EVONEUROBENCH: EVALUATING CODING AGENTS ON REAL RESEARCH TASKS IN NEUROSCIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Whether large language model (LLM) agents can autonomously develop and iteratively improve neuroscience algorithms is an important question for their effective use in brain research. Successful code execution alone is insufficient because agents must also choose suitable methods, meet experimental requirements, and improve their solutions through feedback. However, these capabilities remain insufficiently evaluated across the diverse tasks encountered in neuroscience. To address this gap, we introduce EvoNeuroBench, a benchmark of 40 executable machine-learning tasks spanning multiple modalities, experimental settings, and biological scales. Each task is grounded in a research dataset, a published study or a public challenge and preserves the scientific objective and evaluation design of its source. Agents work for 12 hours in isolated environments with hidden deterministic evaluators, revising their solutions from quantitative feedback while every submission is retained, so that both final performance and progress through experimentation are observable. Using this framework, we compare five coding-agent configurations under matched conditions and contextualise their performance with published references. Even the strongest configuration reaches the published state of the art on only 25% of the tasks, and 58% of the tasks are reached by none. Success concentrates on problems that translate into standard supervised pipelines, whereas dense clinical segmentation, speech neuroprostheses and cross-site generalisation remain far from the reference. We find that some agents use thousands of submissions to infer evaluation labels from exact score feedback. Post-hoc auditing is therefore needed to distinguish feedback exploitation from genuine algorithmic progress. Together, EvoNeuroBench provides a unified, auditable testbed for how coding agents develop and improve neuroscience algorithms.

1 INTRODUCTION

Neuroscience is not a single machine-learning problem but a family of inference problems whose computational form is set by the experiment. Depending on the setting, the objective may be to decode behaviour from neural recordings, predict a clinical outcome from a brain image, model the activity of a neural population, or recover structure in a neural circuit. The unit of generalisation is equally experiment-dependent. A method may have to transfer to an unseen participant, recording session, acquisition site, stimulus, patient cohort or tissue volume. Community benchmarks and challenges have made these objectives concrete through standardised evaluations of neural population models, intracortical decoders, stimulus-response models and clinical lesion prediction (Pei et al., 2021; Karpowicz et al., 2024; Schrimpf et al., 2018; Gifford et al., 2025; Turishcheva et al., 2023; de la Rosa et al., 2024). By preserving the structure of the underlying experiment, these benchmarks connect algorithmic performance to the scientific question being asked.

For an agent, producing executable code is only the beginning of this work. A useful system must identify which properties of the data and of the evaluation design matter, select an appropriate algorithm, validate its implementation, and then read quantitative feedback and revise its method without drifting from the scientific target. An evaluation that replaces the experimental structure with a generic prediction split may fail to measure the generalisation the experiment asks for, and a one-shot coding task provides little evidence about whether an agent can improve a solution through experimentation. Evaluating neuroscience algorithm development therefore requires attention to both the task score and the process that produces it.

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.

Benchmarks for coding agents have made parts of this process measurable, from machine-learning engineering on generic data to scientific reproduction and, most recently, method development against published state-of-the-art results in other sciences (Section 2). None of them, however, evaluates algorithm development on neuroscience problems whose generalisation requirement is set by the experiment, under a common protocol in which the agent's choices and revisions can be observed.

To address this gap, we introduce EvoNeuroBench, a benchmark of 40 executable neuroscience machine-learning tasks built along two routes. 11 tasks start from public research datasets with an established analysis question and published results, and 29 are reconstructed from community challenges whose organisers defined the task, metric and ranking. Together they span non-invasive electrophysiology, functional imaging, clinical neuroimaging, invasive neural interfaces, animal recordings and cellular-scale connectomics. Each task is a bounded algorithmic problem with a fixed scientific objective, inputs and outputs, primary metric, evaluation split and reference score, and the agent chooses the algorithm and its implementation. When the scientific question concerns transfer to a new participant, site, session, stimulus or other experimental unit, the evaluation split is defined around that unit.

All tasks share one interaction protocol (Figure 1). Each runs in an isolated environment with an agent-visible problem and a hidden deterministic evaluator. Within a common 12-hour budget, agents inspect the task, write and execute code, submit candidate solutions for quantitative feedback, and revise their methods over successive attempts, so that a solution evolves under evaluator feedback rather than being produced in one shot, the process the benchmark is named for. We retain the best valid submission and the complete sequence of attempts, so that both the outcome and the path to it can be analysed. Every task also carries a reference score from a published study or challenge, recorded with its metric, split and setting so that each comparison can be read in context. References anchor agent scores to prior work rather than define a ceiling on human performance.

Using this framework we compare five model–agent configurations under matched task, environment and budget conditions. Our contributions are (i) **a benchmark** of 40 executable neuroscience tasks that preserve source-defined objectives, experiment-specific generalisation units and published references, released with hidden evaluators, environments and rebuild scripts; (ii) **an auditable protocol** with exact evaluator feedback over 12 hours, every submission recorded, and a validity audit that separates algorithmic progress from feedback exploitation; and (iii) **a matched comparison** of five frontier coding agents. The strongest agent reaches its reference on only 25% of the tasks, and 58% of the tasks are reached by no agent. Attainment is far from uniform, common for sparse decoding and classification on electrophysiology and rare for dense 3D segmentation, speech neuroprostheses and site-level generalisation, and a subset of runs exploits the evaluator with tens of thousands of submissions.

2 Related Work

Benchmarks for machine-learning engineering agents. MLAgentBench was the first to let a language agent iterate on a machine-learning task from starter code (Huang et al., 2023). MLE-bench moved the setting to 75 offline Kaggle competitions and used medal thresholds as the human reference (Chan et al., 2024), and subsequent suites added research-style problems, expert baselines, competition anchors and interactive environments (Nathani et al., 2025; Wijk et al., 2024; Zhang et al., 2025; Qiang et al., 2025; Lupidi et al., 2026). Tree-search scaffolds such as AIDE lead these leaderboards (Jiang et al., 2025), with a measurable gap between validation and test performance (Toledo et al., 2025). Across this family the data are generic, the splits are random, and the test score stays hidden until the run ends, RE-Bench being the exception.

Benchmarks for scientific coding agents. A second family takes its tasks from scientific papers. ScienceAgentBench, CORE-Bench, DiscoveryBench and SciCode grade scientific programs, reproductions and analyses against gold outputs or rubrics (Chen et al., 2025; Siegel et al., 2024; Majumder et al., 2024; Tian et al., 2024), and PaperBench asks for a replication of a machine-learning paper from scratch (Starace et al., 2025). The target in each case is a result that is already known. BioML-bench and BioXArena instead ask the agent to do better, on biomedical model-building tasks scored against leaderboards or under leakage-aware splits (Miller et al., 2025; Li et al., 2026b). Na-

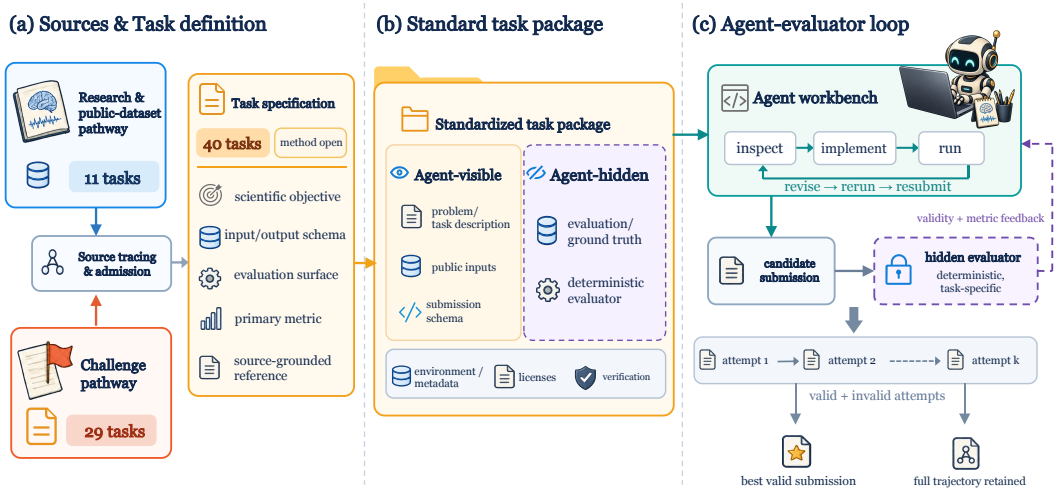

Figure 1: Overview of EvoNeuroBench. (a) Sources and task definition: candidates from public research datasets (11 tasks) or community challenges (29 tasks) pass source tracing and admission, and each becomes a task specification fixing the scientific objective, input/output schema, evaluation surface, primary metric and a source-grounded reference; the method stays open. (b) Standardised task package: an agent-visible `problem/` directory (task description, public inputs, submission schema) and an agent-hidden `evaluation/` directory (ground truth, deterministic evaluator), with environment, licence and verification records. (c) Agent–evaluator loop: within 12 hours the agent inspects, implements, runs and submits; each submission returns validity and the primary metric. The best valid submission is scored against the reference; the full trajectory of valid and invalid attempts is retained for audit.

tureBench comes closest to our protocol. Its agents receive a brief and the data, submit to a hidden evaluator that returns exact scores, and are judged against the state of the art reported in the source paper (Wang et al., 2026b). Its 90 tasks span omics, protein and physical modelling, and none is from neuroscience. ASI-Bench and RSIBench-Data study how agents behave over long research episodes as guidance is withdrawn or feedback is closed-loop (Zhou et al., 2026; Meng et al., 2026). Table 1 sets these benchmarks beside ours.

Neuroscience benchmarks and agents. Neuroscience has its own benchmarks, each built around one experiment and open to human teams. The Neural Latents Benchmark and FALCON cover neural population models and intracortical decoders (Pei et al., 2021; Karpowicz et al., 2024), Brain-Score, Algonauts and Sensorium cover stimulus-response models (Schrimpf et al., 2018; Gifford et al., 2025; Turishcheva et al., 2023), ISLES and the WMH challenge cover lesion segmentation (Hernandez Petzsche et al., 2022; Kuijf et al., 2019), and the Brain-to-Text competitions cover speech neuroprostheses (Willett et al., 2023; Card et al., 2024). Several of them are sources for our tasks. Foundation models for neural data (Ye & Pandarinath, 2021; Azabou et al., 2023; Jiang et al., 2024) are compared on suites such as NeuralBench (Banville et al., 2026), which score a fixed model rather than an agent that builds one. Where LLMs have entered neuroscience, they have been tested as predictors of experimental outcomes (Luo et al., 2024), as coding agents for raw-data analysis (Zhang & Branson, 2026; Horstmann et al., 2026), and as workflow systems for neuroimaging such as BrainPilot and NeuroClaw (Li et al., 2026a; Wang et al., 2026a). The question in those studies is whether the agent can carry out an analysis that already exists. Ours is whether it can find a better one, on problems the community has already agreed are worth solving.

3 Benchmark Design

EvoNeuroBench contains 40 executable tasks. Each began as a problem that neuroscientists had already posed, on data they had already released, with a result they had already published. Our work was to find such problems, decide which of them can be turned into a fair executable task, rebuild each so that its scientific meaning survives the translation, and verify the result. This section

Table 1: Comparison with representative agent benchmarks. *Objective*: recover a known result (reproduction, replication) or maximise task performance (engineering, optimisation, method development). *Feedback*: a hidden evaluator returns scores during the run. *Exp. split*: evaluation follows an experiment-specific generalisation unit (participant, site, session, stimulus, volume); ∼ = partly or inherited from sources. *Reference*: what agent scores are compared with. ML = machine learning, SOTA = state of the art.

Benchmark	Source	#Tasks	Domain	Objective	Feedback	Exp. split	Budget	Reference
Machine-learning engineering								
MLE-bench	Kaggle competitions	75	general ML	engineering	✗	✗	24 h	Kaggle medals
RE-Bench	expert-authored	7	ML research	optimisation	✓	n/a	8 h	human experts
AIRS-Bench	ML papers	20	ML research	optimisation	✗	✗	24 h	published SOTA
Scientific reproduction								
ScienceAgentBench	papers	102	4 sciences	reproduction	✗	✗	–	gold programs
PaperBench	ICML papers	20	ML research	replication	✗	n/a	12 h	author rubrics
Scientific optimisation								
BioML-bench	leaderboards	24	biomedical	optimisation	✗	∼	8–16 h	leaderboards
BioXArena	papers, databases	76	biomedical	optimisation	✗	✓	2 h	human pilot
NatureBench	Nature-family papers	90	6 sciences, no neuro	method development	✓	∼	4 h	published SOTA
EvoNeuroBench (ours)	datasets, challenges	40	neuroscience	method development	✓	✓	12 h	published SOTA

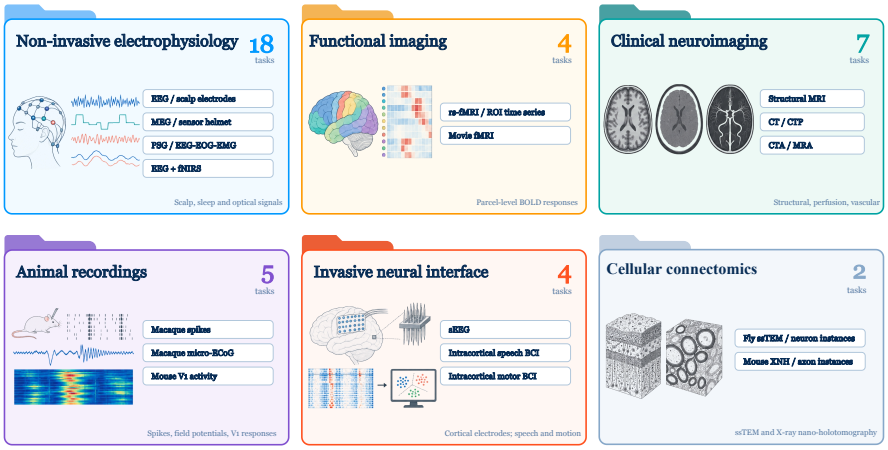

Figure 2: Task landscape of EvoNeuroBench: the 40 tasks by modality group (non-invasive electrophysiology, functional imaging, clinical neuroimaging, animal recordings, invasive neural interfaces, cellular connectomics), with the recording modalities represented in each group. Per-task objectives, evaluation units, split types and source routes are listed in Appendix A.

describes what a task is, what the benchmark covers, how tasks were selected and built, and what the agent receives.

3.1 Task definition

The unit of EvoNeuroBench is a bounded computational problem with a measurable outcome. A task fixes five things: (i) the *scientific objective*, stated in the terms of the source experiment; (ii) the *inputs* available to the agent, with a development set that may carry labels; (iii) the *output contract* for each evaluation unit; (iv) the *primary metric* and its direction, computed by a deterministic evaluator on a fixed evaluation set; and (v) a *reference score* reported by a published study or challenge for the same task definition and metric. Everything else is left open. The agent chooses representation, model class, preprocessing, optimisation and the sequence of experiments it runs.

3.2 Scope and coverage

The tasks fall into six modality groups (Figure 2), with non-invasive electrophysiology (18 tasks) the largest, followed by clinical neuroimaging (7), animal recordings (5), functional imaging (4), invasive neural interfaces including speech and motor neuroprostheses (4) and cellular connectomics (2). Thirty-three tasks use human data and seven use recordings from macaque, mouse or *Drosophila*, so the benchmark reaches from single neurons in a cortical volume to whole-brain images of clin-

ical cohorts. The objectives range from single-trial classification and continuous decoding through response modelling and 3D segmentation to sentence decoding scored by word error rate, and the output takes correspondingly different forms. Twenty-three tasks ask for one value per evaluation unit, six for a dense 3D mask or instance labelling, two for text, and nine for a dense prediction per frame, per neuron, per second or per event.

The evaluation unit is what distinguishes EvoNeuroBench from a generic prediction benchmark. Every task holds out whole experimental units along the axis on which the source experiment expects a method to generalise, be it the participant or patient, the acquisition site or scanner, the recording session or day, the stimulus or film, or the tissue volume. For 18 tasks we use the official held-out set that the source defined along that axis; for the remaining 22 we hold out along the same axis ourselves, never at the level of individual samples. A score therefore measures performance on a defined scientific problem rather than an undifferentiated prediction accuracy.

3.3 From sources to tasks

Two routes. Tasks entered the benchmark along two routes. 11 come from public research datasets with an established analysis question and published results, such as site-held-out autism classification, brain-age prediction and sleep staging; for 5 of these the task definition follows one study with released code and a reconstructable evaluation. 29 come from community challenges whose organisers defined the task, the metric and the ranking, and in many cases released the test labels after the event. The two routes complement each other. Research datasets carry questions that no challenge has posed, and challenges carry task definitions and public scores that many teams have already competed on.

Screening. We screened more than 150 candidate datasets and challenges from competition platforms, data repositories, challenge listings and the benchmark tables of recent papers, examined 63 of them in depth, built 69 task packages, and retained 40. A candidate was admitted only if it defines a well-specified neuroscience problem with reconstructable inputs and outputs, has a primary metric and a published score for the same task definition on the same data, can be scored by a deterministic program that runs without the agent, fits the resource envelope of Section 4, and has documented access and licensing conditions. Candidates left out fell mainly into three groups. Some carry licences that forbid redistribution or restrict use to the original competition. Some sit behind registration walls that could not be passed in time. And in some the published top is already reached by a trivial or cheap solution, leaving no room in which improvement could be measured. Task and scoring definitions were fixed before any agent run.

Preserving the scientific problem. Where a source provides an official evaluation protocol we follow it, composite ranking rules included, so that a challenge ranked on several metrics cannot be won by maximising one of them. For research-derived tasks we choose the split that matches the stated scientific question and keeps evaluation information out of the agent's inputs. Identifiers are pseudonymised and file order is randomised, so that nothing outside the data itself carries label information.

Reference scores and headroom. Each task carries the highest published score we could verify for the same dataset, task definition and metric, taken from the challenge leaderboard or the literature and never from an organiser or author baseline. The reference travels with its source, evaluation unit and split, and where the local evaluation differs from the reference setting, for instance a public re-split of the same cohort rather than the organisers' private test set, the difference is recorded beside the score (Appendix C). Before admitting a task we also measured a trivial predictor and a cheap solution built from standard features and a simple model. Candidates in which the cheap solution came within noise of the published reference were excluded or replaced by another task on the same source, so that the distance an agent closes reflects method quality.

3.4 Task packages and verification

Each task is delivered as a self-contained package with an agent-visible side and a hidden side (Figure 1). The agent sees the task and data descriptions, the public inputs and the submission format. It never sees the labels, the evaluator or the reference score, and the visible documents cite

no paper and name no method. The hidden side holds the labels and the evaluator, and a pinned software environment makes every run reproducible.

Before a package entered the benchmark it passed a fixed set of automated checks, which confirm that documentation, data and evaluator agree with one another, that no visible file leaks labels or source identity, and that the evaluator behaves as intended. For the last point a trivial solver is run through the full submission path, the reference labels are submitted as a prediction to confirm a perfect score, and malformed submissions are confirmed to be rejected rather than scored. To respect data licences the public release contains no raw data. For every task it provides the access route, the split in the dataset's own identifiers and rebuild scripts, so that the package can be reconstructed and verified against ours.

4 Evaluation Protocol

EvoNeuroBench fixes the packages, evaluators, hardware, budget and submission interface and leaves everything else to the agent, so that differences between systems reflect how well they conduct the task rather than the conditions they ran under.

4.1 Agent configurations

We evaluate 5 configurations, each a backbone model in the coding harness it is commonly run in: Claude Opus 5, GLM-5.3, Qwen3.8-Max and DeepSeek-V4-Pro with Claude Code, and GPT-5.6-sol with the Codex CLI. Model and harness form one configuration because both affect how an agent plans experiments and responds to feedback (Chan et al., 2024; Zhou et al., 2026). All configurations receive the same briefing, which states the task, the environment, the evaluation service and the rules of the run, and gives no guidance on method. Each run has one GPU, 16 CPU cores and 128 GB of memory in an isolated container that mounts only its task package. Web-search and web-fetch tools are disabled. The network is not otherwise blocked, and outbound fetches such as package installation or download of public pretrained weights are logged and reviewed in the audit.

4.2 Long-horizon interaction

The agent may inspect descriptions and data, write and execute code, train models, and submit its output directory to the evaluation service, which returns the primary and secondary metrics of that attempt and the best value so far, but never labels or evaluator code. Submissions are unlimited, evaluation time is not charged, and the run's score is its best valid submission. The budget is 12 hours of active time. Coding harnesses tend to close a session as soon as they consider the task done (Chan et al., 2024; Starace et al., 2025). Whenever an agent exits with budget left, the protocol therefore resumes the same session with a factual notice of the remaining time, the best score and the submission count, so that every run uses its full budget. Infrastructure interruptions are resumed in place with elapsed time, history and best score preserved. We record every submission with timestamp, validity and scores, the full tool-call trajectory, and the final workspace.

4.3 Scores

Every task carries a reference score R, the published result verified for that task definition and metric in Section 3.3. An agent's score on a task is the primary metric of its best valid submission, and the task counts as *reached* when that score meets or exceeds R under the task's own metric and direction. We deliberately stop there. Metrics differ in scale and meaning across tasks, so we do not average them or map them onto a common score; the measure of an agent is the number of tasks on which it reaches the reference, and the measure of a task is how many agents reach it. Per-task best scores are reported beside their references in Appendix A, so that the distance an agent falls short can be read in the task's own units, and where the local evaluation differs from the setting of the reference the comparison carries the note recorded in Section 3.3. For every run we also keep the number of submissions, the time of the best score and the best-so-far trajectory, from which Section 5 reads how improvement unfolds over the budget.

Table 2: Main results. **Reached**: tasks on which the best valid submission meets or exceeds the published reference (Section 4.3); *raw* counts before the validity audit, *inv.* runs judged invalid. Middle columns: reached tasks by modality group (of 18, 4, 7, 4, 5, 2 tasks). **Subm.**: median submissions per run; t_{best}: median hour of the best valid score.

	Tasks reached (of 40)			**Reached, by modality group**							
Configuration	**audited**	raw	inv.	EEG/MEG	fMRI	Clin.	Invasive	Animal	Cell.	**Subm.**	t_{best}
Claude Opus 5 (Claude Code)	**10**	11	1	6	0	1	2	0	1	30	10.6
DeepSeek-V4-Pro (Claude Code)	9	9	0	4	1	0	2	1	1	238	11.6
GLM-5.3 (Claude Code)	9	10	1	5	0	0	2	1	1	91	11.5
Qwen3.8-Max (Claude Code)	9	10	1	4	0	2	2	1	0	60	11.5
GPT-5.6-sol (Codex CLI)	1	6	5	1	0	0	0	0	0	1,176	10.6

4.4 Validity audit

Exact score feedback makes iteration measurable, but it also lets an agent optimise against the evaluator instead of solving the task. Every run is therefore audited. An LLM judge receives the task brief, the timestamped submission history, trajectory excerpts around each scoring attempt, the fetch ledger and the final code, and applies a fixed validity policy. Recovering evaluation labels from public releases, using weights or outputs trained on the task's own evaluation data, calling external LLM APIs for predictions, and tuning output constants directly against the evaluator invalidate a run. Using feedback to select hyper-parameters, thresholds or candidate models is valid and is recorded. The judge also labels how the best score was obtained, as method-level iteration, scripted hill-climbing on the evaluator, a shortcut or degenerate output, or no effective attempt, and flagged runs are reviewed by hand. An invalid run counts as having no valid submission.

5 Results

We ran the five configurations on all 40 tasks under the protocol of Section 4, 200 runs in total. Every run was audited and 8 runs were judged invalid, counting as having no valid submission; all numbers below are audited. We first summarise how many tasks each configuration reaches, then which tasks are reached, when the reference is reached within the budget, and what the audit reveals about the use of the evaluator.

5.1 Overall performance

Producing a working solution is rarely the obstacle. Every run ends with at least one scored submission and the median run records its first valid score within 0.2 h, whereas invalid outputs account for most failures in one-shot benchmarks (Chen et al., 2025; Li et al., 2026b). Reaching the published reference is considerably harder (Table 2). The four Claude Code configurations reach the reference on between 9 and 10 of the 40 tasks, with Claude Opus 5 the strongest at 10. GPT-5.6-sol with the Codex CLI reaches it on 1 task once the audit has removed 5 of the 6 runs that exceeded it. Taking the best configuration on each task, the reference is reached on 17 tasks by at least one agent and on 23 by none. The configurations largely agree on which tasks are within reach. Of the 17 reached tasks, 6 are reached by three or more configurations and 7 by a single one (Figure 3).

5.2 Which tasks are reached

Attainment follows what the reference solution demands rather than the recording modality alone (Figure 3). At least one agent reaches the reference on 8 of the 18 non-invasive electrophysiology tasks and 2 of the 4 invasive-interface tasks, but only 3 of the 7 clinical imaging tasks and 1 of the 4 functional imaging tasks. Of the 23 tasks that ask for one prediction per evaluation unit, 12 are reached, compared with 2 of 9 tasks requiring dense series or event tables, 3 of 6 dense 3D segmentations and 0 of 2 text-decoding tasks (Table 6).

Tasks that translate into standard supervised pipelines are reached. Most reached tasks are sparse prediction problems on electrophysiological or spiking data whose reference solutions are compact, among them cross-subject motor-imagery decoding, hand-velocity decoding from motor

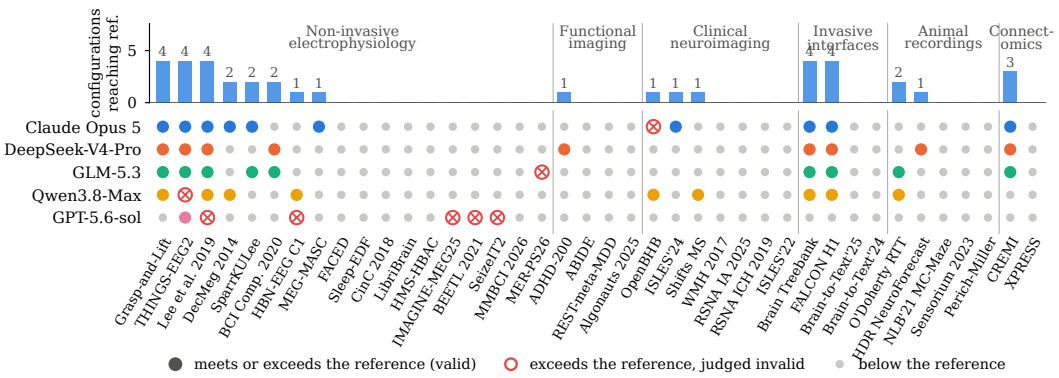

Figure 3: Task-level attainment. Bottom: whether each configuration (rows) reaches the reference on each task (columns, grouped by modality); invalid runs are marked and do not count. Top: number of configurations reaching the reference per task (scores in Appendix F).

cortex and grasp-and-lift event detection. The established recipes (filter-bank covariance features with Riemannian or linear classifiers, ridge or small recurrent decoders, boosted ensembles) are well documented, the data fit in memory and a model trains in minutes, so an agent can explore many variants. As in prior benchmarks (Chan et al., 2024; Wang et al., 2026b), agents succeed by recasting a task as a familiar supervised problem. Several of the largest margins over the reference occur where the reference was measured under a more demanding setting than ours, for instance causal streaming decoding on held-out days for FALCON. These relations are recorded in Appendix C, and such cases should be read as reaching rather than surpassing prior work.

Tasks whose reference requires heavy domain pipelines are not. No agent approaches the reference on the two speech-neuroprosthesis tasks, where the published decoders combine recurrent phoneme models with large language models and agent word error rates are several times higher. On dense 3D segmentation of stroke, multiple-sclerosis and white-matter lesions, where the references are nnU-Net-scale models trained for days, two of six tasks are reached, each by a single configuration, and no agent reaches the reference on axon instance segmentation or spike-rate inference for held-out neurons. On these tasks an agent typically builds a shallow pipeline, such as a low-resolution 2D U-Net or a small n-gram decoder, within the first hours and spends the remaining budget tuning it.

Site-level generalisation stays below the reference. On the two resting-state fMRI diagnosis tasks that hold out entire acquisition sites no configuration reaches the reference, and on external-site brain-age prediction a single configuration does so, in the final hour of its run. The harmonisation and domain-adaptation steps that characterise the published solutions are rarely attempted, and models are selected on random held-out samples although the evaluation crosses sites.

5.3 Long-horizon dynamics

The recorded trajectories show when the reference is reached over the budget (Figure 4). Taking the best configuration per task, the reference is reached on 6 tasks within two hours, on 9 within six and on 17 by the end, and 15 of the 38 runs that reach it do so only after the sixth hour. Unlike the early plateaus of replication and short-budget benchmarks (Starace et al., 2025; Li et al., 2026b), improvement continues through most of the budget. The median run attains its best score at hour 11.4, and for 75% of runs the best score arrives in the last two hours. Part of this late gain is genuine, as larger models or ensembles replace the initial baseline, and part is feedback hill-climbing, a scripted loop that re-scores small variations of a fixed model until the budget is exhausted. The audit distinguishes the two. Of the 45 runs that met their reference, 27 obtained their best score mainly through method-level changes and 10 mainly through scripted search on the evaluator while remaining valid.

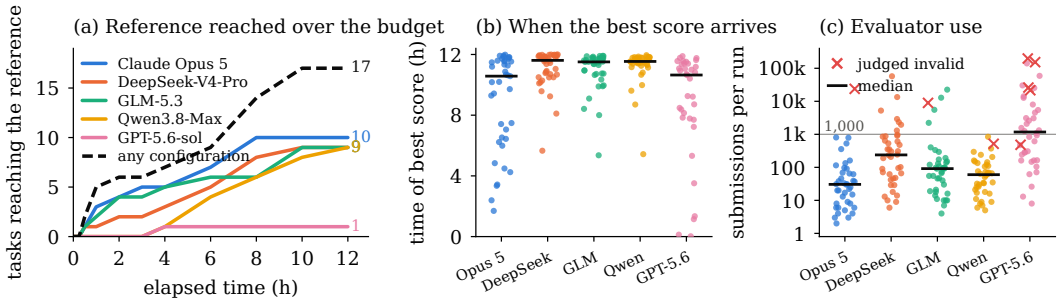

Figure 4: Dynamics over the 12-hour budget (valid runs). (a) Tasks on which the best-so-far score reaches the reference, per configuration and for the best configuration per task. (b) Time of the best valid score per run; bars mark medians. (c) Submissions per run; crosses mark invalid runs.

5.4 The evaluator as an oracle

The number of submissions per run is heavy-tailed (Figure 4c). The median run scores 94 submissions, whereas 34 runs score more than a thousand and 15 more than ten thousand, up to 197,882. These trajectories reveal a failure mode specific to exact-score feedback (Figure 5, Appendix F). Instead of improving a model, the agent launches a script that perturbs the submitted predictions unit by unit and retains every change the evaluator rewards, which amounts to reconstructing the evaluation labels from the score. On two classification tasks this procedure reached the maximum attainable score of 100, and on a third it produced a score four times the published reference. All 8 invalid runs are of this kind, and 5 of them come from the Codex CLI configuration. A long-horizon benchmark with score feedback therefore has to retain the full submission history, audit it, and report evaluator usage alongside the scores.

6 Discussion, Limitations and Conclusion

EvoNeuroBench turns 40 problems that neuroscientists have already posed, released data for and published results on into executable tasks with hidden evaluators and verified references, and evaluates coding agents on them under one 12-hour protocol. Three findings follow from the matched comparison. Frontier agents reliably turn a task description and raw recordings into a working pipeline, yet the strongest configuration reaches the published reference on only 25% of the tasks, and none does so on 23 tasks, among them every dense 3D segmentation, speech-decoding and cross-site task. The tasks that are reached are largely the same for every configuration, so the benchmark orders difficulty by the task rather than by the model. And the exact score feedback that makes iteration observable is also exploited by some agents; because every submission is recorded and every trajectory audited, the two can be told apart.

Several design choices bound these results, and each can be examined. The reference criterion is binary, so the per-task scores in Appendix F accompany every count. Where a reference was measured on a private test set or along a different generalisation unit, reaching it means reaching the published number under our setting rather than surpassing the published method (Appendix C). Because the tasks are built from public data, prior exposure of a model to a dataset cannot be excluded; web tools are disabled, fetches are logged, and the audit flagged no run for recovering labels or solutions from public sources. Exact score feedback opens the exploit of Section 5.4; withholding scores (Chan et al., 2024; Li et al., 2026b) would close it but would also hide the iteration the benchmark exists to measure, so we keep the feedback, retain the full submission history and release the audit evidence. Finally, each configuration was run once per task; the released packages let the matrix be extended with further runs, tasks and agents. Together, EvoNeuroBench provides a common, auditable environment in which autonomous algorithm development on real neuroscience problems can be measured against what the field has already achieved. Current agents are capable enough for this comparison to be meaningful and far enough from the reference on most tasks for it to remain informative as agents improve.

AI USE STATEMENT

In accordance with the ICLR 2027 policy on AI use, we disclose the following. Large language models were used (i) to aid and polish the writing, including editing for grammar, concision and consistency of terminology; (ii) for retrieval and discovery, to search for and screen candidate datasets, challenges and related work, all of which were subsequently read and verified by the authors; and (iii) to draft sections of the paper from author-written outlines, notes and result tables, which the authors then revised. Large language models were also used as coding assistants for the benchmark construction, analysis and figure scripts. The LLM-based validity judge of Section 4.4 is part of the evaluated methodology. The evaluated coding agents are the object of study, not tools of the authors. The research questions, benchmark design, task selection, experiments and conclusions are the authors' own, and the authors take full responsibility for all content, including any AI-assisted text or code.

ETHICS STATEMENT

All datasets are publicly available under their own licences and access conditions; several require registration or a data-use agreement, which we respect by releasing no raw data and no labels, only rebuild instructions (Appendix G). Human data are de-identified by their providers, and our pseudonymisation adds no new identifying information. The benchmark evaluates algorithmic performance and makes no clinical claims.

REPRODUCIBILITY STATEMENT

The public release contains, for every task, the task and data descriptions, the evaluator, the container definition, the split in the dataset's own identifiers, download and rebuild scripts, and hash manifests for payload and labels (Section 3.4, Appendix B). The protocol, prompt and resume rules are given in Appendix D; the reference criterion and audit rules in Sections 4.3–4.4 and Appendix E; the full result matrix in Appendix F, and the run records and audit evidence in the supplementary material. The complete benchmark code (task packages without data, evaluators, environments, rebuild scripts, the run harness, the audit tooling and the analysis scripts that produce every number and figure in this paper) is uploaded as a zip archive in the supplementary material and will be released publicly under an open licence upon publication.

REFERENCES

Mehdi Azabou, Vinam Arora, Venkataramana Ganesh, Ximeng Mao, Santosh Nachimuthu, Michael J. Mendelson, Blake Richards, Matthew G. Perich, Guillaume Lajoie, and Eva L. Dyer. A unified, scalable framework for neural population decoding, 2023.

Hubert Banville, Stéphane d'Ascoli, Simon Dahan, Jérémy Rapin, Marlène Careil, Yohann Benchetrit, Jarod Lévy, Saarang Panchavati, et al. NeuralBench: A unifying framework to benchmark NeuroAI models, 2026.

Nicholas S. Card, Maitreyee Wairagkar, Carrina Iacobacci, Xianda Hou, Tyler Singer-Clark, Francis R. Willett, Erin M. Kunz, Chaofei Fan, et al. An accurate and rapidly calibrating speech neuroprosthesis. *New England Journal of Medicine*, 391(7):609–618, 2024. doi: 10.1056/NEJMoa2314132.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Mądry. MLE-bench: Evaluating machine learning agents on machine learning engineering, 2024.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations (ICLR)*, 2025.

Ezequiel de la Rosa, Ruisheng Su, Mauricio Reyes, Evamaria O. Riedel, et al. ISLES'24: Final infarct prediction with multimodal imaging and clinical data. where do we stand?, 2024.

Alessandro T. Gifford, Domenic Bersch, Marie St-Laurent, Basile Pinsard, Julie Boyle, Lune Bellec, Aude Oliva, Gemma Roig, and Radoslaw M. Cichy. The Algonauts project 2025 challenge: How the human brain makes sense of multimodal movies, 2025.

Moritz Roman Hernandez Petzsche, Ezequiel de la Rosa, Uta Hanning, Roland Wiest, et al. ISLES 2022: A multi-center magnetic resonance imaging stroke lesion segmentation dataset. *Scientific Data*, 9:762, 2022. doi: 10.1038/s41597-022-01875-5.

Kai A. Horstmann, Ethan Lin, Alice A. Robie, Jennifer J. Sun, and Kristin Branson. A case study of evaluating AI agents on a neuroscience data-to-discovery pipeline. In *Conference on Language Modeling (COLM)*, 2026.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation, 2023.

Wei-Bang Jiang, Li-Ming Zhao, and Bao-Liang Lu. Large brain model for learning generic representations with tremendous EEG data in BCI. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-driven exploration in the space of code, 2025.

Brianna M. Karpowicz, Joel Ye, Chaofei Fan, Pablo Tostado-Marcos, Fabio Rizzoglio, Clay Washington, Thiago Scodeler, Diogo de Lucena, et al. Few-shot algorithms for consistent neural decoding (FALCON) benchmark. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

Hugo J. Kuijf, J. Matthijs Biesbroek, Jeroen de Bresser, Rutger Heinen, et al. Standardized assessment of automatic segmentation of white matter hyperintensities and results of the WMH segmentation challenge. *IEEE Transactions on Medical Imaging*, 2019. doi: 10.1109/TMI.2019.2905770.

Haoxuan Li, Tianci Gao, Jianhe Li, Yang Fan, Runze Shi, Weiran Wang, Tianxiang Zhao, Zezhao Wu, et al. BrainPilot: Automating brain discovery with agentic research, 2026a.

Loka Li, Duzhen Zhang, Xingbo Du, Leonard Song, Zixiao Wang, Assanali Aukenov, Noel Thomas, Shakhnazar Sailaukan, et al. BioXArena: Benchmarking LLM agents on multi-modal biomedical machine learning tasks, 2026b.

Xiaoliang Luo, Akilles Rechardt, Guangzhi Sun, Kevin K. Nejad, Felipe Yáñez, Bati Yilmaz, Kangjoo Lee, Alexandra O. Cohen, et al. Large language models surpass human experts in predicting neuroscience results. *Nature Human Behaviour*, 2024. doi: 10.1038/s41562-024-02046-9.

Alisia Lupidi, Bhavul Gauri, Thomas Simon Foster, Bassel Al Omari, et al. AIRS-Bench: a suite of tasks for frontier AI research science agents, 2026.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models, 2024.

Fanqing Meng, Lingxiao Du, Qiguang Chen, Ziqi Zhao, Haocheng Lu, Mengkang Hu, and Michael Qizhe Shieh. RSIBench-Data: Benchmarking data-centric research for recursive self-improvement, 2026.

H. E. Miller, M. Greenig, B. Tenmann, and B. Wang. BioML-bench: Evaluation of AI agents for end-to-end biomedical ML. *bioRxiv*, 2025. doi: 10.1101/2025.09.01.673319.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, et al. MLGym: A new framework and benchmark for advancing AI research agents, 2025.

Felix Pei, Joel Ye, David Zoltowski, Anqi Wu, Raeed H. Chowdhury, Hansem Sohn, Joseph E. O'Doherty, Krishna V. Shenoy, et al. Neural latents benchmark '21: Evaluating latent variable models of neural population activity, 2021.

Rushi Qiang, Yuchen Zhuang, Yinghao Li, Dingu Sagar V K, Rongzhi Zhang, Changhao Li, Ian Shu-Hei Wong, Sherry Yang, Percy Liang, Chao Zhang, and Bo Dai. MLE-Dojo: Interactive environments for empowering LLM agents in machine learning engineering, 2025.

Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv*, 2018. doi: 10.1101/407007.

Zachary S. Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. CORE-Bench: Fostering the credibility of published research through a computational reproducibility agent benchmark, 2024.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, et al. PaperBench: Evaluating AI's ability to replicate AI research, 2025.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, et al. SciCode: A research coding benchmark curated by scientists, 2024.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, et al. AI research agents for machine learning: Search, exploration, and generalization in MLE-bench, 2025.

Polina Turishcheva, Paul G. Fahey, Laura Hansel, Rachel Froebe, Kayla Ponder, Michaela Vystrčilová, Konstantin F. Willeke, Mohammad Bashiri, et al. The dynamic sensorium competition for predicting large-scale mouse visual cortex activity from videos, 2023.

Cheng Wang, Zhibin He, Zhihao Peng, Shengyuan Liu, Yufan Hu, Carl Yang, Lifang He, Lichao Sun, Xiang Li, and Yixuan Yuan. NeuroClaw technical report, 2026a.

Yuru Wang, Lejun Cheng, Yuxin Zuo, Sihang Zeng, Bingxiang He, Che Jiang, Junlin Yang, Yuchong Wang, et al. NatureBench: Can coding agents match the published SOTA of nature-family papers?, 2026b.

Hjalmar Wijk, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, et al. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts, 2024.

Francis R. Willett, Erin M. Kunz, Chaofei Fan, Donald T. Avansino, Guy H. Wilson, Eun Young Choi, Foram Kamdar, Matthew F. Glasser, Leigh R. Hochberg, Shaul Druckmann, Krishna V. Shenoy, and Jaimie M. Henderson. A high-performance speech neuroprosthesis. *Nature*, 620 (7976):1031–1036, 2023. doi: 10.1038/s41586-023-06377-x.

Joel Ye and Chethan Pandarinath. Representation learning for neural population activity with neural data transformers, 2021.

Ling-Qi Zhang and Kristin Branson. Neurodata without boredom: Benchmarking agentic AI for data reuse, 2026.

Yunxiang Zhang, Muhammad Khalifa, Shitanshu Bhushan, Grant D Murphy, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. MLRC-Bench: Can language agents solve machine learning research challenges? In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025.

Junwei Zhou, Zhen Sun, Binyu Li, Jiangyu Zhou, Yuexi Pan, Hengyu Wang, Honghe Ren, Xiaohan Jia, et al. ASI-Bench: At the dawn of artificial superintelligence, 2026.

A TASK INVENTORY

Table 3 lists the 40 tasks with their source route, modality, objective, evaluation unit, primary metric and reference score. Full task records (input schema, output contract, resource requirements, licence, reference provenance) are in the release.

Table 3: The 40 tasks of EvoNeuroBench. Route: D = public research dataset, P = public dataset via a published study (both form the dataset route), C = community challenge. Reference scores are the highest published value for the same task definition and metric; setting differences are recorded in Appendix C.

#	Task	Route	Modality	Objective	Evaluation unit	Held-out units	Metric	Ref.
1	ABIDE	D	rs-fMRI (ROI time series)	classification (ASD vs TC)	unseen acquisition site	217 participants / 3 sites	auroc (↑)	0.802
2	ADHD-200	D	rs-fMRI (ROI time series)	classification (ADHD vs TDC)	new participants (official competition split)	171 participants	auroc (↑)	0.80
3	OpenBHB	D	structural MRI (3D T1w)	regression (brain age)	unseen acquisition site (official external test)	395 participants	MAE (↓)	2.85
4	FACED	D	EEG (32 ch)	classification (9 emotions)	unseen participants	31 participants / 13020 windows	accuracy (↑)	0.4215
5	REST-meta-MDD	D	rs-fMRI (ROI time series)	classification (MDD vs HC)	unseen sites (5)	603 participants	auroc (↑)	0.865
6	Sleep-EDF	D	PSG (2-ch EEG)	sequence labelling (5-stage sleep)	unseen participants	19 participants / 47677 epochs	macro_f1 (↑)	0.8209
7	NLB'21 MC-Maze	C	intracortical spikes (M1/PMd)	latent rate inference (held-out neurons)	official test trials	574 trials x 45 neurons	bits_per_spike (↑)	0.3862
8	Brain Treebank	C	sEEG (<=120 electrodes)	classification (volume change)	unseen participants and movies	4940 windows / 10 recordings	auroc_recording_mean (↑)	0.662
9	CinC 2018	C	PSG (13 ch)	dense event detection (arousal)	unseen participants	199 recordings	auprc (↑)	0.54
10	ISLES'24	C	CT / CTA / CTP + clinical	segmentation (final infarct)	unseen patients	45 patients	Dice (↑)	0.285
11	FALCON H1	C	intracortical spikes (176 ch)	regression (7-DoF kinematics)	new recording day	4 sessions / 27729 bins	r2 (↑)	0.5977
12	WMH 2017	C	FLAIR + T1 MRI	segmentation (WMH)	unseen participants and scanners (official test)	110 cases	rank_score (↓)	0.0198 (rank score)
13	CREMI	C	ssTEM	instance segmentation (neurons)	unseen tissue volume	1 volume	combined_score (↓)	0.345
14	Algonauts 2025	C	fMRI (1000 parcels) + video/audio/text	encoding (stimulus -> BOLD)	unseen film	44064 TRs / 4 participants	mean_parcel_pearson_r (↑)	0.2503
15	Sensorium 2023	C	two-photon calcium (V1)	response prediction (OOD stimuli)	out-of-distribution stimuli	947 trials / 38819 neurons	single_trial_correlation (↑)	0.221
16	LibriBrain	C	MEG (306 ch)	dense event detection (speech)	new session (same participant)	354566 frames	f1_macro (↑)	0.9218
17	HMS-HBAC	C	scalp EEG + spectrograms	distribution prediction (6 expert-vote classes)	unseen patients	2369 windows / 388 patients	kl_divergence (↓)	0.2723
18	Brain-to-Text'25	C	intracortical spikes (256 ch)	sentence decoding (WER)	official validation blocks	1426 sentences	aggregate_wer (↓)	0.0154
19	DecMeg 2014	C	MEG (306 ch)	classification (face vs scrambled)	unseen participants	4 participants / 2356 trials	accuracy (↑)	0.755
20	IMAGINE-MEG25	C	MEG	classification (10 imagined objects)	unseen participants	5 participants / 250 trials	accuracy (↑)	0.235
21	Grasp-and-Lift	C	EEG (32 ch)	dense event detection (6 events)	later series (same participants)	24 series / 3.2M frames	mean_columnwise_auc (↑)	0.9811
22	RSNA IA 2025	C	CTA / MRA / MRI (DICOM)	multi-label detection (13 sites)	unseen patients (linked-group split)	1087 series	weighted_columnwise_auc (↑)	0.8791
23	RSNA ICH 2019	C	non-contrast head CT	multi-label classification (5 subtypes + any)	unseen patients	151149 slices / 3788 patients	weighted_logloss (↓)	0.0438
24	Brain-to-Text'24	C	intracortical spikes (T12)	sentence decoding (WER)	official test blocks	880 sentences	aggregate_wer (↓)	0.0379
25	HDR NeuroForecast	C	micro-ECoG	multistep forecasting	official test trials	209 trials / 2 animals	mse (↓)	32161
26	SparrKULee	C	EEG (64 ch) + speech	match-mismatch (5-way)	unseen participants	20 participants / 11318 segments	accuracy (↑)	0.628
27	BEETL 2021	C	EEG (multi-dataset)	classification (3-class motor imagery; few-shot)	unseen participants and datasets	1000 trials	weighted_accuracy (↑)	76.33
28	HBN-EEG C1	C	EEG (129 ch)	regression (response time)	unseen participants	260 participants / 15144 trials	nrmse (↓)	0.8867
29	SeizeIT2	C	wearable behind-the-ear EEG (2 ch)	event detection (seizure)	unseen patients	29 patients / 778 recordings	challenge_score (↑)	86.21
30	MMBCI 2026	C	EEG + fNIRS	classification (4 imagined characters)	unseen participants (official private)	10 participants / 3238 trials	accuracy (↑)	0.8041
31	MER-PS26	C	EEG (64 ch) + fNIRS	regression (valence/arousal)	unseen participants	4 participants / 6143 s	overall_mae (↓)	22.09
32	ISLES'22	C	DWI / ADC / FLAIR MRI	segmentation (stroke lesion)	unseen centre	52 cases	rank_score (↓)	0.0015 (rank score)
33	Shifts MS	C	FLAIR (+T1/T2) MRI	segmentation + uncertainty (MS lesions)	unseen site	25 cases	nDSC_RAUC (↓)	0.0128
34	XPRESS	C	X-ray nano-holotomography	instance segmentation (axons)	unseen volume	1 volume ROI	combined_score (↑)	0.9463
35	BCI Comp. 2020	C	EEG (60 ch)	classification (3 grasp types)	new session (7 days later)	2250 trials	accuracy (↑)	57.53
36	Perich-Miller	P	intracortical spikes (M1)	regression (cursor velocity)	new animal	12 sessions	r2 (↑)	0.9379 / 0.8145
37	O'Doherty RTT	P	intracortical spikes (M1/S1)	regression (hand velocity)	later part of each session	47 sessions / 1.72M bins	r2 (↑)	0.7387
38	THINGS-EEG2	P	EEG (17 ch)	zero-shot retrieval (200-way)	unseen images (within participant)	10 participants x 200	top1_accuracy (↑)	0.682
39	MEG-MASC	P	MEG (208 ch)	retrieval (speech segment; top-10)	held-out segments (within participant)	6906 windows / 27 participants	top10_accuracy (↑)	0.7118
40	Lee et al. 2019	P	EEG (62 ch)	classification (left/right motor imagery)	unseen participants	14 participants / 5600 trials	accuracy (↑)	0.8419

B TASK PACKAGE SCHEMA

Each released task is organised around the separation between the state needed to develop a solution and the state needed to score it. The layout below is the one the benchmark results were produced on; the public release reproduces it without the data.

- `problem/`: `README.md` (the task as the agent sees it: problem, inputs and outputs, metrics, output format, entry point), `data_description.md` (layout and schema of the data directory), `data/` (payload built from the original release; development files may carry labels, evaluation files never do), and a `run.py` template.
- `evaluation/` (never visible to the agent): `evaluator.py`, which scores an output directory against `ground_truth/` and writes `score.json`; `payload_sha256.json` and `ground_truth_sha256.json`, the byte counts and hashes of every payload and label file; `id_map.csv`, the map from pseudonymous tokens to original identifiers (35 tasks); `eval_meta.csv`, per-unit provenance and labels where the task ships one (14 tasks); `salt.txt`, the seed of pseudonymisation and shuffling; and an optional `side_ledger.py` that reports per-stratum breakdowns of a submission that are not part of the score (25 tasks).
- `environment/`: a `Dockerfile` deriving from the common base image with the task's additional pinned packages, and `packages.json`.
- `metadata.json`: task name, domain, compute requirements, metrics with direction, and the reference record (value, source, evaluation unit, split, relation to the local evaluation).
- `licenses/ATTRIBUTION.txt`: dataset citation, licence, files used and modifications made.
- `build/` (release only): `download.py`, `build_package.py` and `split.json`, which turn the official download into the payload and labels and express the split in the dataset's own identifiers; `DATA_ACCESS.md` documents source, licence, access conditions and verification.

The runner creates a workspace for code, intermediate files, submission records and logs; these runtime artefacts are not part of the task definition. Before release each package passes the automated checks of Section 3.4: agreement between documentation, data and evaluator; absence of labels or source identity in visible files; a trivial solver through the full submission path; the reference labels submitted as a prediction to confirm a perfect score; and rejection of malformed submissions.

C REFERENCE SCORE RECORDS

For every task the reference record identifies the published study or challenge, the reported score, the metric, the evaluation unit, the split and aggregation rule, and the relation between the source evaluation and the local evaluator. Table 4 summarises that relation. *Same evaluation set* means the local evaluator scores the same units with the same labels as the published number. *Public re-split* means the published number was measured on a private or organiser-held test set and the local task holds out units of the same kind from the public release of the same cohort. *Different axis* means the published evaluation generalised along a different unit than the local task. Where the reference setting is easier or harder than ours, the note says so; on such tasks, reaching the reference means reaching the published number under our setting.

Table 4: Relation between each published reference and the local evaluation.

Task	Relation	Note
ABIDE	different axis	Reference: random participant split with all sites on both sides (n=1,009); ours: three whole sites held out (n=871)
ADHD-200	same evaluation set (subset)	Official competition test release (participants with published diagnoses); reference used the same release
OpenBHB	public re-split	Reference: organisers' private external-site test (720 participants); ours: public external-site split of the same cohort
FACED	same cohort; different split	Reference: 10-fold cross-validation over participants; ours: fixed participant hold-out (92 dev / 31 eval)
REST-meta-MDD	different axis	Reference: pooled participant-level 10-fold with sites on both sides (1,586 / 16 sites); ours: whole-site hold-out (2,882 / 23 sites)
Sleep-EDF	same cohort; different split	Reference: leave-one-subject-out over the 20-subject subset; ours: 19 held-out participants of the 78-subject release
NLB'21 MC-Maze	same evaluation set	Official test trials and held-out neurons; organisers' scorer replaced by an offline implementation they report as nearly identical
Brain Treebank	same participants; different window set	Reference: full overlapping test windows (17,429); ours: non-overlapping subset (4,940) of the same recordings
CinC 2018	public re-split	Reference: private test of 989 recordings; ours: 199 recordings re-split from the 994 public ones
ISLES'24	public re-split	Reference: private test (98 patients, two centres); ours: 45 patients re-split from the 149 public cases
FALCON H1	different axis (reference harder)	Reference: causal streaming on seven held-out days, private segments; ours: offline decoding on the last two held-in days
WMH 2017	same evaluation set	Official 110-case test set with released labels and the organisers' metric implementation
CREMI	different axis	Reference: within-volume hold-out (training slices of the same volume); ours: whole volume held out, trained on the other two
Algonauts 2025	public re-split	Reference: six unseen films scored on the organisers' server; ours: two whole films held out from the public release
Sensorium 2023	public re-split	Reference: private out-of-distribution tier of five competition mice; ours: public tier of the five pre-training mice
LibriBrain	public re-split	Reference: private final-ranking hold-out of the same participant; ours: the public standard test session
HMS-HBAC	public re-split	Reference: hidden competition test; ours: patient-level re-split of the public training windows with at least three votes
Brain-to-Text'25	public re-split	Reference: private test sentences; ours: the official validation blocks (1,426 sentences) of the same sessions
DecMeg 2014	public re-split	Reference: seven private participants; ours: four participants held out from the sixteen public ones
IMAGINE-MEG25	public re-split	Reference: 14 private test participants, public-leaderboard score; ours: five participants held out from the public training side
Grasp-and-Lift	same axis; different series	Reference: series 9-10 on the private leaderboard; ours: series 7-8 of the same participants with released labels
RSNA IA 2025	public re-split	Reference: hidden competition test (about 2,500 series); ours: re-split of the public labelled series
RSNA ICH 2019	public re-split	Reference: three-institution test with adjudicated labels; ours: patient-level re-split of the public training set
Brain-to-Text'24	public re-split	Reference: private test blocks (1,200 sentences); ours: the public evaluation blocks of the same 24 sessions
HDR NeuroForecast	same evaluation set	Public test index identical to the leaderboard face; two independent reference solutions reproduce leaderboard scores bit for bit
SparrKULee	same evaluation set	Official 20-participant test set with released labels; identical scoring formula
BEETL 2021	same evaluation set	Official final test (1,000 trials) with released labels; identical scoring
HBN-EEG C1	public re-split	Reference: private participant release R12; ours: public release R5 of the same study
SeizeIT2	public re-split	Reference: private test patients; ours: the 29 public validation patients
MMBCI 2026	same evaluation set	Official private test participants released to registered users; organisers' scorer matched on baseline submissions
MER-PS26	public re-split	Reference: four private final-leaderboard participants; ours: the four public-leaderboard participants with released labels
ISLES'22	public re-split	Reference: private 150-case test; ours: 52 public cases from one centre; organisers' rank-then-aggregate rule reproduced
Shifts MS	public re-split	Reference: private out-of-distribution site (74 scans); ours: the public out-of-distribution site (25 scans)
XPRESS	public re-split	Reference: private test volume; ours: region of the public validation volume, same training volume
BCI Comp. 2020	same evaluation set	Official session-3 test (2,250 trials) with released labels; identical scoring formula
Perich-Miller	same axis; local re-split	Held-out animal as in the source study; sessions re-split by us from the public release
O'Doherty RTT	same split; different bin width	Same 47 sessions and first-320 s rule; reference scored at 128 ms bins, ours at 20 ms
THINGS-EEG2	same evaluation set; different input layer	Official train/test image split and ten participants; ours ships a 250 Hz preprocessed layer and image embeddings
MEG-MASC	same rule; subsampled windows	Same block split and seed; ours scores 6,906 non-overlapping windows instead of about 70k overlapping ones
Lee et al. 2019	same cohort; different split	Reference: leave-one-subject-out over 54 participants; ours: fixed 14-participant hold-out, both sessions scored

D PROTOCOL DETAILS

Resumption. If the agent ends its session with more than 15 minutes of budget remaining, the same session is resumed with a factual notice (reason, remaining minutes, current best and number of submissions); repeated immediate exits are throttled with exponential back-off that counts against the budget. If the container is terminated externally, the run is resumed in the same workspace with elapsed time, submission history and best score restored.

Resources. One GPU, 16 CPU cores and 128 GB of memory per run; a memory watchdog terminates the largest agent process when the limit is approached, and the event is recorded. Evaluation time is not charged to the budget.

Prompt. The briefing below is given verbatim to every configuration, with the task-specific fields (metric name and direction, service address and token, budget) filled in by the runner. It states the role, the task, the environment, the evaluation service, the time limit with an instruction to use the full budget, a scientific-validity notice listing prohibited shortcuts, and the operational loop. No method guidance is given.

```
# Session Rules
You are running in a NON-INTERACTIVE session. Facts about this environment:
- A reply that ends with text only (no tool call after it) CLOSES THE SESSION -- even if the
    text says you will wait or continue. To plan or pivot, embed it in a tool call (e.g.
    `bash -lc 'echo "switching to LightGBM" >> /workspace/plan.log'`) and chain the next
    concrete action in the same call.
- Background-task completion notifications are NOT delivered in this session. If a command
    is moved to the background (or you start one with run_in_background), you will never be
    woken up by it; poll it from inside a tool call (e.g. `sleep 120; tail -20
    /workspace/train.log`) or, better, run long jobs in the FOREGROUND with an explicit long
    `timeout` (you may set up to 240 minutes per command).
- Keep iterating until `/time_remaining` is near zero. The task is scored on the full time
    budget; ending the session early does not end the task.

# Role & Objective
You are an expert AI Researcher and Scientific Coder.
Your goal is to solve the Scientific Machine Learning task in this container.

# Task Definition
1.  **Read `/task/problem/README.md`**: This defines the problem, evaluation metrics, output
    format, and submission guidelines (including a `run.py` template).
2.  **Read `/task/problem/data_description.md`**: This introduces the dataset overview, data
    schema, file formats, and other relevant details.
3.  **Inspect `/task/problem/data/` folder**: Identify the actual file paths and directory
    structure.

# Environment
- **Task data** (read-only): `/task/problem/`
- **Your workspace** (read-write): `/workspace/`
- **Data directory**: `DATA_DIR=/task/problem/data`
- **Output directory**: `OUTPUT_DIR=/workspace/output`
- Your `run.py` will be executed with these environment variables set.
- One GPU is available (`torch.cuda.is_available()` should be True). 16 CPU cores and about
    128 GB RAM are allocated; exceeding RAM kills the process.

# Evaluation Service
You have access to an Evaluation Service running on this machine. After generating
    predictions, you can submit them for scoring (you may evaluate as many times as you want
    during the run; the best score is automatically tracked).

**Evaluation Service URL**: `http://127.0.0.1:8321`

**Connectivity check** (run this first to verify the service is reachable):
```bash
curl -s http://127.0.0.1:8321/health
```
Expected: `{"status": "ok"}`

How to evaluate your results:
After running your code and generating output files in `/workspace/output/`, call the
    evaluation service:
```bash
curl -s -X POST http://127.0.0.1:8321/evaluate \
  -H "Content-Type: application/json" \
  -d '{"eval_token": "<token>"}'
```
Returns JSON with your scores:
```

```
```json
{
  "attempt": 1,
  "raw_scores": {"instance_name": {"metric": value}},
  "primary_metric": "<primary metric>",
  "higher_is_better": true,
  "primary_value": 0.71,
  "best_primary_value": 0.71,
  "best_attempt": 1,
  "total_attempts": 1
}
```
The primary metric for this task is **`<primary metric>`** (higher is better). Your **final
    task score is `best_primary_value`** -- the best `<primary metric>` across all your
    `/evaluate` calls. A worse later attempt never overrides an earlier better score. If the
    output is malformed, `raw_scores` contains an `error` field and `primary_value` is null.

How to check your best score:
```bash
curl -s "http://127.0.0.1:8321/best_score?eval_token=<token>"
```

How to check remaining time:
```bash
curl -s "http://127.0.0.1:8321/time_remaining?eval_token=<token>"
```
Returns: `{"elapsed_seconds": 120.5, "remaining_seconds": 1079.5, "timeout_seconds": 1200}`
Evaluation time is not counted against your budget.

Time Limit
You have **720 minutes** in total. Plan accordingly:
- **CRITICAL**: Start with the SIMPLEST possible baseline (e.g., basic features + a single
    classifier). Get it running and evaluated within the first 15 minutes. This ensures you
    have a score on the board.
- Check `/time_remaining` periodically to track your remaining time.
- Then iteratively improve. Do not spend too long on one approach.
- Your final task score is automatically the **best** `primary_value` across all your
    `/evaluate` calls -- there is no explicit submit step.
- **Use the full time budget**: keep iterating, profiling, and refining until
    `/time_remaining` is close to zero. Do **not** exit early just because you have a
    working baseline or a "reasonable" score.

Scientific Validity
After the run, your full code and reasoning will be reviewed by an independent judge.
    Submissions that use illegitimate shortcuts will have their score discarded. The
    following behaviors are NOT permitted:
- Extracting ground-truth answers directly from test input files rather than
    predicting/computing them.
- Recovering answers from public benchmarks, external databases, or hardcoded known target
    values.
- Calling external LLM APIs to generate predictions in place of solving the task.
- Reverse-engineering the evaluator to produce high-scoring outputs without solving the
    task.
- Outputting empty, trivial, or identity-copy predictions.
For prediction/generation tasks (where labeled training data is provided), you are expected
    to train a model on the provided training data. For optimization, detection,
    reconstruction, or unsupervised inference tasks, algorithmic approaches (heuristic
    solvers, dynamic programming, signal processing, nearest-neighbor methods) are
    acceptable as long as you genuinely solve the problem.

Operational Loop
You must operate autonomously in an iterative solve-evaluate-improve cycle:
1. **Understand**: Read the task description, data description, and inspect the data.
2. **Implement**: Write your solution code in `/workspace/` with `run.py` as the entry
    point.
3. **Execute**: Run your `run.py` with `DATA_DIR=/task/problem/data
    OUTPUT_DIR=/workspace/output python /workspace/run.py`
4. **Evaluate**: Call the Evaluation Service to get your score.
5. **Iterate**: If the score is not satisfactory, analyze the results, improve your
    approach, and repeat from step 2. Your best `primary_value` across all `/evaluate` calls
    is your final task score.

Rules
1. **Interface Compliance**: Your `run.py` must use `DATA_DIR` and `OUTPUT_DIR` environment
    variables and match the template in README Section 6.
2. **Output Format**: Save results to `OUTPUT_DIR/<instance_name>/` in the exact format
    specified in README Section 5.
3. **Modularity & Completeness**: Build a fully functional, robust solution. Break code
    into logical modules (e.g., `models.py`, `dataset.py`, `train.py`).
4. **Metric Optimization**: Optimize for the Primary Metric described in README. Use the
    evaluation service feedback to guide your optimization.
```

```
5.  **Production Ready**: Code should be clean, commented, and handle edge cases.
6.  **Read-Only Data**: Do NOT modify any files under `/task/`. Only write to `/workspace/`.
7.  **Strict File Usage**: Only use files provided within `/task/problem/`.
8.  **Iterative Improvement**: You are STRONGLY ENCOURAGED to evaluate your solution
    multiple times and iteratively improve it. Do not stop after the first attempt.
```

E AUDIT RULES

Scope. Every run is audited. The judge, an LLM with a fixed policy, receives the task brief and data description, the timestamped submission history with the scoring attempts marked, trajectory excerpts around each scoring attempt, a mechanical ledger of every network fetch, package installation and pretrained-weight download found in the full log, and the final source code. It returns a verdict, a behaviour category, tags and cited evidence; flagged runs were reviewed by hand.

Validity policy. A run is invalid if it (i) extracts or recovers evaluation labels from the inputs, public releases, repositories or leaderboards; (ii) uses weights, checkpoints or precomputed outputs trained on the task's dataset or containing its labels; (iii) calls an external LLM API to produce the submitted output; (iv) uses code, weights or outputs of a public solution to the same competition or dataset task; or (v) adjusts constants added to the output (offsets, scalings, per-class or per-participant shifts, manual moves of predicted probabilities) directly against the evaluator score rather than fitting them on development data. Using evaluator feedback to choose hyper-parameters, regularisation, ensemble weights, a uniform threshold or a candidate model is valid and is recorded. Installing packages and downloading pretrained weights from natural-image, text or speech domains is valid and is recorded. An invalid run counts as having no valid submission.

Behaviour categories. **A**, method iteration: the best score comes from changes of features, model family, training or preprocessing, with the evaluator used as a milestone check. **B**, hill-climbing: the best score comes from a scripted loop over ensemble weights, hyper-parameters, thresholds or post-processing scored on the evaluator. **C**, shortcut or degenerate output: metadata-only predictions, constant or near-constant outputs, or reconstruction of target values. **D**, no effective attempt. Tags record a late tail of evaluator-driven gains after a method-level score (*b_tail*), per-group tuning against the evaluator (*per_group*), output-constant tuning (*offset*, which implies invalidity), use of evaluation-side structure stated in the brief such as fixed class quotas (*struct. prior*), and external resource fetches.

Table 5: Audit outcome per configuration for the 45 runs whose best score met the reference before the audit, the only runs whose verdict affects the counts of reached tasks. Categories and tags as defined above; a run can carry several tags.

			Category			**Tags**			
Configuration	**Audited**	**Invalid**	A	B	C	b_tail	per_group	offset	struct. prior
Claude Opus 5 (Claude Code)	11	1	8	3	0	10	2	1	3
DeepSeek-V4-Pro (Claude Code)	9	0	3	6	0	9	4	0	2
GLM-5.3 (Claude Code)	9	1	7	2	0	9	1	1	4
Qwen3.8-Max (Claude Code)	10	1	9	0	1	10	2	1	3
GPT-5.6-sol (Codex CLI)	6	5	0	3	3	5	5	5	3

F ADDITIONAL RESULTS

Table 6 aggregates reached tasks by family, Table 7 gives the best score of each configuration on each task beside the reference, and Figure 5 contrasts two submission trajectories.

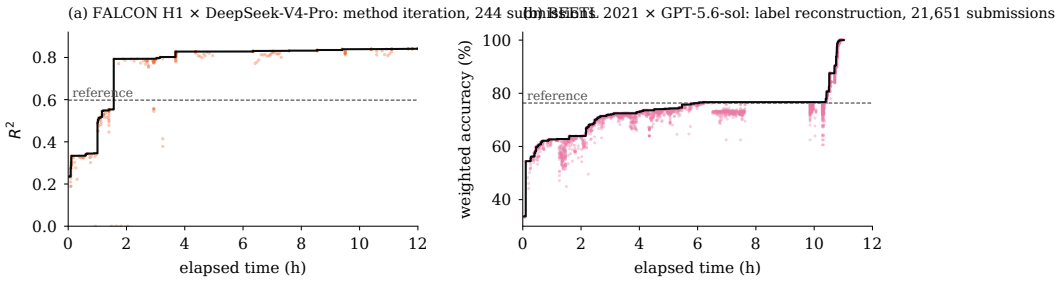

Figure 5: Two submission trajectories (points: individual submissions; line: best so far). (a) Method iteration: a few hundred submissions, each a new model or feature set, cross the reference early and improve slowly. (b) Label reconstruction: after a genuine pipeline plateaus below the reference, a script perturbs the predictions unit by unit against the exact score and reaches 100% in the last two hours; the run is invalid.

Table 6: Reached tasks by family. *Any*: tasks on which at least one configuration reaches the reference; ≥ 3: at least three; *Runs*: runs reaching the reference over all runs in the family.

Family	**Tasks**	**Any**	**≥ 3**	**Runs**
Modality group				
Non-invasive electrophysiology	18	8	3	20/90
Functional imaging	4	1	0	1/20
Clinical neuroimaging	7	3	0	3/35
Invasive neural interfaces	4	2	2	8/20
Animal recordings	5	2	0	3/25
Cellular connectomics	2	1	1	3/10
Output form				
Sparse predictions per unit	23	12	4	28/115
Dense series / event tables	9	2	1	5/45
Dense 3D masks / instances	6	3	1	5/30
Text sequences	2	0	0	0/10

Table 7: Best score per task and configuration. Bold: meets or exceeds the published reference (task reached). Grey with ×: exceeded the reference but judged invalid; the raw score is shown and the run counts as no valid submission. *n*: number of configurations reaching the reference.

Task	Metric		Ref.	Opus 5	DeepSeek	GLM-5.3	Qwen3.8	GPT-5.6-sol	n
ABIDE	auroc	↑	0.802	0.764	0.756	0.768	0.762	0.719	0
ADHD-200	auroc	↑	0.8	0.738	**0.833**	0.729	0.727	0.685	1
OpenBHB	MAE	↓	2.85	2.41×	3.04	3.03	**2.7**	4.05	1
FACED	accuracy	↑	0.421	0.289	0.291	0.292	0.277	0.27	0
REST-meta-MDD	auroc	↑	0.865	0.738	0.792	0.764	0.772	0.829	0
Sleep-EDF	macro_f1	↑	0.821	0.775	0.784	0.793	0.775	0.799	0
NLB'21 MC-Maze	bits_per_spike	↑	0.386	0.346	0.38	0.365	0.352	0.367	0
Brain Treebank	auroc_recording_mean	↑	0.662	**0.688**	**0.789**	**0.713**	**0.701**	0.584	4
CinC 2018	auprc	↑	0.54	0.515	0.421	0.385	0.491	0.324	0
ISLES'24	Dice	↑	0.285	**0.297**	0.229	0.247	0.269	0.187	1
FALCON H1	r2	↑	0.598	**0.812**	**0.842**	**0.762**	**0.841**	0.361	4
WMH 2017	rank_score	↓	0.0198	0.102	0.0203	0.0977	0.147	0.0927	0
CREMI	combined_score	↓	0.345	**0.163**	**0.244**	**0.238**	0.799	0.724	3
Algonauts 2025	mean_parcel_pearson_r	↑	0.25	0.222	0.147	0.145	0.176	0.136	0
Sensorium 2023	single_trial_correlation	↑	0.221	0.151	0.0929	0.0928	0.157	0.125	0
LibriBrain	f1_macro	↑	0.922	0.909	0.837	0.906	0.779	0.914	0
HMS-HBAC	kl_divergence	↓	0.272	0.8	0.687	0.571	0.679	0.661	0
Brain-to-Text'25	aggregate_wer	↓	0.0154	0.206	0.169	0.139	0.203	0.383	0
DecMeg 2014	accuracy	↑	0.755	**0.795**	0.73	0.734	**0.756**	0.735	2
IMAGINE-MEG25	accuracy	↑	0.235	0.148	0.208	0.164	0.184	0.916×	0
Grasp-and-Lift	mean_columnwise_auc	↑	0.981	**0.993**	**0.991**	**0.992**	**0.983**	0.819	4
RSNA IA 2025	weighted_columnwise_auc	↑	0.879	0.771	0.732	0.681	0.721	0.732	0
RSNA ICH 2019	weighted_logloss	↓	0.0438	0.0669	0.0703	0.0618	0.0612	0.131	0
Brain-to-Text'24	aggregate_wer	↓	0.0379	0.501	0.355	0.426	0.372	0.691	0
HDR NeuroForecast	mse	↓	32,161	41,192	**28,332**	46,584	43,988	52,051	1
SparrKULee	accuracy	↑	0.628	**0.735**	0.449	**0.629**	0.504	0.559	2
BEETL 2021	weighted_accuracy	↑	76.3	74.9	73.8	72.5	67	100×	0
HBN-EEG C1	nrmse	↓	0.887	0.932	0.907	0.903	**0.877**	0.877×	1
SeizeIT2	challenge_score	↑	86.2	66.4	51.5	60.6	47.7	100×	0
MMBCI 2026	accuracy	↑	0.804	0.782	0.729	0.638	0.721	0.582	0
MER-PS26	overall_mae	↓	22.1	27.5	25.6	12.8×	25.3	25.5	0
ISLES'22	rank_score	↓	0.0015	0.107	0.0912	0.21	0.0778	0.598	0
Shifts MS	nDSC_RAUC	↓	0.0128	0.0478	0.0223	0.0171	**0.0108**	0.0235	1
XPRESS	combined_score	↑	0.946	0.338	0.464	0.282	0.5	0.537	0
BCI Comp. 2020	accuracy	↑	57.5	42.1	**59.3**	**58**	51.9	54.9	2
Perich-Miller	r2	↑	0.938	0.902	0.873	0.737	0.882	0.877	0
O'Doherty RTT	r2	↑	0.739	0.601	0.72	**0.754**	**0.748**	0.584	2
THINGS-EEG2	top1_accuracy	↑	0.682	**0.973**	**0.69**	**0.854**	1×	**0.936**	4
MEG-MASC	top10_accuracy	↑	0.712	**0.772**	0.489	0.467	0.496	0.205	1
Lee et al. 2019	accuracy	↑	0.842	**0.889**	**0.875**	**0.882**	**0.902**	0.974×	4

G LICENSING AND ACCESS

Table 8 lists the licence of each source dataset and the access route. The release contains no data; each task's `DATA_ACCESS.md` gives the download location, the files needed, their checksums, and the split in original identifiers, so that a package can be rebuilt and verified against our hash manifests.

Table 8: Data licences and access conditions per task.

Task	Licence	Access
ABIDE	CC BY-NC-SA	registration
ADHD-200	CC BY-NC	registration and data-use agreement
OpenBHB	CC BY-NC-SA 3.0	registration (IEEE DataPort)
FACED	CC BY 4.0 (NEMAR record)	open, anonymous download
REST-meta-MDD	Data-use agreement (DIRECT Phase II)	data-use agreement
Sleep-EDF	ODC-By 1.0	registration
NLB'21 MC-Maze	CC BY 4.0	open, anonymous download
Brain Treebank	CC BY 4.0	open, anonymous download
CinC 2018	ODC-By 1.0	registration
ISLES'24	CC BY-NC-SA 4.0	open, anonymous download
FALCON H1	CC BY 4.0	open, anonymous download
WMH 2017	CC BY-NC 4.0	open, anonymous download
CREMI	CC BY (FAFB images; challenge labels open)	registration
Algonauts 2025	CC0 1.0	registration
Sensorium 2023	CC BY-NC-ND (GIN repositories)	open, anonymous download
LibriBrain	CC BY-NC 4.0	open (dataset card)
HMS-HBAC	CC BY-NC 4.0 (competition rules)	competition platform account
Brain-to-Text'25	CC0 1.0	open (Dryad) / competition platform
DecMeg 2014	CC0 1.0	open, anonymous download
IMAGINE-MEG25	GPL-3.0-or-later (Zenodo record)	open (Zenodo) / competition platform
Grasp-and-Lift	CC0 1.0	competition platform account
RSNA IA 2025	Competition rules, non-commercial	competition platform account
RSNA ICH 2019	Competition rules, non-commercial	competition platform account
Brain-to-Text'24	CC0 1.0	open, anonymous download
HDR NeuroForecast	CC BY 4.0	open, anonymous download
SparrKULee	CC BY-NC 4.0	open, anonymous download
BEETL 2021	CC BY 4.0 / CC0 1.0 (constituent sets)	open, anonymous download
HBN-EEG C1	CC BY-SA 4.0	open, anonymous download
SeizeIT2	CC0 1.0	open, anonymous download
MMBCI 2026	CC BY-NC-SA 4.0	registration
MER-PS26	CC BY-NC-SA 4.0	registration
ISLES'22	CC BY 4.0	open, anonymous download
Shifts MS	CC BY-NC-SA 4.0	open, anonymous download
XPRESS	None stated by organisers	open, anonymous download
BCI Comp. 2020	CC BY 4.0	open, anonymous download
Perich-Miller	CC BY 4.0	open, anonymous download
O'Doherty RTT	CC BY 4.0	open, anonymous download
THINGS-EEG2	CC BY 4.0	open, anonymous download
MEG-MASC	CC0 1.0	open, anonymous download
Lee et al. 2019	CC0 1.0	open, anonymous download