EEGUnity: Open-Source Tool in Facilitating Unified EEG Datasets Toward Large-Scale EEG Model

Chengxuan Qin, Rui Yang, *Senior Member, IEEE*, Wenlong You, Zhige Chen, *Student Member, IEEE*, Longsheng Zhu, Mengjie Huang, *Member, IEEE*, and Zidong Wang, *Fellow, IEEE*

***Abstract*—The increasing number of dispersed EEG dataset publications and the advancement of large-scale Electroencephalogram (EEG) models have increased the demand for practical tools to manage diverse EEG datasets. However, the inherent complexity of EEG data, characterized by variability in content data, metadata, and data formats, poses challenges for integrating multiple datasets and conducting large-scale EEG model research. To tackle the challenges, this paper introduces EEGUnity, an open-source tool that incorporates modules of "EEG Parser", "Correction", "Batch Processing", and "Large Language Model Boost". Leveraging the functionality of such modules, EEGUnity facilitates the efficient management of multiple EEG datasets, such as intelligent data structure inference, data cleaning, and data unification. In addition, the capabilities of EEGUnity ensure high data quality and consistency, providing a reliable foundation for large-scale EEG data research. EEGUnity is evaluated across 25 EEG datasets from different sources, offering several typical batch processing workflows. The results demonstrate the high performance and flexibility of EEGUnity in parsing and data processing. The project code is publicly available at github.com/Baizhige/EEGUnity.**

***Index Terms*—Brain-computer-interface, electroencephalogram data integration, large-scale model, open-source software.**

Received 23 September 2024; revised 21 March 2025; accepted 24 April 2025. Date of publication 28 April 2025; date of current version 5 May 2025. This work was supported in part by the University Ethics Committee of Xi'an Jiaotong-Liverpool University (XJTLU) under Grant EXT20-01-07 on March 31 2020; in part by the National Natural Science Foundation of China under Grant 72401233; in part by Jiangsu Provincial Qinglan Project, Natural Science Foundation of Jiangsu Higher Education Institutions of China under Grant 23KJB520038; and in part by the Research Enhancement Fund of XJTLU under Grant REF-23-01-008. *(Corresponding author: Rui Yang.)*

Chengxuan Qin, Wenlong You, and Longsheng Zhu are with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China, and also with the School of Electrical Engineering, Electronics and Computer Science, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: C.Qin8@liverpool.ac.uk; Wenlong.You23@student.xjtlu.edu.cn; Longsheng.Zhu22@alumni.xjtlu.edu.cn).

Rui Yang is with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: R.Yang@xjtlu.edu.cn).

Zhige Chen is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, SAR (e-mail: zhige.chen@connect.polyu.hk).

Mengjie Huang is with the Design School, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China (e-mail: Mengjie.Huang@xjtlu.edu.cn).

Zidong Wang is with the Department of Computer Science, Brunel University London, UB8 3PH Uxbridge, U.K. (e-mail: Zidong.Wang@brunel.ac.uk).

Digital Object Identifier 10.1109/TNSRE.2025.3565158

I. Introduction

BRAIN-COMPUTER interfaces (BCI) are really pushing the limits, being able to record over not just days, weeks, but months, years at a time [29]. The proposal of BCI systems has encouraged many researchers to actively explore brain signals and apply BCI systems in various fields: game interaction entertainment, robot control, emotion recognition, fatigue detection, sleep quality assessment, and clinical fields [2], [10], [30], [36], [38], [42]. With the increasing exploration of BCI systems, the demand for neuro-monitoring capabilities has significantly increased. EEG is applied in various BCI systems to collect active electrical signals from the brain. To meet diverse requirements, numerous EEG-based BCI system paradigms have been proposed [1]. The paradigms allow for the selection of an appropriate approach to gather extensive brain electrical signals to form EEG datasets that provide foundational knowledge of specific patterns in the brain [35], [50].

In recent years, there has been a marked increase in both the quantity and demand for EEG data publications. On the one hand, online databases such as Zenodo [7] and PhysioNet [8] have a vast and growing collection of diverse EEG datasets. On the other hand, a study [15] on large EEG models employing over 20 datasets demonstrates the ever-increasing need to process large-scale EEG data. Nevertheless, the inherent characteristics of EEG data present a challenge in data processing [22], [51]. The challenge is primarily attributed to three key factors:

- **Differences in content data**: Variations in the configuration of electrodes, characteristics of the sensor, and circuit structures can lead to significant differences in the dimensions and distribution of EEG data [31], [44], [45];
- **Differences in metadata**: Variations in the labeling criteria for metadata (such as channel names and events

TABLE I
OVERVIEW OF COMMONLY USED EEG DATA PROCESSING SOFTWARE

Tool	Openness	Platform
SPM [24]	open source	MATLAB
EEGLAB [4]	open source	MATLAB
FieldTrip [32]	open source	MATLAB
Brainstorm [37]	open source	MATLAB
MNE-Python [9]	open source	Python
NeuroScan CURRY [41]	commercial	Windows
BESA [17]	commercial	Windows

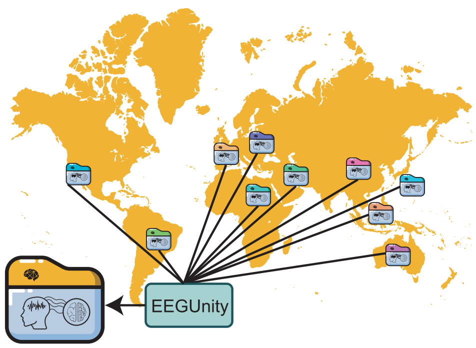

Fig. 1. Objective of Proposed EEGUnity.

annotation), along with the absence or errors in standard information, can lead to inconsistencies in annotation [9];

- **Differences in data formats**: Variations in data formats (such as gdf, edf, mat, csv, txt) complicate the standardization and processing steps of data, demonstrating that the processing pipeline must be tailored for each study [9], [32].

The challenges above indicate an urgent need for an efficient large-scale EEG data management tool to simplify data standardization and processing steps, eventually improving data processing efficiency. Table I summarizes highly integrated EEG data processing tools, covering their openness and supported platforms. Various EEG data processing tools with comprehensive functionality are widely used in multiple EEG research. Some of the common open-source tools include SPM [24], EEGLAB [4], FieldTrip [32], Brainstorm [37], and MNE-Python [9], offering flexible and powerful analysis capabilities to meet diverse research needs. Additionally, commercial software such as NeuroScan CURRY [41] and BESA [17] are also available.

Despite the significant advantages in functionality and broad applicability offered by existing tools, the tools listed in Table I exhibit shortcomings when handling large-scale data. Specifically, the existing tools offer flexible processing for individual datasets, but lack specialized management methods to simultaneously handle multiple datasets with varying content data, metadata, and formats. Therefore, current tools are limited in effectively managing and analyzing heterogeneous EEG data from various sources. As the quantity and demand for EEG datasets continue to increase, the shortcomings become particularly evident, affecting research efficiency and the reliability of results. Therefore, an urgent need is to enhance EEG data processing capability for handling large-scale EEG data.

In this paper, a novel EEG data processing tool named EEGUnity is proposed, aiming to unify diverse EEG datasets from all over the world, as reflected in its name. As illustrated in Fig. 1, the objective of EEGUnity is to manage multiple datasets, efficiently process large-scale data, and enhance data processing efficiency. The introduction of EEGUnity addresses the limitations of existing tools in handling large-scale EEG data and provides a new approach to the unified management and processing of EEG data. EEGUnity offers several innovative features: (1) intelligent data structure inference technology to address the challenge of data heterogeneity; (2) a user-friendly interface for reviewing and modifying EEG dataset annotations to ensure accurate analysis; (3) a comprehensive and unified interface for large-scale data processing to establish a solid foundation for subsequent analyses. These features allow EEGUnity to effectively manage and process heterogeneous EEG data from different sources, enhancing data consistency and comparability.

Based on the above discussions, the main contributions of this paper can be summarized as follows:

- The proposal of a new concept for efficiently managing and processing large-scale EEG data through a unified platform;
- The proposal of EEGUnity, a tool specifically designed for EEG data processing, supporting the management of multiple datasets, thereby addressing the challenges of data heterogeneity;
- The intelligent integration of features into EEGUnity, including data structure inference, correction, cleaning, and unification, ensuring high data quality and consistency, thereby providing a reliable data foundation for EEG research.

The remaining structure of this paper is arranged as follows: Section II provides a detailed introduction to EEGUnity, including overview and implementation details, giving readers comprehensive background information about the tool. Section III demonstrates several typical batch process workflows of EEGUnity, such as dataset management, data correction, data cleaning, and data unification, through specific use cases, helping readers understand the practical application of the tool. Section IV comprehensively analyzes the advantages and limitations of EEGUnity in practical applications, and Section V concludes this paper and proposes future research directions.

II. INTRODUCING EEGUNITY: OVERVIEW AND IMPLEMENTATION

In this section, an overview of EEGUnity is initially presented, including the implementation platform, components, usage, and innovative features. Subsequently, the architectural design of two core components of EEGUnity, `UnifiedDataset` and `Locator`, is detailed.

A. Overview of EEGUnity

EEGUnity is currently a Python package focused on managing multiple EEG datasets. The EEGUnity package includes a core Python class—`UnifiedDataset`, along

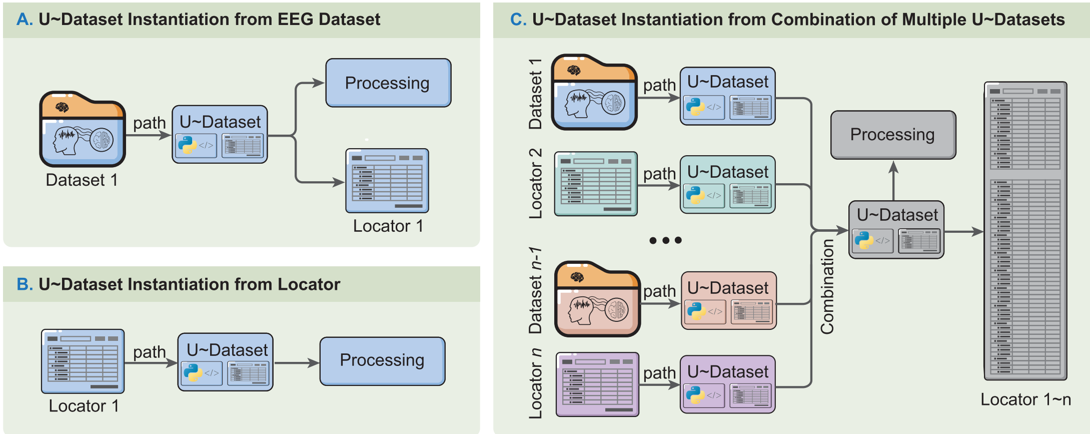

Fig. 2. Three Approaches for Managing Datasets in EEGUnity. "U∼Dataset" refers to `UnifiedDataset`.

with functions for creating, modifying, and merging on the `UnifiedDataset`. The Python class `UnifiedDataset` is designed to provide a unified interface for different operations to datasets. Users can efficiently manage multiple datasets through `UnifiedDataset`, thereby gaining the capability to handle large-scale EEG data.

The usage for managing datasets in EEGUnity is very convenient, following a two-step process: 1) instantiating a `UnifiedDataset`; 2) performing batch processing based on the `UnifiedDataset`. There are three approaches for managing datasets in EEGUnity, as shown in Fig. 2:

- The first approach is illustrated in Fig. 2A. EEGUnity supports the instantiation of `UnifiedDataset` by providing an accessible path to EEG dataset for the initial accessing. After instantiating the `UnifiedDataset`, users can utilize the interface of `UnifiedDataset` for batch processing or export a `Locator`, which is a file recording the essential metadata that required to access the dataset, while parser processes are automatically employed by `UnifiedDataset`.
- The second approach is illustrated in Fig. 2B. EEGUnity supports the instantiation of `UnifiedDataset` by utilizing a pre-existing `Locator`. If the dataset referenced by the `Locator` remains accessible on the user's system, the resulting `UnifiedDataset` instance is equivalent to one that would have been instantiated directly via an EEG dataset address. This design allows users to quickly reload a dataset and save any modification made to the metadata.
- The third approach is illustrated in Fig. 2C. EEGUnity supports the instantiation of `UnifiedDataset` by integrating multiple `UnifiedDataset` instances, thereby allowing users to effectively manage multiple datasets according to specific requirements.

Compared to other processing tools, EEGUnity offers the following innovative features:

- **Data Scale Capabilities**: EEGUnity is designed to process large-scale EEG datasets, supporting the integration of multiple datasets and large-scale data operations;
- **Functional Scope**: EEGUnity processes diverse functions for data cleaning, data correction, data unification, and custom batch processing operations, significantly improving EEG data processing efficiency;
- **Compatibility and Flexibility**: EEGUnity supports intelligent data structure inference using preset code and large language models (LLM), enhancing the flexibility and accuracy of data processing.

B. Implementation Details of UnifiedDataset

In EEGUnity, the parser for EEG data and the unified interface are integrated within a Python class `UnifiedDataset`. `UnifiedDataset` includes a comprehensive set of methods and attributes to provide a unified interface, as shown in Fig. 3. The functions can be categorized into four modules:

- **EEG parser module**: This module facilitates the intelligent parsing of EEG data across diverse formats, including but not limited to edf, gdf, mat, txt, and csv. The EEG data files are initially processed using standard readers, such as parser functions in MNE-Python. If standard readers are insufficient, non-standard readers are employed, which are boosted by pre-defined code and another module in EEGUnity—large language model boost module. EEGUnity supports various multimodal data by classifying them into different channel types, including electrooculography (EOG), electromyography (EMG), electrocardiography (ECG), stimulation (STIM), and other biological signals.
- **Correction module**: This module offers a user-friendly interface to facilitate the correction of dataset annotations. On the one hand, the module allows users to visually inspect and modify annotations within a spreadsheet-like environment. On the other hand, the module provides several correction methods, by built-in functions and the large language model boost module, to systematically review and refine dataset annotations.
- **Batch processing interface module**: This module provides batch processing functionality. On the one hand, this module enables users to customize batch processing pipelines according to specific needs, with implementation details presented in Table V in Appendix A. On the other hand, this module includes a variety of built-in functions for EEG data processing, such as data cleaning,

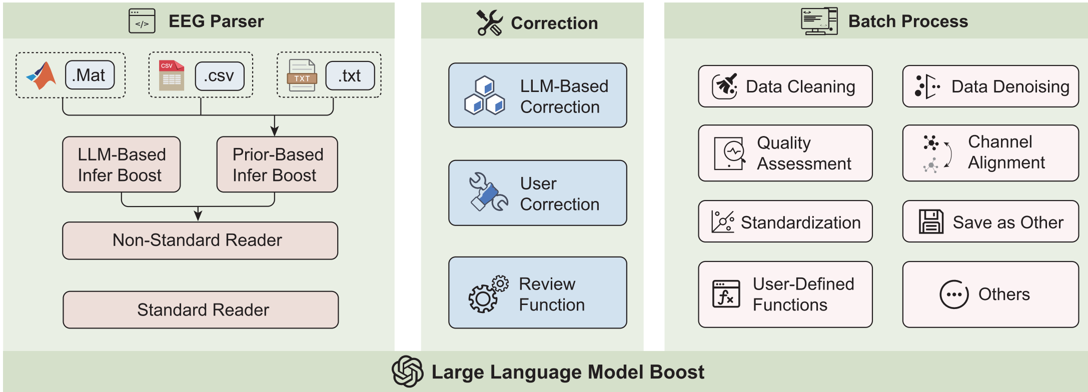

Fig. 3. Structure of the UnifiedDataset.

denoising, quality assessment, channel alignment, and standardization.

- **Large language model boost module**: This module enhances the capabilities of other modules within `UnifiedDataset` by leveraging existing large language models, such as ChatGPT and DeepSeek. It assists with tasks like parsing metadata when traditional programming-based methods are insufficient, improving efficiency in these cases.

C. Implementation Details of Locator

In the design of EEGUnity, a critical component supporting the functionality of EEGUnity is the `Locator`. During the instantiation of `UnifiedDataset`, attributes for each EEG data file are stored in the `Locator`, which is structured using a `DataFrame` from the Pandas package—a widely adopted Python library for data science applications [26]. The attributes within the `Locator` are categorized into basic and advanced attributes for each data file:

- **Basic attributes** include file path, domain tag, file type, EEG channel configuration, sampling rate, duration, and completeness check;
- **Advanced attributes** are linked to specific submodules for functionalities such as data quality scoring. Advanced attributes can be added by subsequent community developers or customized according to user requirements.

Based on the design of `Locator`, users can calibrate the dataset by visually inspecting and modifying the `Locator` file in a spreadsheet-like environment, either programmatically or manually. The metadata specified by the `Locator` takes precedence over those in source data. In such a design, EEGUnity allows users to quickly correct metadata through the locator without modifying the source files.

D. Difference between EEGUnity and Other Toolboxes

Existing EEG analysis toolboxes, such as SPM, EEGLAB, FieldTrip, Brainstorm, MNE-Python, NeuroScan CURRY, and BESA, provide a range of features tailored to diverse research requirements. However, none of these toolboxes fully address the increasing need for large-scale EEG data analysis optimization and the integration of LLMs. EEGUnity is specifically designed to bridge this gap, offering unique capabilities that enhance the efficiency and automation of EEG data processing.

TABLE II
DIFFERENCE BETWEEN EEGUNITY AND OTHER TOOLBOXES

Feature	Multi-Modal Support	GUI Support	Large Data Analysis Optimization	LLMs Boosting Functionality
EEGUnity	✓		✓	✓
SPM	✓	✓		
EEGLAB		✓		
FieldTrip	✓			
Brainstorm	✓	✓		
MNE-Python	✓			
NeuroScan CURRY	✓	✓		
BESA	✓	✓		

A comparative summary of features across these toolboxes is presented in Table II. As demonstrated in the table, EEGUnity distinguishes itself through its specialized support for large data analysis optimization and LLM-boosted functionality, which are absent in existing toolboxes. These features underscore the necessity of EEGUnity within the EEG research community, particularly in the context of modern advancements in artificial intelligence and large-scale data processing.

III. TYPICAL BATCH PROCESSING WORKFLOW IN EEGUNITY

This section introduces typical batch processing workflows in EEG data processing, including dataset management, data correction, data cleaning, and data unification. The described typical batch processing workflows comprise multiple functions of EEGUnity and are not fixed pipelines. Users can customize the pipelines according to specific requirements (such as reducing computational load and minimizing aliasing effects). Table III lists the datasets used in this study, covering diverse paradigms such as disease classification, auditory attention, motor imagery, and sleep pattern detection. For large-scale model training, EEGUnity can automatically process all datasets. The total storage of effective EEG data exceeds 2TB (with the entire database around 4TB), and the processed EEG data spans 35,489 hours. These datasets also

TABLE III
DATASETS USED IN THIS STUDY

File Name	Paradigms	EEG System	Institute Area	Total Hours	Dataset Size (GB)
zenodo-saa [6]	audio attention	ActiveTwo(BioSemi)	USA	15.0	13.7
tuh-eeg-seizure [14]	seizures	NicoletOne(Natus)	USA	1473.6	78.8
tuh-eeg-events [14]	epilepsy	NicoletOne(Natus)	USA	148.7	7.5
tuh-eeg-artifact [14]	artifacts	NicoletOne(Natus)	USA	100.0	5.4
tuh-eeg-abnormal [14]	normal/abnormal	NicoletOne(Natus)	USA	1137.3	58.2
tuh-eeg [14]	clinical recordings	NicoletOne(Natus)	USA	27065.0	1638.9
physionet-sleepedfx [19]	sleep patterns	unknown	Netherlands	3849.1	8.1
physionet-eegmmidb [33]	motor imagery	BCI2000(open-source)	USA/Germany	48.5	3.4
physionet-eegmat [48]	mental arithmetic	Neurocom(XAI-MEDICA)	Ukraine	2.4	0.2
physionet-chbmit [11]	intractable seizures	Neurostimulator (detailed on [54])	USA	106	45.8
physionet-capslpdb [40]	cyclic alternating pattern	unknown	Italy	1004.0	40.1
physionet-auditoryeeg [20]	audio attention	Ganglion Board(OpenBCI) with gold electrode	Italy	9.8	1.4
physionet-erpbci [21]	event-related potentials	ActiveTwo(BioSemi)	UK	2.3	2.19
other-seed [47]	various	ESI NeuroScan	China	172.3	76.3
other-openbmi [23]	various	BrainAmp with Ag/AgCl	South Korea	43.0	60.4
other-migrainedb [46]	migraine	ActiveTwo(BioSemi)	USA	21.2	15.7
other-highgammadataset [34]	motor imagery	NeurOne(Mega Electronics)/WaveGuard(ANT Neuro)/BCI2000	Germany	28.7	12.8
openneuro-ds004015 [13]	audio attention	SMARTING(mBrainTrain)/Abralyt HiCl(Easycap)	Germany	47.3	0.3
openneuro-ds003516 [12]	audio attention	A device with a Ag/AgCl cap(Easycap)	Germany	22.6	0.2
figshare-stroke [25]	motor imagery	A EEG System (ZhenTec)	China	4.4	0.5
figshare-shudb [16]	motor imagery	a amplifier(Brickcom) with Ag/AgCl	China	24.8	1.3
figshare-meng2019 [27]	motor imagery	modified device(Emotiv)	USA	49.6	4.8
figshare-largemi [18]	motor imagery	Neurofax EEG-1200(Nihon Kohden)	Turkey	70.2	4.9
bcic-iv-2b [39]	motor imagery	a device with Ag/AgCl	Austria	26.3	0.3
bcic-iv-2a [39]	motor imagery	a device with Ag/AgCl	Austria	13.4	0.6
bcic-iv-1 [39]	motor imagery	BrainAmp MR plus with Ag/AgCl (Brain Vision)	Germany	3.7	1.5
bcic-iii-1 [3]	motor imagery	A 8x8 ECoG platinum electrode grid	Germany	0.3	0.1

support various research domains, including clinical applications, cognitive neuroscience, and neuro-engineering.

A. Dataset Management and Integration

EEGUnity offers three primary approaches when initiating the processing of one or multiple EEG datasets: (1) specifying the path to an available dataset; (2) specifying the path to an available locator; and (3) integrating multiple datasets. The detailed implementation for each approach is provided in Table IV in Appendix A.

In data management, differences often exist across datasets. To standardize or compensate for the differences, EEGUnity is capable of handling some common discrepancies. Whether EEGUnity can automatically resolve inconsistencies in metadata across datasets depends on the severity of these inconsistencies and the capabilities of its current version. For cases where automatic processing is not available, EEGUnity provides convenient options to inspect and manually correct metadata.

B. Data Correction

EEGUnity supports data correction for one or multiple EEG datasets, including a user-friendly interface for inspecting, modification, diagnosis, and visualization for dataset integrity. The implementation details of the aforementioned functions are outlined in Table VI in Appendix A and elaborated below:

- **Interface for reviewing and modification**: EEGUnity stores metadata in the `Locator`, which users can easily review and modify using tools like PyCharm, Microsoft Excel, and Pandas;
- **Dataset diagnosis**: EEGUnity supports dataset diagnosis through built-in functions that generate detailed dataset reports, including the ratio of file types, domain tags, channel configurations, sampling rates, and completeness checks;
- **Dataset visualization**: EEGUnity includes built-in functions to visualize datasets, such as displaying magnitude-frequency curves for alpha, beta, theta, and gamma waves and channel correlations for each data. Two visualization results are shown in Fig. 4 and Fig. 6 in Appendix B.

C. Data Cleaning

EEGUnity supports data cleaning for one or multiple EEG datasets, with the detailed implementation outlined in Table VII in Appendix A. The suggested data cleaning process is divided into the following steps:

- **Data completeness check**: EEGUnity classifies the completeness of EEG data into three levels and records the results in basic attributes of the `Locator`: (1) completed, indicating that the data includes the original EEG sequence and all basic attributes in the `Locator` have been correctly parsed; (2) acceptable, indicating that further parsing of EEG information may affect the accuracy of analyses; (3) unavailable: indicating the original EEG sequence is unavailable or any basic attributes in the `Locator` cannot be filled. Based on data completeness check, quality assessment, and sample filtering, a statistical result for multiple datasets is presented in Fig. 5 in Appendix B.

- **Quality assessment**: EEGUnity is capable of supporting established methodologies [9], [28] to comprehensively assess the quality of EEG data. The EEG quality assessment method from [28] is employed for data cleaning, with results presented as scores ranging from 0 to 100, as detailed in Algorithm 1 in Appendix C. Additionally, EEGUnity supports the artifact-free ratio, adapted from an existing artifact detection algorithm [9]. The pseudocode and corresponding results are presented in Algorithm 2 in Appendix C and Fig. 7 in Appendix B.
- **Sample filtering**: EEGUnity supports filtering the data based on the attributes in the `Locator` to meet the specific requirements. The filtering process can take into account the quality and completeness level of the data, as well as custom criteria.
- **Denoising**: EEGUnity supports the use of independent component analysis and filtering techniques to remove noise [49] from the EEG data [9]. The denoising process can significantly improve the quality of the data and ensure the reliability of subsequent analysis.

D. Data Unification

Data unification is a novel batch processing workflow in EEGUnity, aiming to transform heterogeneous EEG data into a unified entity. The specific implementation of data unification varies according to different requirements and standards. A detailed outline of the implementation for data unification is illustrated in Table VII in Appendix A. Currently, EEGUnity supports the following unification methods:

- **Save as unified format**: EEGUnity supports saving EEG data in a specified unified format to facilitate subsequent processing and analysis.
- **Resampling**: EEGUnity supports resampling each EEG dataset to a specific target sampling rate.
- **Channel alignment**: EEGUnity supports aligning EEG channels according to a specified order, with or without interpolation. While interpolation assumes consistent electrode placement, the effectiveness depends on task sensitivity (e.g., motor imagery is more affected than sleep or emotion detection) and variability across datasets (e.g., ear electrodes differ for audio detection and SSVEP, whereas similar tasks have more consistent placements). To assess the practical impact of channel alignment, an experiment is conducted, with results presented in Table IX and Table X in Appendix D.
- **Normalization**: EEGUnity supports computing normalization transformation factors, including the variance and mean of samples/channels, and records the results in the `Locator`. The transformation factors are stored in the `Locator`, making it convenient for users to inspect and correct the factors. This normalization process can be performed when retrieving data.
- **Infer unit**: EEGUnity supports inferring the units for each channel in EEG data. The inferred units are stored in the `Locator`, enabling users to inspect and correct inferred units easily. Users can convert the units when retrieving data.
- **Extract event**: EEGUnity supports extracting events for each data and storing the results in the `Locator`. Users can quickly inspect, correct, and supplement the events in `Locator`. The events in `Locator` can be directly used with other epoch functions in EEGUnity.
- **Epoch by event**: EEGUnity supports extracting epochs (segments) from each EEG dataset based on specified events listed in the `Locator`.
- **Epoch for pretraining**: EEGUnity supports extracting epochs (segments) from each EEG dataset by controlling segment parameters.

In the final stage of data processing, the selection of an appropriate dataset structure and read/write policy is critical for accelerating subsequent analysis. EEGUnity supports both individual file formats (e.g., FIF, EDF) organized within a directory and a compact HDF5-based format, with optional multithreading optimization, specifically designed to improve efficiency in input/output operations and storage. A comparative analysis of processing speed with and without multithreading optimization for epoching is presented in Table XI.

E. Large Language Model Boosting

The integration of LLMs can enhance the efficiency and automation of data processing pipelines, but their use is not strictly necessary. Traditional data processing workflows typically depend on predefined algorithms and manual intervention for tasks such as metadata extraction, formatting, and calibration. EEGUnity leverages LLMs to intelligently interpret and structure EEG data, thereby reducing the need for manual corrections and improving overall accuracy.

In the current version of EEGUnity, this functionality is implemented through API-Key authentication. LLMs are employed to facilitate the extraction and calibration of critical metadata, including channel order and sampling frequency, from descriptive files. The results of metadata extraction using LLMs are presented in Table XII in Appendix D

IV. Discussion

EEG data processing capability has been significantly improved by software tools such as EEGLAB, FieldTrip, and MNE-Python. Initially, EEGLAB revolutionized EEG analysis by providing a user-friendly MATLAB-based GUI with advanced capabilities like independent component analysis. Following EEGLAB, FieldTrip offered a modular and flexible MATLAB-based toolbox that emphasized customized and in-depth analyses script-based approach. Subsequently, MNE-Python expanded the landscape by introducing Python packages for EEG data processing, with an emphasis on the reproducibility of data processing pipelines. The progression of established software tools reflects a trajectory toward increasingly sophisticated, flexible, and integrated EEG analysis methods.

Compared with established software, the novel Python package EEGUnity proposed in this paper provides notable advantages in multiple EEG datasets management and large-scale

TABLE IV
IMPLEMENTATION OF THREE APPROACHES FOR MANAGING DATASETS IN EEGUNITY

```
# This script demonstrates three approaches to instantiate and manage EEG datasets using EEGUnity toolkit.
# Example:
from eegunity import UnifiedDataset, con_udatasets
# Approach (1) - Instantiate UnifiedDataset by providing an accessible EEG dataset address.
u_ds = UnifiedDataset(dataset_path='ds')
u_ds.save_locator('locator.csv')
# Approach (2) - Instantiate UnifiedDataset by providing a Locator.
u_ds = UnifiedDataset(locator_path='locator.csv')
# Approach (3) - Instantiate UnifiedDataset by integrating multiple UnifiedDataset instances.
u_ds_1 = UnifiedDataset(locator_path='locator1.csv')
u_ds_2 = UnifiedDataset(dataset_path='ds2')
u_ds_list = [unified_dataset_1, unified_dataset_2]
com_u_ds = con_udatasets(unified_dataset_list)
```

TABLE V
CUSTOM BATCH PROCESSING INTERFACE FOR EEG DATA IN EEGUNITY

```
# This script demonstrates custom batch processing interface in EEGUnity.
from eegunity import UnifiedDataset
from eegunity.utils import Pipeline, get_data_row
# Define a custom processing pipeline
custom_pipeline = Pipeline([
    lambda raw: raw.pick_types(eeg=True),
    lambda raw: raw.filter(0.1, 75),
    custom_process1 # define by yourself
    ...
    custom_process2 # define by yourself
]
# Custom processing function
def custom_process(row):
    raw = get_data_row(row)
    processed = custom_pipeline.forward(raw)
     # define the output path by yourself
    processed.save(out_path)
    return out_path
# Custom condition function
custom_condition = lambda row: row['Completeness Check'] == 'Completed'
# Instantiate and process EEG dataset
u_ds = UnifiedDataset(dataset_path=input_path)
u_ds.eeg_batch.batch_process(
    con_func=custom_condition,
    app_func=custom_process,
    is_patch=False)
```

data processing, including various functions such as data correction, data cleaning, data unification, and customized batch operations. In addition to functionality, the flexibility of EEGUnity is another critical aspect: (1) the `Locator` design facilitates easy inspection and modification of dataset descriptions, ensuring accurate and reliable data processing; (2) the package features a user-friendly batch process interface, allowing researchers to develop customized batch processes tailored to specific requirements.

The ability to process large-scale data has become increasingly significant in the field of EEG, as foundation models require vast amounts of data for pretraining [53]. This trend is similar to computer vision, where the availability of large and annotated datasets like ImageNet [5] have significantly boosted the development of foundational models. Similarly, the ability of EEGUnity to handle and integrate multiple datasets can drive significant advancements in EEG research, enabling the creation of powerful and accurate foundation models. The large foundational models are essential for

TABLE VI
IMPLEMENTATION FOR DIAGNOSING AND VISUALIZING DATASETS IN EEGUNITY

```
# This script demonstrates how to diagnose and visualize EEG datasets using the EEGUnity toolkit. The script sequentially implements the following functions:
# 1. Instantiate a UnifiedDataset by providing an accessible EEG dataset address.
# 2. Generate and display a report summarizing EEG dataset.
# 3. Visualize the frequency spectrum of EEG signals for a subset of samples.
# 4. Visualize the correlation between EEG channels for a subset of samples.
# Example:
from eegunity import UnifiedDataset
u_ds = UnifiedDataset(dataset_path='ds')
u_ds.eeg_correction.report()
u_ds.eeg_correction.visualization_frequency(max_sample=10)
u_ds.eeg_correction.visualization_channels_corr(max_sample=16)
```

TABLE VII
IMPLEMENTATION FOR DATA CLEANING IN EEGUNITY

```
# This script demonstrates the process of cleaning EEG data using the EEGUnity toolkit. The script sequentially implements the following functions:
# 1. Instantiate a UnifiedDataset object.
# 2. Filter EEG samples based on completeness.
# 3. Save the dataset state after filtering.
# 4. Filter sample based on quality assessment.
# 5. Apply a bandpass filter.
# 6. Perform Independent Component Analysis (ICA).
# 7. Save the final dataset state.
# Example:
from eegunity import UnifiedDataset
u_ds = UnifiedDataset(ocator_path="./test_dataset")
u_ds.eeg_batch.sample_filter(completeness_check="Completed")
u_ds.save_locator("locator_completed.csv")
u_ds.eeg_batch.get_quality()
locator = u_ds.get_locator()
u_ds.set_locator([locator['Quality Score'] > 80])
u_ds.eeg_batch.filter(output_path="ds_1_49Hz", filter_type='bandpass', l_freq=1, h_freq=49)
u_ds.save_locator("locator_1_49Hz.csv")
u_ds.eeg_batch.ica(output_path="ds_1_49Hz_ica", max_components=20, method='fastica')
u_ds.save_locator("locator_1_49Hz_ica.csv")
```

TABLE VIII
IMPLEMENTATION FOR DATA UNIFICATION IN EEGUNITY

```
# This script demonstrates the process of unifying and preparing EEG data using the EEGUnity toolkit. The script sequentially implements the following functions:
# 1. Instantiate a UnifiedDataset object by specifying the path to the EEG dataset and a domain tag for identification.
# 2. Save the EEG data in a different format (e.g., FIF format).
# 3. Resample the EEG data to a new sampling frequency.
# 4. Align EEG channels to a specified order.
# 5. Infer the units of the EEG data.
# 6. Extract and display events from the EEG data and segment it into epochs.
# 7. Prepare segmented EEG data for pretraining.
# Example:
from eegunity import UnifiedDataset
u_ds = UnifiedDataset(dataset_path="ds")
u_ds.eeg_batch.save_as_other(output_path="ds_fif")
u_ds.eeg_batch.resample(output_path="ds_resample", new_sfreq=128)
u_ds.eeg_batch.align_channel(output_path="ds_epoch", channel_order=["Cz", "C3", "C4"])
u_ds.eeg_batch.infer_units()
u_ds.eeg_batch.get_events()
u_ds.eeg_batch.epoch_by_event(output_path="ds_epoch", seg_sec=1)
u_ds.eeg_batch.epoch_for_pretraining(output_path="ds_epoch", seg_sec=1)
```

extending advanced EEG tasks, eventually facilitating the application of BCI systems. Despite the advantages provided

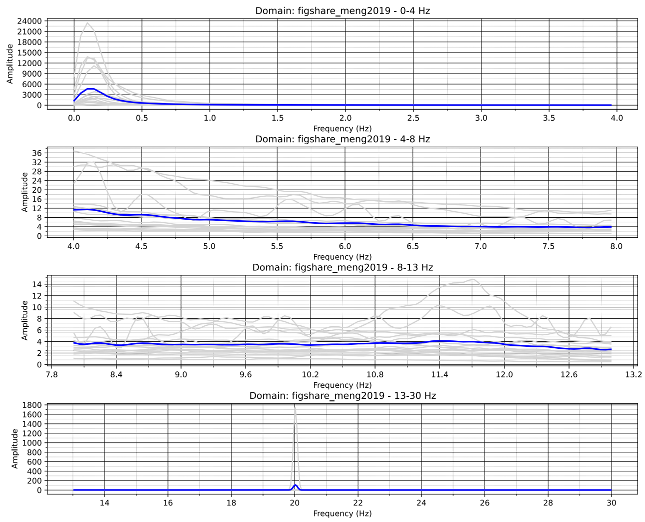

Fig. 4. Frequency Visualization Results of Correction Module. The figure displays magnitude-frequency curves across four subfigures, with each subfigure corresponding to one of the frequency bands: alpha, beta, theta, and gamma. The samples visualized are randomly selected within a domain specified by a "domain tag". The average curve for each band is represented in blue, while individual curves for each data sample are depicted in grey.

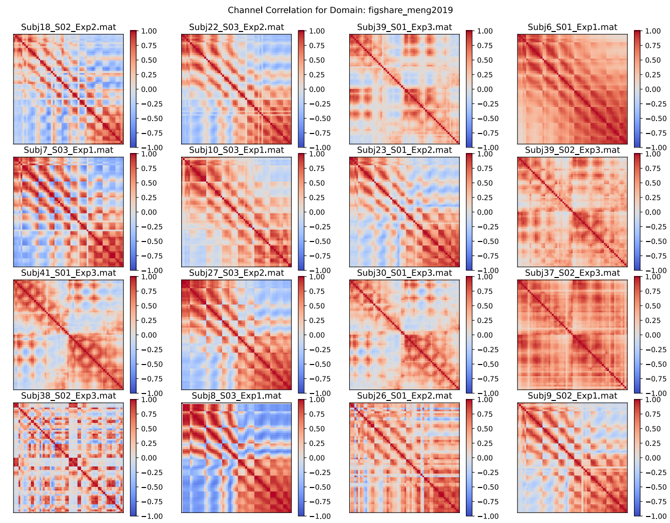

Fig. 6. Channel Correlation Visualization Results of Correction Module. The figure presents channel correlation for samples that are randomly selected within a domain identified by a "domain tag". The number of samples to be visualized is adjustable via a parameter. Subfigure placement is automatically optimized for ease of review. This figure allows users to inspect specific aspects of the samples, such as low-frequency noise or channel noise, facilitating a detailed analysis [52].

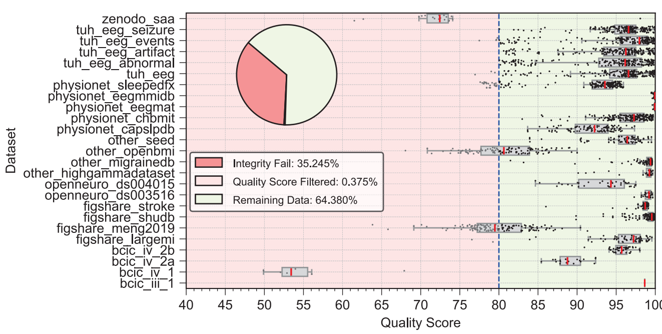

Fig. 5. A Statistical Result of Data Integrity Checks and Quality Assessments for Filtering. The figure illustrates the results of sample filtering using two types of charts: a pie chart and a box plot. The pie chart displays the proportion of data filtered by completeness (i.e., fully completed data) and quality (i.e., data with scores above 80), along with the proportion of the remaining data. The pie chart provides an overview for users to understand the extent of data retained after filtering. The box plot shows the distribution of scores across various datasets. The red line within each box plot represents the median score, while individual dots around the box represent the scores of specific data points, with the maximum number of dots adjustable via parameters. These dots provide insight into the score distribution within each dataset.

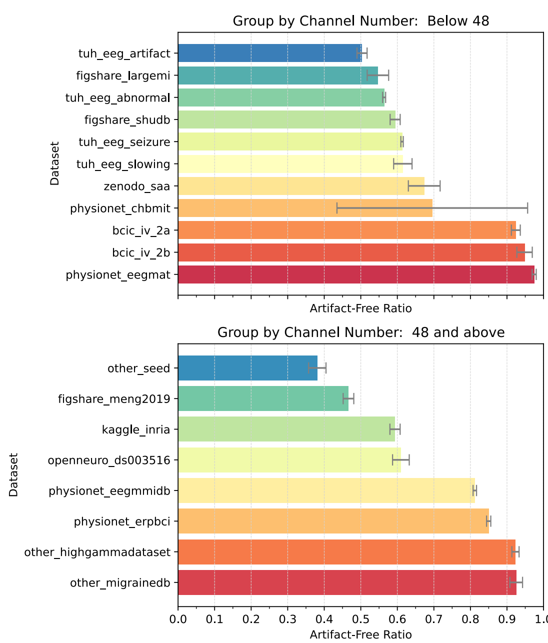

Fig. 7. The Artifact-free Ratio Score for Available Datasets. The dataset scores presented in this figure differ slightly from those in Fig. 5 due to variations in the availability conditions.

by EEGUnity, the construction of large-scale EEG datasets faces three challenges: data permission, privacy concerns, and data validation [22], [43]. Addressing these challenges requires coordinated efforts from the research community to establish guidelines and frameworks that facilitate data sharing while safeguarding individual privacy and ensuring data validation.

V. Conclusion

This paper introduces EEGUnity, a new Python package designed to manage multiple EEG datasets, representing a significant step forward in the management of large-scale EEG datasets. In the management of large-scale EEG datasets, EEGUnity provides powerful data processing capabilities, including data correction, data cleaning, and data unification, all of which can be applied flexibly through the `Locator` design and custom batch processing interface. The comprehensive capabilities and flexible design position EEGUnity as a valuable tool for researchers. The integration of large-scale, high-quality datasets facilitated by EEGUnity has the potential to help researchers gain a deep understanding of brain patterns and develop powerful EEG foundation models. As EEGUnity continues to evolve, the contribution of EEGUnity to efficiency

TABLE IX
CHANNEL ALIGNMENT EVALUATION: CROSS-DATASET BY CHANNEL ALIGNMENT

Channel Alignment Space	Test Datasets	Pre-Train Models					
		figshare-meng2019			*physionet-eegmmidb*		
		EEGNet	*FBCNet*	*TSCeption*	*EEGNet*	*FBCNet*	*TSCeption*
bcic-iv-2a	*figshare-meng2019*	78.87	73.73	76.11	73.92	69.60	62.32
	physionet-eegmmidb	74.56	63.47	64.38	79.20	73.10	72.70
physionet-eegmmidb	*figshare-meng2019*	80.60	78.29	79.59	77.01	71.87	64.62
	physionet-eegmmidb	78.30	62.10	72.13	84.15	78.86	79.74
figshare-meng2019-64	*figshare-meng2019*	82.26	78.32	81.18	75.91	70.57	64.08
	physionet-eegmmidb	78.62	65.35	72.43	85.91	78.52	79.00
figshare-meng2019-62	*figshare-meng2019*	81.93	79.30	79.52	76.05	72.67	67.79
	physionet-eegmmidb	80.81	71.06	67.31	83.94	75.95	78.25

and scalability will become pronounced. Future work includes addressing the challenges in data permission, privacy concerns, and data validation. Addressing these challenges requires coordinated efforts, potentially led by a representative organization that obtains necessary permissions and leverages EEGUnity to create a public, large, and standardized EEG dataset.

APPENDIX A
IMPLEMENTATION OF EEGUNITY

See Tables IV–VIII.

APPENDIX B
FIGURES

See Figs 4–7.

APPENDIX C
ALGORITHMS

Algorithm 1 EEG Signal Quality Assessment via the Shady Method *(Adapted and summarized from original literature [28])*

Input: EEG data $\mathbf{X}$data; Sampling rate fs
Output: Lists of EEG quality scores S

Function `Score`($\mathbf{X}$)**:**
- $(\mathcal{S}2, \mathcal{L}, \mathcal{R}) = \text{ClassifyChannels}(\mathcal{C})$;
  `// ` $\mathcal{S}2$`, channels for score ` sh
  `// ` $\mathcal{L}/\mathcal{R}$`, left/right-side channels`
- $\mathbf{X}\beta = \text{BandpassFilter}(\mathbf{X}\text{data}, \beta\text{low}, \beta\text{high})$;
- $\mathbf{X}\theta = \text{BandpassFilter}(\mathbf{X}\text{data}, \theta\text{low}, \theta\text{high})$;
- $sg = \text{GeneralAmplitudeScore}(\mathbf{X}\text{data})$;
- $s\beta = \text{BetaAmplitudeScore}(\mathbf{X}\beta)$;
- $ss = \text{BetaSinusoidalScore}(\mathbf{X}\beta)$;
- $s\theta = \text{ThetaAmplitudeScore}(\mathbf{X}\theta)$;
- **if** $|\mathcal{S}2| + |\mathcal{L}| + |\mathcal{R}| >= |\mathcal{C}|$ ***and*** $\mathcal{C} \neq \emptyset$ **then**
  - $\mathbf{X}\alpha = \text{BandpassFilter}(\mathbf{X}\text{data}, \alpha\text{low}, \alpha\text{high})$;
  - $sh = \text{HighestAmplitudeScore}(\mathbf{X}\alpha, \mathcal{S}2)$;
  - $sd = \text{SymmetryScore}(\mathbf{X}\alpha, \mathcal{L}, \mathcal{R}, fs)$;
  - $S = [sg, sh, sd, s\beta, ss, s\theta]$;
- **else**
  - $S \leftarrow [sg, s\beta, ss, s\theta]$;
- **return** S;

Algorithm 2 EEG Signal Artifact-free Ratio Score Assessment via the ICA Method (*EOG/EMG/ECGArtifact-Detection is based on MNE-Python library* [9])

Input: EEG data **Xdata**, ICA parameters θICA, Filter parameters θfilter
Output: EEG quality scores S

Function `Score`($\mathbf{X}, \theta ICA$)**:**
- $\mathbf{X}filter = \text{Filter}(\mathbf{X}, \theta\text{filter})$;
- Set $n$components $\leftarrow$ Number of EEG channels;
- Initialize ICA with $n$components and θICA;
- $\mathbf{X}ICA = \text{ICA}(\mathbf{X}filter)$;
- Initialize $\mathcal{E}\text{EOG} = \emptyset$, $\mathcal{E}\text{EMG} = \emptyset$, $\mathcal{E}\text{ECG} = \emptyset$;
- **if** *EOG channels detected in* $\mathbf{X}$ **then**
  - $\mathcal{E}\text{EOG}, s\text{EOG} = \text{EOGArtifactDetection}(\mathbf{X}ICA)$;
- **if** *EMG channels detected in* $\mathbf{X}$ **then**
  - $\mathcal{E}\text{EMG}, s\text{EMG} = \text{EMGArtifactDetection}(\mathbf{X}ICA)$;
- **if** *ECG channels detected in* $\mathbf{X}$ **then**
  - $\mathcal{E}\text{ECG}, s\text{ECG} = \text{ECGArtifactDetection}(\mathbf{X}ICA)$;
- $\mathcal{E}\text{artifact} = \mathcal{E}\text{EOG} \cup \mathcal{E}\text{EMG} \cup \mathcal{E}\text{ECG}$;
- $r\text{artifact} = \frac{|\mathcal{E}\text{artifact}|}{n\text{components}}$;
- $S = 1 - r\text{artifact}$;
- **return** S;

APPENDIX D
TABLES

Table X presents the classification accuracy of models trained on a mixed dataset comprising bcic-iv-2a, physionet-eegmmidb, figshare-meng2019-64, and figshare-meng2019-62. The mixed dataset is aligned into four versions with different channel configurations. Datasets figshare-meng2019-64 and figshare-meng2019-62 share the same task and stimulus, while the others involve similar motor imagery tasks. The results underscore the effectiveness of channel alignment in motor imagery datasets despite the inherent sensitivity of motor imagery to spatial patterns.

Table IX presents the classification accuracy of models trained on figshare-meng2019 and physionet-eegmmidb, then tested on similar tasks without fine-tuning. Channel orders were aligned using four configurations. Results show that despite initial differences, alignment improves cross-dataset generalization, confirming its effectiveness in preserving shared patterns.

TABLE X
CHANNEL ALIGNMENT EVALUATION: MIX DATASETS TRAINING BY CHANNEL ALIGNMENT

Channel Alignment Space	EEGNet	FBCNet	TSCeption
bcic-iv-2a	75.46	69.58	72.68
physionet-eegmmidb	78.79	74.07	77.02
figshare-meng2019-64	76.42	73.20	77.48
figshare-meng2019-62	74.77	72.79	76.84

TABLE XI
PERFORMANCE COMPARISON OF MULTITHREADING (W. MT) AND WITHOUT MULTITHREADING (W.O. MT) IN EEG DATASET PROCESSING

Datasets	w. MT (s)	w.o. MT (s)	Speedup Ratio
tuh-eeg-slowing	13.72313976	20.79512858	34.01%
tuh-eeg-epilepsy	521.8052375	718.3637364	27.36%
tuh-eeg-abnormal	552.7808495	748.7200813	26.17%
tuh-eeg-abnormal	706.4396095	1253.417531	43.64%
physionet-eegmat	4.065684319	4.821900606	15.68%
physionet-chbmit	394.7779763	424.0668974	6.91%

TABLE XII
EEG METADATA EXTRACTION TASK BY LLMS

Task	Datasets	LLMs				
		gpt3.5	*gpt-4o*	*hunyuan*	*qwen*	*deepseek*
Sample Rate	*physionet-auditoryeeg*	✓	✓	✓	✓	✓
	physionet-chbmit	✓	✓	✓	✓	✓
	physionet-erpbci	✓	✓	✓	✓	✓
Channel Order	*other-openbmi*	✓	✓	✓	✓	✓
	physionet-auditoryeeg	✓	✓	✓	✓	✓
	physionet-chbmit	✓	✓	✓	✓	✓

Table XII presents EEG metadata extraction results using the LLM boost module of EEGUnity. The extraction process follows six steps: 1) Potential description files are filtered by attributes like file size, format, and name. 2) EEGUnity generates a predefined prompt from each identified file. 3) The prompt is sent to remote LLMs via API-Key authentication. 4) Responses are retrieved in JSON format. 5) Sampling rate and channel order are extracted. 6) If inconsistencies arise, an interactive answer selection resolves conflicts. The results from XII validate the effectiveness of LLMs in extracting EEG metadata.

Table XI compares processing times for epoching in EEG dataset processing. In the multithreading setting, datasets are grouped by predefined domain tags, with each group processed in a separate thread. Processing speed depends on factors like thread count and available CPU cores. The speedup ratio ranges from 6.91% to 43.64%, demonstrating the effectiveness of multithreaded design in EEG processing.

REFERENCES

[1] R. Abiri, S. Borhani, E. Sellers, Y. Jiang, and X. Zhao, "A comprehensive review of EEG-based braincomputer interface paradigms," *J. Neural Eng.*, vol. 16, no. 1, 2019, Art. no. 011001.

[2] J. Berdell, S. Kudernatsch, and H. Ferdowsi, "AI-driven solid-state device to enable natural control of upper-extremity robotic exoskeletons," *Syst. Sci. Control Eng.*, vol. 12, no. 1, Dec. 2024, Art. no. 2331061.

[3] B. Blankertz et al., "The BCI competition III: Validating alternative approaches to actual BCI problems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 153–159, Jun. 2006.

[4] A. Delorme and S. Makeig, "EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *J. Neurosci. Methods*, vol. 134, no. 1, pp. 9–21, Mar. 2004.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[6] Y. Deng, I. Choi, B. Shinn-Cunningham, and R. Baumgartner, "Impoverished auditory cues limit engagement of brain networks controlling spatial selective attention," *NeuroImage*, vol. 202, Nov. 2019, Art. no. 116151.

[7] *European Organization For Nuclear Research and OpenAIRE*, Zenodo, Geneva, Switzerland, 2013.

[8] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[9] A. Gramfort, "MEG and EEG data analysis with MNE-Python," *Frontiers Neurosci.*, vol. 7, p. 267, Dec. 2013.

[10] X. Gu et al., "EEG-based brain–computer interfaces (BCIs): A survey of recent studies on signal sensing technologies and computational intelligence approaches and their applications," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 18, no. 5, pp. 1645–1666, Sep. 2021.

[11] A. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. Dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[12] B. Holtze, M. Jaeger, S. Debener, K. Adiloğlu, and B. Mirkovic, "Are they calling my name? Attention capture is reflected in the neural tracking of attended and ignored speech," *Frontiers Neurosci.*, vol. 15, 2021.

[13] B. Holtze, M. Rosenkranz, M. Jaeger, S. Debener, and B. Mirkovic, "Ear-EEG measures of auditory attention to continuous speech," *Frontiers Neurosci.*, vol. 16, 2022.

[14] *The Temple University Hospital EEG Corpus*. Accessed: 2024. [Online]. Available: https://isip.piconepress.com/projects/nedc/html/tuh

[15] W. Jiang, L. Zhao, and B. L. Lu, "Large brain model for learning generic representations with tremendous EEG data in BCI," in *Proc. Int. Conf. Learn. Represent.*, 2024, pp. 1–18.

[16] J. Ma, B. Yang, W. Qiu, Y. Li, S. Gao, and X. Xia, "A large EEG dataset for studying cross-session variability in motor imagery brain-computer interface," *Sci. Data*, vol. 9, no. 1, p. 531, 2022.

[17] A. Kabbara, N. Forde, C. Maumet, and M. Hassan, "Successful reproduction of a large EEG study across software packages," *Neuroimage, Rep.*, vol. 3, no. 2, Jun. 2023, Art. no. 100169.

[18] M. Kaya, M. K. Binli, E. Ozbay, H. Yanar, and Y. Mishchenko, "A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces," *Sci. Data*, vol. 5, no. 1, pp. 1–16, Oct. 2018.

[19] B. Kemp, A. Zwinderman, B. Tuk, H. Kamphuisen, and J. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000.

[20] N. Abo Alzahab et al., "Auditory evoked potential electroencephalography-biometric dataset," *Data Brief*, vol. 57, Dec. 2024, Art. no. 111065.

[21] L. Citi, R. Poli, and C. Cinel, "Documenting, modelling and exploiting P300 amplitude changes due to variable target delays in Donchin's speller," *J. Neural Eng.*, vol. 7, no. 5, Oct. 2010, Art. no. 056006.

[22] H. Khan, X. Wang, and H. Liu, "A study on relationship between prediction uncertainty and robustness to noisy data," *Int. J. Syst. Sci.*, vol. 54, no. 6, pp. 1243–1258, Apr. 2023.

[23] M.-H. Lee et al., "EEG dataset and OpenBMI toolbox for three BCI paradigms: An investigation into BCI illiteracy," *GigaScience*, vol. 8, no. 5, 2019, Art. no. giz002.

[24] V. Litvak et al., "EEG and MEG data analysis in SPM8," *Comput. Intell. Neurosci.*, vol. 2011, no. 1, 2011, Art. no. 852961.

[25] H. Liu et al., "An EEG motor imagery dataset for brain computer interface in acute stroke patients," *Sci. Data*, vol. 11, no. 1, p. 131, 2024.

[26] W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python Sci. Conf.*, S. van der Walt and J. Millman, Eds., 2010, pp. 51–56.

[27] J. Meng and B. He, "Exploring training effect in 42 human subjects using a non-invasive sensorimotor rhythm based online BCI," *Frontiers Human Neurosci.*, vol. 13, p. 128, Apr. 2019.

[28] S. Mohamed, S. Haggag, S. Nahavandi, and O. Haggag, "Towards automated quality assessment measure for EEG signals," *Neurocomputing*, vol. 237, pp. 281–290, May 2017.

[29] M. Naddaf, "Mind-reading devices are revealing the brain's secrets," *Nature*, vol. 626, no. 8000, pp. 706–708, Feb. 2024.

[30] M. Y. M. Naser and S. Bhattacharya, "Towards practical BCI-driven wheelchairs: A systematic review study," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 1030–1044, 2023.

[31] G. Niso, E. Romero, J. T. Moreau, A. Araujo, and L. R. Krol, "Wireless EEG: A survey of systems and studies," *NeuroImage*, vol. 269, Apr. 2023, Art. no. 119774.

[32] R. Oostenveld, P. Fries, E. Maris, and J.-M. Schoffelen, "FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data," *Comput. Intell. Neurosci.*, vol. 2011, no. 1, 2011, Art. no. 156869.

[33] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: A general-purpose brain-computer interface (BCI) system," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1034–1043, 2004.

[34] R. T. Schirrmeister et al., "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, no. 11, pp. 5391–5420, Nov. 2017.

[35] S. Shivaprasad, K. Chadaga, C. C. Dias, N. Sampathila, and S. Prabhu, "An interpretable schizophrenia diagnosis framework using machine learning and explainable artificial intelligence," *Syst. Sci. Control Eng.*, vol. 12, no. 1, Dec. 2024, Art. no. 2364033.

[36] R. Somarathna, T. Bednarz, and G. Mohammadi, "Virtual reality for emotion elicitation – a review," *IEEE Trans. Affect. Comput.*, vol. 14, no. 4, pp. 2626–2645, Apr. 2023.

[37] F. Tadel, S. Baillet, J. C. Mosher, D. Pantazis, and R. M. Leahy, "Brainstorm: A user-friendly application for MEG/EEG analysis," *Comput. Intell. Neurosci.*, vol. 2011, pp. 1–13, 2011, Art. no. 879716.

[38] J. Tan, Y. Li, Q. Xie, and X. Wang, "Modeling and causality analysis of human sensorimotor control system based on NVAR method," *Int. J. Netw. Dyn. Intell.*, vol. 2, no. 4, Dec. 2023, Art. no. 100014.

[39] M. Tangermann et al., "Review of the BCI competition IV," *Frontiers Neurosci.*, vol. 6, p. 55, Jul. 2012.

[40] M. G. Terzano et al., "Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep," *Sleep Med.*, vol. 2, no. 6, pp. 537–553, Nov. 2001.

[41] D. Toutant, G. McLeod, P. Abbasian, A. Ghassemi, M. Darbandi, and M. Ng, "A novel customized representation of source space in epilepsy pre-surgical evaluation (P5–8.003)," *Neurology*, vol. 98, no. 18, p. 3751, May 2022.

[42] J. Wang, L. Bi, and W. Fei, "EEG-based motor BCIs for upper limb movement: Current techniques and future insights," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 4413–4427, 2023.

[43] W. Wang, L. Ma, Q. Rui, and C. Gao, "A survey on privacy-preserving control and filtering of networked control systems," *Int. J. Syst. Sci.*, vol. 55, no. 11, pp. 2269–2288, Aug. 2024.

[44] M. Wei, M. Huang, and J. Ni, "Cross-subject EEG channel selection method for lower limb brain–computer interface," *Int. J. Netw. Dyn. Intell.*, vol. 2, no. 3, Sep. 2023, Art. no. 100008.

[45] K. Yi, Y. Wang, K. Ren, and D. Li, "Learning topology-agnostic EEG representations with geometry-aware modeling," in *Proc. Conf. Neural Inf. Process. Syst.*, 2023, pp. 1–16.

[46] A. Chamanzar, S. M. Haigh, P. Grover, and M. Behrmann, "Abnormalities in cortical pattern of coherence in migraine detected using ultra high-density EEG," *Brain Commun.*, vol. 3, no. 2, 2021, Art. no. fcab061.

[47] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks," *IEEE Trans. Autonomous Mental Develop.*, vol. 7, no. 3, pp. 162–175, Sep. 2015.

[48] I. Zyma et al., "Electroencephalograms during mental arithmetic task performance," *Data*, vol. 4, no. 1, p. 14, Jan. 2019.

[49] H. Li, J. Tang, W. Li, W. Dai, Y. Liu, and Z. Zhou, "Multi-task collaborative network: Bridge the supervised and self-supervised learning for EEG classification in RSVP tasks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 32, pp. 638–651, 2024.

[50] K. Guan, Z. Zhang, T. Liu, and H. Niu, "Cross-task mental workload recognition based on EEG tensor representation and transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 2632–2639, 2023.

[51] J. Zhou, Y. Duan, Y. Zou, Y.-C. Chang, Y.-K. Wang, and C.-T. Lin, "Speech2EEG: Leveraging pretrained speech model for EEG signal recognition," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 2140–2153, 2023.

[52] W. Yan, B. He, J. Zhao, Y. Wu, C. Du, and G. Xu, "Frequency domain filtering method for SSVEP-EEG preprocessing," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 2079–2089, 2023.

[53] E. Eldele, M. Ragab, Z. Chen, M. Wu, C.-K. Kwoh, and X. Li, "Self-supervised learning for label-efficient sleep stage classification: A comprehensive evaluation," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 1333–1342, 2023.

[54] A.-T. Avestruz et al., "A 5 μW/Channel spectral analysis IC for chronic bidirectional brain–machine interfaces," *IEEE J. Solid-State Circuits*, vol. 43, no. 12, pp. 3006–3024, Dec. 2008.