Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/neuroimage

HAPPILEE: HAPPE In Low Electrode Electroencephalography, a standardized pre-processing software for lower density recordings

K.L. Lopez[a], A.D. Monachino[a], S. Morales[b], S.C. Leach[b], M.E. Bowers[b], L.J. Gabard-Durnam[a,*]

[a] Northeastern University, 360 Huntington Ave, Boston, MA, United States
[b] University of Maryland, College Park, MD, United States

ARTICLE INFO

Keywords:
Electroencephalography
Processing pipeline
Low-density EEG
HAPPILEE
Automated pre-processing

ABSTRACT

Lower-density Electroencephalography (EEG) recordings (from 1 to approximately 32 electrodes) are widely-used in research and clinical practice and enable scalable brain function measurement across a variety of settings and populations. Though a number of automated pipelines have recently been proposed to standardize and optimize EEG pre-processing for high-density systems with state-of-the-art methods, few solutions have emerged that are compatible with lower-density systems. However, lower-density data often include long recording times and/or large sample sizes that would benefit from similar standardization and automation with contemporary methods. To address this need, we propose the HAPPE In Low Electrode Electroencephalography (HAPPILEE) pipeline as a standardized, automated pipeline optimized for EEG recordings with lower density channel layouts of any size. HAPPILEE processes task-free (e.g., resting-state) and task-related EEG (including event-related potential data by interfacing with the HAPPE+ER pipeline), from raw files through a series of processing steps including filtering, line noise reduction, bad channel detection, artifact correction from continuous data, segmentation, and bad segment rejection that have all been optimized for lower density data. HAPPILEE also includes post-processing reports of data and pipeline quality metrics to facilitate the evaluation and reporting of data quality and processing-related changes to the data in a standardized manner. Here the HAPPILEE steps and their optimization with both recorded and simulated EEG data are described. HAPPILEE's performance is then compared relative to other artifact correction and rejection strategies. The HAPPILEE pipeline is freely available as part of HAPPE 2.0 software under the terms of the GNU General Public License at: https://github.com/PINE-Lab/HAPPE.

1. Introduction

Electroencephalography (EEG) recordings are a useful and noninvasive tool for interrogating human brain function across the lifespan. Advancements in neuroimaging technology and computer science have allowed for rich data collection in laboratories through the use of high-density channel layouts, but it is not always feasible or optimal to rely on these dense layouts. Low-density channel layouts (fewer than approximately 32 channels) continue to be heavily used, particularly with clinical populations, both in clinical research (Brito et al., 2019, 2016; Gu et al., 2018; Pellinen et al., 2020; van den Munckhof et al., 2018) and diagnostic testing (Aeby et al., 2021; Cassani et al., 2017; Paul et al., 2019; Tiwari et al., 2017), as well as in low-resource areas (Kariuki et al., 2016; Siddiqi et al., 2015; Sokolov et al., 2020; Williams et al., 2019). A low-density EEG approach also provides the flexibility for researchers to travel to participants for testing in natural contexts (e.g. school-based or home-based studies, Troller-Renfree et al., 2021) or in the event that participants cannot come to the lab. Low-density EEG will also be instrumental in future research, given the current momentum towards large-scale neuroscience studies that achieve community implementation and the focus on precision medicine through brain-based biomarkers (e.g., potential EEG-based screening for Autism Spectrum Disorder at well-child doctor's visits), where high-density recordings may be neither practical nor necessary. Indeed, a number of wearable, ultra-low-cost, low-density EEG hardware solutions are emerging in industry to facilitate such measurement. A key impediment to the use of low-density EEG in these contexts is the fact that the raw EEG signal is contaminated by both environmental and physiological artifacts. Up to this point, researcher selection of uncontaminated EEG data has been standard practice, but even with low-density data, this method is time-consuming, subjective, and does not allow for the efficient processing of a large number of data sets. Low-density EEG collected in clinical contexts that can span hours may also preclude manual inspection due to recording length. As a result, there remains a current and growing need for software that standardizes and automates the processing and removal of artifacts in low-density EEG data.

[*] Corresponding author.
E-mail address: l.gabard-durnam@northeastern.edu (L.J. Gabard-Durnam).

https://doi.org/10.1016/j.neuroimage.2022.119390.
Received 18 March 2022; Received in revised form 3 June 2022; Accepted 13 June 2022
Available online 8 July 2022.
1053-8119/

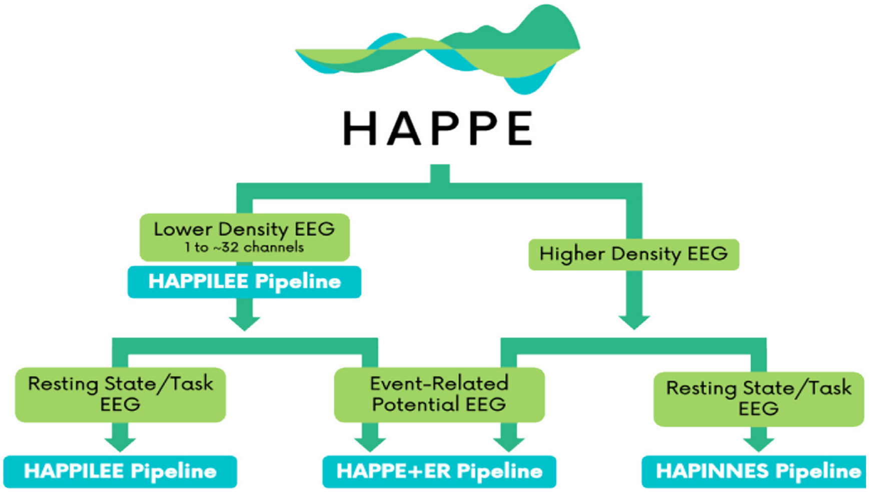


Fig. 1. Schematic illustrating the various pipelines within the HAPPE software package.

There is now an extensive collection of automated EEG processing pipelines (e.g., Andersen 2018; APP, da Cruz et al. 2018; MADE, Debnath et al. 2020; EEG-IP-L, Desjardins et al. 2021; HAPPE, Gabard-Durnam et al. 2018; Hatz et al. 2015; FASTER, Nolan et al. 2010; Automagic, Pedroni et al. 2019; EPOS, Rodrigues et al. 2020). However, their reliance on independent component analysis (ICA) to segregate and correct artifacts makes them unsustainable for low-density data, as the limited number of channels provides insufficient independent components for robust artifact isolation. Many of these pipelines also use standard deviation-dependent approaches to identify outlier data or channels as artifact-contaminated. These approaches may require modification to scale down to low-density setups with few channels. Other software tools built into these fully-automated pipelines to aid in different stages of artifact detection are most effective when used with high-density data or have not been validated in low-density data (PREP, Bigdely-Shamlo et al. 2015a; SASICA, Chaumon et al. 2015; Adjusted-ADJUST, Leach et al. 2020; ADJUST, Mognon et al. 2011; ASR, Mullen et al. 2013; MARA, Winkler et al. 2014). A recent automated mega-analysis by Bigdely-Shamlo et al. (2020) introduced a pipeline that supports both high-density and low-density data, but only at the upper bound of low-density channel layouts. Specifically, they tested their pipeline using data sets ranging from a Neuroscan 30-channel headset (Compumedics Neuroscan) to a Biosemi 256-channel headset (Biosemi B.V.), but found that the density of the headset accounted for variability in the channel amplitudes across datasets after processing. Several pipelines automate the processing of strictly low-density data. Of these options, some are made specifically for a particular population (Cassani et al., 2017) or acquisition systems (e.g., James Long EEG Analysis System software, Whedon et al. 2020). Others use independent component analysis to correct artifacts (Hajra et al., 2020), which cannot support many low-density setups. Still others offer only artifact-rejection approaches (e.g., channel and/or segment rejection), some of which target only specific classes of artifact (e.g., eye-blink artifacts), and these artifact-rejection methods can cause significant data loss without artifact-correction in continuous data first (e.g., EEG Analysis System software (James Long Company), MINIMADE, Troller-Renfree et al. 2021). Thus, there remains a need for standardized processing solutions to serve the range of low-density EEG configurations in use and the range of artifacts that occur in EEG data.

To address this need, we propose a novel pipeline for low-density EEG data (fewer than 32 channels) called HAPPILEE (Harvard Automated Pre-processing Pipeline Including Low-Electrode Encephalography). We apply contemporary approaches to optimize line noise reduction, bad channel detection, artifact correction from continuous data, segmentation, and bad segment rejection methods to suit low density datasets. HAPPILEE is embedded in the HAPPE software package, which facilitates pre-processing for a variety of analyses and data types (Fig. 1). To facilitate ERP analyses on low density data, HAPPILEE interfaces with the HAPPE+ER pipeline within HAPPE (Monachino et al., 2021). Because modifications for ERP analyses are not density-dependent (with the exception of the optional bad channel detection step, which uses HAPPILEE optimization criteria for low density ERP analyses), specific details on the optimization of this software for ERP analyses can be found in Monachino et al. (2021). The following sections of this manuscript describe HAPPILEE's processing steps and outputs, assess optimization of these steps for low-density EEG, and demonstrate HAPPILEE's effectiveness with a low-density developmental EEG dataset and simulated EEG signals relative to other pre-processing approaches (see Fig. 2 for full pipeline schematic).

1.1. Optimization dataset

The various steps of the HAPPILEE automated pipeline were optimized using a subset of developmental EEG files from the Bucharest Early Intervention Project (BEIP) (for full study design, see Zeanah et al. 2003). The EEG files contributing to this example dataset may be freely assessed at: https://zenodo.org/record/5,088,346 (Lopez et al., 2021). We selected the BEIP dataset as its study design facilitated testing HAPPILEE on EEG data from children across a range of caregiving conditions, behavioral/clinical phenotypes, and ages (Zeanah et al., 2009). The optimization dataset includes resting-state EEG from three groups of children living in Romania starting in 2001. The first group, referred to as the Care as Usual Group (CAUG), is composed of children living across six institutionalized care facilities throughout Bucharest, Romania. The second group is the Foster Care Group (FCG), which is composed of children who were removed from these institutions through random assignment and placed in a foster care intervention. The final group is the Never Institutionalized Group (NIG), made up of a community sample of children living with their biological families who have never been placed in institutionalized care or foster care. We selected a subset of thirty EEG files across the three groups from the greater dataset (CAUG, $n = 8$; FCG, $n = 8$, NIG, $n = 14$). The average age at the start of the baseline assessment across the three groups was 17.40 months, with a range of 6.28–29.98 months (averages per group: CAUG=18.30; FCG=16.21; NIG=17.58). The resting-state EEG for all children was recorded with the James Long system from twelve scalp sites (F3, F4, Fz, C3, C4, P3, P4, Pz, O1, O2, T7 and T8) using a lycra Electro-Cap (Electro-Cap International Inc., Eaton, OH) with sewn-in tin electrodes (the two mastoid sites, M1 and M2, were removed due to poor recording quality across the dataset).

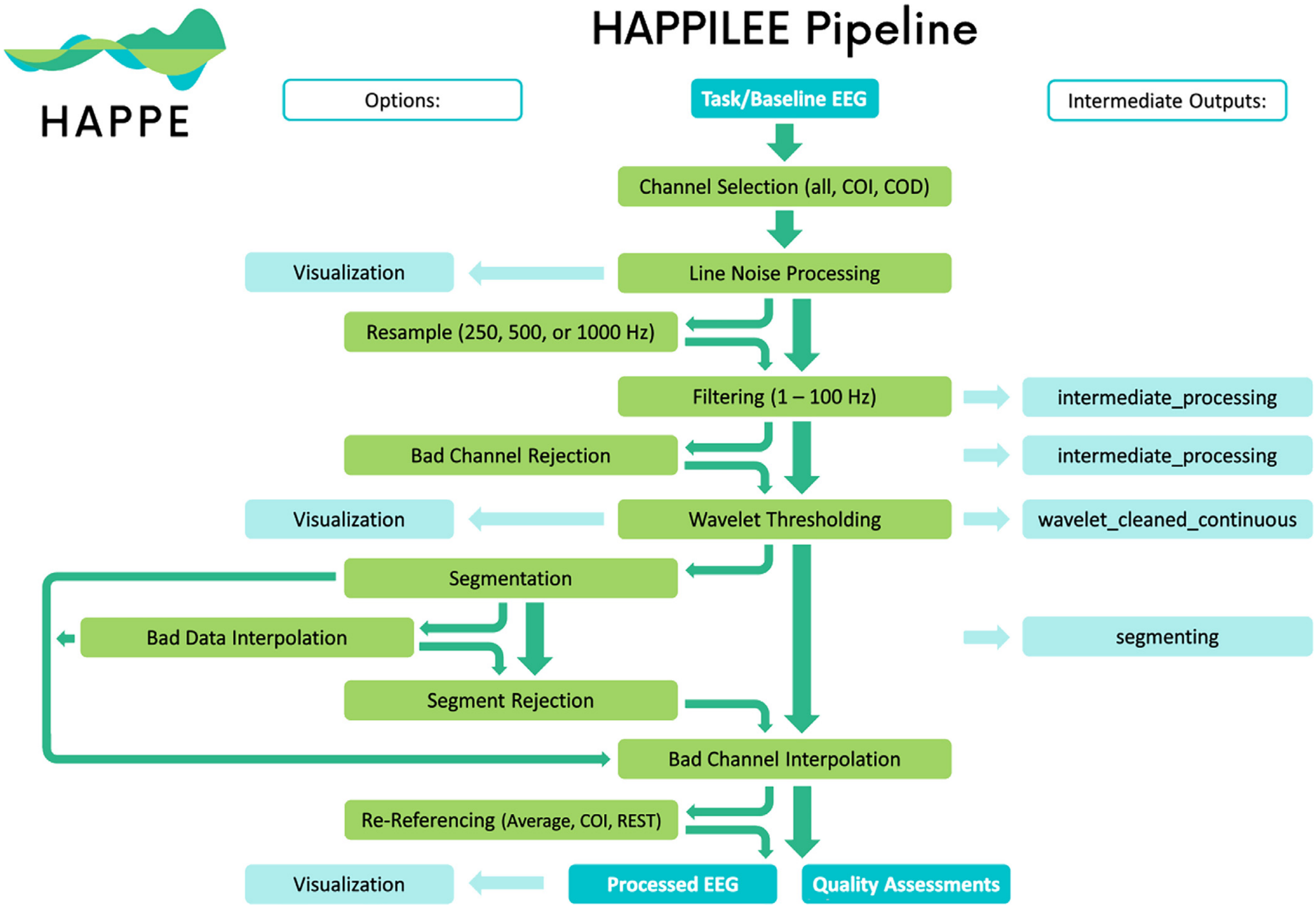


Fig. 2. Schematic illustrating the HAPPILEE pipeline's processing steps. The intermediate output EEG files are indicated by the suffix added after that specific processing step in the light blue boxes. The user options for resampling, segmentation, bad data interpolation, segment rejection, and re-referencing steps and visualizing several steps in HAPPILEE with the semi-automated setting are also indicated.

1.2. HAPPILEE data inputs

HAPPILEE accommodates multiple types of EEG files with different acquisition layouts as inputs. A single run will support only a single file type across files, specified by the user. For .set formated files, the correct channel locations should be pre-set and embedded in the file (e.g. by loading it into EEGLAB and confirming the correct locations) prior to running through HAPPILEE. When running .mat formated files, you must have a file with channel locations specified in your folder in order to run all steps in the HAPPILEE pipeline. If channel locations are not provided, you will not be able to do the following: filter to channels of interest, detect and reject bad channels, interpolate bad channels, or re-reference your data. Each batch run of HAPPILEE must include files collected with the same channel layout (company, net type, and electrode number) and paradigm (resting-state or event-related), each of which users must specify for a given run. HAPPILEE processes data collected with any sampling rate, and files within a single run may differ in their individual sampling rates (if this is the case, we strongly recommend selecting the option to resample data to the same frequency to ensure subsequent steps perform comparably across files regardless of original sampling rate).

1.3. Line noise processing

HAPPILEE addresses electrical noise (e.g., 60 or 50 Hz artifact signal) through the multi-taper regression approach implemented by the CleanLineNoise program (Mullen, 2012). Multi-taper regression can detect and subtract regular sinusoidal signal at a given frequency (e.g. electrical noise) without sacrificing or distorting the underlying EEG signal at that frequency or nearby frequencies, drawbacks of the notch-filtering approach to line-noise processing (Mitra and Pesaran, 1999). Specifically, HAPPILEE applies the updated version of CleanLine's multi-taper regression (called CleanLineNoise, implemented in the PREP pipeline; Bigdely-Shamlo et al. 2015b) which is more effective in addressing line noise than the original CleanLine version present in HAPPE 1.0 (Gabard-Durnam et al., 2018a) software (purportedly a bug fix in the CleanLine code, see Makoto's pipeline page for unpublished evidence: https://sccn.ucsd.edu/wiki/Makoto%27s_preprocessing_pipeline#Why_does_IC_rejection_increase_gamma_power.2C_or_why_is_an_IC_not_broadband-independent). The legacy CleanLine version from HAPPE 1.0 (Gabard-Durnam et al., 2018) is available as an option to the user, however the updated version is registered as the default. CleanLineNoise's multi-taper regression scans for line-noise signal near the user-specified frequency ± 2 Hz, a 4-s window with a 1-s step size and a smoothing tau of 100 during the fast Fourier transform, and a significance threshold of $p = 0.01$ for sinusoid regression coefficients. This process is highly specific to the frequency of electrical noise, which the user can specify to be 60 Hz or 50 Hz. The user may also specify line-noise harmonic frequencies to be similarly cleaned (e.g. 30, 15 Hz, etc.) or neighboring frequencies to be cleaned (e.g. 59 and 61 Hz for a 60 Hz electrical signal). Quality control metrics for the degree of regular sinusoidal signal removal at line noise frequency/frequencies are automatically generated in HAPPILEE and discussed in detail as part of the subsequent "HAPPILEE Pipeline Quality Assessment Report" section of this manuscript.

1.4. Filtering

Filtering the EEG signal is important for isolating frequencies of interest and improving signal-to-noise (e.g., for ERP analyses or isolating frequencies within the range produced by EEG's electrophysiological sources) but can also distort the data in undesirable ways if attention is not paid to filtering settings both alone and in combination with other pre-processing steps (e.g., baseline correction, line-noise reduction steps; see Widmann et al. 2015 and Tanner et al. 2015 for thorough discussion of issues related to filtering EEG signals). For example, HAPPILEE uses CleanLine for line-noise removal rather than band-stop filtering (aka notch filtering) to avoid signal distortion (e.g., Luck 2005).

HAPPILEE uses the EEGLab filter pop_eegfiltnew (a zero-phase Hamming-windowed sinc FIR filter) to apply preliminary filtering prior to detecting bad channels (if user-selected) and employing artifact correction methods for all files. If processing resting-state EEG or task-based data for time-frequency analyses, the filtering at this stage is a band-pass filter from 1 to 100 Hz. Filtering at this stage of pre-processing allows the bad channel and artifact correction steps to assess the relevant frequencies for these types of analyses and optimizes performance in the higher frequency range. If the user selects the ERP option when entering user-inputs at the start of HAPPE 2.0, HAPPILEE applies a low-pass 100 Hz filter to aid the artifact correction steps that follow but does not apply any low-pass filtering at this stage. The liberal 100 Hz low-pass filter optimizes bad channel detection and artifact-correction performance with respect to EMG and other high-frequency artifact contamination in the data that occurs within but also beyond the higher frequencies typically-included in filtered ERPs. HAPPILEE then automatically interfaces with the HAPPE+ER pipeline for further filtering specific to ERPs after artifact correction (e.g., high-pass and low-pass filtering at user-specified values like 0.1 to 30 Hz, with options for filter type. See Monachino et al. 2021) for specifics on ERP filtering. Users may assess or explore the effects of different filter settings, i.e., filter type and frequency boundaries, on artifact-corrected data to optimize their ERP filtering using HAPPE's rerun functionality).

1.5. Bad channel detection (Optional)

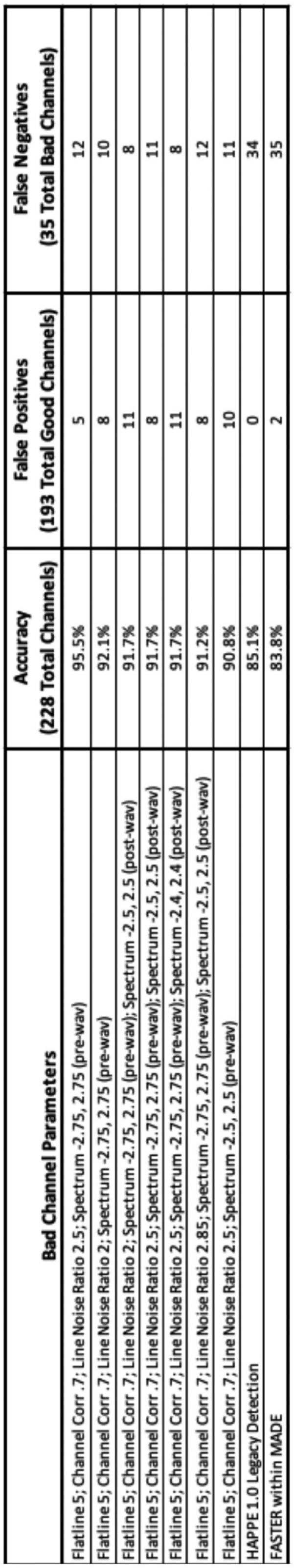
HAPPILEE includes an option to detect channels that do not contribute usable brain data due to high impedances, damage to the electrodes, insufficient scalp contact, and excessive movement or electromyographic (EMG) artifact throughout the recording. Users have the option to run bad channel detection or not (in which case all channels are subjected to subsequent processing steps). Various methods are currently used to detect and remove bad channels across automated pipelines. However, some common automated detection methods used for high-density EEG may not be optimal for low-density EEG without modification, especially those relying heavily on standard deviation-related metrics of activity to detect outlier channels. For example, in HAPPE 1.0 (Gabard-Durnam et al., 2018), bad channel detection is achieved by evaluating the normed joint probability of the average log power from 1 to 125 Hz across the user-specified subset of included channels. Channels whose probability falls more than 3 standard deviations from the mean are removed as bad channels in two iterations of this bad channel detection step. However, removing channels that are three or more standard deviations from the mean activity assumes a normal distribution of channel activities (via the Central Limit Theorem) that we cannot assume with low-density channel numbers (Altman and Bland, 1995). Similarly, the FASTER algorithm (Nolan et al., 2010) used in the MADE pipeline (Debnath et al., 2020) flags channels by measuring each channel's Hurst exponent, correlation with other channels, and channel variance and standardizing the three values with an absolute Z-score (subject to the same constraints as standard deviations with very small samples). Thus, these algorithms require validation before implementation in low-density EEG data.

Other methods like EEGLab's Clean Rawdata algorithm (Kothe and Makeig 2013, additional code developed by Makoto Miyakoshi, Arnaud Delorme with Scott Makeig) may more readily translate to low-density EEG data. Specifically, Clean Rawdata's 'Flatline Criterion,' can detect channels with flat recording lengths longer than a user-specified threshold of seconds (indicating no data collected at that location). If the channel contains a flatline that lasts longer than the threshold, the channel is marked bad. Similarly, 'Channel Correlation Criterion' sets the minimally acceptable correlation value between the channel in question and all other channels. If a channel is correlated at less than the preset value to an estimate based on other channels, it is considered abnormal and marked bad. But features like the Line Noise Ratio Criterion, which identifies whether a channel has more line noise relative to neural signal than a predetermined value, in standard deviations based on the total channel population, should be assessed in the low-density EEG context.

To test the efficacy of FASTER (Nolan et al., 2010), HAPPE 1.0 (Gabard-Durnam et al., 2018), and Clean Raw data functions and determine the optimal criterion values for the detection of bad channels in low density data, we compared a series of thirty-three automated options to a set of manually identified bad channels for nineteen files in the BEIP dataset. For manual identification of bad channels, we took the ratings of three field experts and only selected files where agreement across reviewers was reached, ensuring that we were using clear-cut cases of good and bad channels within the optimization dataset. These files included channels that were bad for a variety of reasons and had variability in how many bad channels existed per file. For automated bad channel rejection, the files were run through the HAPPE 1.0 (Gabard-Durnam et al., 2018) legacy detection method for bad channels and the FASTER (Nolan et al., 2010) detection method used in the MADE pipeline (Debnath et al., 2020), as well as a number of iterations of the Clean Rawdata function and combinations of Clean Rawdata with spectrum evaluation to optimize channel classification (shown in Table 1). Note that for iterations of Clean Rawdata with Flatline Criterion included, the Flatline default of 5 s was determined to be sufficient for detecting flat channels and was not manipulated further. We evaluated the outputs from each criterion for bad channel detection relative to the manually selected channels by summing the number of false negatives and false positives for each file and calculating the overall accuracy rate across files for that set of automated parameters. False negatives refer to channels that were manually marked as bad but not flagged as bad by the pipeline. False positives refer to channels that were manually marked 'good' but were marked bad by the pipeline. An extra emphasis was placed on finding the settings with high accuracy that produced the lowest number of false positives in order to avoid getting rid of usable channels in the low-density dataset. HAPPILEE's optimal settings produced 12 false negative and 5 false positive channels across all 19 files (228 total channels), with an overall accuracy rate of 95.5%.

HAPPILEE combines EEGLab's Clean Rawdata functions with power spectral evaluation steps as follows. HAPPILEE first runs the Clean Rawdata 'Flatline Criterion,' to detect bad channels with flat recording lengths longer than 5 s (indicating no data collected at that location). After flat channels have been removed, HAPPILEE uses Clean Rawdata's 'Line Noise Ratio Criterion' with a threshold of 2.5 standard deviations (channels with line noise: neural data ratios greater than 2.5 standard deviations are marked as bad) and 'Channel Correlation Criterion' with a minimal acceptable correlation of 0.7 to detect additional bad channel cases. Finally, HAPPILEE includes a spectrum-based bad channel detection step following the Clean Rawdata functions. While the HAPPE 1.0 (Gabard-Durnam et al., 2018) method of legacy detection proved to be insufficient for our low-density dataset (see Table 1), evaluating the joint probability of average log power from 1 to 100 Hz was useful for optimizing bad channel detection alongside Clean Rawdata. A spectrum evaluation step with thresholds of −2.75 and 2.75 was included to optimize bad channel detection accuracy. Thus, HAPPILEE achieves bad channel detection that is suitable for low density data and expands the classes of bad channels that can be detected relative to HAPPE 1.0

Table 1
Performance of bad channel detection parameters tested on nineteen files from the example dataset. Pre-wav denotes that spectrum evaluation was run prior to wavelet thresholding and post-wav denotes that spectrum evaluation was run following wavelet thresholding.



Bad Channel Parameters	Accuracy (228 Total Channels)	False Positives (193 Total Good Channels)	False Negatives (35 Total Bad Channels)
Flatline 5; Channel Corr .7; Line Noise Ratio 2.5; Spectrum -2.75, 2.75 (pre-wav)	95.5%	5	12
Flatline 5; Channel Corr .7; Line Noise Ratio 2; Spectrum -2.75, 2.75 (pre-wav)	92.1%	8	10
Flatline 5; Channel Corr .7; Line Noise Ratio 2; Spectrum -2.75, 2.75 (pre-wav); Spectrum -2.5, 2.5 (post-wav)	91.7%	11	8
Flatline 5; Channel Corr .7; Line Noise Ratio 2.5; Spectrum -2.75, 2.75 (pre-wav); Spectrum -2.5, 2.5 (post-wav)	91.7%	8	11
Flatline 5; Channel Corr .7; Line Noise Ratio 2.5; Spectrum -2.75, 2.75 (pre-wav); Spectrum -2.4, 2.4 (post-wav)	91.7%	11	8
Flatline 5; Channel Corr .7; Line Noise Ratio 2.85; Spectrum -2.75, 2.75 (pre-wav); Spectrum -2.5, 2.5 (post-wav)	91.2%	8	12
Flatline 5; Channel Corr .7; Line Noise Ratio 2.5; Spectrum -2.5, 2.5 (pre-wav)	90.8%	10	11
HAPPE 1.0 Legacy Detection	85.1%	0	34
FASTER within MADE	83.8%	2	35

(Gabard-Durnam et al., 2018) and MADE pipelines' (Debnath et al., 2020) prior automated pipeline options.

1.6. *Artifact correction in continuous data*

Raw EEG data may contain a number of artifacts (e.g., from participant motion, electromyogenic activity, eye movements/blinks) that must be addressed during processing. Historically, artifact removal has been achieved via manual data inspection, where artifact-laden timepoints are deleted from the data, including artifact-free data from unaffected electrodes (i.e. artifact rejection approach). HAPPILEE instead uses wavelet thresholding methods for artifact correction (correcting artifacts without removing any timepoints) first to allow for fewer segments or trials to be rejected in subsequent artifact rejection steps and address artifacts that would survive segment rejection but could still impact the integrity of further analyses. This artifact correction approach is performed on each electrode independently in HAPPILEE, so it is appropriate for all channel densities down to single-electrode recordings, and its performance is channel-density independent (i.e., wavelet-thresholding will not perform differently in higher- vs. lower-electrode densities). These properties make wavelet-thresholding an excellent option for serving a variety of low-density EEG layouts.

Wavelet-thresholding refers to a series of three steps performed on each electrode:

Step 1) Apply the wavelet transform. Each electrode's time series is subjected to a wavelet transform by fitting a wavelet function to the EEG data to represent the signal and parse it into multiple frequency ranges (akin to frequency bands). Wavelet functions are orthonormal basis functions that come in a variety of shapes (called families) and are used for a variety of signal compression, denoising, and representation applications (including the authors' personal favorite of detecting Van Gogh forgery paintings by fitting wavelet functions, Jafarpour et al., 2009). Wavelet functions used with electroencephalography resemble the oscillatory shape of EEG data (see Fig. 5 for examples in optimization section below). These wavelet functions also have excellent temporal resolution, so they can accurately represent both time and frequency information simultaneously. The wavelet transform passes the selected wavelet function over the EEG signal and produces a series of coefficients to describe how the wavelet function changed to fit the EEG signal's fluctuations across the entire timeseries. Importantly, poor function selection could result in poor fitting of the EEG signals, so the type of wavelet function used is important for the integrity of artifact-correction. The transform also separates the EEG signal into multiple frequency ranges and coefficients evaluate fluctuations in the signal within each frequency range separately. In this way, the wavelet transform operates like a frequency filter on the EEG signal, though one that is subsequently completely reversed and leaves no trace in the retained EEG data. The wavelet function has a resolution level (i.e., level of decomposition) associated with it that determines how many of these frequency ranges the EEG signal should be separated out into, with increasing resolution levels here parsing the lower frequencies into finer frequency bins. If the resolution level is not determined appropriately, unnecessary loss of low-frequency information can occur during thresholding (described below). Once the wavelet transform separates out and fits the EEG signal in these frequency bins in Step 1, artifacts may be detected in Step 2.

Step 2) Threshold the data to isolate artifact signal. The wavelet transform coefficients that describe the EEG signal within each frequency bin are subjected to a thresholding procedure to separate out artifact signal from neural signal. This feature allows for frequency-band-specific artifact detection relative to neural data in those same frequencies, which is a key component of wavelet-thresholding's success in EEG signals. Given that artifact signal is larger than the neural signal found at the same frequencies across all classes of artifact, and occurs more inconsistently throughout the timeseries than neural-related fluctuations, the wavelet coefficients reflecting these properties are separated out as artifact-signal to be removed from the data using a thresholding method and rule. If the wavelet resolution level is not set appropriately, low-frequency neural data, which typically have higher amplitudes than higher-frequency data in EEG signals, can be mistakenly removed as artifacts. There are multiple methods for determining the threshold for determining signal to be artifact vs. neural. HAPPILEE uses an empirical Bayesian method that learns from each individual's EEG where to set the threshold and is less sensitive to outlier effects (Clyde and George, 2000). This threshold may be determined and applied across all frequency ranges (level-independent threshold) or determined individually for each frequency range based on the characteristics of the signal in that frequency range (level-dependent threshold). HAPPILEE uses a level-dependent threshold to best fit the artifact properties occurring within each frequency range. Once the threshold has been applied to determine which wavelet coefficients (reflecting parts of the EEG signal) are above the threshold, they can be removed as artifact-related signal. There are also multiple methods for treating these thresholded coefficients in terms of how they are separated from the rest of the signal called threshold rules, but HAPPILEE uses a hard threshold that completely removes the sub-threshold coefficients from the data (i.e., completely separates the signal classified as neural from the artifact-related signal). Other rules like the soft threshold instead downweight the sub-threshold coefficients closest to the threshold cutoff in the data, for example. If these steps together are not optimized for EEG signals, incomplete artifact correction, or attrition of neural signal can both occur, so great care was taken in optimizing these parameters for HAPPILEE (described below). Once the artifact-signal is fully separated from the neural signal, it can be removed from the electrode timeseries without disturbing the underlying neural signal at those timepoints via the inverse wavelet transform and subtraction.

Step 3) Apply the inverse wavelet transform and subtract thresholded (artifact) signal. Finally, the artifact-related coefficients are transformed back from wavelet-coefficients to the electrode's signal timeseries using the inverse of the wavelet transform function. The HAPPILEE wavelet function family allows for this transform without distortion to the data in phase, amplitude, or frequency space. This inverse transform thus results in an artifact-timeseries that is then simply subtracted from the electrode's original timeseries, resulting in an artifact-corrected timeseries. Because waveleting is both a time- and frequency-specific method, the artifact timeseries will be 0 where no artifact is temporally present in the data, and thus this subtraction does not disturb the clean EEG signal surrounding the artifact-contaminated segment at all (this distinguishes wavelet-thresholding from other artifact-correction strategies like independent component analysis that in practice do not consistently meet this standard in the artifact-labeled components). The wavelet-thresholded artifact-corrected signal may then continue through pre-processing steps like segmentation.

Through only three steps, the wavelet-thresholding process contains many parameters that may be optimized to improve artifact-correction performance. The final wavelet-thresholding parameters implemented in HAPPILEE are as follows: 'Wavelet Family,' 'coif4;' 'Level of Decomposition,' '10;' 'Noise Estimate,' 'Level Dependent;' 'Thresholding Method,' 'Bayes;' 'Threshold Rule,' 'Hard'. The approaches and steps to optimize wavelet-thresholding artifact correction are detailed below, with additional details about each component of the wavelet-thresholding process.

1.6.1. Wavelet threshold optimization approaches

Three approaches were taken to test and optimize automated wavelet thresholding-based artifact correction in HAPPILEE using the BEIP dataset. Prior to artifact correction testing, all files were initially filtered and subjected to line noise correction. The approaches are detailed below.

The first approach (the clean vs. artifact approach) involved selecting two 30 s segments within each participant's EEG file. The first 30 s segment was heavily artifact laden while the second 30 s segment was determined to be clean and without considerable artifact by experienced researchers. This approach facilitated testing whether artifact was effectively and accurately removed via wavelet thresholding. That is, optimal artifact correction performance would be characterized by (1) substantial (artifact) signal removal for the artifact-laden 30 s file, indicating sensitivity to artifacts, and (2) minimal signal removal for the clean 30 s file from the same individual, indicating specificity in signal removal constrained to artifact. High levels of signal removal in the clean 30 s files would indicate unnecessary data loss, while low levels of signal removal in the artifact-laden 30 s files could indicate insufficient performance. Additionally, to ensure that artifact removal was not biased to certain frequencies, data correlations pre- and post-processing were evaluated at key frequencies spanning all canonical frequency bands in the clean and artifact-laden segments. Examples of the clean and artifact signals within an individual (before and after wavelet thresholding) are provided in Fig. 3.

The second approach (artifact-addition approach) used known artifact signals to establish how much artifact could be removed without distorting underlying neural signals during wavelet thresholding. In the absence of a ground-truth neural signal, we used the 30 s clean files from the first approach as the signal to be recovered during artifact-correction. We then isolated artifact timeseries by running ICA on the artifact laden 30 s files and selecting approximately 2 components per individual that were determined to be artifact with minimal neural data via visual inspection and automated classification through both the ICLabel and Multiple Artifact Rejection Algorithm options. We subsequently added those artifact timeseries to the clean 30 s data segment from the same individual (see Supplemental File 1 for types of artifacts added for each file). These 30 s files with added artifact were then subjected to wavelet-thresholding and compared (via correlation coefficients) to the clean 30 s files run through wavelet-thresholding to determine how much of the added artifact was removed during the artifact-correction step (example from single participant in Fig. 4). Higher correlation coefficients would indicate better recovery of the clean EEG signal and better removal of the added artifact during the wavelet-thresholding step. (The post-wavelet thresholded clean files were used as the comparison because even these clean files had some minor artifacts that could be removed during artifact correction, so the post-wavelet-thresholded versions were the cleanest option for comparisons).

For the third approach, we used simulated EEG data with artifact added to it in order to have a "ground truth" signal that we could attempt to recover with waveleting. To create the simulated EEG data, we used code from Bridwell et al. (2018). In short, the simulated EEG consisted of four signals. The four signals had distinct spatial patterns and frequency ranges (1.00–3.91, 3.91–7.81, 7.81–15.62, and 15.62–31.25 Hz). For a more thorough description on how the simulated signals were created, (see Bridwell et al. 2018). After creating the simulated EEG data, we added various developmental artifacts and adult eye blinks to the data to test multiple waveleting settings and wavelet thresholding in general. To get the blink artifact, we used a clear blink independent component (IC) from an adult participant (see Leach et al. 2020 for the specific study details). This IC was selected based on both an automated artifactual IC detection algorithm and visual inspection by two researchers with over five years of EEG and at least two years of ICA experience. For the de-

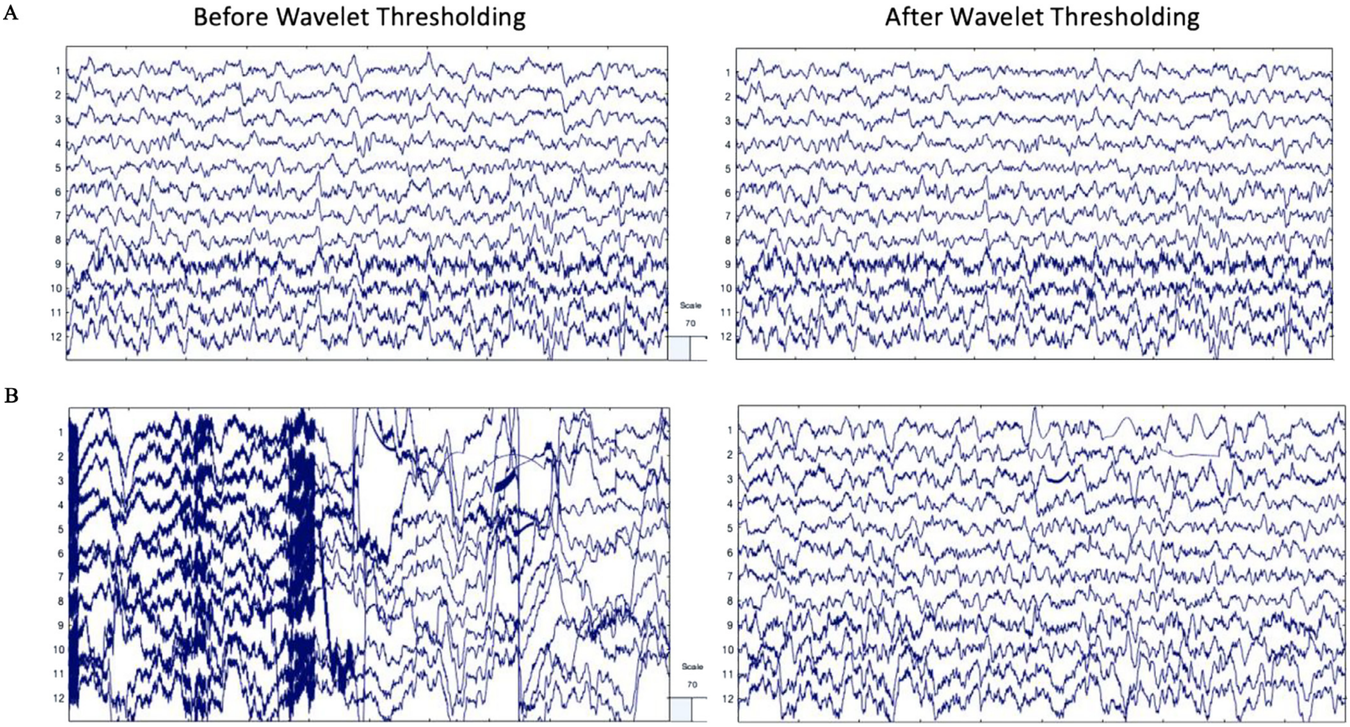


Fig. 3. EEG signal before and after wavelet thresholding with the following parameters that were optimized for low density data: 'Wavelet Family,' 'coif4;' 'Level of Decomposition,' '10;' 'Noise Estimate,' 'Level Dependent;' 'Thresholding Method,' 'Bayes;' 'Threshold Rule,' 'Hard.' Two files from the same participant in the example dataset are shown with 10 s of data extracted from the clean 30 s segment (A) and artifact-laden 30 s segment (B). The EEG signal before processing is shown in the left panel. The EEG signal after wavelet thresholding is shown in the right panel. All scales are in microvolts.

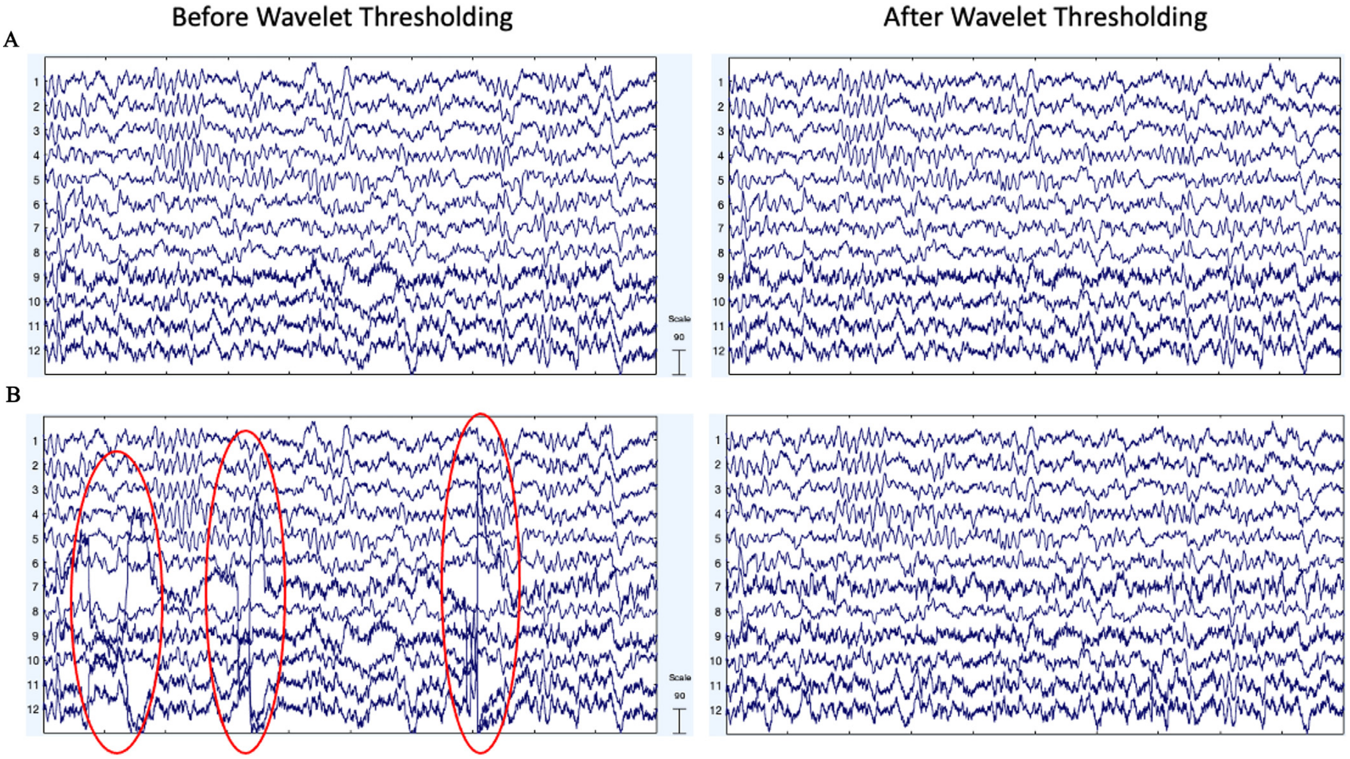


Fig. 4. EEG signal before and after wavelet thresholding with the following parameters that were optimized for low density data: 'Wavelet Family,' 'coif4;' 'Level of Decomposition,' '10;' 'Noise Estimate,' 'Level Dependent;' 'Thresholding Method,' 'Bayes;' 'Threshold Rule,' 'Hard.' Two files from the same participant in the example dataset are shown with 10 s of data extracted from the clean 30 s segment (A) and artifact-added 30 s segment (B). The EEG signal before processing is shown in the left panel. The EEG signal after wavelet thresholding is shown in the right panel. All scales are in microvolts.

velopmental artifact, we pulled eight ICs from the BEIP dataset used above in the artifact addition approach. After adding the artifact to the simulated EEG data, we did a 1 Hz highpass and 35 Hz lowpass filter. Following this, we epoched the data into two-second epochs (50% overlap) to prepare the data for wavelet thresholding and/or artifact rejection. For artifact rejection, we used a −100 to 100 μV voltage threshold to identify bad epochs. We also required both frontal electrodes to exceed this threshold in order to classify an epoch as containing a blink. In order to test how well wavelet thresholding removes artifact without removing neural activity, we computed the power spectral density (PSD) on the original simulated signal (no artifact added) and compared that to the PSD of the signal after pre-processing with various wavelet thresholding parameters when either one or two developmental artifactual independent components and an adult eye blink component are added to the simulated signal. In addition, we looked at how the preprocessed data might differ when only artifact rejection is run compared to when waveleting is run before artifact rejection. This gave us two preprocessing conditions: (1) Artifact rejection only (no wavelet thresholding) and (2) Wavelet thresholding followed by artifact rejection.

These three distinct approaches facilitated optimizing and evaluating the wavelet thresholding method for artifact removal in low-density data in multiple ways. Importantly, the wavelet thresholding approach broadly includes decomposing the EEG signal via a wavelet transform, determining a threshold value or values used to dissect data into the portion to be retained and the portion to be rejected, removal of the rejected data components, and reconstruction of the remaining signal. Each of these steps may be accomplished multiple ways across wavelet thresholding methods. Here, five key parameters in the wavelet thresholding process were manipulated and tested to optimize wavelet thresholding performance in this context, specifically: wavelet family, wavelet resolution (i.e., level of data decomposition), noise estimate method, thresholding level, and threshold rule. Each parameter was manipulated one at a time within a default set of wavelet thresholding parameters and tested using the clean vs. artifact approach and artifact-added approach in the BEIP dataset. The initial default set of wavelet thresholding parameters was chosen based on preliminary visual inspection of performance across a broader range of parameters prior to optimization and was as follows: ‘Wavelet Family,’ ‘coif5;’ ‘Level of Decomposition,’ ‘8;’ ‘Noise Estimate,’ ‘Level Dependent;’ ‘Thresholding Method,’ ‘Bayes;’ ‘Threshold Rule,’ ‘Soft.’ Subsequent optimization of each parameter is described in detail below.

1.6.2. Wavelet family

The wavelet-thresholding method first subjects each electrode’s time series to wavelet transform by fitting a wavelet function to the data. The wavelet transform produces a series of coefficients to describe the EEG signal’s fluctuations across multiple frequency ranges. The wavelet function consists of both a wavelet family, dictated by the mother wavelet shape (e.g., the Coiflet mother wavelet is more symmetric than the Daubechies mother wavelet), and the wavelet order, which modifies the mother wavelet shape (see Fig. 5; i.e., Coiflet order 4 wavelet has 8 vanishing moments in the function). To find the optimal wavelet function to carry out stationary wavelet transform, we tested the Coiflets family (orders 3, 4, and 5), the Daubechies family (orders 4 and 10), and the Symlets family (order 4). We selected these family/order combinations as they share shapes similar to those found in EEG signals and they are all orthogonal wavelet functions, which optimizes decomposition and reconstruction of the EEG signal from the wavelet transform (Strang and Nguyen, 1996). Moreover, prior literature indicates these wavelet families and specific orders have performed well on electrophysiological data (Al-Qazzaz et al., 2015; Alyasseri et al., 2017; Harender and Sharma, 2018; Lema-Condo et al., 2017; Nagabushanam et al., 2020). Using these wavelet families, we evaluated whether there was biased data removal at any of the data frequencies in the clean 30 s segments by evaluating correlations between data pre-waveleting and post-waveleting at specific canonical frequencies. We also compared data removal rates between the clean files and the 30 s artifact laden files. As a second analysis, we evaluated which wavelet family and order combination removed the most added artifact across frequencies (i.e., which wavelet facilitated the greatest correlation between the artifact added files and the clean files post-processing).

After running the various wavelet family/order options, we found there were not meaningful differences between several wavelet family/order options (Tables 2 and 3). Specifically, performance did not differ between coif4, coif5, db4, and sym4 options across the clean vs. artifact-laden and artifact addition tests (e.g., correlation values in the artifact-addition tests were identical to the hundredths place). Coif4 was selected as the wavelet implemented in HAPPILEE as this wavelet/order also performed very well in data collected from a saline-based system (EGI), suggesting its performance may generalize more broadly than the other options tested here (Monachino et al., 2021).

1.6.3. Wavelet resolution/level of data decomposition

Following wavelet family/order selection, we manipulated the resolution of the wavelet that affects the level of data decomposition in wavelet thresholding. Specifically, this level of decomposition determines how fine-grained the frequency bands are in which data correction occurs. Importantly, in the current code, the data sampling rate (not for example, the frequencies retained through initial filtering) determines which frequencies fall into different levels of decomposition. For example, the first level of decomposition for a file sampled at 1000 Hz (regardless of frequency filtering) would split data into two halves around 500 Hz. If that file had been resampled to 500 Hz prior to wavelet decomposition, the first level of decomposition would now split data into halves around 250 Hz. The default decomposition level splits data down to ≤1 Hz. We tested decomposition levels of ∼4, 2, and 1. The 4 Hz decomposition resulted in increased data removal in the lower frequencies relative to other frequencies of the clean data, resulting in data correlations pre-/post-thresholding of less than 0.5 (e.g., $r = 0.48$ at 2 Hz). This pattern indicated the need for further decomposition levels to avoid over-rejecting low frequency data that is not artifact-laden (low frequencies were rejected at similar rates in the artifact-laden data). In the clean vs. artifact files, we saw no difference in which data was rejected and retained when comparing levels 2 and 1 Hz, though 1 Hz provides coverage down to the filtering cutoff for time-frequency analyses. Therefore, HAPPILEE decomposes data into detail coefficients for frequencies above approximately 1 Hz to evaluate artifacts within each frequency range for time-frequency-related analyses. For event-related-potential decomposition optimization where signals below 1 Hz are relevant, HAPPILEE interfaces with the HAPPE+ER pipeline and uses those settings, so we refer readers to the HAPPE+ER manuscript (Monachino et al., 2021).

1.6.4. Noise (Artifact) estimation level

Once the level of decomposition is set, the noise estimate parameter is chosen to establish either a threshold for each level of decomposition (level dependent threshold) or establish a threshold that operates across all of the levels of decomposition (level independent threshold). The threshold(s) determine which wavelet coefficients describe data that is artifact-laden (i.e., coefficients describing larger amplitude changes, or noise) and will be removed from the EEG data during this artifact correction step. Due to the unavailability of level dependent thresholding for the wavelet function used at the time of its conception, HAPPE 1.0 (Gabard-Durnam et al., 2018) employed level independent thresholding (with a different threshold method and rule as well). We anticipated that level dependent thresholding would improve artifact detection specificity because of its ability to scale within each frequency range (e.g., artifacts in gamma frequencies have smaller amplitudes than artifacts in delta frequencies), rather than apply across all frequencies at once (which may over-penalize low-frequency clean EEG data that has higher amplitudes than higher-frequency clean EEG data). The default

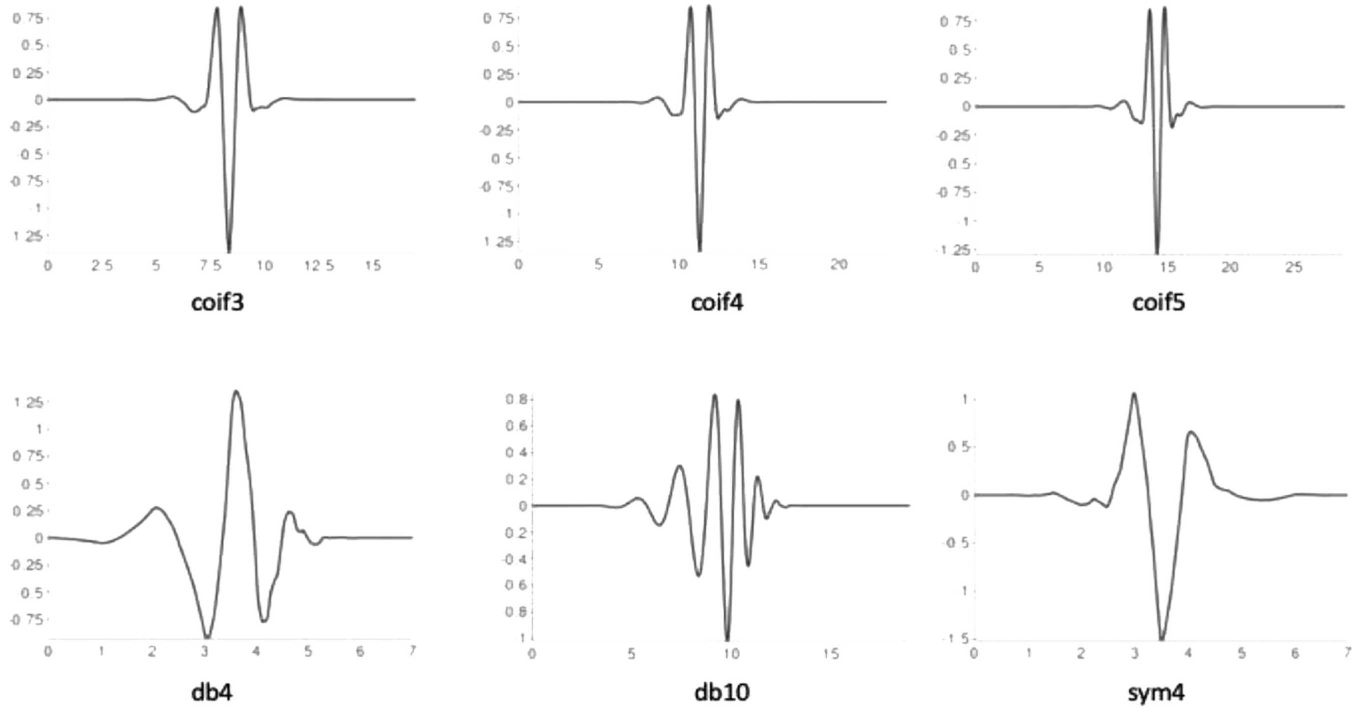


Fig. 5. Visualization of the three wavelet families with selected orders tested to find the optimal wavelet function to carry out stationary wavelet transform on low density data.

Table 2
Artifact-laden vs. Clean Approach. Correlations between the EEG signal before wavelet thresholding and the EEG signal after wavelet thresholding for the wavelet parameters tested on the artifact-laden and clean 30 s segments. The *r* values of the wavelet parameters that are included in the final code are bolded. Asterisks denote default wavelet parameters.

Wavelet Parameters	Artifact-Laden									Clean								
Wavelet Family	Average r	2hz r	5hz r	8hz r	12hz r	20hz r	30hz r	45hz r	70hz r	Average r	2hz r	5hz r	8hz r	12hz r	20hz r	30hz r	45hz r	70hz r
coif3	0.6495386	0.8341557	0.9039538	0.9124215	0.8737657	0.9257608	0.9319297	0.8818718	0.9209791	0.8537765	0.9805714	0.9942900	0.9946382	0.9944396	0.9966822	0.9963891	0.9928368	0.9925715
coif4	**0.6409072**	**0.8224776**	**0.9108023**	**0.9119056**	**0.8958911**	**0.9317483**	**0.9272825**	**0.8977289**	**0.9105859**	**0.8502602**	**0.9708517**	**0.9935547**	**0.9951477**	**0.9969815**	**0.9960891**	**0.9951767**	**0.9932113**	**0.9943216**
coif5*	0.6317364	0.8042749	0.9112935	0.9148003	0.9120963	0.9395317	0.9472025	0.8771795	0.9280485	0.8452176	0.9560839	0.9916856	0.9950517	0.9980797	0.9969268	0.9968790	0.9926212	0.9934005
db4	0.6745547	0.8710232	0.8943305	0.9103185	0.8795906	0.9198904	0.9381887	0.8938951	0.9051647	0.8611718	0.9886946	0.9911826	0.9968351	0.9947068	0.9965864	0.9968433	0.9948863	0.9951650
db10	0.6723805	0.8680295	0.9070212	0.9379687	0.9118920	0.9343418	0.9473784	0.8934602	0.9262520	0.8554125	0.9852761	0.9963632	0.9960585	0.9942808	0.9982296	0.9978062	0.9944235	0.9922931
sym4	0.6645012	0.8491330	0.8886317	0.9131185	0.8758145	0.9270180	0.9423294	0.8911181	0.9225114	0.8590634	0.9869279	0.9901779	0.9954765	0.9930299	0.9948369	0.9963707	0.9915858	0.9934201
Level of Data Decomposition																		
Level 7	0.4706001	0.4568898	0.9105142	0.9142371	0.9110843	0.9388060	0.9466663	0.8768213	0.9274731	0.7070777	0.4835164	0.9911996	0.9949208	0.9976923	0.9966372	0.9966664	0.9924659	0.9930409
Level 8*	0.6317364	0.8042749	0.9112935	0.9148003	0.9120963	0.9395317	0.9472025	0.8771795	0.9280485	0.8452176	0.9560839	0.9916856	0.9950517	0.9980797	0.9969268	0.9968790	0.9926212	0.9934005
Level 9	0.7343196	0.8044431	0.9116424	0.9151761	0.9125530	0.9396270	0.9474459	0.8773602	0.9284372	0.9211428	0.9566750	0.9917687	0.9951122	0.9982365	0.9970986	0.9969594	0.9927146	0.9935756
Level 10	0.7565080	0.8043941	0.9116741	0.9152546	0.9126432	0.9397223	0.9475151	0.8774575	0.9285303	0.9320311	0.9567118	0.9918028	0.9951221	0.9982588	0.9971213	0.9969784	0.9927402	0.9936028
Level 11	0.7593162	0.8044069	0.9116858	0.9152531	0.9126547	0.9397121	0.9475120	0.8775217	0.9285750	0.9326935	0.9567205	0.9918067	0.9951225	0.9982542	0.9971241	0.9969778	0.9927416	0.9936031
Noise Estimate Method																		
Level Independent	0.0119531	0.0687366	0.0632083	0.0689387	0.0571533	0.0656005	0.0638888	0.0640006	0.5991267	0.0192180	0.0614340	0.0673882	0.0777877	0.0620493	0.0694187	0.0673803	0.0620892	0.5678356
Level Dependent*	**0.6317364**	**0.8042749**	**0.9112935**	**0.9148003**	**0.9120963**	**0.9395317**	**0.9472025**	**0.8771795**	**0.9280485**	**0.8452176**	**0.9560839**	**0.9916856**	**0.9950517**	**0.9980797**	**0.9969268**	**0.9968790**	**0.9926212**	**0.9934005**
Thresholding Level																		
Bayes*	**0.6317364**	**0.8042749**	**0.9112935**	**0.9148003**	**0.9120963**	**0.9395317**	**0.9472025**	**0.8771795**	**0.9280485**	**0.8452176**	**0.9560839**	**0.9916856**	**0.9950517**	**0.9980797**	**0.9969268**	**0.9968790**	**0.9926212**	**0.9934005**
SURE	0.5449277	0.6324712	0.7956184	0.8398051	0.8147391	0.8727297	0.8813146	0.8067312	0.8455665	0.7744327	0.7795935	0.9024834	0.9463993	0.9466939	0.9756644	0.9755786	0.9616694	0.9712791
Univeral Threshold	0.6921037	0.9021545	0.9488209	0.9389146	0.9389948	0.9616815	0.9627782	0.9054370	0.9486801	0.8567709	0.9949188	0.9974719	0.9978736	0.9994394	0.9981523	0.9983733	0.9962493	0.9965857
Minimax	0.6631253	0.8515140	0.9199850	0.9170071	0.9098444	0.9366292	0.9460976	0.8805530	0.9274095	0.8509405	0.9764616	0.9914494	0.9944892	0.9950825	0.9953290	0.9949455	0.9898739	0.9909732
Threshold Rule																		
Soft*	0.6317364	0.8042749	0.9112935	0.9148003	0.9120963	0.9395317	0.9472025	0.8771795	0.9280485	0.8452176	0.9560839	0.9916856	0.9950517	0.9980797	0.9969268	0.9968790	0.9926212	0.9934005
Median	0.4977552	0.5418288	0.7692647	0.8255840	0.7900075	0.8392417	0.8637228	0.7758505	0.8155977	0.7981215	0.7915035	0.9417516	0.9731402	0.9702290	0.9837920	0.9798027	0.9621283	0.9677733
Hard	0.4278164	0.3920570	0.6961517	0.7790864	0.7520299	0.8090674	0.8273337	0.7401785	0.7599699	0.7506100	0.6286745	0.8964708	0.9542559	0.9431892	0.9744735	0.9703364	0.9484776	0.9579156
Hard (level 9)	0.4419678	0.3919517	0.6962050	0.7791285	0.7520823	0.8090638	0.8273489	0.7401665	0.7600296	0.76646649	0.6287149	0.8965116	0.9542537	0.9432158	0.9744641	0.9703503	0.9484915	0.9579139
Hard (level 10)	**0.4437741**	**0.3919728**	**0.6962142**	**0.7791177**	**0.7520895**	**0.8090850**	**0.8273534**	**0.7401521**	**0.7600314**	**0.76800752**	**0.6287386**	**0.8965451**	**0.9542584**	**0.9432336**	**0.9744747**	**0.9703579**	**0.9484932**	**0.9579214**
Hard (level 11)	0.4445122	0.3920010	0.6961893	0.7790936	0.7521045	0.8091205	0.827366	0.7401616	0.7600506	0.76834755	0.6287675	0.8965547	0.9542651	0.9432359	0.9744798	0.9703611	0.9484892	0.9579185

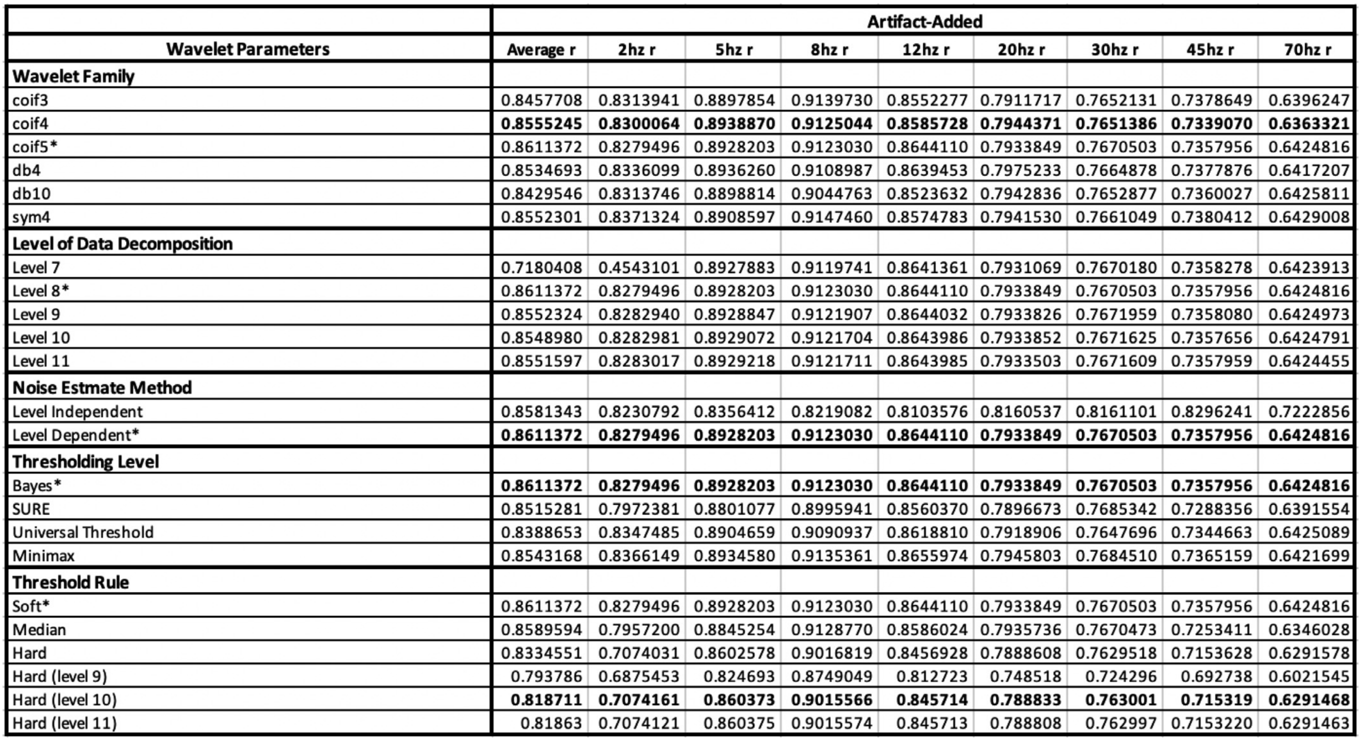
set of parameters was run with both level dependent and level independent thresholding and confirmed our prediction. With the improved thresholding method and rule included in the default parameters, the level independent threshold now heavily over-rejected the clean data (resulting in a correlation pre-/post-thresholding of $r = 0.02$), removing nearly all of the data (and a similar level of data removal was observed in the artifact-laden data). (Note: this performance differs from HAPPE 1.0 (Gabard-Durnam et al., 2018) due to the other wavelet thresholding parameter changes included in the new default settings of HAPPILEE, and does not reflect the functionality of wavelet-independent thresholding in HAPPE 1.0). There was no meaningful difference between level independent and level dependent thresholding in how much artifact was removed in the artifact-added approach. This pattern of results suggested the level dependent threshold was just as effective at removing artifact as the level independent threshold without also removing underlying clean neural signal. As a result, a level dependent threshold was chosen in order to preserve data without compromising artifact correction success.

1.6.5. *Thresholding method and threshold rule*

Wavelet coefficients are then subjected to thresholding, (here, in a level-dependent way) such that coefficients with values smaller than a determined threshold for that level have their contribution to the data substantially suppressed (similar to Jansen, 2001; You and Chen, 2005). For EEG data, this effectively isolates the artifact signals within each frequency level (which are then subtracted out of the original EEG signal to clean it). A number of high performing options to determine the thresholds separating artifact from clean EEG

Table 3

Artifact-added approach. Correlations between the EEG signal before wavelet thresholding and the EEG signal after wavelet thresholding for the wavelet parameters tested on the artifact-added 30 s segments. The *r* values of the wavelet parameters that are included in the final code are bolded. Asterisks denote default wavelet parameters.



	Artifact-Added								
Wavelet Parameters	**Average r**	**2hz r**	**5hz r**	**8hz r**	**12hz r**	**20hz r**	**30hz r**	**45hz r**	**70hz r**
Wavelet Family									
coif3	0.8457708	0.8313941	0.8897854	0.9139730	0.8552277	0.7911717	0.7652131	0.7378649	0.6396247
coif4	**0.8555245**	**0.8300064**	**0.8938870**	**0.9125044**	**0.8585728**	**0.7944371**	**0.7651386**	**0.7339070**	**0.6363321**
coif5*	0.8611372	0.8279496	0.8928203	0.9123030	0.8644110	0.7933849	0.7670503	0.7357956	0.6424816
db4	0.8534693	0.8336099	0.8936260	0.9108987	0.8639453	0.7975233	0.7664878	0.7377876	0.6417207
db10	0.8429546	0.8313746	0.8898814	0.9044763	0.8523632	0.7942836	0.7652877	0.7360027	0.6425811
sym4	0.8552301	0.8371324	0.8908597	0.9147460	0.8574783	0.7941530	0.7661049	0.7380412	0.6429008
Level of Data Decomposition									
Level 7	0.7180408	0.4543101	0.8927883	0.9119741	0.8641361	0.7931069	0.7670180	0.7358278	0.6423913
Level 8*	0.8611372	0.8279496	0.8928203	0.9123030	0.8644110	0.7933849	0.7670503	0.7357956	0.6424816
Level 9	0.8552324	0.8282940	0.8928847	0.9121907	0.8644032	0.7933826	0.7671959	0.7358080	0.6424973
Level 10	0.8548980	0.8282981	0.8929072	0.9121704	0.8643986	0.7933852	0.7671625	0.7357656	0.6424791
Level 11	0.8551597	0.8283017	0.8929218	0.9121711	0.8643985	0.7933503	0.7671609	0.7357959	0.6424455
Noise Estmate Method									
Level Independent	0.8581343	0.8230792	0.8356412	0.8219082	0.8103576	0.8160537	0.8161101	0.8296241	0.7222856
Level Dependent*	**0.8611372**	**0.8279496**	**0.8928203**	**0.9123030**	**0.8644110**	**0.7933849**	**0.7670503**	**0.7357956**	**0.6424816**
Thresholding Level									
Bayes*	**0.8611372**	**0.8279496**	**0.8928203**	**0.9123030**	**0.8644110**	**0.7933849**	**0.7670503**	**0.7357956**	**0.6424816**
SURE	0.8515281	0.7972381	0.8801077	0.8995941	0.8560370	0.7896673	0.7685342	0.7288356	0.6391554
Universal Threshold	0.8388653	0.8347485	0.8904659	0.9090937	0.8618810	0.7918906	0.7647696	0.7344663	0.6425089
Minimax	0.8543168	0.8366149	0.8934580	0.9135361	0.8655974	0.7945803	0.7684510	0.7365159	0.6421699
Threshold Rule									
Soft*	0.8611372	0.8279496	0.8928203	0.9123030	0.8644110	0.7933849	0.7670503	0.7357956	0.6424816
Median	0.8589594	0.7957200	0.8845254	0.9128770	0.8586024	0.7935736	0.7670473	0.7253411	0.6346028
Hard	0.8334551	0.7074031	0.8602578	0.9016819	0.8456928	0.7888608	0.7629518	0.7153628	0.6291578
Hard (level 9)	0.793786	0.6875453	0.824693	0.8749049	0.812723	0.748518	0.724296	0.692738	0.6021545
Hard (level 10)	**0.818711**	**0.7074161**	**0.860373**	**0.9015566**	**0.845714**	**0.788833**	**0.763001**	**0.715319**	**0.6291468**
Hard (level 11)	0.81863	0.7074121	0.860375	0.9015574	0.845713	0.788808	0.762997	0.7153220	0.6291463

have been established in the literature (Anumala and Kumar Pullakura, 2018; Estrada et al., 2011; Geetha and Geethalakshmi, 2011a; Geetha and Geethalakshmi, 2011b; Guo et al., 2020; Jiang et al., 2007), specifically ‘Empirical Bayes’ (Johnstone and Silverman, 2004), ‘SURE’ (Donoho and Johnstone, 1995), ‘Universal’ (Donoho, 1995), and ‘Minimax’ (Donoho and Johnstone, 1998) thresholding approaches. HAPPE 1.0 (Gabard-Durnam et al., 2018) originally included a Universal Threshold approach, but we aimed to examine these additional high-performing options as well to find the best fit for the current version of the pipeline. Relatedly, we evaluated available threshold rules for the various thresholding methods, specifically ‘Soft’ (Guo et al., 2002; Donoho, 1995), ‘Median’ (Abramovich et al., 1998; Clyde and George, 2000; Johnstone and Silverman, 2005), and ‘Hard’ (Guo et al., 2002). Of note, not all thresholding methods are compatible with all thresholding rules. For example, the median rule is specific to the thresholding method ‘Bayes’ as it involves using the median posterior generated by the Bayesian algorithm to determine the threshold. Soft and hard threshold rules are different in how they treat the coefficients near the threshold (in soft thresholding, these coefficients are shrunk while they are unaffected in hard thresholding). For thresholding options, we found minimal difference between options when tested on the artifact-added data, but we were able to eliminate ‘SURE’ due to over-rejecting data in the clean files (resulting in a correlation pre-/post-thresholding of $r = 0.77$). ‘Bayes’ narrowly outperformed ‘Minimax’ on the artifact-added data (Table 3), but the reverse was true for our clean and artifact-laden data (Table 2). ‘Bayes’ was chosen as the thresholding method as it considers the uncertainty of the potential artifact and has been shown to result in more accurate denoising of signals generally. Moreover, the Bayes algorithm increases performance with increased data samples (as it can adjust its certainty estimates about artifact from more data), so performance on these 30 s files is a conservative reflection of this general performance. For threshold rule, on average, the median and hard thresholds removed more of the clean and artifact-laden file data, especially at the lower frequencies (e.g., at 2 Hz in clean data: soft threshold $r = 0.84$, median threshold $r = 0.79$, hard threshold $r = 0.63$, see Table 2). However, visual inspection of data cleaned using soft vs. hard thresholds revealed that hard thresholds appeared to better remove artifact (see Fig. 6). Therefore, this step was further explored using the simulated signals as described below and ultimately a Bayesian method with hard thresholding rule was implemented.

For the third approach with simulated data that had real artifact added to it, we tested which combination of noise estimation level and threshold rule performed best in terms of removing artifact while retaining the ground-truth underlying simulated signal. We compared combinations of level dependent and level independent estimation methods with soft, median, and hard threshold rules (Fig. 7). We tested on the simulated signal with an adult eye blink component added and either one developmental artifactual IC added or a combination of two developmental artifactual ICs added that reflected a variety of different artifact types and combinations to ensure generalization of the results across artifact conditions (The specific ICs added to the simulated signal are provided above each plot in Fig. 7). We found that level dependent thresholding far outperformed level independent thresholding. This is not surprising given that level dependent thresholding scales within each frequency range (and thus may be more sensitive to different artifact profiles across frequencies) rather than evaluating all frequencies together as is the case for level independent thresholding (see Fig. 7, level-independent results all lie along the x-axis). Moreover, level dependent thresholding with a hard or median threshold rule outperformed level dependent thresholding with a soft threshold rule in terms of visually returning the simulated signal’s spectrum profile. The hard threshold rule narrowly outperformed the median threshold rule (es-

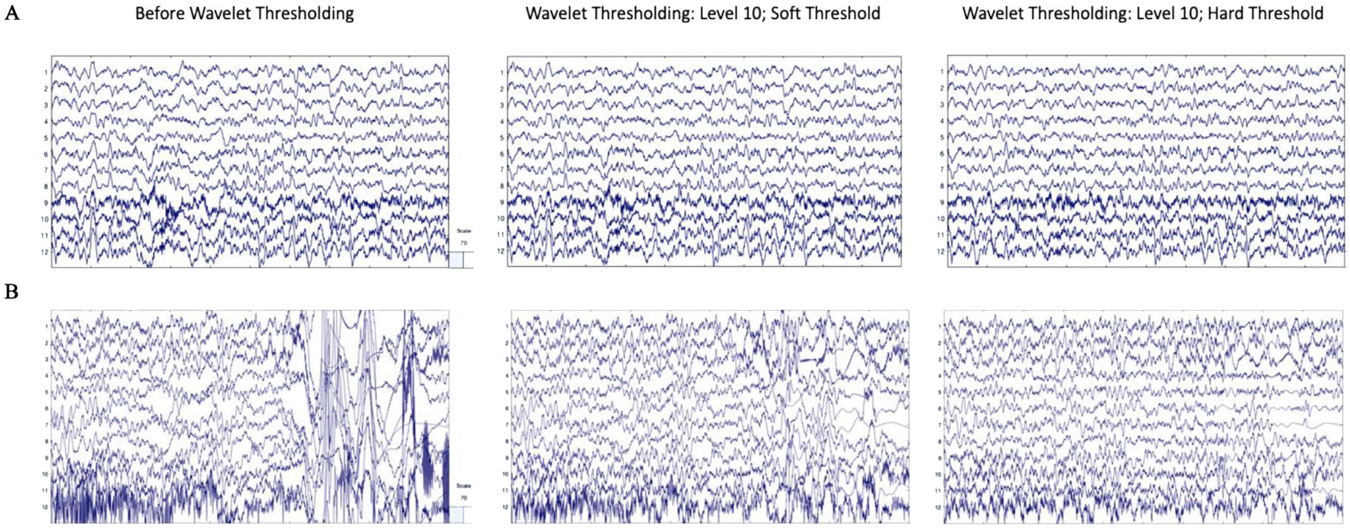


Fig. 6. EEG signal before and after wavelet thresholding with the following parameters: 'Wavelet Family,' 'coif5;' 'Level of Decomposition,' '10;' 'Noise Estimate,' 'Level Dependent;' 'Thresholding Method,' 'Bayes;' 'Threshold Rule,' 'Soft' (middle panel) and 'Wavelet Family,' 'coif5;' 'Level of Decomposition,' '10;' 'Noise Estimate,' 'Level Dependent;' 'Thresholding Method,' 'Bayes;' 'Threshold Rule,' 'Hard' (right panel). Two files from the example dataset are shown with 10 s of data extracted from the clean 30 s segment (A) and artifact-laden 30 s segment (B). The EEG signal before processing is shown in the left panel. The EEG signal after wavelet thresholding with a soft threshold is shown in the middle panel. The EEG signal after wavelet thresholding with a hard threshold is shown in the right panel. All scales are in microvolts.

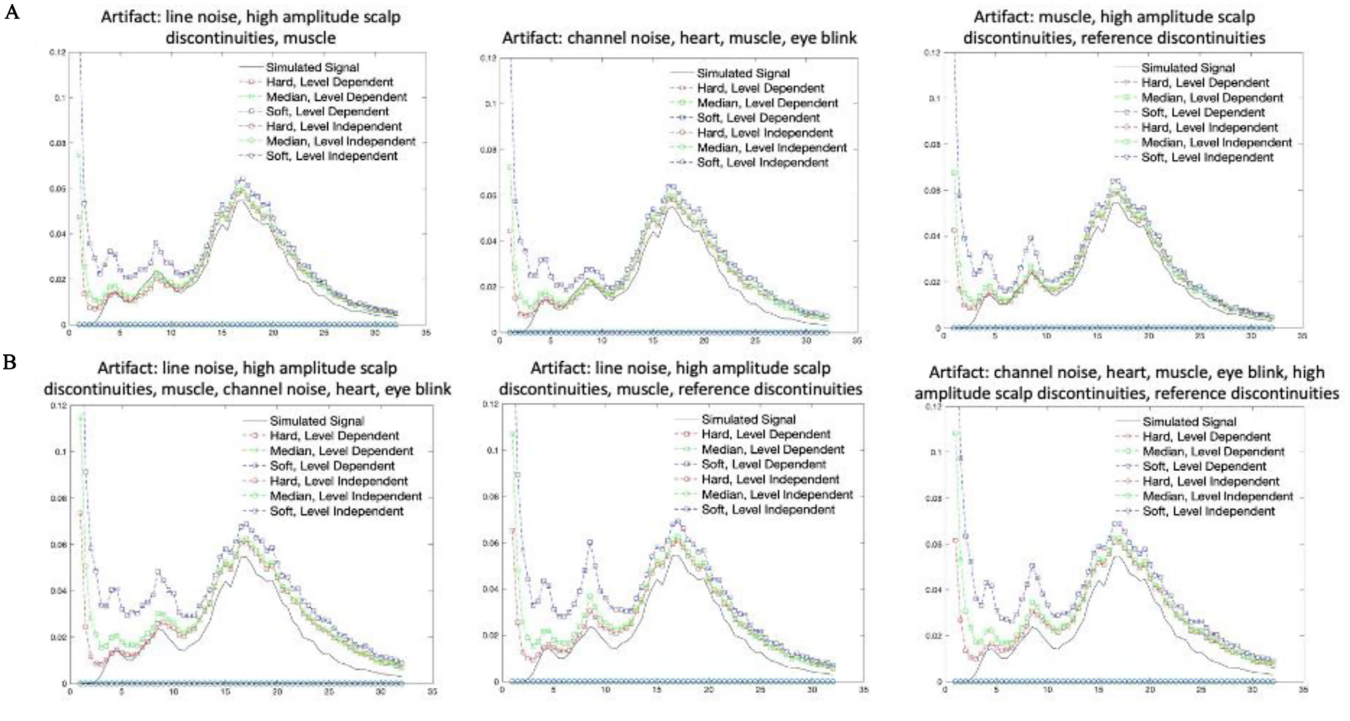


Fig. 7. Plots of the power spectral density for the simulated signal without artifact and the simulated signal with an adult eye blink component added and either one developmental artifactual IC added (A; top panel) or two developmental artifactual ICs added (B; bottom panel) following wavelet thresholding.

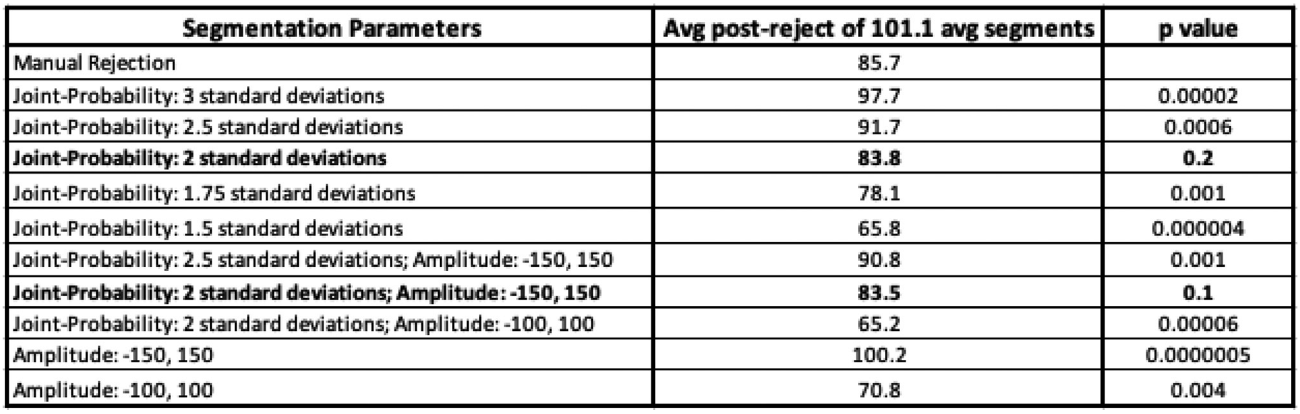
pecially for lower frequencies). Together these results helped solidify the decision to use level dependent thresholding with a hard threshold rule as the combination of parameters that best removed artifact while retaining the ground-truth underlying signal.

Taken together, the final wavelet-thresholding optimized parameters implemented in HAPPILEE are again as follows: 'Wavelet Family,' 'coif4;' 'Level of Decomposition,' '10;' 'Noise Estimate,' 'Level Dependent;' 'Thresholding Method,' 'Bayes;' 'Threshold Rule,' 'Hard'. These parameters ensure optimized wavelet-thresholding based artifact correction occurs in EEG data.

1.7. Segmentation (Optional)

After artifact correction, HAPPILEE includes an optional data segmentation step along with several additional artifact rejection steps to further optimize processing. For data without event markers (e.g., resting-state EEG), regularly marked segments of any duration specified by the user are generated for the duration of the recording (e.g., 1 s segments). For low density data with event markers (e.g., event-related EEG data or ERP designs), data can be segmented around events as specified by user inputs (ERP-processing is supported, including base-

Table 4
Comparison of various segment rejection options tested on nineteen files from the example dataset. *P* values are calculated from t-tests comparing the number of remaining segments for each parameter following automated rejection to the number of segments remaining following manual rejection.



Segmentation Parameters	Avg post-reject of 101.1 avg segments	p value
Manual Rejection	85.7	
Joint-Probability: 3 standard deviations	97.7	0.00002
Joint-Probability: 2.5 standard deviations	91.7	0.0006
Joint-Probability: 2 standard deviations	**83.8**	**0.2**
Joint-Probability: 1.75 standard deviations	78.1	0.001
Joint-Probability: 1.5 standard deviations	65.8	0.000004
Joint-Probability: 2.5 standard deviations; Amplitude: -150, 150	90.8	0.001
Joint-Probability: 2 standard deviations; Amplitude: -150, 150	**83.5**	**0.1**
Joint-Probability: 2 standard deviations; Amplitude: -100, 100	65.2	0.00006
Amplitude: -150, 150	100.2	0.0000005
Amplitude: -100, 100	70.8	0.004

line and timing offset correction; see Monachino et al., 2021 for more information).

Users with data files where segment rejection would lead to an unacceptably low remaining number of segments for analysis may choose an optional post-segmentation step involving the interpolation of data within individual segments for channels determined to be artifact-contaminated during that segment, as implemented by FASTER software (Nolan et al., 2010). Each channel in each segment is evaluated on the four FASTER criteria (variance, median gradient, amplitude range, and deviation from mean amplitude; Nolan et al. 2010), and the Z score (a measure of standard deviation from the mean) for each channel in that segment is generated for each of the four metrics. Any channels with one or more Z scores that are greater than 3 standard deviations from the mean for an individual segment are marked bad for that segment. These criteria may identify segments with residual artifacts. Subsequently, for each segment, the channels flagged as bad in that segment have their data interpolated with spherical splines, as in FASTER (Nolan et al., 2010). This allows users to maintain the maximum number of available segments, while still maximizing artifact rejection within individual segments. However, we caution users from implementing this option in cases where channels are distributed with significant distance between them as the interpolation process would pull data from distal channels that does not reflect the appropriate activity profile for that scalp space. Effects of interpolation on data may depend on experiment, layout (Melnik et al., 2017; Bigdely-Shamlo et al., 2015a), and interpolation method (Courellis et al., 2016; Petrichella et al., 2016; Robeson, 1997).

The majority of users, including those who wish to avoid interpolating data within individual segments, may instead choose to reject segments that are determined to still be artifact-contaminated. HAPPILEE includes three segment rejection options. Criteria for rejection include a choice of joint-probability criteria, amplitude-based criteria, or a combination of joint-probability criteria with amplitude-based criteria. Joint-probability criteria considers how likely a segment's activity is given the activity of other segments for that same channel, as well as other channels' activity for the same segment. The assumption is that artifact segments should be the rare segments with activity several standard deviations apart relative to the rest of the data. Amplitude-based criteria sets a minimum and maximum signal amplitude as the artifact threshold, with segments being removed when their amplitude falls on either side of this threshold. HAPPILEE allows the user to specify their minimum and maximum allowable amplitudes. Users may also specify a combined approach with joint-probability criteria and amplitude-based criteria that removes outlier segments that fail either standard deviations or the signal amplitude criteria.

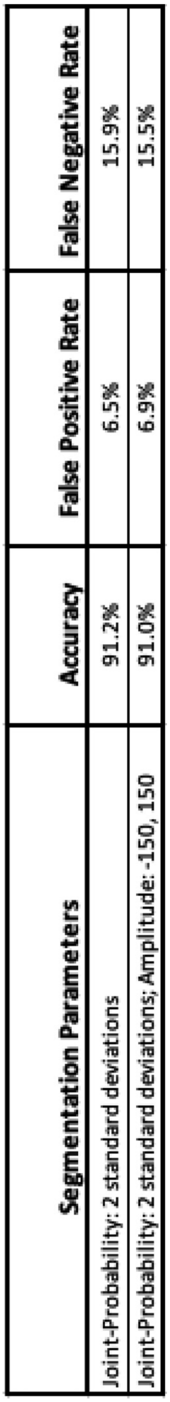
To test the efficacy of the three segment rejection options and determine the optimal criterion values for the rejection of segments in low density EEG data, we compared a series of ten automated options to a set of manually rejected segments for fourteen files in the BEIP dataset. We manipulated the standard deviation values for joint-probability rejection, the amplitude values for amplitude-based rejection, and tested combinations of different joint-probability standard deviations with amplitude criteria. The number of segments rejected for each of these automated rejection approaches was compared to the segments rejected via manual inspection as the gold standard approach using paired t-tests (Table 4). Amplitude-based rejection alone did not sufficiently match the manual rejection rates. However, the number of segments rejected using joint-probability criteria of 2 standard deviations alone or in combination with amplitude criteria (here, −150 and 150 microvolts) were not significantly different from the number of segments rejected manually (both $p > 0.1$).

Segment rejection performance was further evaluated for these two approaches by comparing the identity of segments rejected via the automated approach to the manually rejected segments by summing the number of false negatives and false positives for each file and calculating the overall accuracy rate across files compared to the manual rejection classification (i.e., did HAPPILEE reject the same segments that were rejected manually). The joint probability criterion alone (using 2 standard deviations) achieved the higher accuracy rate of 91.2% across all files but joint-probability with amplitude also did well (91.0% accuracy) (see Table 5). HAPPILEE therefore includes three segment rejection options with the following recommendations. For data with sufficient channels (e.g., here 12 was sufficient but we did not test performance on sparser configurations), segment rejection via joint-probability criteria is recommended. This setting is also recommended for users combining data collected across different systems or ages where the overall signal amplitude may differ across files. Users performing analyses in the time domain (as for ERP paradigms) may opt to include amplitude-based criteria. For users with very low-density configurations where the joint-probability criteria relying on standard deviations may not perform as well as it did here, amplitude-only criteria may be used for segment rejection.

1.8. *Interpolation of bad channels (if bad channel detection was run)*

For all HAPPILEE runs that included bad channel detection, channels marked as bad are now subject to spherical interpolation (with Legendre polynomials up to the 7th order) to repopulate their signal. The identity of all interpolated channels, if any, for a file are recorded in the HAPPILEE processing report for users who wish to monitor the percentage or identity of interpolated channels in their datasets before further anal-

Table 5
Performance of segment rejection parameters tested on nineteen files from the example dataset.



Segmentation Parameters	Accuracy	False Positive Rate	False Negative Rate
Joint-Probability: 2 standard deviations	91.2%	6.5%	15.9%
Joint-Probability: 2 standard deviations; Amplitude: -150, 150	91.0%	6.9%	15.5%

ysis. This interpolation step is available for all files formats except for .mat formats without channel locations as the interpolation step requires electrode location information to interpolate appropriately spatially. However, similar to segment interpolation outlined above, we caution users against including these interpolated channels in analyses in cases where channels are distributed with significant distance between them as the interpolation process would pull data from distal channels that does not reflect the appropriate activity profile for that scalp space. Effects of interpolation on data may depend on experiment, layout (Melnik et al., 2017; Bigdely-Shamlo et al., 2015a), and interpolation method (Courellis et al., 2016; Petrichella et al., 2016; Robeson, 1997). Because this step is not optional when bad channel detection is run, if a user wishes to omit these interpolated channels from analyses, a list of interpolated channels can be found on the data quality assessment report.

1.9. Re-referencing (Optional)

HAPPILEE offers users the choice to re-reference the EEG data if they wish. While there is no ideal reference for EEG, re-referencing with one of several practical if imperfect options can reduce artifact signals that exist consistently across electrodes, including residual line-noise, and recover signal from online reference channels of interest. If re-referencing, the user may specify either re-referencing using an average across all channels (i.e., average re-reference), using a channel subset of one or multiple channels, or re-referencing to a point at infinity using the reference electrode standardization technique (REST) (for additional information on REST, see Yao 2001). It is important to carefully consider which re-referencing method is best suited for a particular dataset, especially when using a low-density layout (e.g. Junghofer et al. 1999). Users may re-process data with different re-referencing options selected to assess the effect of re-reference scheme on their results.

A major concern that stems from low density layouts is inter-electrode distance. Given the limited number of channels, there may be cases where the electrodes used are far apart spatially on the scalp, leading to biases when re-referencing to a channel subset of one or multiple channels. If the reference electrode is spatially close to some electrodes, but far from others, then it will not be representative of the signal as a whole across the scalp and be disproportionately influenced by the immediately surrounding electrodes (Lei and Liao, 2017). If the user chooses to re-reference to a channel subset, they must ensure that the amplitude of that subset is representative of the broader signal across other electrode sites and that the signal at the chosen electrode(s) is not correlated with task-induced activity (Kim, 2018).

To avoid biases associated with re-referencing to electrodes on the scalp, average re-referencing is often used, averaging across all scalp electrodes. However, this can still be challenging with low density datasets when the number of electrodes are limited and the distribution of the electrodes are uneven across the scalp (Dien, 1998). Average re-referencing is recommended when the EEG layout is dense (some recommend over 100 channels) and evenly distributed, allowing the overall activity to average to 0 (Peng and Peng, 2019). Users should consider the distribution of their electrodes before using average re-referencing on low density data.

REST, proposed as a neutral reference to a point at infinity (Yao, 2001), is another option for low density data. In a study considering re-referencing for layouts with 32, 64, and 128 channels, Lei and Liao (2017) found that the relative error was lowest for REST, followed by average re-reference, and then referencing to a single electrode (FCz, Oz), regardless of the number of electrodes and signal-to-noise ratio. An additional study looking at a graph-based analysis of brain connectivity using a 19-channel layout found that REST can minimize contamination and reduce effects of volume conduction (Olejarczyk and Jernajczyk, 2017). With an even broader range of electrodes (21, 34,

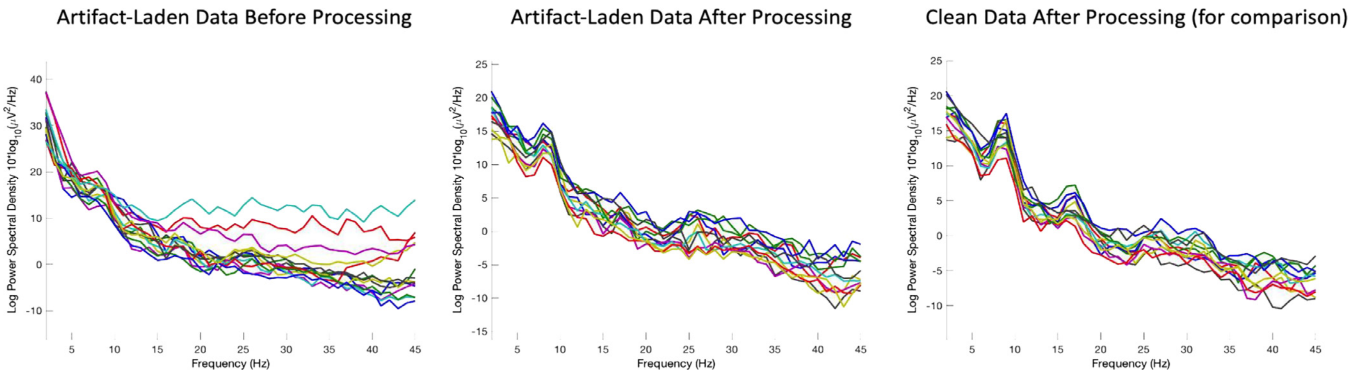


Fig. 8. EEG spectrum before and after complete processing through HAPPILEE with the final optimizations for each step. Two files from the same participant in the example dataset are shown from the artifact-laden 30 s segment and clean 30 s segment. The artifact-laden EEG spectrum before processing is shown in the left panel. The artifact-laden EEG spectrum after processing is shown in the middle panel. The clean EEG spectrum after processing is shown in the right panel for comparison.

74, 128 electrodes), Chella et al. (2016) found that REST reduces biases associated with referencing to a singular electrode and average re-referencing across all electrode density layouts, although REST's performance was further improved with high-density layouts. REST has been implemented in HAPPILEE using the open-source MATLAB toolbox for REST of scalp EEG (Dong et al., 2017).

Note that for re-referencing to a subset of channels and average re-referencing, only channels within the user-specified channel subset selected for HAPPILEE processing can be used for re-referencing. During re-referencing, if there is a prior reference channel (e.g., an online reference channel), that channel's data is recovered and included in the re-referenced dataset. For example, EGI (Electrical Geodesics, Inc., Eugene, OR) data is typically online-referenced to channel CZ. In this example, users could now recover data at channel CZ by re-referencing to any other channel or channels (or average rereference) in this step.

An example file pre- and post-processing with HAPPILEE is shown in Fig. 8 to demonstrate the effectiveness of the pipeline. Clean 30 s segment of data pre- and post-processing is shown as well as an artifact-laden 30 s segment of data pre- and post-processing. The clean and artifact-laden power spectra are much more similar post-processing compared to pre-processing.

1.10. HAPPILEE outputs

HAPPILEE outputs include the processed EEG and the HAPPILEE processing reports. These outputs are generated in several folders that are located within the user-specified folder of files for processing. EEG files are saved out after several intermediate processing steps so that users can explore in-depth and visualize how those steps affected the EEG signal in their own datasets. The intermediate files are separated into folders based on the level of processing performed on the data and include: (1) data after filtering to 100 Hz and line-noise reduction, (2) data post-bad channel rejection (if selected), and (3) post-wavelet-thresholded data. If segmenting is enabled, HAPPILEE outputs one to two additional intermediate files: (5) post-segmented EEG data (always) and (6) interpolated data (if bad data interpolation is enabled). If segment rejection is selected, HAPPILEE saves the data post-segment rejection as well.

HAPPILEE outputs fully processed files that are suitable inputs for further analyses in one of several formats, selected by the user at the start of the HAPPILEE run, to increase compatibility with other software for data visualizations or statistical analyses. Options include mat, .set, and .txt formats. Alongside the fully processed data, HAPPILEE also outputs the HAPPE Data Quality Assessment Report and the HAPPE Pipeline Quality Assessment Report, each described in detail below. Finally, if HAPPILEE is run in the semi-automated setting, the software generates an image for each file containing the fully processed data's power spectrum.

2. HAPPILEE data quality assessment report

HAPPILEE generates a report table of descriptive statistics and data metrics for each EEG file in the batch in a single spreadsheet to aid in quickly and effectively evaluating data quality across participants within or across studies. The report table with all these metrics is provided as a .csv file in the "quality_assessment_outputs" folder generated during HAPPILEE. We describe each of these metrics below to facilitate their use to determine and report data quality (for an example data quality assessment report, see Table 6).

2.1. File length in seconds

HAPPILEE outputs the length, in seconds, of each file prior to processing.

2.2. Number of user selected channels

HAPPILEE outputs the number of channels included for each file as determined by the user.

2.3. Number of good channels selected and percent of good channels selected

The number of channels contributing data ("good channels") and the percentage of channels remaining following rejection of bad channels are provided.

2.4. Bad channel IDs

The identity of channels that are marked bad during the bad channel detection step and subsequently interpolated are provided. Users wishing to limit the amount of interpolated data in further analyses can easily identify channels for exclusion using this metric. Users may also reject files from further analysis on the basis of too high a percentage of bad channels.

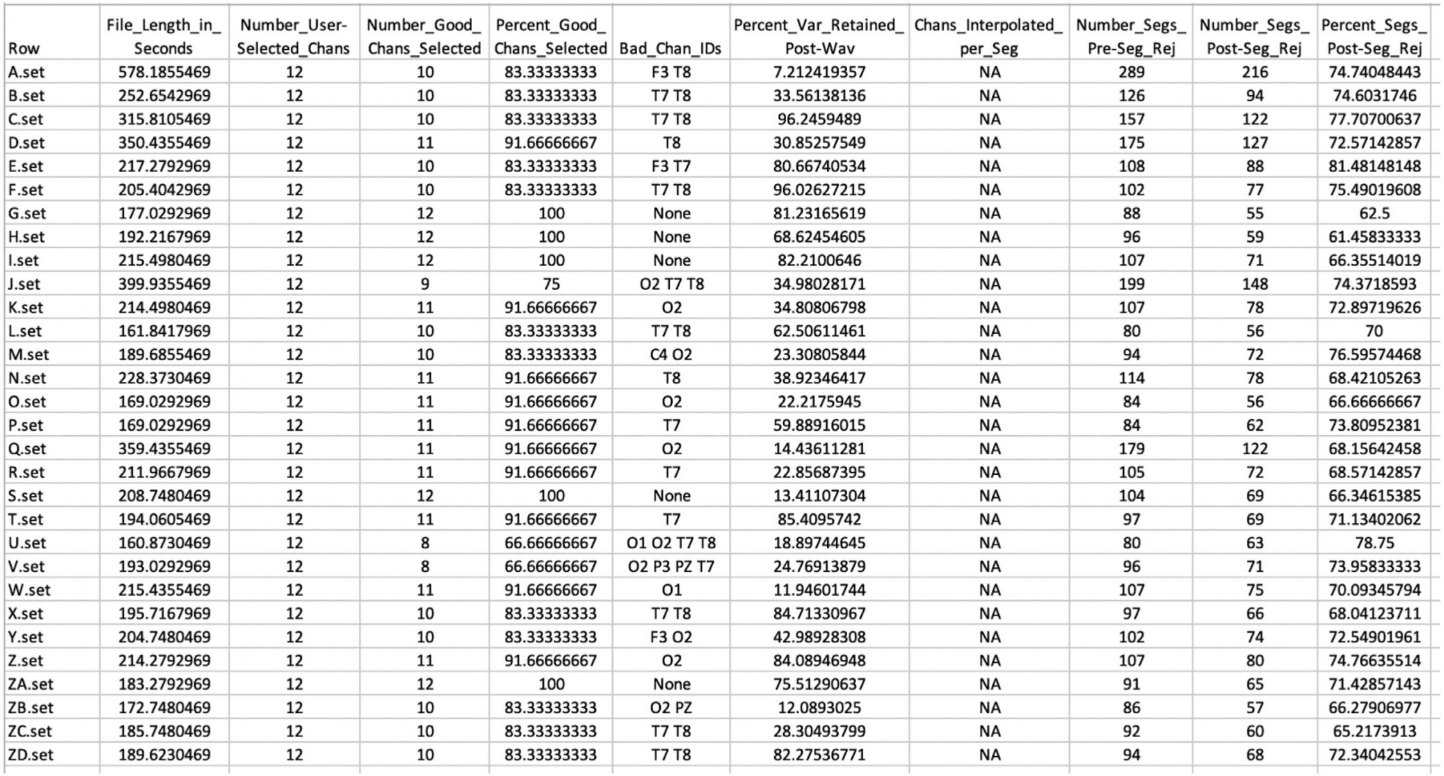
2.5. Percent variance retained post-wavelet

The percent change of the variance of the signal following wavelet thresholding compared to before wavelet thresholding is provided for the user to evaluate how much data is retained following this step of artifact correction.

2.6. Channels interpolated per segment

Users that choose to perform bad data interpolation within segments (as in FASTER, Nolan et al. 2010) will be provided with the list of channels interpolated for each segment.

Table 6
Example HAPPILEE data quality assessment report for the 30 files in the example dataset.



Row	File_Length_in_ Seconds	Number_User- Selected_Chans	Number_Good_ Chans_Selected	Percent_Good_ Chans_Selected	Bad_Chan_IDs	Percent_Var_Retained_ Post-Wav	Chans_Interpolated_ per_Seg	Number_Segs_ Pre-Seg_Rej	Number_Segs_ Post-Seg_Rej	Percent_Segs_ Post-Seg_Rej
A.set	578.1855469	12	10	83.33333333	F3 T8	7.212419357	NA	289	216	74.74048443
B.set	252.6542969	12	10	83.33333333	T7 T8	33.56138136	NA	126	94	74.6031746
C.set	315.8105469	12	10	83.33333333	T7 T8	96.2459489	NA	157	122	77.70700637
D.set	350.4355469	12	11	91.66666667	T8	30.85257549	NA	175	127	72.57142857
E.set	217.2792969	12	10	83.33333333	F3 T7	80.66740534	NA	108	88	81.48148148
F.set	205.4042969	12	10	83.33333333	T7 T8	96.02627215	NA	102	77	75.49019608
G.set	177.0292969	12	12	100	None	81.23165619	NA	88	55	62.5
H.set	192.2167969	12	12	100	None	68.62454605	NA	96	59	61.45833333
I.set	215.4980469	12	12	100	None	82.2100646	NA	107	71	66.35514019
J.set	399.9355469	12	9	75	O2 T7 T8	34.98028171	NA	199	148	74.3718593
K.set	214.4980469	12	11	91.66666667	O2	34.80806798	NA	107	78	72.89719626
L.set	161.8417969	12	10	83.33333333	T7 T8	62.50611461	NA	80	56	70
M.set	189.6855469	12	10	83.33333333	C4 O2	23.30805844	NA	94	72	76.59574468
N.set	228.3730469	12	11	91.66666667	T8	38.92346417	NA	114	78	68.42105263
O.set	169.0292969	12	11	91.66666667	O2	22.2175945	NA	84	56	66.66666667
P.set	169.0292969	12	11	91.66666667	T7	59.88916015	NA	84	62	73.80952381
Q.set	359.4355469	12	11	91.66666667	O2	14.43611281	NA	179	122	68.15642458
R.set	211.9667969	12	11	91.66666667	T7	22.85687395	NA	105	72	68.57142857
S.set	208.7480469	12	12	100	None	13.41107304	NA	104	69	66.34615385
T.set	194.0605469	12	11	91.66666667	T7	85.4095742	NA	97	69	71.13402062
U.set	160.8730469	12	8	66.66666667	O1 O2 T7 T8	18.89744645	NA	80	63	78.75
V.set	193.0292969	12	8	66.66666667	O2 P3 PZ T7	24.76913879	NA	96	71	73.95833333
W.set	215.4355469	12	11	91.66666667	O1	11.94601744	NA	107	75	70.09345794
X.set	195.7167969	12	10	83.33333333	T7 T8	84.71330967	NA	97	66	68.04123711
Y.set	204.7480469	12	10	83.33333333	F3 O2	42.98928308	NA	102	74	72.54901961
Z.set	214.2792969	12	11	91.66666667	O2	84.08946948	NA	107	80	74.76635514
ZA.set	183.2792969	12	12	100	None	75.51290637	NA	91	65	71.42857143
ZB.set	172.7480469	12	10	83.33333333	O2 PZ	12.0893025	NA	86	57	66.27906977
ZC.set	185.7480469	12	10	83.33333333	T7 T8	28.30493799	NA	92	60	65.2173913
ZD.set	189.6230469	12	10	83.33333333	T7 T8	82.27536771	NA	94	68	72.34042553

2.7. Number of segments pre-segment rejection and number of segments post-segment rejection

HAPPILEE reports the number of segments before segment rejection and post segment rejection (using joint probability, amplitude-based rejection, or both). If segment rejection is not enabled, these numbers are identical. If the user enabled segment rejection in HAPPILEE, they may evaluate the number of data segments remaining post-rejection for each file to identify any files that cannot contribute enough clean data to be included in further analyses (user discretion). The user may also easily tabulate the descriptive statistics for remaining segments to report in their manuscript's Methods section (e.g., the mean and standard deviation of the number of usable data segments per file in their study).

2.8. Percent of segments post-segment rejection

The percentage of segments that remain following segment rejection (using joint probability, amplitude-based rejection, or both) are provided for the user.

3. HAPPILEE pipeline quality assessment report

For each run, HAPPILEE additionally generates a report table of descriptive statistics and data metrics for each EEG file in the batch in a single spreadsheet to aid in quickly and effectively evaluating how well the *pipeline* performed across participants within or across studies. Note that these metrics may also be reported in manuscript methods sections as indicators of how data manipulations changed the signal during preprocessing. The report table with all these metrics is provided as a .csv file in the "quality_assessment_outputs" folder generated during HAPPILEE processing (for an example pipeline quality assessment report, see Tables 7 and 8).

3.1. r pre/post linenoise processing

HAPPILEE automatically outputs cross-correlation values at and near the specified line noise frequency (correlation between data at each frequency before and after line noise processing). These cross-correlation values can be used to evaluate the performance of line noise attenuation, as the correlation pre- and post-line noise alogirthm should be lower at the specified frequency or frequencies, but not at the surrounding frequencies beyond 1 to 2 Hz (unless those are also specified by the user). HAPPILEE will automatically adjust which frequencies are reported depending on the user-identified line noise frequency. This metric can also be used to detect changes in how much line noise is present during the recordings (e.g., if generally cross-correlation values are high when study protocol is followed, indicating low line-noise removal from the data, but a staff member forgets and leaves their cell phone on the amplifier for several testing sessions, the degree of line noise removal for those files summarized by this metric could be used as a flag to check in on site compliance with acquisition protocols).

3.2. r pre/post wav-threshold

HAPPILEE automatically outputs the cross-correlation values before and after wavelet thresholding across all frequencies and specifically at 0.5, 1, 2, 5, 8, 12, 20, 30, 45, and 70 Hz. These specific frequencies were selected to cover all canonical frequency bands across the lifespan from delta through high-gamma as well as the low-frequencies retained in ERP analyses. These cross-correlation values can be used to evaluate the performance of waveleting on the data for each file. For example, if cross-correlation values are below 0.2 for all participants in the sample, the wavelet thresholding step has not progressed as intended (users are advised to first check their sampling rate in this case and visualize several raw data files). Note that this measure may also be used to exclude individual files from further analysis based on dramatic signal change during waveleting (indicating high degree of artifact), for example if the 8 Hz or all-data cross-correlations are below some threshold set by the user (e.g., 3 standard deviations from the median or mean, *r* values below 0.2).

Through these quality assessment reports, HAPPILEE aims to provide a rich, quantifiable, yet easily accessible way to effectively evaluate data quality for even very large datasets in the context of automated processing. Visual examination of each file is not required, although it is available. We also hope to encourage more rigorous reporting of data quality metrics in manuscripts by providing these outputs already tabulated and easily transformed into descriptive statistics for inclusion in reports. Users may also wish to include one or several of these metrics as continuous nuisance covariates in statistical analyses to better account for differences in data quality between files or verify whether there are statistically significant differences in data quality post-processing between study groups of interest.

Several metrics may also be useful in evaluating study progress remotely to efficiently track the integrity of the system and data collection protocols. For example, the r pre/post linenoise removal metric may indicate environmental or protocol deviations that cause significant increases in line noise in the data, and the Percent Good Channels Selected and Interpolated Channel ID metrics can be used to track whether the net/cap is being applied and checked for signal quality prior to paradigms or whether a channel (or channels) is in need of repair. For example, if the T6 electrode starts consistently returning bad data for a specific net/cap, it may need to be examined for repair. For further guidance about using the processing report metrics to evaluate data, users may consult the User Guide distributed with HAPPILEE software.

4. HAPPILEE compared to other low-density pre-processing approaches

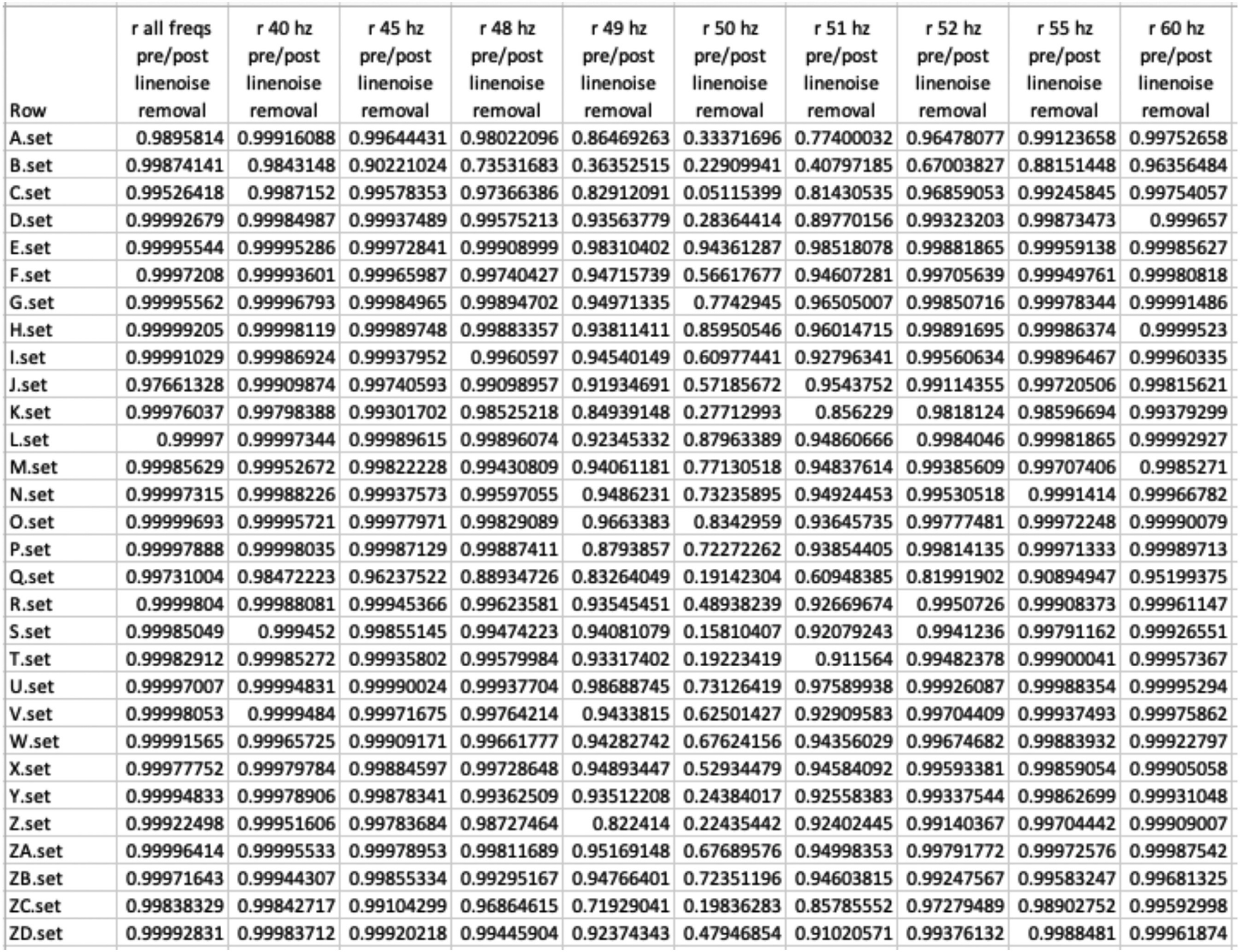
The HAPPILEE pipeline uses wavelet-thresholding based artifact-correction methods to improve pre-processing capabilities for low-density EEG. Below we compare HAPPILEE's approach to two other common pre-processing methods, independent component analysis (ICA) for artifact correction, and voltage-thresholding segment rejection for artifact rejection, to justify the choice of wavelet-thresholding in HAPPILEE's pre-processing sequence.

4.1. HAPPILEE's wavelet-thresholding vs. ICA

Automated artifact correction approaches in high-density EEG pipelines to date have relied on independent component analysis (ICA) and wavelet thresholding methods instead as they can successfully remove artifact while retaining the entire length of the data file. ICA clusters data across electrodes into independent components that can segregate artifacts from neural data, while wavelet-thresholding parses data within frequency ranges using coefficients that can detect artifact fluctuations in either electrode data or independent components. ICA is included as an artifact rejection approach in many pipelines, including MADE (Debnath et al., 2020) and HAPPE 1.0 (Gabard-Durnam et al., 2018) software. Wavelet thresholding was also implemented in HAPPE 1.0 (Gabard-Durnam et al., 2018) as part of an initial wavelet-enhanced ICA (W-ICA) artifact rejection step (see Castellanos and Makarov 2006 for details).

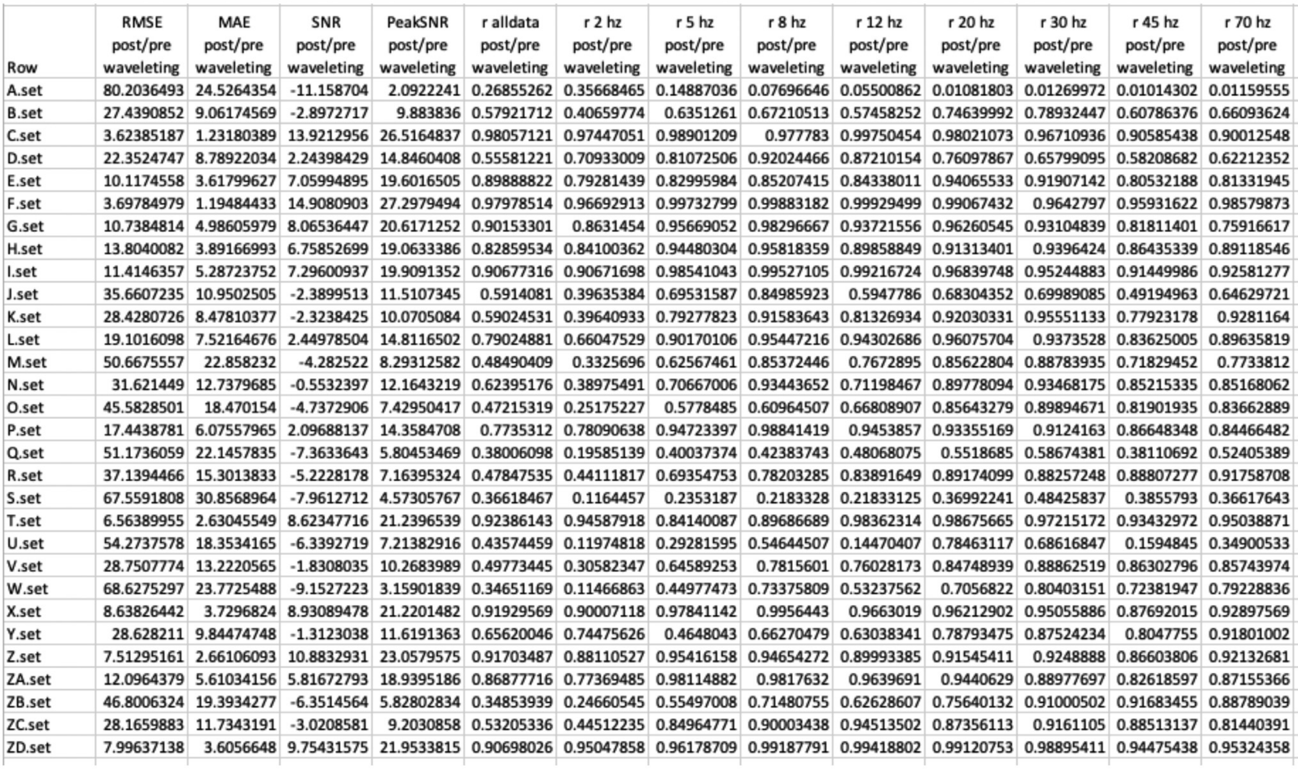
Importantly, prior literature suggests that ICA is not an optimal artifact rejection tool for low-density EEG configurations, as the number of channels determines the number of independent components to be generated, and many low-density configurations have too few channels to adequately segregate artifact from neural components sufficiently. For example, Troller-Renfree and colleagues have indicated through their MiniMADE pipeline that ICA performs poorly on their low-density data to remove artifacts (the MiniMADE pipeline instead uses threshold-based rejection to remove eye blinks) (Troller-Renfree et al., 2021).

Table 7
Example HAPPILEE pipeline quality assessment report with line noise removal metrics for the 30 files in the example dataset.



Row	r all freqs pre/post linenoise removal	r 40 hz pre/post linenoise removal	r 45 hz pre/post linenoise removal	r 48 hz pre/post linenoise removal	r 49 hz pre/post linenoise removal	r 50 hz pre/post linenoise removal	r 51 hz pre/post linenoise removal	r 52 hz pre/post linenoise removal	r 55 hz pre/post linenoise removal	r 60 hz pre/post linenoise removal
A.set	0.9895814	0.99916088	0.99644431	0.98022096	0.86469263	0.33371696	0.77400032	0.96478077	0.99123658	0.99752658
B.set	0.99874141	0.9843148	0.90221024	0.73531683	0.36352515	0.22909941	0.40797185	0.67003827	0.88151448	0.96356484
C.set	0.99526418	0.9987152	0.99578353	0.97366386	0.82912091	0.05115399	0.81430535	0.96859053	0.99245845	0.99754057
D.set	0.99992679	0.99984987	0.99937489	0.99575213	0.93563779	0.28364414	0.89770156	0.99323203	0.99873473	0.999657
E.set	0.99995544	0.99995286	0.99972841	0.99908999	0.98310402	0.94361287	0.98518078	0.99881865	0.99959138	0.99985627
F.set	0.9997208	0.99993601	0.99965987	0.99740427	0.94715739	0.56617677	0.94607281	0.99705639	0.99949761	0.99980818
G.set	0.99995562	0.99996793	0.99984965	0.99894702	0.94971335	0.7742945	0.96505007	0.99850716	0.99978344	0.99991486
H.set	0.99999205	0.99998119	0.99989748	0.99883357	0.93811411	0.85950546	0.96014715	0.99891695	0.99986374	0.9999523
I.set	0.99991029	0.99986924	0.99937952	0.9960597	0.94540149	0.60977441	0.92796341	0.99560634	0.99896467	0.99960335
J.set	0.97661328	0.99909874	0.99740593	0.99098957	0.91934691	0.57185672	0.9543752	0.99114355	0.99720506	0.99815621
K.set	0.99976037	0.99798388	0.99301702	0.98525218	0.84939148	0.27712993	0.856229	0.9818124	0.98596694	0.99379299
L.set	0.99997	0.99997344	0.99989615	0.99896074	0.92345332	0.87963389	0.94860666	0.9984046	0.99981865	0.99992927
M.set	0.99985629	0.99952672	0.99822228	0.99430809	0.94061181	0.77130518	0.94837614	0.99385609	0.99707406	0.9985271
N.set	0.99997315	0.99988226	0.99937573	0.99597055	0.9486231	0.73235895	0.94924453	0.99530518	0.9991414	0.99966782
O.set	0.99999693	0.99995721	0.99977971	0.99829089	0.9663383	0.8342959	0.93645735	0.99777481	0.99972248	0.99990079
P.set	0.99997888	0.99998035	0.99987129	0.99887411	0.8793857	0.72272262	0.93854405	0.99814135	0.99971333	0.99989713
Q.set	0.99731004	0.98472223	0.96237522	0.88934726	0.83264049	0.19142304	0.60948385	0.81991902	0.90894947	0.95199375
R.set	0.9999804	0.99988081	0.99945366	0.99623581	0.93545451	0.48938239	0.92669674	0.9950726	0.99908373	0.99961147
S.set	0.99985049	0.999452	0.99855145	0.99474223	0.94081079	0.15810407	0.92079243	0.9941236	0.99791162	0.99926551
T.set	0.99982912	0.99985272	0.99935802	0.99579984	0.93317402	0.19223419	0.911564	0.99482378	0.99900041	0.99957367
U.set	0.99997007	0.99994831	0.99990024	0.99937704	0.98688745	0.73126419	0.97589938	0.99926087	0.99988354	0.99995294
V.set	0.99998053	0.9999484	0.99971675	0.99764214	0.9433815	0.62501427	0.92909583	0.99704409	0.99937493	0.99975862
W.set	0.99991565	0.99965725	0.99909171	0.99661777	0.94282742	0.67624156	0.94356029	0.99674682	0.99883932	0.99922797
X.set	0.99977752	0.99979784	0.99884597	0.99728648	0.94893447	0.52934479	0.94584092	0.99593381	0.99859054	0.99905058
Y.set	0.99994833	0.99978906	0.99878341	0.99362509	0.93512208	0.24384017	0.92558383	0.99337544	0.99862699	0.99931048
Z.set	0.99922498	0.99951606	0.99783684	0.98727464	0.822414	0.22435442	0.92402445	0.99140367	0.99704442	0.99909007
ZA.set	0.99996414	0.99995533	0.99978953	0.99811689	0.95169148	0.67689576	0.94998353	0.99791772	0.99972576	0.99987542
ZB.set	0.99971643	0.99944307	0.99855334	0.99295167	0.94766401	0.72351196	0.94603815	0.99247567	0.99583247	0.99681325
ZC.set	0.99838329	0.99842717	0.99104299	0.96864615	0.71929041	0.19836283	0.85785552	0.97279489	0.98902752	0.99592998
ZD.set	0.99992831	0.99983712	0.99920218	0.99445904	0.92374343	0.47946854	0.91020571	0.99376132	0.9988481	0.99961874

Table 8
Example HAPPILEE pipeline quality assessment report with waveleting metrics for the 30 files in the example dataset.



Row	RMSE post/pre waveleting	MAE post/pre waveleting	SNR post/pre waveleting	PeakSNR post/pre waveleting	r alldata post/pre waveleting	r 2 hz post/pre waveleting	r 5 hz post/pre waveleting	r 8 hz post/pre waveleting	r 12 hz post/pre waveleting	r 20 hz post/pre waveleting	r 30 hz post/pre waveleting	r 45 hz post/pre waveleting	r 70 hz post/pre waveleting
A.set	80.2036493	24.5264354	-11.158704	2.0922241	0.26855262	0.35668465	0.14887036	0.07696646	0.05500862	0.01081803	0.01269972	0.01014302	0.01159555
B.set	27.4390852	9.06174569	-2.8972717	9.883836	0.57921712	0.40659774	0.6351261	0.67210513	0.57458252	0.74639992	0.78932447	0.60786376	0.66093624
C.set	3.62385187	1.23180389	13.9212956	26.5164837	0.98057121	0.97447051	0.98901209	0.977783	0.99750454	0.98021073	0.96710936	0.90585438	0.90012548
D.set	22.3524747	8.78922034	2.24398429	14.8460408	0.55581221	0.70933009	0.81072506	0.92024466	0.87210154	0.76097867	0.65799095	0.58208682	0.62212352
E.set	10.1174558	3.61799627	7.05994895	19.6016505	0.89888822	0.79281439	0.82995984	0.85207415	0.84338011	0.94065533	0.91907142	0.80532188	0.81331945
F.set	3.69784979	1.19484433	14.9080903	27.2979494	0.97978514	0.96692913	0.99732799	0.99883182	0.99929499	0.99067432	0.9642797	0.95931622	0.98579873
G.set	10.7384814	4.98605979	8.06536447	20.6171252	0.90153301	0.8631454	0.95669052	0.98296667	0.93721556	0.96260545	0.93104839	0.81811401	0.75916617
H.set	13.8040082	3.89166993	6.75852699	19.0633386	0.82859534	0.84100362	0.94480304	0.95818359	0.89858849	0.91313401	0.9396424	0.86435339	0.89118546
I.set	11.4146357	5.28723752	7.29600937	19.9091352	0.90677316	0.90671698	0.98541043	0.99527105	0.99216724	0.96839748	0.95244883	0.91449986	0.92581277
J.set	35.6607235	10.9502505	-2.3899513	11.5107345	0.5914081	0.39635384	0.69531587	0.84985923	0.5947786	0.68304352	0.69989085	0.49194963	0.64629721
K.set	28.4280726	8.47810377	-2.3238425	10.0705084	0.59024531	0.39640933	0.79277823	0.91583643	0.81326934	0.92030331	0.95551133	0.77923178	0.9281164
L.set	19.1016098	7.52164676	2.44978504	14.8116502	0.79024881	0.66047529	0.90170106	0.95447216	0.94302686	0.96075704	0.9373528	0.83625005	0.89635819
M.set	50.6675557	22.858232	-4.282522	8.29312582	0.48490409	0.3325696	0.62567461	0.85372446	0.7672895	0.85622804	0.88783935	0.71829452	0.7733812
N.set	31.621449	12.7379685	-0.5532397	12.1643219	0.62395176	0.38975491	0.70667006	0.93443652	0.71198467	0.89778094	0.93468175	0.85215335	0.85168062
O.set	45.5828501	18.470154	-4.7372906	7.42950417	0.47215319	0.25175227	0.5778485	0.60964507	0.66808907	0.85643279	0.89894671	0.81901935	0.83662889
P.set	17.4438781	6.07557965	2.09688137	14.3584708	0.7735312	0.78090638	0.94723397	0.98841419	0.9453857	0.93355169	0.9124163	0.86648348	0.84466482
Q.set	51.1736059	22.1457835	-7.3633643	5.80453469	0.38006098	0.19585139	0.40037374	0.42383743	0.48068075	0.5518685	0.58674381	0.38110692	0.52405389
R.set	37.1394466	15.3013833	-5.2228178	7.16395324	0.47847535	0.44111817	0.69354753	0.78203285	0.83891649	0.89174099	0.88257248	0.88807277	0.91758708
S.set	67.5591808	30.8568964	-7.9612712	4.57305767	0.36618467	0.1164457	0.2353187	0.2183328	0.21833125	0.36992241	0.48425837	0.3855793	0.36617643
T.set	6.56389955	2.63045549	8.62347716	21.2396539	0.92386143	0.94587918	0.84140087	0.89686689	0.98362314	0.98675665	0.97215172	0.93432972	0.95038871
U.set	54.2737578	18.3534165	-6.3392719	7.21382916	0.43574459	0.11974818	0.29281595	0.54644507	0.14470407	0.78463117	0.68616847	0.1594845	0.34900533
V.set	28.7507774	13.2220565	-1.8308035	10.2683989	0.49773445	0.30582347	0.64589253	0.7815601	0.76028173	0.84748939	0.88862519	0.86302796	0.85743974
W.set	68.6275297	23.7725488	-9.1527223	3.15901839	0.34651169	0.11466863	0.44977473	0.73375809	0.53237562	0.7056822	0.80403151	0.72381947	0.79228836
X.set	8.63826442	3.7296824	8.93089478	21.2201482	0.91929569	0.90007118	0.97841142	0.9956443	0.9663019	0.96212902	0.95055886	0.87692015	0.92897569
Y.set	28.628211	9.84474748	-1.3123038	11.6191363	0.65620046	0.74475626	0.4648043	0.66270479	0.63038341	0.78793475	0.87524234	0.8047755	0.91801002
Z.set	7.51295161	2.66106093	10.8832931	23.0579575	0.91703487	0.88110527	0.95416158	0.94654272	0.89993385	0.91545411	0.9248888	0.86603806	0.92132681
ZA.set	12.0964379	5.61034156	5.81672793	18.9395186	0.86877716	0.77369485	0.98114882	0.9817632	0.9639691	0.9440629	0.88977697	0.82618597	0.87155366
ZB.set	46.8006324	19.3934277	-6.3514564	5.82802834	0.34853939	0.24660545	0.55497008	0.71480755	0.62628607	0.75640132	0.91000502	0.91683455	0.88789039
ZC.set	28.1659883	11.7343191	-3.0208581	9.2030858	0.53205336	0.44512235	0.84964771	0.90003438	0.94513502	0.87356113	0.9161105	0.88513137	0.81440391
ZD.set	7.99637138	3.6056648	9.75431575	21.9533815	0.90698026	0.95047858	0.96178709	0.99187791	0.99418802	0.99120753	0.98895411	0.94475438	0.95324358

ICA has also failed to remove classes of artifact like line noise in low-density data (Xue et al., 2006). In contrast, wavelet-thresholding operates on each EEG channel separately, and thus should have density-independent artifact-correction performance. Indeed, wavelet thresholding has been found to outperform denoising methods that could apply to low-density data, including Empirical Mode Decomposition, Kalman filtering (Salis et al., 2013), and ICA. Specifically, prior studies have found that wavelet thresholding outperforms ICA as an artifact removal approach on data with fourteen channels (Bajaj et al., 2020). Krishnaveni and colleagues also found that waveleting performs better than ICA (using the JADE algorithm) in removing eye blinks in the low frequency range while preserving brain signal in high frequencies (Krishnaveni et al., 2006).

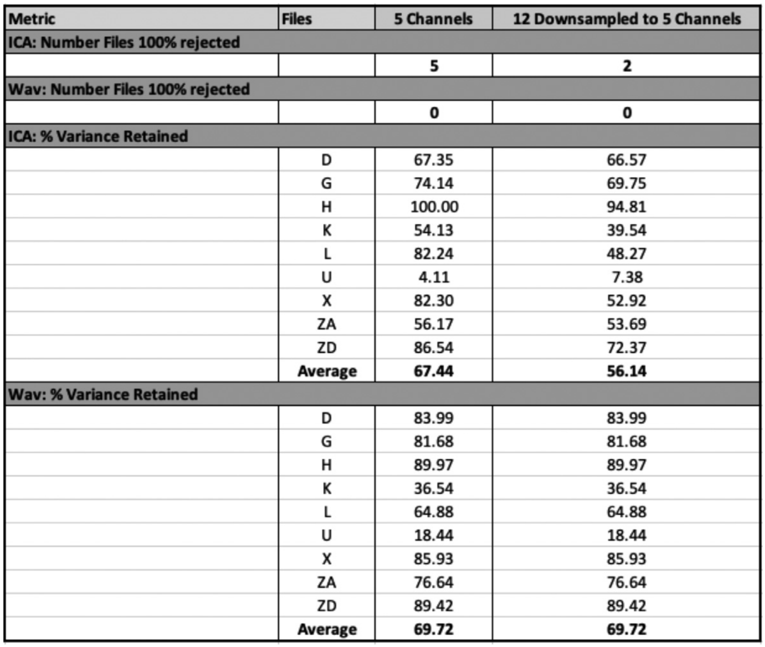
Wavelet thresholding has several additional advantages beyond channel-density independent performance when compared to other artifact correction and rejection approaches. First, wavelet-thresholding can be applied down to the level of single-channel EEG recordings, as it has no minimum channel number to run effectively. Additionally, wavelet thresholding corrects artifact without removing discrete time-points from the EEG recording, an issue that is inherent to manual artifact rejection where whole EEG segments are rejected (and so data from clean channels during that segment are sacrificed). Relatedly, wavelet thresholding can detect and remove isolated and non-stereotyped artifacts easily without removing neural data from other points in time or other channels, whereas ICA must reject an entire timeseries (the component) to remove the artifact, even if it occurs relatively rarely over the recording (or ICA must be paired with initial segment rejection, which sacrifices good data from other electrodes during those artifact-contaminated segments). Wavelet thresholding is also computationally more efficient than either ICA or manual inspection (with results that are perfectly reproducible each time the method is applied to the same dataset). All of these features make wavelet thresholding appealing as a strategy for removing artifact from low-density EEG layouts.

In addition to the prior literature and conceptual reasons provided so far in favor of wavelet-thresholding as an artifact-correction strategy for low-density recordings, we ran a series of empirical tests to provide evidence for our decision to implement wavelet thresholding, rather than ICA, as our artifact correction method. In order to test ICA's performance as a function of channel density, we ran ICA under two conditions: on a subset of 5 channels (F3 F4 FZ C3 C4) and on 12 channels downsampled to the same 5 channels following ICA correction. We predicted that ICA would perform differently as a result of varying the number of channels run on artifact correction for the two datasets. Confirming our prediction, we found that ICA run on five channels rejected all independent components on a total of five files, while ICA run on twelve channels did so for only two files. That is, processing performance differed as a function of channel density during ICA.

Moreover, downsampling before vs. after artifact correction led to differing performances of ICA, as measured by percent variance retained following ICA (t(8) = 2.61, $p = 0.03$); see Table 9 below. Further, these findings are supported by clear visual differences between the post-artifact corrected signal when downsampling occurs before vs. after ICA (Fig. 9). Notably, bad channel detection was not run for this assessment, but these findings raise concerns that removing bad channels and thus changing the channel density for ICA across files within a dataset would vary ICA performance within that dataset. Meanwhile, running the same files under the above conditions through waveleting made no difference as the artifact correction runs within each channel independently (and thus is agnostic to channel density differences within or across files). That is, wavelet thresholding for artifact correction within a dataset is never channel-density dependent, so correction will operate similarly if a file has many, few, or no bad channels flagged.

As one additional test of ICA's performance on low density datasets under conditions where bad channels could more commonly alter channel density, we ran ICA on a subset of 8 channels (F3 F4 C3 C4 P3 P4 O1 O2) and on 12 channels downsampled to the same 8 channels following

Table 9
Percent variance retained following ICA and waveleting for files with 5 channels (F3 F4 FZ C3 C4) and files with 12 channels downsampled to the same 5 channels.



Metric	Files	5 Channels	12 Downsampled to 5 Channels
ICA: Number Files 100% rejected			
		5	**2**
Wav: Number Files 100% rejected			
		0	**0**
ICA: % Variance Retained			
	D	67.35	66.57
	G	74.14	69.75
	H	100.00	94.81
	K	54.13	39.54
	L	82.24	48.27
	U	4.11	7.38
	X	82.30	52.92
	ZA	56.17	53.69
	ZD	86.54	72.37
	Average	**67.44**	**56.14**
Wav: % Variance Retained			
	D	83.99	83.99
	G	81.68	81.68
	H	89.97	89.97
	K	36.54	36.54
	L	64.88	64.88
	U	18.44	18.44
	X	85.93	85.93
	ZA	76.64	76.64
	ZD	89.42	89.42
	Average	**69.72**	**69.72**

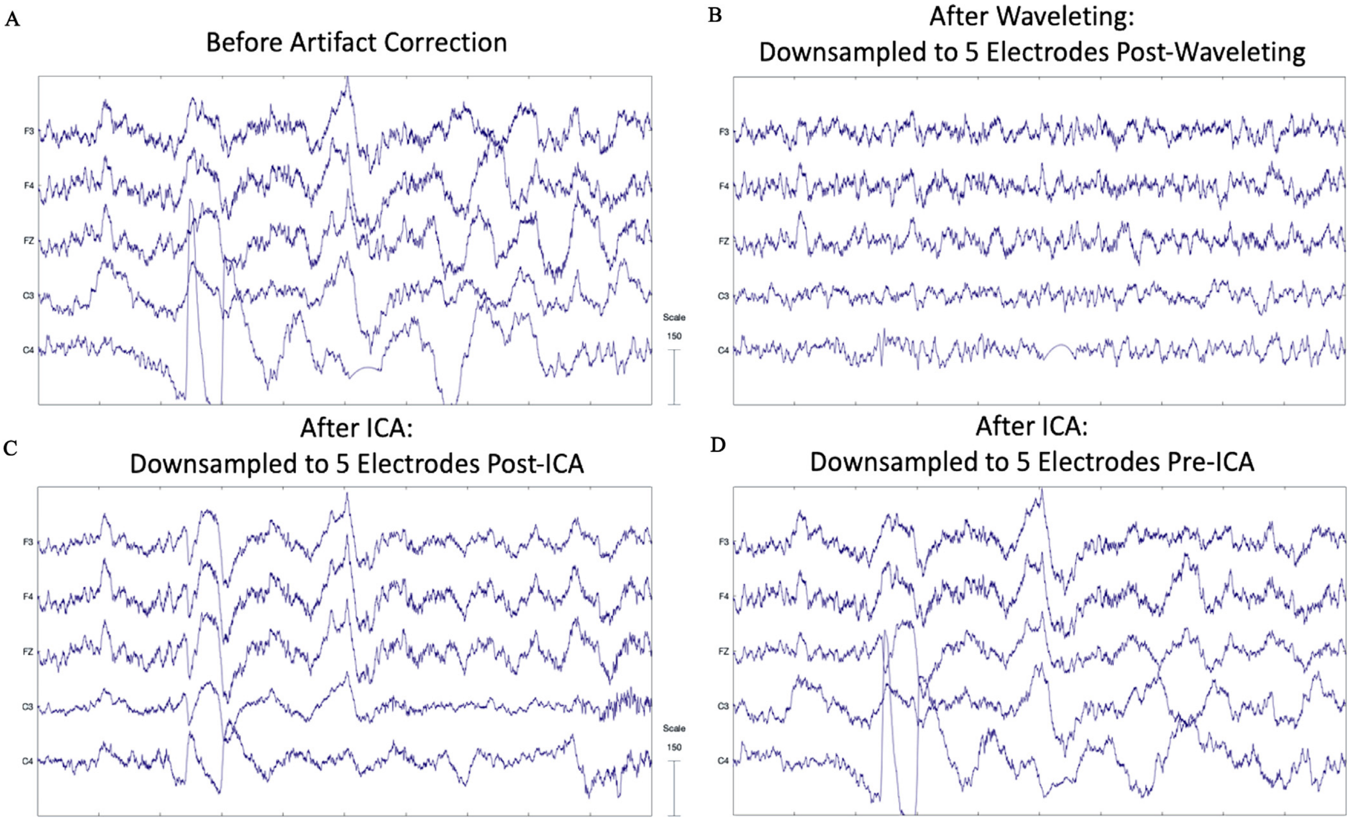
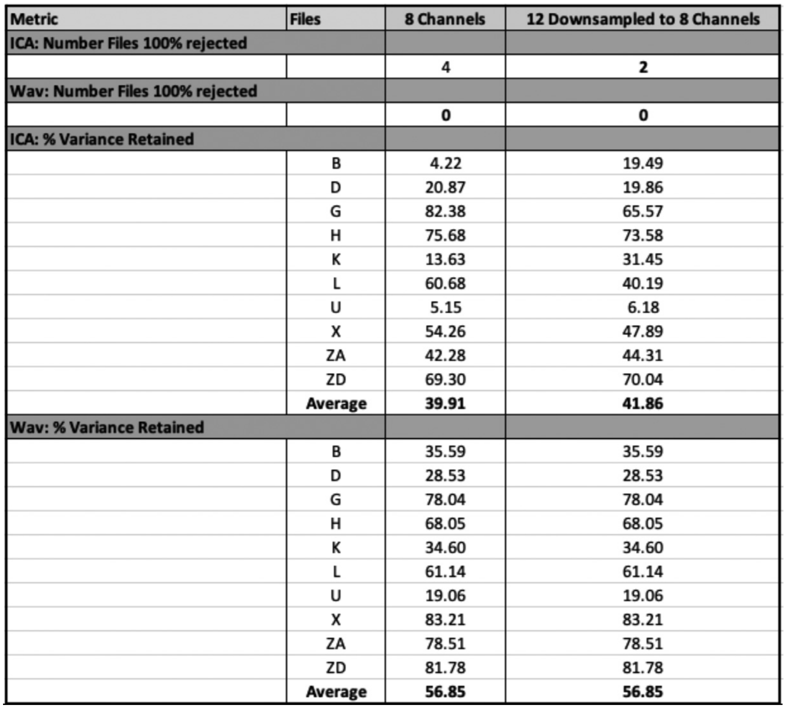


Fig. 9. EEG signal before artifact correction (A), after waveleting (B) and after ICA (C) when the file is downsampled to 5 electrodes following artifact correction, and after ICA when the file is downsampled to 5 electrodes before artifact correction (D). All scales are in microvolts.

Table 10
Percent variance retained following ICA and waveleting for files with 8 channels (F3 F4 C3 C4 P3 P4 O1 O2) and files with 12 channels downsampled to the same 8 channels.



Metric	Files	8 Channels	12 Downsampled to 8 Channels
ICA: Number Files 100% rejected			
		4	2
Wav: Number Files 100% rejected			
		0	0
ICA: % Variance Retained			
	B	4.22	19.49
	D	20.87	19.86
	G	82.38	65.57
	H	75.68	73.58
	K	13.63	31.45
	L	60.68	40.19
	U	5.15	6.18
	X	54.26	47.89
	ZA	42.28	44.31
	ZD	69.30	70.04
	Average	**39.91**	**41.86**
Wav: % Variance Retained			
	B	35.59	35.59
	D	28.53	28.53
	G	78.04	78.04
	H	68.05	68.05
	K	34.60	34.60
	L	61.14	61.14
	U	19.06	19.06
	X	83.21	83.21
	ZA	78.51	78.51
	ZD	81.78	81.78
	Average	**56.85**	**56.85**

ICA correction (several files in the BEIP dataset had 3–4 bad channels resulting in ICA on 8–9 good channels instead of the full 12 channels). Once again, as predicted, we found that downsampling before vs. after artifact correction led to differences in the performance of ICA, as ICA run on 8 channels removed all components on four files, while downsampling to 8 channels rejected all components on only two files. For the subset of files retained by both processing runs, although there were not significant differences between processing conditions measured by percent variance retained following ICA ($p > 0.05$), several individual files experienced dramatic changes in variance retained as a function of channel density (e.g., 13 vs 31%, 60 vs 40%); see Table 10 below. Once again, there are clear visual differences between the post-artifact corrected signal when downsampling occurs before vs. after ICA (Fig. 10; significant signal loss in multiple channels observable in the case of downsampling before running ICA).

This pattern may indicate inconsistent ICA results across these channel densities. Again, as expected, the percent variance retained for waveleting and files rejected were identical for 8 channels compared to 12 channels downsampled to 8 channels. These results together further solidify our decision to use waveleting instead of ICA to ensure consistent performance across files regardless of differences in bad channel number with low density data in HAPPILEE.

Even with the complete set of channels in a low-density context, ICA underperformed in artifact correction relative to wavelet-thresholding visually (Fig. 11). For example, ICA and wavelet thresholding perform similarly on the clean segment where there is very little artifact to correct, but high amplitude artifact clearly remains in the artifact laden segment after it is run through ICA (Fig. 11). Meanwhile, wavelet thresholding effectively removes the high amplitude artifact.

Beyond the inconsistency of ICA's performance as a function of channel density with low density datasets, ICA is also far less computationally efficient than waveleting, reflected both mathematically (e.g., complexity of a learning algorithm vs. wavelet function) and in differences in processing time on the same machine. When run on twelve full length BEIP files averaging roughly three and a half minutes in length and containing 12 channels, ICA was much slower to complete than wavelet-thresholding. ICA averaged 165 s per file while waveleting averaged 1 s per file for artifact correction. Both the prior literature and these additional assessments suggest that waveleting-based approaches will provide the more efficient and consistent performance relative to ICA across the range of low-density layouts considered here.

We do note that how wavelet-thresholding (spatially-independent approach) and spatial filtering approaches (e.g. ICA, PCA) perform relative to each other in correcting artifacts in EEG may depend on several contextual parameters that the field has yet to systematically explore. For example, the inter-channel distance, total spatial coverage on the head across channels, number of channels, and type of artifact signal may all affect how ICA performs relative to wavelet-thresholding. Here, channels were spatially distributed across the scalp and a variety of artifact types were included but not compared directly to each other. Further research is required to systematically quantify how these factors impact artifact correction performance across layout configurations and age.

4.2. HAPPILEE's wavelet-thresholding correction vs. segment rejection

Another option for addressing artifact involves no artifact-correction steps (e.g., wavelet-thresholding, ICA), instead just artifact rejection through segment/trial rejection. Although no manipulations are performed on the retained data segments in the "segment rejection only" approach, there is loss of good data from channels without artifact in the segments that are rejected. Here we examined whether the inclusion of wavelet thresholding reduced the need to reject segments or trials of data downstream in the pipeline. To do so, we compared segment rejection rates in both the simulated and real EEG datasets.

With respect to the simulated data that had adult eye-blink and other developmental artifacts added, an automated amplitude-based voltage threshold (−100 to 100 μV threshold) was used for segment rejection

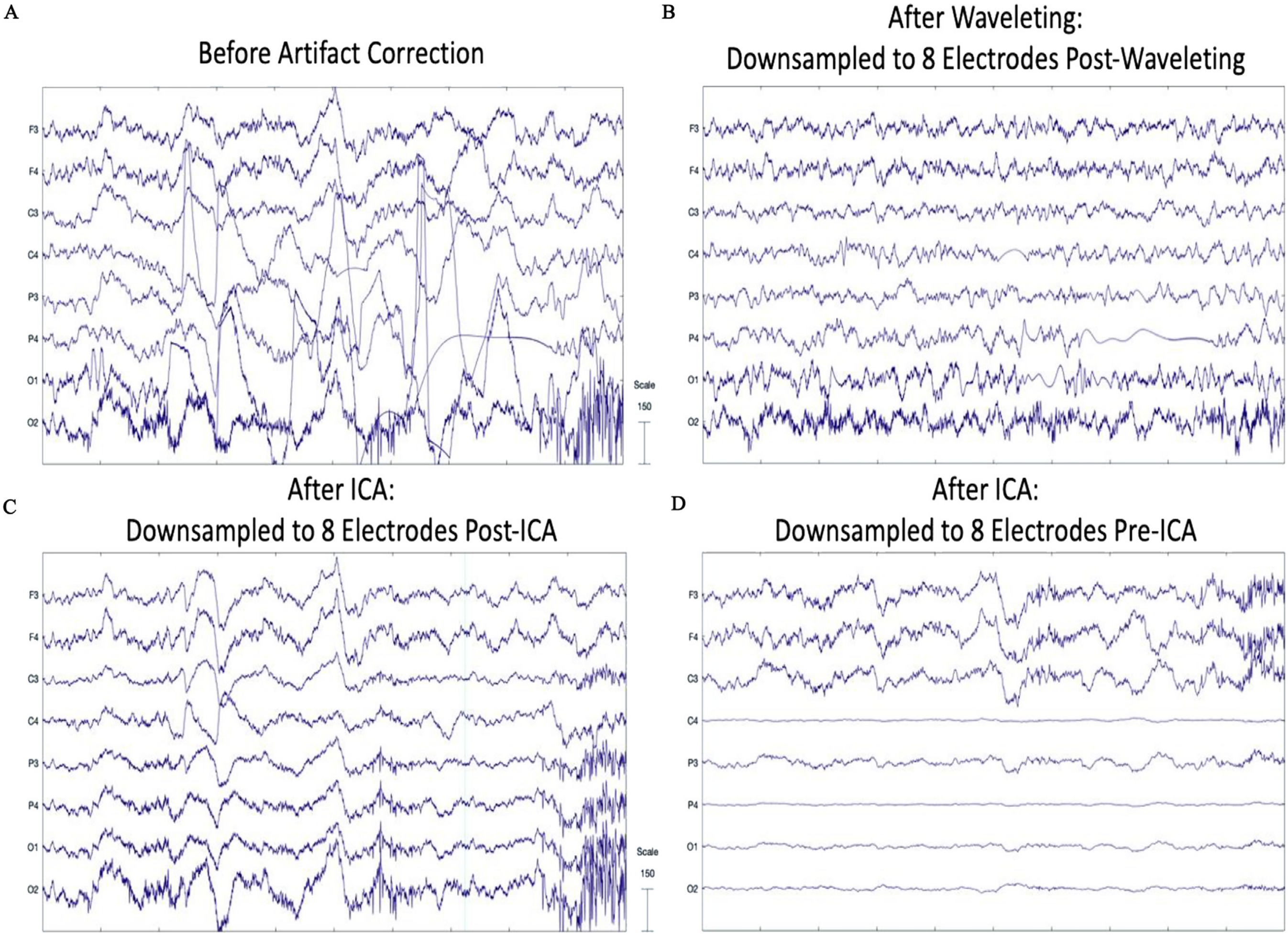


Fig. 10. EEG signal before artifact correction (A), after waveleting (B) and after ICA (C) when the file is downsampled to 8 electrodes following artifact correction, and after ICA when the file is downsampled to 8 electrodes before artifact correction (D). All scales are in microvolts.

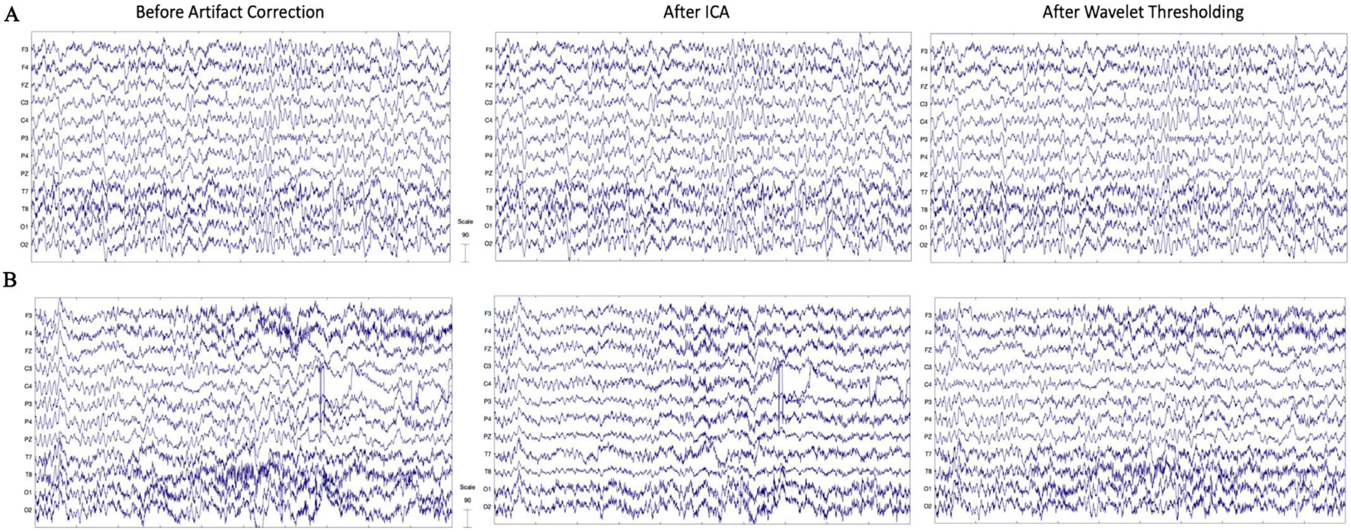
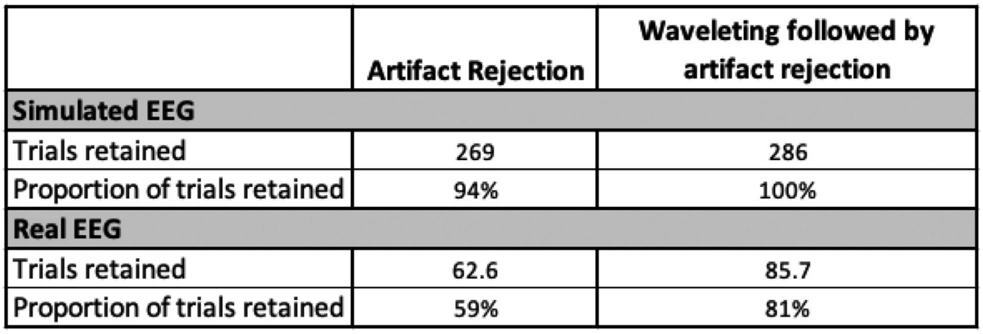


Fig. 11. EEG signal before artifact correction, after ICA, and after wavelet thresholding. Two files from the same participant in the example dataset are shown with 10 s of data extracted from the clean 30 s segment (A) and artifact-laden 30 s segment (B). The EEG signal before artifact correction is shown in the left panel. The EEG signal after ICA is shown in the middle panel. The EEG signal after wavelet thresholding is shown in the right panel. All scales are in microvolts.

Table 11
Performance of two pre-processing sequences on simulated EEG data and real EEG data.



	Artifact Rejection	Waveleting followed by artifact rejection
Simulated EEG		
Trials retained	269	286
Proportion of trials retained	94%	100%
Real EEG		
Trials retained	62.6	85.7
Proportion of trials retained	59%	81%

comparisons. Only the wavelet thresholding approach retained all trials of simulated signal (Table 11) and appeared (based on visual inspection) to have removed all blinks and most of the other artifacts. In contrast, when using artifact rejection only, epochs exceeding the voltage threshold were removed, including some epochs with blinks, but not all. A total of 116 epochs were contaminated with blink artifact, but only 17 epochs were removed with voltage threshold artifact rejection when additional artifact was included in the simulated data (in a separate test, not reported in detail here, we ran the data with just blink artifact included and 107/116 epochs with blinks were removed with voltage threshold artifact rejection). Visual inspection of the data suggested that some of the blinks in noisier epochs were retained when subjected to only artifact-rejection thresholding because the noise stemming from muscle activity included negative deflections that lowered the amplitude of the blinks, which caused the blink artifacts to fall within our acceptable voltage threshold range. Of note, some muscle activity still remained in the simulated data in all pre-processing conditions in this test. These simulated data results strongly support the use of HAPPILEE's wavelet-thresholding as an artifact-correction approach prior to segment rejection to improve both artifact removal and segment retention.

To further evaluate the performance of segment retention only compared to wavelet-thresholding, the real EEG data in the BEIP study was leveraged with manual segment rejection. Manual segment rejection here addressed the effect observed above in the simulated data where some artifacts in the segment rejection only approach passed through automated voltage thresholds when combined with other artifacts (as often occurs in real EEG data). Specifically, manual segment rejection rates were compared in 14 files of the BEIP dataset that were processed twice, once with and once without wavelet-thresholding on the data (mean number of segments before rejection = 105.6). The mean number of clean segments retained after manual rejection on post-waveleted data (85.7 segments) was significantly higher than the mean number of clean segments retained without wavelet thresholding (62.6 segments; $t(18) = -9.07$, $p = 0.00000004$, Table 11). That is, artifact-correction via wavelet-thresholding improved segment retention by 37% relative to no artifact-correction before segment rejection in real EEG data. This pattern of results across comparisons with ICA and artifact rejection via voltage-thresholding provides consistent evidence in support of using wavelet-thresholding for artifact-correction in pre-processing EEG data in low-density contexts.

5. Conclusion

The field of cognitive neuroscience has been rapidly moving toward the use of automated EEG pre-processing pipelines that make use of contemporary artifact correction and rejection approaches like wavelet-thresholding and ICA as effective, efficient, standardized alternatives to subjective and labor-intensive manual pre-processing. Here we provide a solution suitable for lower-density layouts from approximately 32 channels down to single channel EEG with the current automated pipeline, HAPPILEE. HAPPILEE supports processing resting-state and task-related EEG, as well as ERP data (see Monachino et al. 2021 for details on optimization for ERP analyses). HAPPILEE is suitable for configurations with any number of channels, though it may perform best on data with 1 to around 32 channels (those with higher-density configurations may consider pipelines optimized for high-density data, including the companion HAPINNES or HAPPE+ER (Monachino et al., 2021) pipelines within HAPPE 2.0 software.

There are several limitations to HAPPLIEE that should also be considered. First, HAPPILEE was optimized and validated using developmental EEG and simulated signals, so it remains to be validated in adult EEG data or other populations. Though the authors do not anticipate specific reasons HAPPILEE would not perform well in other populations, and have run HAPPILEE in adult EEG data themselves, researchers with EEG data from populations not validated in this manuscript are encouraged to carefully verify performance themselves before using HAPPILEE. Second, bad channel detection was tested on a dataset with twelve channels. As a result, those working with layouts with substantially fewer electrodes may consider verifying for themselves that the default settings work sufficiently well for their datasets. Alternatively, the bad channel detection step is optional, so if it is unsuitable or is not desired for a dataset, the user may opt-out of this step of the pipeline. Furthermore, the appropriate amplitude threshold for performing segment rejection by amplitude will vary across datasets collected on different ages or systems and should be verified through visual inspection of several files (via running HAPPILEE in the semi-automated setting with visualizations). Lastly, HAPPILEE was optimized using a single, gel-based (low-impedance) system (James Long) given the data available to the researchers, but others should independently verify performance on the other systems and recording contexts to confirm compatibility with HAPPILEE. Although the authors do not currently foresee any difference in performance between datasets collected in laboratory environments (i.e., the current optimization dataset) and datasets collected in the home or clinics, further testing will be necessary in order to fully verify its efficacy on data collected in non-laboratory settings.

The HAPPILEE pipeline is freely available as part of the HAPPE software (first released with HAPPE version 2.0), covered under the terms of the GNU General Public License (version 3) (Free Software Foundation, 2007). HAPPILEE's sequence of processing steps are automatically triggered within HAPPE 2.0+ software when the user indicates they have data with fewer than 32 channels. HAPPILEE may be accessed at: https://github.com/PINE-Lab/HAPPE. The HAPPE 2.0+ software download includes a user guide to aid in the set-up and implementation of the pipeline. The subset of BEIP EEG data used to optimize the HAPPILEE pipeline, including the files used in the clean vs. artifact and artifact addition approaches and simulated signals are publicly available at: https://zenodo.org/record/5088346 (Lopez et al., 2021).

Funding

This project was supported via a grant from the Bill and Melinda Gates Foundation to LGD.

Code and data availability

All code used in the HAPPILEE pipeline is freely available under the terms of the GNU General Public License at: https://github.com/PINE-Lab/HAPPE.

All EEG and simulated EEG data used to optimize the HAPPILEE pipeline in this manuscript are freely available as a dataset through Zenodo as Lopez et al., 2021: https://zenodo.org/record/5,088,346.

Credit authorship contribution statement

K.L. Lopez: Data curation, Formal analysis, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **A.D. Monachino:** Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **S. Morales:** Data curation, Formal analysis, Validation, Writing – original draft, Writing – review & editing. **S.C. Leach:** Data curation, Formal analysis, Validation, Writing – original draft, Writing – review & editing. **M.E. Bowers:** Data curation, Formal analysis, Validation, Writing – original draft, Writing – review & editing. **L.J. Gabard-Durnam:** Conceptualization, Funding acquisition, Methodology, Project administration, Software, Supervision, Validation, Writing – original draft, Writing – review & editing.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.neuroimage.2022.119390.

References

Abramovich, F., Sapatinas, T., Silverman, B.W., 1998. Wavelet thresholding via a Bayesian approach. J. R. Stat. Soc. Ser. B Stat. Methodol. 60, 725–749. doi:10.1111/1467-9868.00151.

Aeby, A., Santalucia, R., Van Hecke, A., Nebbioso, A., Vermeiren, J., Deconinck, N., De Tiège, X., Van Bogaert, P., 2021. A qualitative awake EEG score for the diagnosis of continuous spike and waves during sleep (CSWS) syndrome in self-limited focal epilepsy (SFE): a case-control study. Seizure 84, 34–39. doi:10.1016/j.seizure.2020.11.008.

Al-Qazzaz, N.K., Ali, M., Bin, S.H., Ahmad, S.A., Islam, M.S., Escudero, J., 2015. Selection of mother wavelet functions for multi-channel EEG signal analysis during a working memory task. Sensors (Switzerland) 15, 29015–29035. doi:10.3390/s151129015.

Altman, D.G., Bland, J.M., 1995. Statistics notes: the normal distribution. BMJ 310, 298. doi:10.1136/bmj.310.6975.298.

Alyasseri, Z.A.A., Khader, A.T., Al-Betar, M.A., 2017. Electroencephalogram signals denoising using various mother wavelet functions: a comparative analysis. In: Proceedings of the ACM International Conference Proceeding Series. Association for Computing Machinery, pp. 100–105. doi:10.1145/3132300.3132313.

Andersen, L.M., 2018. Group analysis in MNE-python of evoked responses from a tactile stimulation paradigm: a pipeline for reproducibility at every step of processing, going from individual sensor space representations to an across-group source space representation. Front. Neurosci. 12, 6. doi:10.3389/fnins.2018.00006.

Anumala, V., Kumar Pullakura, R., 2018. Removal of ocular artifacts in single channel EEG by EMD, EEMD and CEEMD methods inspired by wavelet thresholding. Int. J. Image Graph. Signal Process. 10, 45–55. doi:10.5815/ijigsp.2018.05.05.

Bajaj, N., Requena Carrión, J., Bellotti, F., Berta, R., De Gloria, A., 2020. Automatic and tunable algorithm for EEG artifact removal using wavelet decomposition with applications in predictive modeling during auditory tasks. Biomed. Signal Process. Control 55, 101624. doi:10.1016/j.bspc.2019.101624.

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K.M., Robbins, K.A., 2015a. The PREP pipeline: standardized preprocessing for large-scale EEG analysis. Front. Neuroinform. 16. doi:10.3389/FNINF.2015.00016.

Bigdely-Shamlo, N., Mullen, T., Kothe, C., Su, K.M., Robbins, K.A., 2015b. The PREP pipeline: standardized preprocessing for large-scale EEG analysis. Front. Neuroinform. 9, 1–19. doi:10.3389/fninf.2015.00016.

Bigdely-Shamlo, N., Touryan, J., Ojeda, A., Kothe, C., Mullen, T., Robbins, K., 2020. Automated EEG mega-analysis I: spectral and amplitude characteristics across studies. Neuroimage 207, 116361. doi:10.1016/j.neuroimage.2019.116361.

Bridwell, D.A., Rachakonda, S., Silva, R.F., Pearlson, G.D., Calhoun, V.D., 2018. Spatiospectral Decomposition of multi-subject EEG: evaluating blind source separation algorithms on real and realistic simulated data. Brain Topogr. 31, 47–61. doi:10.1007/s10548-016-0479-1.

Brito, N.H., Elliott, A.J., Isler, J.R., Rodriguez, C., Friedrich, C., Shuffrey, L.C., Fifer, W.P., 2019. Neonatal EEG linked to individual differences in socioemotional outcomes and autism risk in toddlers. Dev. Psychobiol. 61, 1110–1119. doi:10.1002/dev.21870.

Brito, N.H., Fifer, W.P., Myers, M.M., Elliott, A.J., Noble, K.G., 2016. Associations among family socioeconomic status, EEG power at birth, and cognitive skills during infancy. Dev. Cogn. Neurosci. doi:10.1016/j.dcn.2016.03.004.

Cassani, R., Falk, T.H., Fraga, F.J., Cecchi, M., Moore, D.K., Anghinah, R., 2017. Towards automated electroencephalography-based Alzheimer's disease diagnosis using portable low-density devices. Biomed. Signal Process. Control 33, 261–271. doi:10.1016/j.bspc.2016.12.009.

Castellanos, N.P., Makarov, V.A., 2006. Recovering EEG brain signals: artifact suppression with wavelet enhanced independent component analysis. J. Neurosci. Methods 158, 300–312. doi:10.1016/j.jneumeth.2006.05.033.

Chaumon, M., Bishop, D.V.M., Busch, N.A., 2015. A practical guide to the selection of independent components of the electroencephalogram for artifact correction. J. Neurosci. Methods 250, 47–63. doi:10.1016/j.jneumeth.2015.02.025.

Chella, F., Pizzella, V., Zappasodi, F., Marzetti, L., 2016. Impact of the reference choice on scalp EEG connectivity estimation. J. Neural Eng. 13, 036016. doi:10.1088/1741-2560/13/3/036016.

Clyde, M., George, E.I., 2000. Flexible empirical bayes estimation for wavelets. J. R. Stat. Soc. Ser. B 62, 681–698. doi:10.1111/1467-9868.00257.

Courellis, H.S., Iversen, J.R., Poizner, H., Cauwenberghs, G., 2016. EEG channel interpolation using ellipsoid geodesic length. In: Proceedings of the IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 540–543. doi:10.1109/BIOCAS.2016.7833851 2016.

da Cruz, J.R., Chicherov, V., Herzog, M.H., Figueiredo, P., 2018. An automatic preprocessing pipeline for EEG analysis (APP) based on robust statistics. Clin. Neurophysiol. 129, 1427–1437. doi:10.1016/j.clinph.2018.04.600.

Debnath, R., Buzzell, G.A., Morales, S., Bowers, M.E., Leach, S.C., Fox, N.A., 2020. The Maryland analysis of developmental EEG (MADE) pipeline. Psychophysiology 57, e13580. doi:10.1111/PSYP.13580.

Desjardins, J.A., van Noordt, S., Huberty, S., Segalowitz, S.J., Elsabbagh, M., 2021. EEG Integrated Platform Lossless (EEG-IP-L) pre-processing pipeline for objective signal quality assessment incorporating data annotation and blind source separation. J. Neurosci. Methods 347, 108961. doi:10.1016/j.jneumeth.2020.108961.

Dien, J., 1998. Issues in the application of the average reference: review, critiques, and recommendations. Behav. Res. Methods Instrum. Comput. 301 (30), 34–43. doi:10.3758/BF03209414, 1998.

Dong, L., Li, F., Liu, Q., Wen, X., Lai, Y., Xu, P., Yao, D., 2017. MATLAB toolboxes for reference electrode standardization technique (REST) of scalp EEG. Front. Neurosci. 11, 601. doi:10.3389/FNINS.2017.00601/BIBTEX.

Donoho, D.L., 1995. De-Noising by Soft-Thresholding. IEEE Trans. Inf. Theory 41, 613–627. doi:10.1109/18.382009.

Donoho, D.L., Johnstone, I.M., 1998. Minimax estimation via wavelet shrinkage. Ann. Stat. 26, 879–921. doi:10.1214/aos/1024691081.

Donoho, D.L., Johnstone, I.M., 1995. Adapting to unknown smoothness via wavelet shrinkage. J. Am. Stat. Assoc. 90, 1200–1224. doi:10.1080/01621459.1995.10476626.

Estrada, E., Nazeran, H., Sierra, G., Ebrahimi, F., Setarehdan, S.K., 2011. Wavelet-based EEG denoising for automatic sleep stage classification. In: Proceedings of the CONIELECOMP 2011, 21st International Conference on Electronics Communications and Computers, pp. 295–298. doi:10.1109/CONIELECOMP.2011.5749325.

Gabard-Durnam, L.J., Mendez Leal, A.S., Wilkinson, C.L., Levin, A.R., 2018. The harvard automated processing pipeline for electroencephalography (happe): standardized processing software for developmental and high-artifact data. Front. Neurosci. 97. doi:10.3389/FNINS.2018.00097.

Geetha, G., Geethalakshmi, S.N., 2011a. DE-noising of EEG signals using Bayes shrink based on Coiflet transform. IET Semin. Dig. 41–46. doi:10.1049/ic.2011.0048, IET.

Geetha, G., Geethalakshmi, S.N., 2011b. EEG De-noising using SURE Thresholding based on Wavelet Transforms. Int. J. Comput. Appl..

Gu, Y., Cleeren, E., Dan, J., Claes, K., Van Paesschen, W., Van Huffel, S., Hunyadi, B., 2018. Comparison between scalp EEG and behind-the-ear EEG for development of a wearable seizure detection system for patients with focal epilepsy. Sensors (Switzerland) 18, 29. doi:10.3390/s18010029.

Guo, J., Wang, H., Zhang, C., Bai, X., 2020. EEG signals denoising using bayesian estimation. In: Proceedings of the IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA. Institute of Electrical and Electronics Engineers Inc., pp. 317–320. doi:10.1109/AEECA49918.2020.9213615 2020.

Guo, D.F., Zhu, W.H., Gao, Z.M., Zhang, J.Q., 2002. A Study of Wavelet Thresholding Denoising Institute of Electrical and Electronics Engineers (IEEE), pp. 329–332.

Hajra, S.G., Gopinath, S., Liu, C.C., Pawlowski, G., Fickling, S.D., Song, X., D'Arcy, R.C.N., 2020. Enabling event-related potential assessments using low-density electrode arrays: a new technique for denoising individual channel EEG data. In: Proceedings of the IEMTRONICS 2020 - International IOT, Electronics and Mechatronics Conference. Proceedings. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/IEMTRONICS51293.2020.9216365.

Harender, B., Sharma, R.K., 2018. DWT based epileptic seizure detection from EEG signal using k-NN classifier. In: Proceedings of the International Conference on Trends in Electronics and Informatics, ICEI. Institute of Electrical and Electronics Engineers Inc., pp. 762–765. doi:10.1109/ICOEI.2017.8300806 2017.

Hatz, F., Hardmeier, M., Bousleiman, H., Rüegg, S., Schindler, C., Fuhr, P., 2015. Reliability of fully automated versus visually controlled pre- and post-processing of resting-state EEG. Clin. Neurophysiol. 126, 268–274. doi:10.1016/j.clinph.2014.05.014.

Jafarpour, S., Polatkan, G., Brevdo, E., Hughes, S., Brasoveanu, A., Daubechies, I., 2009. Stylistic analysis of paintings using wavelets and machine learning. In: Proceedings of the 17th European Signal Processing Conference.

Jansen, M., 2001. Noise reduction by wavelet thresholding. Lecture Notes in Statistics. Springer New York doi:10.1007/978-1-4613-0145-5.

Jiang, X., Mahadevan, S., Adeli, H., 2007. Bayesian wavelet packet denoising for structural system identification. Struct. Control Heal. Monit. 14, 333–356. doi:10.1002/stc.161.

Johnstone, I.M., Silverman, B.W., 2005. Empirical bayes selection of wavelet thresholds. Ann. Stat. 33, 1700–1752. doi:10.1214/009053605000000345.

Johnstone, I.M., Silverman, B.W., 2004. Needles and straw in haystacks: empirical BAYES estimates of possibly sparse sequences. Ann. Stat. 32, 1594–1649. doi:10.1214/009053604000000030.

Junghöfer, M., Elbert, T., Tucker, D.M., Braun, C., 1999. The polar average reference effect: a bias in estimating the head surface integral in EEG recording. Clin. Neurophys. 110, 1149–1155.

Kariuki, S.M., White, S., Chengo, E., Wagner, R.G., Ae-Ngibise, K.A., Kakooza-Mwesige, A., Ngugi, A.K., Sander, J.W., Neville, B.G., Newton, C.R., Wagner, R., Twine, R., Connor, M., Olivé, F.X.G., Collinson, M., Kahn, K., Tollman, S., Masanja, H., Mathew, A., Kakooza, A., Pariyo, G., Peterson, S., Ndyomughenyi, D., Odhiambo, R., Chabi, M., Bauni, E., Kamuyu, G., Odera, V.M., ala, Mageto, J.O., Ae-Ngibise, K., Akpalu, B., Akpalu, A., Agbokey, F., Adjei, P., Owusu-Agyei, S., Bottomley, C., Kleinschmidt, I., Doku, V.C.K., Odermatt, P., Neville, B., Nutman, T., Wilkins, P., Noh, J., 2016. Electroencephalographic features of convulsive epilepsy in Africa: a multicentre study of prevalence, pattern and associated factors. Clin. Neurophysiol. 127, 1099–1107. doi:10.1016/j.clinph.2015.07.033.

Kim, SP., 2018. Preprocessing of EEG. In: Im, CH. (Ed.), Computational EEG Analysis. Biological and Medical Physics, Biomedical Engineering. Springer, Singapore doi:10.1007/978-981-13-0908-3_2.

Kothe, C.A., Makeig, S., 2013. BCILAB: a platform for brain–computer interface development. J. Neural Eng. 10, 056014. doi:10.1088/1741-2560/10/5/056014.

Krishnaveni, V., Jayaraman, S., Anitha, L., Ramadoss, K., 2006. Removal of ocular artifacts from EEG using adaptive thresholding of wavelet coefficients. J. Neural Eng. 3. doi:10.1088/1741-2560/3/4/011.

Leach, S.C., Morales, S., Bowers, M.E., Buzzell, G.A., Debnath, R., Beall, D., Fox, N.A., 2020. Adjusting ADJUST: optimizing the ADJUST algorithm for pediatric data using geodesic nets. Psychophysiology 57, e13566. doi:10.1111/psyp.13566.

Lei, X., Liao, K., 2017. Understanding the influences of EEG reference: a large-scale brain network perspective. Front. Neurosci. 11, 205. doi:10.3389/FNINS.2017.00205/BIBTEX.

Lema-Condo, E.L., Bueno-Palomeque, F.L., Castro-Villalobos, S.E., Ordonez-Morales, E.F., Serpa-Andrade, L.J., 2017. Comparison of wavelet transform symlets (2-10) and daubechies (2-10) for an electroencephalographic signal analysis. In: Proceedings of the IEEE 24th International Congress on Electronics, Electrical Engineering and Computing, INTERCON. Institute of Electrical and Electronics Engineers Inc doi:10.1109/INTERCON.2017.8079702, 2017.

Lopez, Kelsie, Monachino, Alexa, Morales, Santiago, Leach, Stephanie, Bowers, Maureen, Zeanah, Charles, Marshall, Peter, Fox, Nathan, Nelson, Charles, & Gabard-Durnam, Laurel. (2021). Low Density EEG Files from Bucharest Early Intervention Project for HAPPILEE Software [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5088346.

Luck, S.J., 2005. An introduction to event-related potentials and their neural origins. In: An Introduction to the Event-Related Potential Technique. MIT Press, pp. 1–50.

Melnik, A., Legkov, P., Izdebski, K., Kärcher, S.M., Hairston, W.D., Ferris, D.P., König, P., 2017. Systems, subjects, sessions: to what extent do these factors influence eeg data? Front. Hum. Neurosci. 150. doi:10.3389/FNHUM.2017.00150.

Mitra, P.P., Pesaran, B., 1999. Analysis of dynamic brain imaging data. Biophys. J. 76, 691–708. doi:10.1016/S0006-3495(99)77236-X.

Mognon, A., Jovicich, J., Bruzzone, L., Buiatti, M., 2011. ADJUST: an automatic EEG artifact detector based on the joint use of spatial and temporal features. Psychophysiology 48, 229–240. doi:10.1111/j.1469-8986.2010.01061.x.

Monachino, A.D., Lopez, K.L., Pierce, L.J., Gabard-Durnam, L.J., 2021. The HAPPE plus event-related (HAPPE+ER) software: a standardized processing pipeline for event-related potential analyses. Developmental Cognitive Neuroscience.

Mullen, T., 2012. CleanLine EEGLAB Plugin.

Mullen, T., Kothe, C., Chi, Y.M., Ojeda, A., Kerth, T., Makeig, S., Cauwenberghs, G., Jung, T.P., 2013. Real-time modeling and 3D visualization of source dynamics and connectivity using wearable EEG. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society. EMBS, pp. 2184–2187. doi:10.1109/EMBC.2013.6609968.

Nagabushanam, P., George, S.T., Dolly, D.R.J., Radha, S., 2020. Artifact cleaning of motor imagery EEG by statistical features extraction using wavelet families. Int. J. Circuit Theory Appl. 48, 2219–2241. doi:10.1002/cta.2856.

Nolan, H., Whelan, R., Reilly, R.B., 2010. FASTER: fully automated statistical thresholding for EEG artifact rejection. J. Neurosci. Methods 192, 152–162. doi:10.1016/j.jneumeth.2010.07.015.

Olejarczyk, E., Jernajczyk, W., 2017. Graph-based analysis of brain connectivity in schizophrenia. PLoS ONE 12, e0188629. doi:10.1371/JOURNAL.PONE.0188629.

Paul, J.K., Iype, T., R, D., Hagiwara, Y., Koh, J.E.W., Acharya, U.R., 2019. Characterization of fibromyalgia using sleep EEG signals with nonlinear dynamical features. Comput. Biol. Med. 111, 103331. doi:10.1016/j.compbiomed.2019.103331.

Pedroni, A., Bahreini, A., Langer, N., 2019. Automagic: standardized preprocessing of big EEG data. Neuroimage 200, 460–473. doi:10.1016/j.neuroimage.2019.06.046.

Pellinen, J., Carroll, E., Friedman, D., Boffa, M., Dugan, P., Friedman, D.E., Gazzola, D., Jongeling, A., Rodriguez, A.J., Holmes, M., 2020. Continuous EEG findings in patients with COVID-19 infection admitted to a New York academic hospital system. Epilepsia 61, 2097–2105. doi:10.1111/epi.16667.

Peng, W., Peng, W., 2019. EEG Preprocessing and Denoising. In: Extr.EEG Signal Processing and Feature Extraction. Springer Nature Singapore Pte Ltd., pp. 71–87. doi:10.1007/978-981-13-9113-2_5.

Petrichella, S., Vollere, L., Ferreri, F., Guerra, A., Määtta, S., Könönen, M., Di Lazzaro, V., Iannello, G., 2016. Channel interpolation in TMS-EEG: a quantitative study towards an accurate topographical representation. In: Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBS, pp. 989–992. doi:10.1109/EMBC.2016.7590868 2016-October.

Robeson, S.M., 1997. Spherical methods for spatial interpolation: review and evaluation. Cartogr. Geogr. Inf. Sci. 24, 3–20. doi:10.1559/152304097782438746.

Rodrigues, J., Weiss, M., Hewig, J., Allen, J., 2020. EPOS: EEG Processing Open-Source Standardization. PsychArchives.

Salis, C.I., Malissovas, A.E., Bizopoulos, P.A., Tzallas, A.T., Angelidis, P.A., Tsalikakis, D.G., 2013. Denoising simulated EEG signals: a comparative study of EMD, wavelet transform and Kalman filter. In: Proceedings of the 13th IEEE International Conference on BioInformatics and BioEngineering, IEEE BIBE doi:10.1109/BIBE.2013.6701613, 2013.

Siddiqi, O.K., Elafros, M.A., Sikazwe, I., Birbeck, G.L., Kalungwana, L., Potchen, M.J., Bositis, C.M., Koralnik, I.J., Theodore, W.H., 2015. Acute EEG findings in HIV-infected: zambian adults with new-onset seizure. Neurology 84, 1317–1322. doi:10.1212/WNL.0000000000001411.

Sokolov, E., Abdoul Bachir, D.H., Sakadi, F., et al., 2020. Tablet-based electroencephalography diagnostics for patients with epilepsy in the West African Republic of Guinea. Eur. J. Neurol. 27, 1570–1577. doi:10.1111/ene.14291.

Strang, G., Nguyen, T., 1996. Wavelets and Filter Banks. Wellesley-Cambridge Press.

Tanner, D., Morgan-Short, K., Luck, S.J., 2015. How inappropriate high-pass filters can produce artifactual effects and incorrect conclusions in ERP studies of language and cognition. Psychophysiology 52, 997–1009. doi:10.1111/PSYP.12437.

Tiwari, A.K., Pachori, R.B., Kanhangad, V., Panigrahi, B.K., 2017. Automated diagnosis of epilepsy using key-point-based local binary pattern of EEG signals. IEEE J. Biomed. Heal. Inform. 21, 888–896. doi:10.1109/JBHI.2016.2589971.

Troller-Renfree, S.V., Morales, S., Leach, S.C., Bowers, M.E., Debnath, R., Fifer, W.P., Fox, N.A., Noble, K.G., 2021. Feasibility of assessing brain activity using mobile, in-home collection of electroencephalography: methods and analysis. Dev. Psychobiol. doi:10.1002/dev.22128.

van den Munckhof, B., Alderweireld, C., Davelaar, S., van Teeseling, H.C., Nikolakopoulos, S., Braun, K.P.J., Jansen, F.E., 2018. Treatment of electrical status epilepticus in sleep: clinical and EEG characteristics and response to 147 treatments in 47 patients. Eur. J. Paediatr. Neurol. 22, 64–71. doi:10.1016/j.ejpn.2017.08.006.

Whedon, M., Perry, N.B., Bell, M.A., 2020. Relations between frontal EEG maturation and inhibitory control in preschool in the prediction of children's early academic skills. Brain Cogn. 146, 105636. doi:10.1016/j.bandc.2020.105636.

Widmann, A., Schröger, E., Maess, B., 2015. Digital filter design for electrophysiological data – a practical approach. J. Neurosci. Methods 250, 34–46. doi:10.1016/J.JNEUMETH.2014.08.002.

Williams, J.A., Cisse, F.A., Schaekermann, M., Sakadi, F., Tassiou, N.R., Hotan, G.C., Bah, A.K., Hamani, A.B.D., Lim, A., Leung, E.C.W., Fantaneanu, T.A., Milligan, T.A., Khatri, V., Hoch, D.B., Vyas, M.V., Lam, A.D., Cohen, J.M., Vogel, A.C., Law, E., Mateen, F.J., 2019. Smartphone EEG and remote online interpretation for children with epilepsy in the Republic of Guinea: quality, characteristics, and practice implications. Seizure 71, 93–99. doi:10.1016/j.seizure.2019.05.025.

Winkler, I., Brandl, S., Horn, F., Waldburger, E., Allefeld, C., Tangermann, M., 2014. Robust artifactual independent component classification for BCI practitioners. J. Neural Eng. 11, 035013. doi:10.1088/1741-2560/11/3/035013.

Xue, Z., Li, J., Li, S., Wan, B., 2006. Using ICA to Remove Eye Blink and Power Line Artifacts in EEG. First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06), 3, pp. 107–110.

Yao, D., 2001. A method to standardize a reference of scalp EEG recordings to a point at infinity. Physiol. Meas. 22, 693. doi:10.1088/0967-3334/22/4/305.

You, R.Y., Chen, Z., 2005. Blind source separation of multichannel electroencephalogram based on wavelet transform and ICA. Chin. Phys. 14, 2176–2180. doi:10.1088/1009-1963/14/11/006.

Zeanah, C.H., Egger, H.L., Smyke, A.T., Nelson, C.A., Fox, N.A., Marshall, P.J., Guthrie, D., 2009. Institutional rearing and psychiatric disorders in romanian preschool children. Am. J. Psychiatry 166, 777–785. doi:10.1176/appi.ajp.2009.08091438.

Zeanah, C.H., Nelson, C.A., Fox, N.A., Smyke, A.T., Marshall, P., Parker, S.W., Koga, S., 2003. Designing research to study the effects of institutionalization on brain and behavioral development: the bucharest early intervention project. Dev. Psychopathol. 15, 885–907. doi:10.1017/S0954579403000452.