTP 400/401/402/403/404 caused by access, quota, routing, endpoint, or content-inspection restrictions	✓	✗
Container initialization failure	Docker image startup failure or Python-kernel initialization failure	✓	✗
Invalid generated code	Syntax errors, runtime exceptions, or invalid data-processing operations in CodeAct	✗	✓
Incorrect tool invocation	Inappropriate tool selection, invalid arguments, or unresolved tool dependencies in BrainAgent	✗	✓
Execution-budget exhaustion	Excessive computation, hard instance timeout, maximum interaction rounds, or repeated execution failures	✗	✓
Incomplete report or artifact	Missing final report, omitted requested result, absent artifact, or malformed deliverable	✗	✓

E Additional Experimental Results and Analyses

E.1 Overall Score Distributions

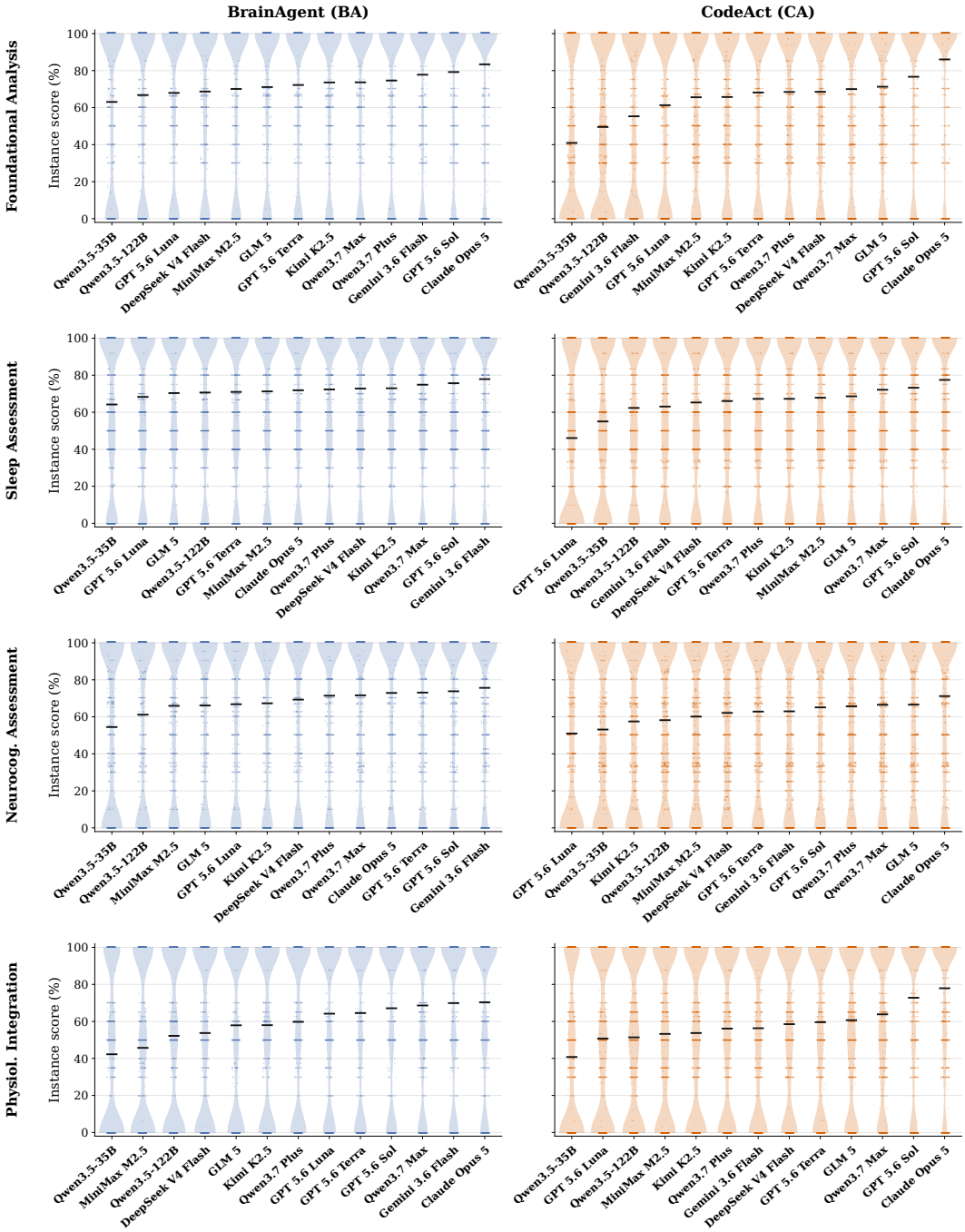


Figure 8: **Instance-level score distributions across models and execution paradigms.** Violin densities and translucent points show the unweighted normalized instance scores across four subsets. Black horizontal markers denote arithmetic means. Models are sorted independently within each panel by their mean instance score; task-difficulty weights are not applied.

Figure 8 reveals that aggregate performance masks substantial heterogeneity at the instance level across all four BrainBench subsets. Both execution paradigms exhibit broad score distributions,

indicating that strong average performance does not translate into uniformly reliable behavior across individual analyses. Compared with flexible LLM-driven coding, structured agentic execution generally raises the performance of weaker models and reduces between-model dispersion, making overall performance less sensitive to the choice of underlying LLM. This effect is particularly evident in domains where execution can be effectively organized through predefined analytical capabilities. CodeAct, in contrast, shows stronger model dependence: weaker models degrade more substantially, while stronger models can match or surpass the leading structured-agent configurations in some domains. Overall, structured workflows primarily improve robustness across model capabilities, whereas autonomous coding preserves greater performance variability together with a higher potential ceiling for sufficiently capable models.

E.2 Difficulty-Conditioned Effects of Execution Paradigms

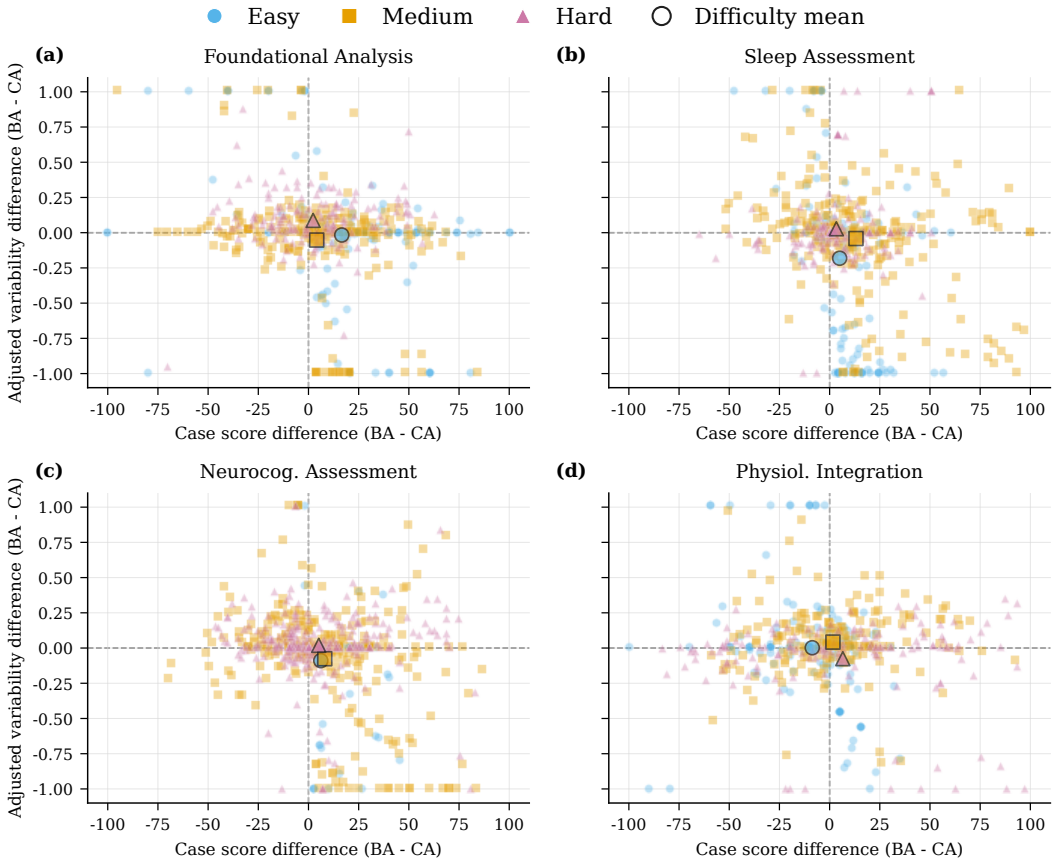


Figure 9: **Difficulty-conditioned performance–stability trade-off between BrainAgent (BA) and CodeAct (CA).** Results are shown across the four BrainBench subsets. Each translucent marker represents a paired model–task comparison, with color and shape denoting task difficulty; outlined markers indicate difficulty-level means. Positive horizontal values indicate higher task scores under BA, while negative vertical values indicate lower adjusted within-task variability under BA.

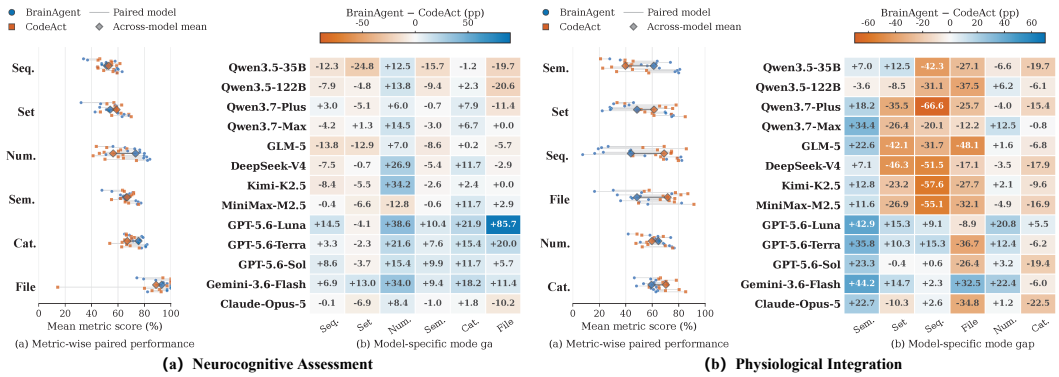
To quantify how the relative benefit of structured execution changes with difficulty, let $\bar{S} * p, d$ denote the task-macro score under paradigm $p \in \mathrm{BA}, \mathrm{CA}$ at difficulty $d \in \mathrm{E}, \mathrm{M}, \mathrm{H}$, and define the paradigm gap as $G_d = \bar{S} * \mathrm{BA}, d - \bar{S} * \mathrm{CA}, d$. The **Difficulty Advantage Index** (DAI) is

$$\mathrm{DAI} = \frac{(G * \mathrm{M} - G_{\mathrm{E}}) + (G_{\mathrm{H}} - G_{\mathrm{M}})}{2} = \frac{G_{\mathrm{H}} - G_{\mathrm{E}}}{2}. \tag{11}$$

DAI characterizes how the relative advantage of BrainAgent over CodeAct evolves with task difficulty, with positive and negative values indicating widening and narrowing advantages, respectively.

Figure 9 further examines how this difficulty-conditioned performance effect relates to within-task stability. Across the four subsets, the difficulty-level means reveal no universal trajectory for structured execution. In Foundational Analysis, the BrainAgent advantage weakens as difficulty increases, whereas Sleep Assessment shows its clearest benefit at intermediate difficulty and substantially less separation on the hardest tasks. Neurocognitive Assessment exhibits a comparatively stable advantage across difficulty levels, accompanied by generally favorable stability. In contrast, Physiological Integration displays an increasing benefit from structured execution as task difficulty rises, suggesting that explicit workflow organization becomes more valuable as multimodal coordination demands grow. The broad dispersion around these aggregate trends further highlights substantial heterogeneity across individual model–task combinations. Taken together, these results show that task difficulty does more than reduce overall performance: it changes how execution strategy affects both analytical effectiveness and stability. The benefit of structured agentic workflows is therefore not monotonic with difficulty, but depends on the nature of the analytical demands. **Structured execution can provide substantial support when complexity can be organized through domain-oriented workflows, whereas its advantage may diminish when the dominant bottleneck shifts toward long-horizon reasoning and evidence integration.**

E.3 Additional Validation-Unit-Level Analysis



(a) Neurocognitive Assessment **(b) Physiological Integration**

Figure 10: **Validation-unit-specific effects of execution paradigms.** Within each subset, paired markers show model-wise unweighted mean validation scores under BA and CA, diamonds indicate across-model means, and heatmaps report the corresponding Δ in percentage points.)

Figure 10 extends the validation-unit analysis to Neurocognitive Assessment and Physiological Integration. The overall pattern is consistent with the ***Finding 3***: BrainAgent retains a clear and broadly consistent advantage in numerical validation across models, further supporting the finding that structured agentic execution most reliably strengthens quantitative EEG analysis. In contrast, the remaining validation units exhibit substantially more heterogeneous paradigm effects, with the relative benefit of BrainAgent and CodeAct varying across models and analytical domains. This heterogeneity is particularly evident in semantic, categorical, set, sequence, and artifact validation, where neither execution paradigm shows a uniform advantage across the two subsets. Taken together with the Foundational Analysis and Sleep Assessment results, these findings indicate that the benefit of structured execution is selective rather than universal, with quantitative analysis emerging as its most consistent cross-domain strength.

E.4 Additional Within-Task Stability Analysis

For a given model, execution paradigm, and task, let $s_1, \ldots, s_n \in [0, 100]$ denote the normalized scores of its n instances. We quantify within-task variability using the *mean pairwise absolute difference* (MPAD):

$$\mathrm{MPAD} = \frac{2}{n(n-1)} \sum_{i<j} \frac{|s_i - s_j|}{100}. \tag{12}$$

MPAD ranges from 0 to 1, with lower values indicating more consistent performance across instances sharing the same analytical objective and higher values indicating greater within-task variability.

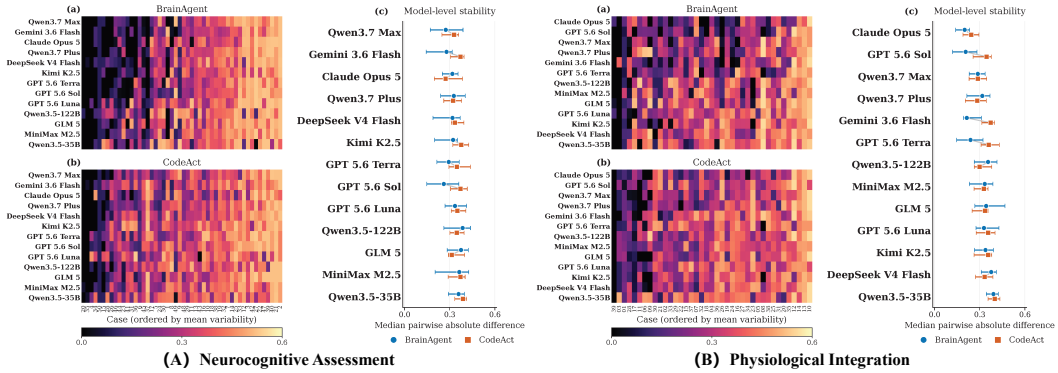


Figure 11: **Cross-instance stability within reusable tasks.** For each model–task pair, the heatmap shows the mean pairwise absolute difference among within-task normalized instance scores. Model-level panels report the median across tasks with task-bootstrap 95% confidence intervals.

Because MPAD measures consistency rather than correctness, it should be interpreted together with task-level performance. For model-level reporting, we take the median MPAD across tasks and estimate 95% confidence intervals using 10,000 task-level bootstrap resamples.

Figure 11 extends the within-task stability analysis to Neurocognitive Assessment and Physiological Integration. Consistent with the Foundational Analysis and Sleep Assessment results in the Sec. 4.4, structured agentic execution generally exhibits lower within-task variability than flexible LLM-driven coding across both additional subsets. Although individual model–task pairs remain heterogeneous and occasional reversals occur, the model-level distributions show an overall shift toward lower MPAD under BrainAgent. Importantly, this pattern persists across substantially different analytical settings, from neurocognitive inference to multimodal physiological analysis, suggesting that the stability benefit of structured workflows is not confined to a particular EEG domain. Taken together with the ***Finding 4***, these findings further support that structured agentic workflows improve the cross-context reliability of EEG understanding by promoting more consistent execution across recordings that share the same analytical objective.

E.5 Effects of Reasoning Effort and Token Efficiency

Figure 12 further examines how increasing reasoning effort affects test-time computation across all four BrainBench subsets. BrainAgent and CodeAct exhibit markedly different token-scaling behaviors: under BrainAgent, target-token usage remains comparatively stable across reasoning-effort levels, whereas CodeAct shows a substantially stronger increase in token consumption as reasoning effort grows. This difference reflects how the two execution paradigms organize additional inference-time computation, with structured agentic workflows constraining reasoning within a relatively stable execution process and flexible coding allowing additional reasoning budget to expand autonomous reasoning and code-generation trajectories more freely. When considered together with the corresponding performance trends, **enabling reasoning provides the dominant performance gain, while further scaling generally requires greater computation for smaller and increasingly task-dependent improvements.** This effect is particularly evident under CodeAct, where higher reasoning effort can continue to improve performance in several settings but is accompanied by pronounced growth in token usage. Reasoning efficiency is therefore strongly execution-dependent: structured workflows stabilize computational demand, whereas flexible coding exposes a clearer trade-off between reasoning budget and analytical performance. These results provide a computational perspective on ***Finding 5***, further supporting that **LLM reasoning is an effective but bounded amplifier of EEG understanding rather than a capability that improves proportionally with inference-time computation.**

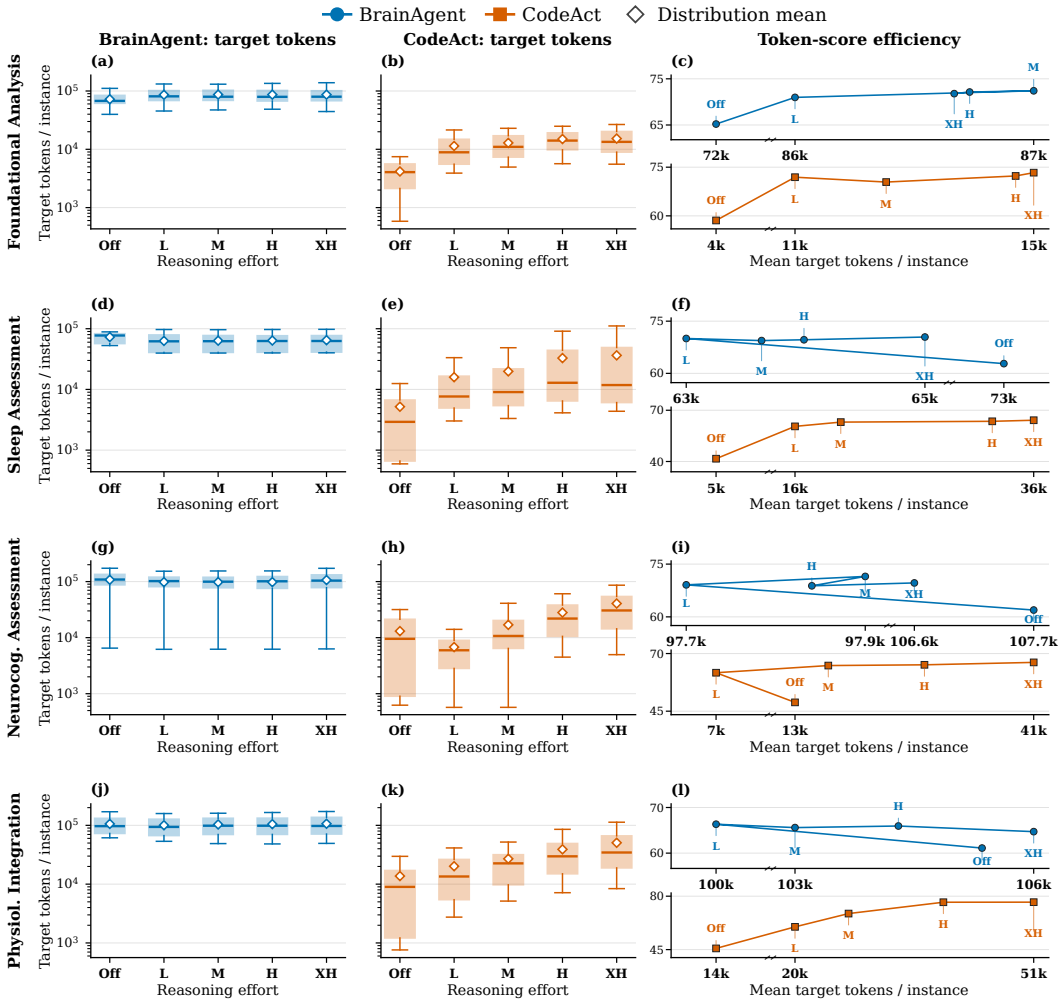


Figure 12: **Reasoning-effort scaling and token–score efficiency across execution paradigms.** Results are shown for all four BrainBench subsets. The left and middle columns report the distributions of target-token usage per instance across reasoning-effort levels for BrainAgent and CodeAct, respectively, with diamonds indicating distribution means. The right column relates mean target-token usage to the corresponding EEG understanding score, illustrating how additional inference-time computation translates into performance under each execution paradigm.

E.6 Model-Family Structure in Task-Level Performance

Figure 13 examines the consistency of task-level performance profiles both within and across model families. Across both BrainAgent and CodeAct, variants from the same model family exhibit consistently strong agreement, indicating that tasks that are relatively easy or difficult for one family member tend to retain similar relative difficulty for another. Cross-family comparisons remain positively correlated but show noticeably weaker and more variable agreement, suggesting that different model families share a common task structure while exhibiting more distinct capability profiles. This contrast persists across execution paradigms, although the strength of agreement can vary with how the model is executed. Overall, task-level EEG understanding exhibits a clear model-family structure: **capability profiles are more strongly preserved within families than across families, while execution paradigms further modulate how these underlying capabilities are expressed.** At the same time, deviations from the identity line indicate that stronger models do not improve uniformly across all tasks, reinforcing that model scaling changes the magnitude of performance without simply preserving a fixed task-wise advantage.

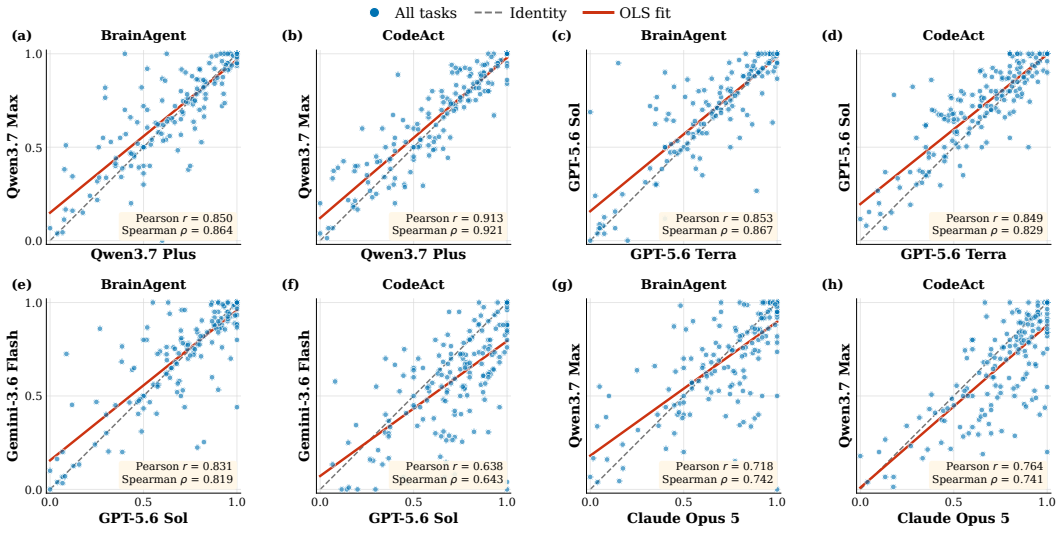
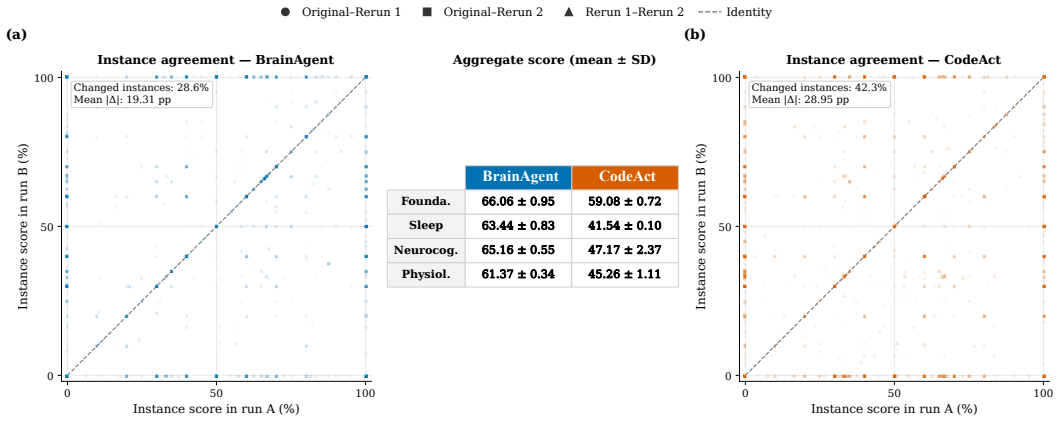


Figure 13: **Model-family consistency of task-level EEG understanding.** The top row evaluates consistency between model variants from the same family, whereas the bottom row compares strong models from different families. Each point represents the unweighted mean instance score of the same task for a pair of models. The gray dashed line denotes identity, and the red line shows the ordinary least-squares fit. Insets report Pearson and Spearman correlations computed over the 172 pooled tasks. Panels (a)–(d) therefore characterize within-family consistency, while panels (e)–(h) provide a cross-family reference for comparison.

E.7 Consistency of Results Across Repeated Runs



	BrainAgent	CodeAct
Founda.	66.06 ± 0.95	59.08 ± 0.72
Sleep	63.44 ± 0.83	41.54 ± 0.10
Neurocog.	65.16 ± 0.55	47.17 ± 2.37
Physiol.	61.37 ± 0.34	45.26 ± 1.11

Figure 14: **Consistency across repeated runs.** Pairwise instance-level agreement across three runs under (a) BrainAgent and (b) CodeAct. The central table reports aggregate benchmark scores as mean ± SD across the three runs.

Repeated runs produced observable instance-level variation, but BrainAgent showed greater consistency than CodeAct, with fewer changed instances (28.6% vs. 42.3%) and a lower mean absolute score difference (19.31 vs. 28.95 percentage points). Nevertheless, aggregate benchmark scores remained closely clustered across the three runs, with standard deviations of 0.10–2.37 points across subsets and execution modes. These results indicate that aggregation over diverse benchmark instances attenuates run-level sampling variability, while BrainAgent's constrained workflow further improves instance-level reproducibility.

E.8 Token Usage and Execution Efficiency

Figure 15 compares total token consumption under the two execution paradigms. Across both subsets and all evaluated models, BrainAgent consistently uses more tokens than CodeAct, reflecting the additional communication and context required for agent coordination, tool selection, and intermediate result synthesis. CodeAct is substantially more token-efficient because much of the analytical computation is delegated directly to executable code. Nevertheless, token consumption is not monotonically associated with benchmark performance: several highly ranked CodeAct configurations achieve strong scores with comparatively modest token budgets, whereas larger token usage does not necessarily yield a higher rank. These results reveal a clear performance–efficiency trade-off between the two paradigms: structured agentic execution generally provides stronger and more reliable EEG analysis at higher token overhead, while autonomous coding reduces token consumption but exhibits less consistent performance. As token accounting and pricing differ across model providers, the comparison should be interpreted as execution overhead rather than a direct estimate of monetary cost.

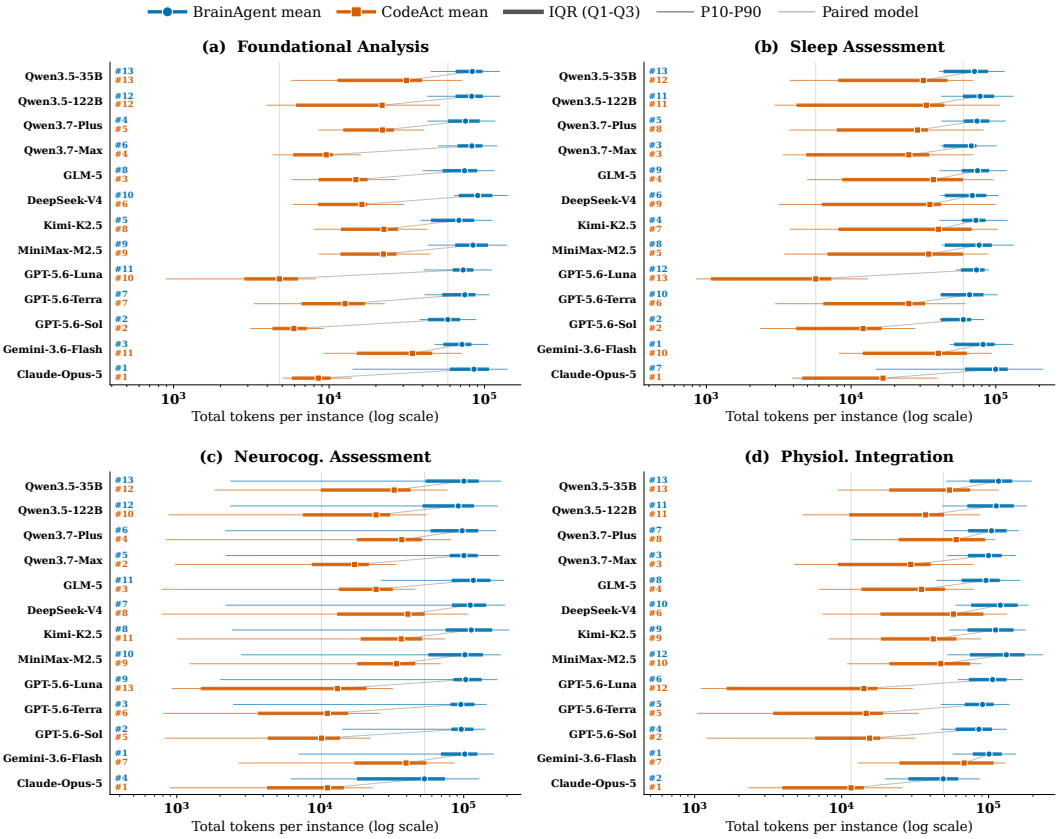


Figure 15: **Token usage and execution efficiency.** Total tokens per instance are compared between BrainAgent and CodeAct across four subsets. Markers denote means, thick bars show the interquartile range, and thin whiskers indicate the 10th–90th percentiles. Gray lines connect the two execution paradigms for the same model, and the colored annotations report the corresponding performance ranks. The horizontal axis uses a logarithmic scale.

E.9 Parser Stability and Scoring Robustness

The Parser Agent separates free-form reporting from structured validation, but extraction errors could introduce evaluator-side noise into the final scores. We therefore audit its outputs on the complete Sleep Assessment subset across ten evaluation runs. For each of the 1,025 instances in a run, an independent auditor compares the extracted JSON with the parser prompt and original report, judging only whether the extraction faithfully represents the explicitly reported answer without assessing its scientific correctness.

Table 12: Parser-audit results on Sleep Assessment across ten complete evaluation runs. Agree and Agreement are reported as mean $\pm$ standard deviation across runs.

Audited	**Agree**	**Agreement (%)**
1,025	$1{,}015.8 \pm 8.9$	99.10 ± 0.87

Across 10,250 audited extractions, 10,158 agree with the independent audit, yielding an aggregate agreement of 99.10%. The consistently high run-level agreement indicates that converting flexible reports into structured fields contributes limited evaluator-side noise in the Sleep Assessment results and supports the separation between EEG understanding and rigid format compliance. As this audit is restricted to one subset, it should be interpreted as a focused robustness check rather than a benchmark-wide estimate of parser accuracy.

E.10 Representative Failure Cases Across BrainBench

To complement the aggregate evaluation, we present representative failure cases across the four BrainBench subsets under both BrainAgent and CodeAct. All cases use Claude Opus 5 to control for differences in underlying model capability, allowing the analysis to focus on how analytical demands and execution paradigms shape failure behavior. Rather than treating failures as isolated scoring errors, we examine where the analytical pipeline breaks and how the error propagates to the final interpretation. Errors are highlighted in bold red within each listing for clarity.

BrainAgent failure case1. This trace illustrates the evaluation of Physiological Integration instance `PI-27-Instance26` on a SEED-VII recording. The corresponding instruction is:

> *Analyze the third chronologically ordered event_code=1–event_code=2 interval. Use the following EEG ROIs: { "frontal": ["FP1", "FP2", "F3", "F4"], "posterior": ["P3", "P4", "O1", "O2"] }. Compare mean alpha power in [8, 13) Hz between the two ROIs. Report the ROI with lower alpha power as lower_alpha_roi. Report the proportion of finite gaze samples in the central 50% × 50% screen region as central_gaze_fraction. Briefly state which ROI has lower alpha power, whether gaze is centrally concentrated or relatively peripheral using the 0.5 threshold, and describe the two findings together as a descriptive EEG–gaze spatial profile. Do not interpret lower alpha as a direct measure of engagement or imply causal EEG–gaze coupling. Report only lower_alpha_roi and central_gaze_fraction, together with the brief descriptive interpretation.*

```
[Supervisor dispatch]
Target subagent: MultiAgent
Task: analyze the third event-code 1--2 interval using
SEEDVII EEG and eye-tracking recordings

[Planning round 1]
No data are registered.  Load the EEG, eye-tracking recording, and event annotations while
     preserving their native sampling rates.  The initial eye-tracking modality hint is
     rejected, so retry the EDF without that hint.
01  MultirateRecordingLoader  [EEG CNT, 1000 Hz]
02  MultirateRecordingLoader  [eye-tracking EDF, 250 Hz]
03  NumericAnnotationLoader   [40 event rows]

[Planning round 2]
The event table contains alternating codes 1 and 2.  Resolve the intervals and select the
     third interval, [483.170, 672.749) s.  Compute alpha power for both EEG ROIs and obtain
     an initial gaze summary.
04  EventTrialResolver        [codes 1--2, third interval]
05  EEGPatternAnalyzer        [alpha 8--13 Hz, frontal/posterior]
06  EyeTrackingAnalyzer       [GazePointX, GazePointY]

[Planning round 3]
The EEG result is frontal=-25.2661 and posterior=-26.0357, so posterior has lower alpha.  The
     initial gaze call returned validity and dispersion features rather than central
     fraction; request central_fraction explicitly.
07  EyeTrackingAnalyzer       [feature=central_fraction, fraction=0.5]

[Planning round 4]
The second gaze result contains valid_ratio=18447/47395=0.3892.  Interpret this value as
     central_gaze_fraction; because it is below 0.5, classify gaze as relatively peripheral
     and generate the report.  ERROR: valid_ratio is not central_gaze_fraction; usable=false
     and insufficient_valid_eye_samples are ignored.

[Final output]
lower_alpha_roi: posterior
central_gaze_fraction: 0.3892

The posterior ROI exhibits lower mean alpha power (-26.04 dB) than
the frontal ROI (-25.27 dB). With a central gaze fraction of 0.39,
gaze is relatively peripheral rather than centrally concentrated.
Together, these findings provide a descriptive EEG--gaze spatial
profile without implying engagement or causal EEG--gaze coupling.
```

Listing 7: Condensed BrainAgent execution trace for PI-27-Instance26 with Claude Opus 5. Planning summaries are color-coded by round.

Error analysis. The EEG branch is correct: BrainAgent selects the intended interval and correctly identifies the posterior ROI as having lower alpha power. The failure occurs in the eye-tracking branch, where `EyeTrackingAnalyzer` does not compute the requested `central_fraction` because the required screen dimensions are missing. BrainAgent nevertheless interprets the returned

`valid_ratio=0.3892` as the central-gaze fraction and ignores the associated validity warning, leading to the opposite gaze interpretation. The correct central-gaze fraction is approximately 0.9360, indicating centrally concentrated gaze. This case therefore reflects a *tool-output interpretation error*. The parser and evaluator operate correctly. **Score: 35/100.** The lower-alpha ROI metric scores 35/35, while the central-gaze fraction and semantic interpretation metrics score 0/35 and 0/30, respectively.

BrainAgent failure case2. This trace illustrates the evaluation of Sleep Assessment instance `SA-32-Instance01` on an ISRUC overnight PSG recording. The corresponding instruction is:

> *Please use the provided sleep EEG/PSG file and paired sleep-stage label file to calculate the whole-night Apnea-Hypopnea Index (AHI). Count all apnea events and all hypopnea events across the full recording from the sleep signal. Compute total sleep time (TST) from the label file as the number of non-Wake epochs (N1, N2, N3, or R/REM) multiplied by 30 seconds, converted to hours. AHI is defined as (apnea_count + hypopnea_count) divided by TST in hours, reported as events per hour of total sleep time. Respiratory channels for ISRUC: X5 = snore signal; X6 = Flow-1, pressure-based airflow; DC3 = Flow-2, second airflow channel; X7 and X8 = abdominal respiratory-effort channels. Please clearly report the final AHI value.*

```
[Supervisor dispatch]
Target subagent: SleepAgent
Task: compute whole-night AHI from ISRUC PSG respiratory
signals and paired sleep-stage labels

[Planning round 1]
No data are registered.  Load the PSG recording and sleep-stage labels.  Detect apnea and
    hypopnea events using X6/DC3 airflow and X7/X8 respiratory-effort channels, and compute
    TST from the non-Wake sleep-stage epochs.
01  RecordingLoader             [19 channels, 200 Hz, 7.33 h]
02  SleepLabelLoader            [880 epochs, 30 s each]
03  RespiratoryEventAnalyzer    [X6, DC3 airflow; X7, X8 effort]
04  SleepArchitectureAnalyzer   [full-night labels]

Respiratory events:
apnea     = 130
hypopnea  = 334
total     = 464

Sleep-stage counts:
W=264, N1=73, N2=194, N3=231, R=118
non-Wake epochs = 616
TST = 308.0 min

[Planning round 2]
The respiratory analyzer reports 130 apneas and 334 hypopneas, while the labels give 616
    non-Wake epochs and TST=5.133 h.  Compute AHI=(130+334)/5.133=90.4 events/h and report
    the result.  ERROR: the respiratory-event count is substantially overestimated;
    BrainAgent accepts the 464 detected events without validating the event count against
    the underlying respiratory evidence.

[Final output]
AHI = 90.4 events/hour

Apneas: 130
Hypopneas: 334
TST: 5.133 h

The resulting AHI indicates a very severe level of
sleep-disordered breathing.
```

Listing 8: Condensed BrainAgent execution trace for SA-32-Instance01 with Claude Opus 5. Planning summaries are color-coded by round.

Error analysis. The sleep-time estimation and AHI formula are correct, but the failure arises from the respiratory-event count. `RespiratoryEventAnalyzer` reports 464 apnea-plus-hypopnea events, which substantially overestimates the reference event burden and consequently inflates the final AHI. Accurate whole-night respiratory-event counting is itself a challenging problem because event detection must remain reliable over long recordings with variable signal quality and respiratory morphology. In this case, BrainAgent accepts the intermediate event count without further validation and propagates it directly to the final result, representing an *intermediate-result validation failure*. The parser and evaluator operate correctly. **Score: 0/100.** The reported AHI of 90.4 events/h falls outside the tolerance around the reference value of approximately 32.14 events/h.

CodeAct failure case1. This trace illustrates the evaluation of Physiological Integration instance `PI-32-Instance02` on a Simultaneous Dataset B recording. The corresponding instruction is:

> *Compare `Go` (event code `16`) and `NoGo` (event code `32`) events within* $[184.307, 253.202)$ *seconds, using only events with complete EEG and fNIRS epochs. For EEG channels `FP1`, `AFz`, and `F1`, compare the baseline-corrected theta power in* $[4, 8)$ *Hz between the two conditions, using* $[-1, 0)$ *seconds as baseline and* $[0, 1)$ *seconds as response. Report the stronger condition as `eeg_stronger_condition`. For fNIRS pair `S03D02`, compare the baseline-corrected HbO response between the two conditions, using* $[-2, 0)$ *seconds as baseline and* $[4, 8)$ *seconds as response. Report the stronger condition as `hbo_stronger_condition`. Briefly state whether EEG and HbO favor the same condition or different conditions. Treat this as descriptive cross-modal evidence without making causal or inhibition-specific claims.*

```
[Task assignment]
Execution backend: persistent IPython kernel
Task: compare Go and NoGo EEG theta and S03D02 HbO responses
within [184.307, 253.202) s

[Execution round 1]
Load the event labels as a serialized dictionary and begin the complete EEG and fNIRS
    analysis.  ERROR: the label file is an array, so .item() raises ValueError.
01  PythonExecutor            [np.load(...).item()]
ValueError: can only convert an array of size 1 to a scalar

[Execution round 2]
The label file is a numerical array rather than a dictionary.  Inspect its structure, extract
    event times and codes, and load both EDF recordings.
02  PythonExecutor            [inspect labels: shape=(128,4)]
03  PythonExecutor            [time=column 0, code=column 2]
04  PythonExecutor            [load EEG and fNIRS EDF files]

[Execution round 3]
Select codes 16 and 32 within the requested interval and retain events with complete EEG and
    fNIRS epochs.  Six Go events and fourteen NoGo events satisfy the requirements.
05  PythonExecutor            [filter 184.307--253.202 s]
06  PythonExecutor            [Go=6, NoGo=14, all complete]

[Execution round 4]
For FP1, AFz, and F1, compute Welch theta power in 4-8 Hz for the baseline and response
    windows.  The response-minus-baseline mean is larger for NoGo.
07  PythonExecutor            [EEG theta calculation]
08  PythonExecutor            [print high-precision means]

Go theta mean   = -1.452706e-11
NoGo theta mean =  8.800841e-12
EEG stronger    = NoGo

[Execution round 5]
Treat the S03D02 760-nm channel as HbO. Compare its mean response-minus-baseline value
    between conditions; the resulting value is larger for Go.  ERROR: raw 760-nm intensity
    is not HbO; optical-density conversion and MBLL are omitted.
09  PythonExecutor            [S03D02_760 as HbO]

fn_hbo = fn.get_data(picks=["S03D02_760"])[0]

Go "HbO" mean   = -1.151377e-04
NoGo "HbO" mean = -2.765590e-04
HbO stronger    = Go

[Execution round 6]
EEG favors NoGo while the computed fNIRS quantity favors Go.  Report that the two modalities
    favor different conditions.

[Final output]
eeg_stronger_condition: NoGo
hbo_stronger_condition: Go

EEG and fNIRS favor different conditions. Frontal theta power is
stronger for NoGo trials, while the reported S03D02 HbO response is
stronger for Go trials. The two modalities therefore provide
divergent descriptive cross-modal evidence.
```

Listing 9: Condensed CodeAct execution trace for PI-32-Instance02 with Claude Opus 5. Execution summaries are color-coded by round.

Error analysis. The event selection and EEG analysis are correct: CodeAct retains the valid Go and NoGo epochs and correctly identifies stronger frontal theta responses for NoGo. The failure occurs in the fNIRS branch, where the raw `S03D02_760` intensity signal is incorrectly treated as HbO. Proper HbO estimation requires both 760- and 850-nm channels, conversion to optical density, and a modified Beer–Lambert transformation. Omitting these steps causes CodeAct to analyze raw intensity rather than HbO, leading to the wrong condition assignment and an incorrect cross-modal conclusion. The files, execution environment, parser, and evaluator operate normally; the error originates from the model-generated analysis code. **Score: 30/100.** The EEG condition metric scores 30/30, while the HbO condition and cross-modal interpretation metrics score 0/30 and 0/40, respectively.

CodeAct failure case2. This trace illustrates the evaluation of Sleep Assessment instance `SA-34-Instance05` on an SHHS recording. The corresponding instruction is:

> *Use the provided sleep EEG/PSG recording and paired sleep-stage labels to identify the longest apnea event over the complete recording. Report its duration in seconds, its start time in minutes from the beginning of the recording, and the sleep stage at event onset. The stage must be one of `W`, `N1`, `N2`, `N3`, or `R`. The available respiratory channels are `THOR RES`, `ABDO RES`, `NEW AIR`, and `SaO2`.*

```
[Task assignment]
Execution backend: persistent IPython kernel
Task: identify the longest apnea event and report its duration,
start time, and sleep stage at onset

[Execution round 1]
Inspect the sleep-stage labels, load the EDF recording, and identify the respiratory channels
    and their sampling rate.
01  PythonExecutor            [load sleep-stage labels]
02  PythonExecutor            [load EDF recording]
03  PythonExecutor            [inspect respiratory channels]

Recording duration = 32519.992 s
Sampling frequency = 125 Hz
Sleep-stage epochs = 1084

[Execution round 2]
Construct a per-second airflow-amplitude signal from NEW AIR and compare it with a local
    rolling baseline to identify sustained reductions.
04  PythonExecutor            [filter and downsample NEW AIR]
05  PythonExecutor            [compute airflow envelope]
06  PythonExecutor            [estimate rolling baseline]
07  PythonExecutor            [group consecutive low-amplitude seconds]

Threshold   Events   Maximum duration
0.10        0        0 s
0.20        0        0 s
0.25        0        0 s
0.30        1        25 s

[Execution round 3]
The initial detector finds no event under the strict thresholds.  Test alternative rolling
    baselines and inspect airflow, respiratory-effort, and oximetry signals.
08  PythonExecutor            [test five-minute percentile baseline]
09  PythonExecutor            [recompute airflow amplitude]
10  PythonExecutor            [inspect THOR RES and ABDO RES]
11  PythonExecutor            [detect SaO2 desaturations]

Trailing-baseline threshold 0.20: maximum duration = 29 s
Trailing-baseline threshold 0.25: maximum duration = 30 s
Trailing-baseline threshold 0.30: maximum duration = 34 s

[Execution round 4]
Bandpass-filter NEW AIR, compute a Hilbert envelope, average it into one-second bins, and
    retain runs below 30% of a trailing median baseline.
12  PythonExecutor            [bandpass NEW AIR at 0.05--1 Hz]
13  PythonExecutor            [compute and smooth Hilbert envelope]
14  PythonExecutor            [apply 30\% rolling-baseline threshold]

Candidate events:
start=8147 s,  duration=34 s
start=28061 s, duration=15 s
start=3 s,     duration=12 s

ERROR: the detector relaxes the airflow criterion to 30% of the rolling baseline and
    quantizes event boundaries into one-second bins.  This heuristic is not
    boundary-equivalent to the annotated apnea interval and underestimates its duration.
```

```
[Execution round 5]
Inspect the longest candidate, verify the associated desaturation and respiratory-effort
    reduction, and map its onset to the paired sleep-stage label.
15  PythonExecutor            [inspect SaO2 around 8147 s]
16  PythonExecutor            [inspect thoracic effort]
17  PythonExecutor            [map onset to sleep-stage epoch]

Detected start = 8147 s = 135.78 min
Detected duration = 34.0 s
Sleep stage = R

[Execution round 6]
Report the longest detected airflow-cessation event together with its onset time and sleep
    stage.

[Final output]
duration_sec: 34.0
start_time_min: 135.78
sleep_stage: R
```

Listing 10: Condensed CodeAct execution trace for SA-34-Instance05 with Claude Opus 5. Execution summaries are color-coded by round.

Error analysis. CodeAct correctly identifies the target event region, onset time, and sleep stage; the failure is confined to event duration. The model-generated detector uses relaxed airflow thresholds and one-second temporal aggregation, which coarsens the event boundaries and yields a duration of 34.0 s instead of the 39.4-s reference interval. This case therefore reflects a *temporal discretization error* introduced by the model-generated detection procedure. The files, execution environment, parser, and evaluator operate normally. **Score: 60/100.** The duration metric scores 0/40, while the onset-time and sleep-stage metrics score 40/40 and 20/20, respectively.

F Discussion

Our results suggest that EEG understanding should be viewed as a system-level capability shaped jointly by the LLM, analytical domain, and execution paradigm. Structured workflows provide explicit capability boundaries, reproducible operations, and auditable execution traces, making them particularly suitable for clinical or standardized analyses in which data security, procedural consistency, and accountability are essential. Flexible coding, by contrast, allows strong models to exploit their general reasoning and programming capabilities more fully, offering greater adaptability for unfamiliar workflows, long-horizon analyses, and exploratory scientific discovery. Neither paradigm is universally superior: structured execution may constrain analyses beyond its predefined tools, whereas autonomous coding can introduce greater execution variability and weaker auditability. A promising direction is therefore to combine both paradigms, using validated tools for safety-critical operations while permitting sandboxed code generation for open-ended exploration. Such hybrid systems may offer a practical path toward more reliable and generalizable EEG understanding.

G Limitations and Future Work

Evaluation cost is a key practical consideration in large-scale LLM benchmarking. Inspired by lightweight benchmark maintenance strategies such as HeaRT-Lighting, we sample five subjects per dataset to balance subject diversity with evaluation efficiency. This design keeps large-scale evaluation tractable while retaining broad coverage across datasets, recordings, and analytical tasks. Nevertheless, the resulting subject coverage remains insufficient to characterize population-level variability. Future releases could adopt more scalable evaluation tracks that broaden subject coverage without substantially increasing evaluation cost. More broadly, we plan to extend BrainBench toward a more comprehensive benchmark for brain-science understanding by incorporating richer analytical tasks and additional neural modalities, including MEG and fMRI. Such extensions will enable systematic evaluation of whether LLM-based systems can generalize analytical competence across modalities, cohorts, experimental settings, and scientific workflows.