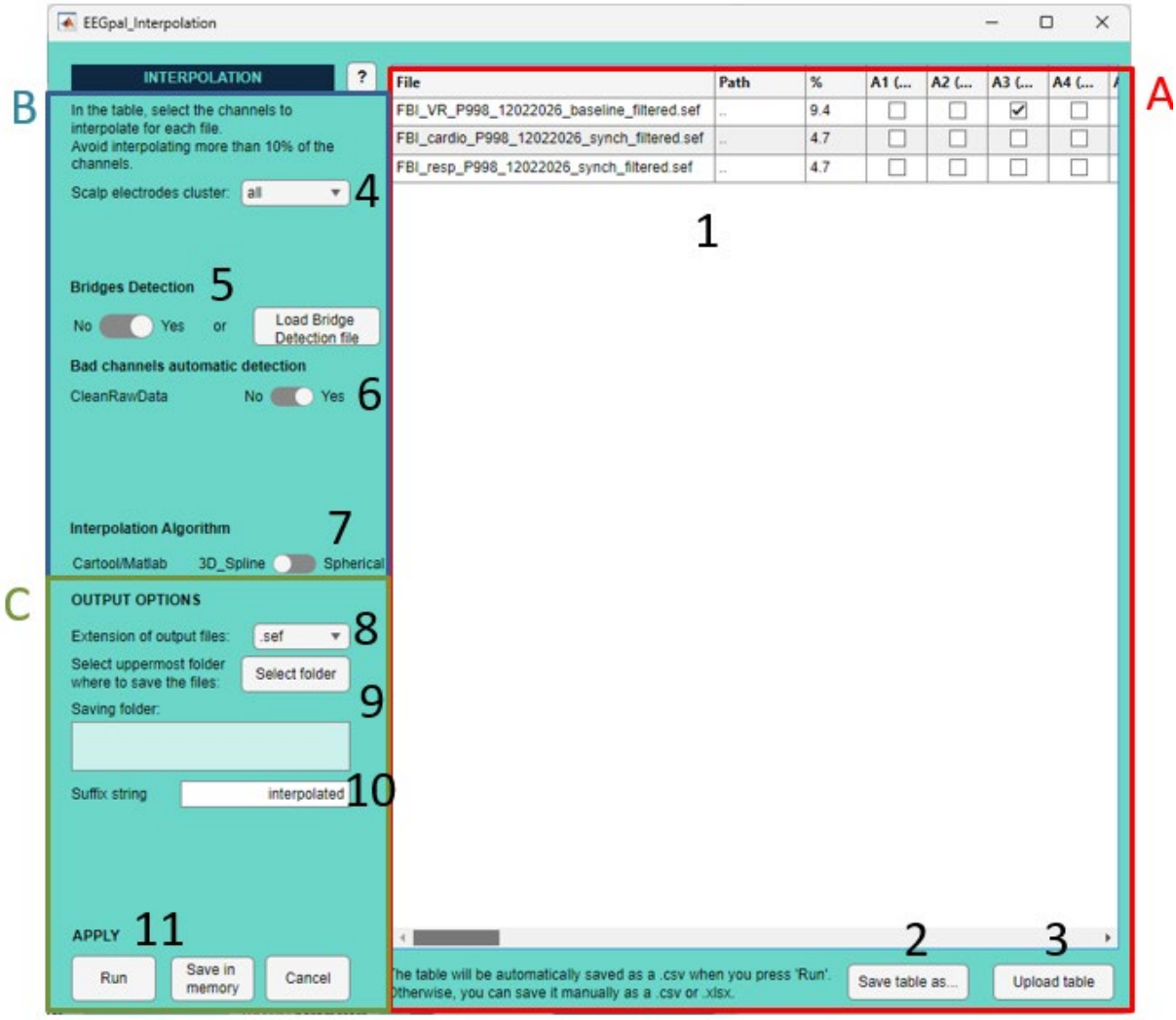
EEGpal: **Interpolation module**

Version 2.01, 20.11.2025

Video tutorial: https://youtu.be/TwBe_8CH-i8

The interpolation module removes bad electrodes/channels by replacing them with an estimate of the signal using neighboring channels. Users can choose from two different algorithms to perform this interpolation.



Pannel A:

1. This table lists all the files and electrodes. You can manually tick or untick the channel you want to interpolate. However, we advise you to first save the table (point **2**), complete it in Excel and upload the new table (point **3**), which is much easier (see FAQ **How to determine which electrodes to interpolate?**). The % column automatically indicates the percentage of

the electrode to be interpolated for each file. As a general rule, try to avoid interpolating more than 10% per file.
2. Save the table as '.csv' or '.xlsx' file. It would be much easier to work on it in Excel
3. Upload the '.csv' or '.xlsx' after filling it

Pannel B:

4. Select to work only within an electrode cluster if you have defined one in the Electrode Setting panel of the main EEGpal window.
5. Bridge detection: This occurs when two channels are connected (usually due to gel contact), resulting in them recording the same signal. This is something we want to avoid, and these channels need to be interpolated.
   The slider button on the left is available only if you have run the **Bridge Detection** module from EEGpal before to open the I**nterpolation module**, or if you have loaded a *Bridge_detection.xlsx* file using the button 'Load Bridge Detection file' on the right. It will automatically fill the interpolation table (Panel A) with the bridged channel. You can activate or deactivate this feature if you want to remove these channels from the interpolation table.
6. Enable or disable the suggestion of bad electrodes made by CleanRawData. This option is only available if the interpolation module detects that you have activated the **Flag bad channels** option in the Clean Rawdata panel of the **Filtration+** module. These bad channels could be added here in the interpolation table (Yes) or not (No).
   Note from the EEGpal authors: we advise you not to use this option as we are not convinced by the results (see in FAQ **how to determine which electrodes to interpolate?)**.
7. Here, you can choose from two interpolation algorithms:
   a. 3D_Spline is the method used by the Cartool software. It is based on the article: *Perrin, Pernier, Bertrand, and Echallier (1989). PubMed #2464490*.
   b. Spherical is the default method used by eeglab

   See FAQ for more details

Pannel D:

8. Select the format for the output files.
9. Select the destination folder where the results files will be saved (note: it reproduces the input structure. For example, a folder per participant if the input files where in subfolder).
10. The suffix added to the input filename to obtain the output filename
11. There are three validation buttons:
    a. The **Run** button will carry out the processing parameterized in the Interpolation module.
    b. The **Save in memory** button will store all the parameters in memory and close the Interpolation module without performing the processing.
    c. The button **Cancel** closes the module without processing and without keeping the entered parameters in memory. The same effect will be achieved by closing the Interpolation module window.

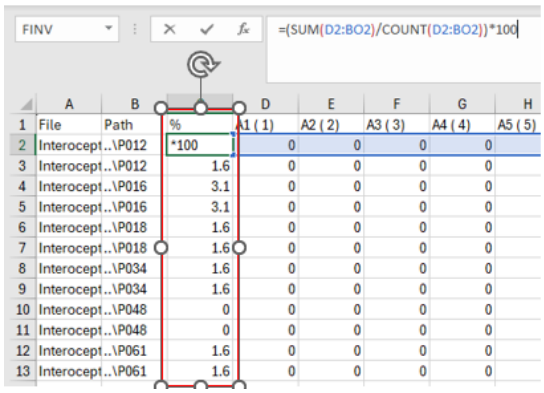
FAQ

How can you determine which electrodes to interpolate?

The choice of which electrode to interpolate depends largely on the user, making this the most difficult step in the EEG pre-processing pipeline. Although people have tried to implement an automatic method, the results have been disappointing (see point 6). The best method is to perform a visual inspection. Here, I will explain how I perform this step, but I don't claim that it is the best way. We will use Cartool software to visualise our data (https://cartool.cibm.ch/download/).

Before starting, I would open the Interpolation module and perform the following steps:

a. Activate the Bridge detection (see point 5).
b. Press on the "Save Table as" to save the table on your hard drive (see point 2).
c. Open this file with Excel.
d. This file will help you decide which channels to interpolate. Leave it at 0 to keep a channel, or put 1 when you want to interpolate it. To perform an automatic calculation of the percentage of channel rejection, enter the following formula into cell C2: `=(SUM(D2:BO2)/COUNT(D2:BO2))*100 ;` This formula is for a 64-electrode configuration, but it must be adjusted if a different number of electrodes is used. Then copy and paste it into all the cells in the third column (column C).



	A	B	C	D	E	F	G	H
1	File	Path	%	A1 (1)	A2 (2)	A3 (3)	A4 (4)	A5 (5)
2	Interocept	..\P012	*100	0	0	0	0	
3	Interocept	..\P012	1.6	0	0	0	0	
4	Interocept	..\P016	3.1	0	0	0	0	
5	Interocept	..\P016	3.1	0	0	0	0	
6	Interocept	..\P018	1.6	0	0	0	0	
7	Interocept	..\P018	1.6	0	0	0	0	
8	Interocept	..\P034	1.6	0	0	0	0	
9	Interocept	..\P034	1.6	0	0	0	0	
10	Interocept	..\P048	0	0	0	0	0	
11	Interocept	..\P048	0	0	0	0	0	
12	Interocept	..\P061	1.6	0	0	0	0	
13	Interocept	..\P061	1.6	0	0	0	0	

e. Remember that it is recommended to interpolate a maximum of 10% of the signal (i.e. six electrodes for a 64-channel EEG cap).

The authors now propose a strategy for detecting visually defective electrodes and interpolating them, depending on the type of study:

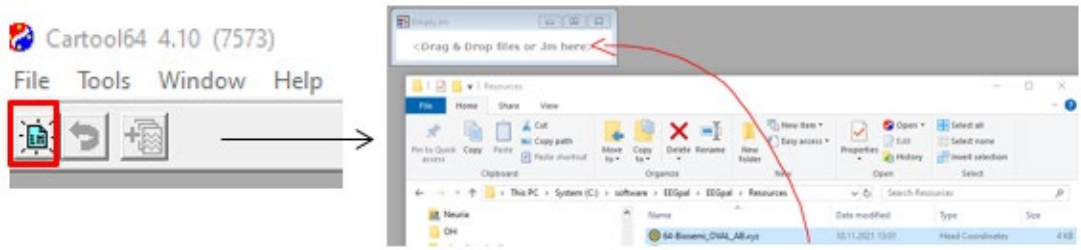
Case 1: Evoked-related potential study (ERP study).

To make things easier, we are going to temporarily perform Epoching to create ERP files from the current file. Please refer to the Epoching module manual to learn how to generate averaged files per trigger.

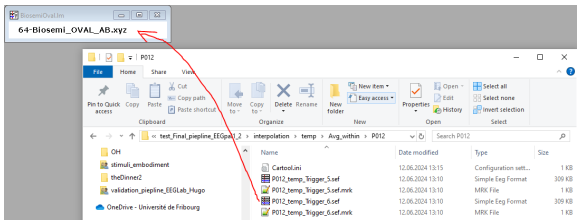
Now you will inspect the ERP to detect potential bad channels after averaging.

a. Open Cartool

b. Create a link any file (.lm) and drap_and_drop a electrode file (for example the file *64-Biosemi_Oval_DePretto.xyz* inside of the Resources sub-folder of EEGpal folder. Save it somewhere easy to access, because you will need to open it several times.



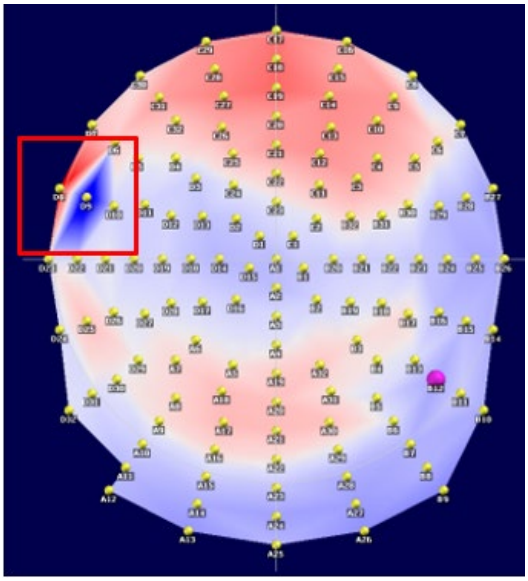
c. Visualization:
   i. Drap and drop an ERP file into the lm file



   ii. Press on the topography windows, then 1x  ,then 1x  , then 1x 

   iii. Press on the electrode windows, then 2x 

d. **Test visually the topography**
   i. Put the cursor at the origin
   ii. Use the right arrow on the keyboard to move the cursor and observe the change in topography carefully. Check that the electric charge of an electrode is not incoherent with neighboring electrodes during a stable period of time (and not at the ERP end). For example, the electrode highlighted by the red rectangle in the picture below has an abnormal polarity that remains stable over time. This electrode should be interpolated.

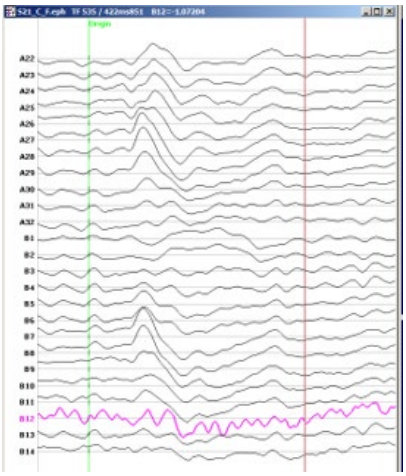


e. **test visually on the eeg-track**
   i. Click on the electrode window

ii. With the keyboard, press *on "SHIFT"* + *"-"* keys, then press 5x on *"+"* to see only 5 electrodes on the screen

iii. With the keyboard, press on *"CTRL"* + *"↑"* key to increase the amplitude (make the comparison easier)

iv. Change the channels on the screen with *"↑"* and *"↓"* key. Carefully look at the trace of electrodes. If there is a weird electrode in comparison to **SPATIAL** neighbors (huge difference of oscillations) = electrode to interpolate.
For example, as can be seen in the picture below, B12 is unusual compared to its neighboring squares.



v. Further tools to help decide whether a chanell needs to be interpolated or not:

f. Close the .lm file and reopen the one you saved in point **b**. Repeat points **c** to **e** with the other ERP files of the same participant. Decide which electrode you will actually interpolate, then fill in the interpolation file according to your choice. Remember the following rules when making your decision:
   i. Max 10% of electrodes per participant to interpolate. If you really need to have more, consider the participant as an outlier.
   ii. interpolate the SAME channels across one participant
   iii. Then bridged channel detects need absolutely to be interpolated
   iv. If two neighbors are bad, interpolate only one of them (preference to the electrode which is not on the border of the hat).

g. Fill the interpolation file with your choices and save it (keep the .xlsx format)

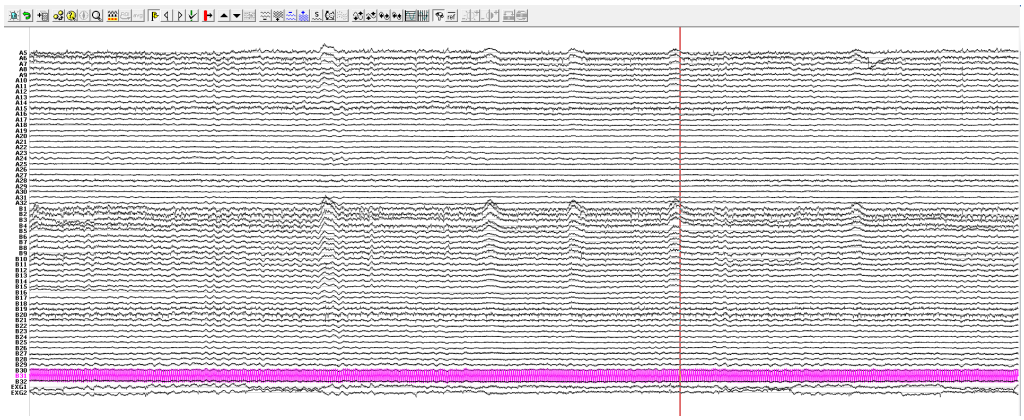
<u>Case 2: Non-ERP study like Resting state</u>

In this case, you cannot use temporary epoching to help you. You have to look at the raw signal instead.

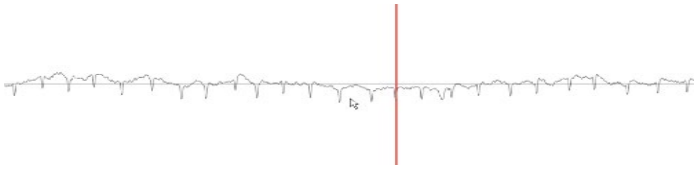
Look at the channels visually. Only note the channels that are clearly bad.

Examples of bad channels:

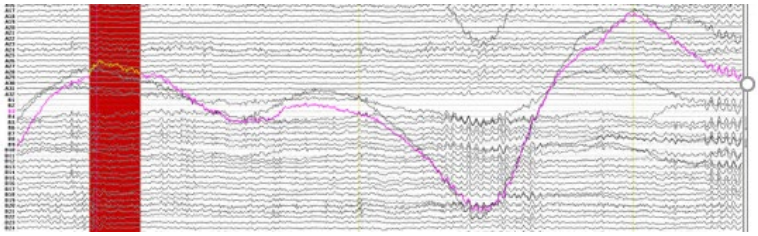
a. Electrodes with continuous oscillation that differ from neighbouring electrodes



b. A regular spike indicates that it is a technical artefact and not a physiological signal.



c. A single channel with an abnormally high amplitude that differs from its neighbours over a long period of time.



Focus on the technical artefact rather than the participant's transient movements. If you notice any such channels, make a note of them!

Section written by Michael Mouthon, FNDlab, University of Fribourg

Which functions are used for interpolation?

As explained in point 7, there are two possible algorithms for performing the interpolation:

a. For the 3D_spline (based *Perrin, Pernier, Bertrand, and Echallier (1989). PubMed #2464490),* EEGpal used the function **interpolate_perrinX.m** written by Mike X Cohen which is available in the material associate to his book *Analyzing Neural Time Series Data: Theory and Practice* (MIT Press, 2014) https://github.com/mikexcohen/AnalyzingNeuralTimeSeries
b. For the Spherical option, EEGpal use the eeglab function **pop_interp** developed by Arnaud Delorme, CERCO, CNRS, 2009.

Which of these two solutions should I use?

Regarding the choice of interpolation method, both options have been fully validated. The authors prefer the 3D_Spline method, which, unlike the spherical method, does not assume that the EEG electrode has spherical coverage (because the human head is oval, not spherical). However, using the eeglab version may be easier to justify to a reviewer.

Can I trust CleanRawData's automatic suggestion to flag bad channels?

The flagged channel from the CleanRawData algorithm is described in the following webpage: https://eeglab.org/tutorials/06_RejectArtifacts/cleanrawdata.html

Basically, the bad detection is based on three criterium:

1. Check if the channel has the same value during a long period of time. It is considered flat channel with a recording problem.
2. This is the most important method. It has for purpose to estimate the level of high-frequency noise.
    a. First, it estimate how noisy is the signal using this formula:

$$\text{noisiness} = \frac{\text{MAD}(high-pass\ filtered\ data)}{\text{MAD}(low-pass\ filtered\ data)}$$

[MAD: median absolute deviation]

    b. It estimate the standard deviation of this noise using this formula:

$$\text{znoise} = (\text{noisiness} - \text{median}(\text{noisiness}))./\sigma$$

$$\sigma = \text{MAD}(\text{noisiness},1)*1.4826$$

    c. Check if this standard deviation reach the specified threshold (by default 4) :

$$\text{noise\_mask} = \text{znoise} > \text{noise\_threshold}$$

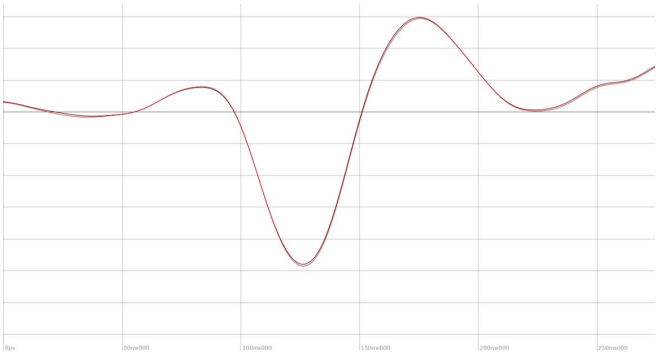
3. Check if there are channels which have a high value of correlation. It basically does the same work as the Bridge Detection module.

Source: https://sccn.ucsd.edu/githubwiki/files/cleanRawData_30thEEGLABWorkshop.pdf

The authors compared the results of an ERP analysis using either a visual inspection of the bad electrode (see the 'How to determine which electrodes to interpolate?' section of this manual for procedure details) or automatic bad channel detection using the CleanRawData EEGLAB plugin. This study was conducted with a cohort of 40 participants. The dataset is freely available at this address: https://osf.io/pfde9/. It is a visual face recognition paradigm that generates an N170 response around the parieto-occipital position (PO8). These data were recorded with a Biosemi EEG system.

Firstly, we compared the number of bad electrodes with the number of interpolated electrodes. We observed that CleanRawData interpolates around 75% more channels than humans do. This means that for every electrode flagged by human inspection, CleanRawData will flag 2.5 electrodes, which is much more conservative. On the positive side, in 90% of cases, the bad channel chosen by a human was also chosen by CleanRawData. Changing the 'Max acceptable high frequency std dev' threshold from 4 to 7 does not seem efficient because it still removes more channels than a human and the coherence between human and CleanRawData decreases.

Now if we compare the ERP result, with the grand average across participants, we observed that the N170 peak are statistically identical with both methods:



Black = interpolation after human visual inspection ; Red = interpolation with channels flag by CleanEawData

So, despite the fact that significantly more channels are interpolated with CleanRawData, the results are the same as if you were to perform a visual inspection yourself in this case. Whether to trust the automatic flagging of bad channels with CleanRawData is a matter of personal choice. It saves a lot of time, and the final results seem to be OK. The drawback is that you remove a lot of acceptable EEG signals, replacing them with artificial mathematical reconstructions.

Michaël Mouthon, the 15.09.2025