Supplementary Material: Different Algorithms *(Might)* Uncover Different Patterns

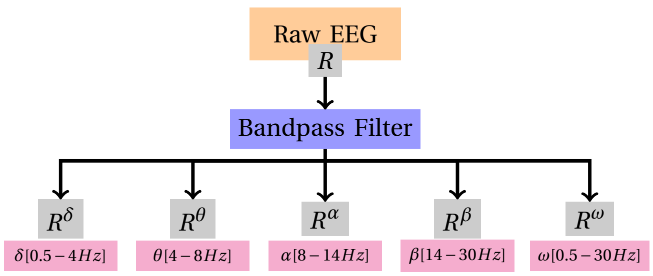

Fig. A.1: Illustration on how a EEG recording is filtered into the frequency bands of interest.

I. Feature Extraction Pipeline

Our feature extraction pipeline's architecture closely aligns with the procedural framework outlined in the work of [1], [2]. The focal frequency bands for our feature extraction encompass delta (0.5 - 4 Hz), theta (4 - 7 Hz), alpha (7 - 14 Hz), beta (14 - 30 Hz), as well as the whole spectrum ω (0.5 - 30 Hz) encompassing all frequency bands. To initiate the process, we implement a band-pass filter on recording R for each distinct frequency band. This yields filtered recordings denoted as R^{fb}, where $fb \in \delta, \theta, \alpha, \beta, \omega$ defines the frequency band specific filter applied on R (see Fig. A.1). Following this step, each signal R^{fb} is divided into a sequence of epochs (see Fig. A.2).

Lets denote a trail of $n \in \mathbb{N}$ epochs obtained by splitting a recording R^{fb} as $E^{fb} = (R_1^{fb}, R_2^{fb}, ..., R_n^{fb})$. In this representation, each epoch $R_i^{fb} = (c_{i1}, ..., c_{im})$ consists of a collection of channels denoted as $c_{ij} = (t_k, t_{k+1}, ..., tk + \Delta s)$. Here, Δs is defined as the product of the *sampling rate* and the *epoch duration*, encapsulating a span of time points.

We set the epoch duration to 2s for eyes open and to 4s for eyes closed to obtain 10 epochs from each recording. For feature extraction we used the *extract_features* method from the *mne_features* python library [3].

Let us symbolize a sequence of features derived from a series of epochs E^{fb}, as $F^{fb} = (F_1^{fb}, F_2^{fb}, ..., F_n^{fb})$. Within this context, the notation $F_i^{fb} = (f_{i1}, f_{i2}, ..., f_{im})$ takes on the role of representing the unique features intrinsic to an epoch R_i^{fb}, whereas f_{ij} represents a vector that captures a range of different features extracted from the signal $c_{ij} \in R_i^{fb}$ from the i-th epoch in channel j.

After extracting the sequence of features F^{fb} from a trail of epochs E^{fb}, we averaged the feature vectors across all epochs, given by $\overline{F}^{fb} = (\overline{f_1}, \overline{f_2}, ..., \overline{f_m})$ with $\overline{f_j} \in \overline{F}^{fb}$ being the mean feature vector over all n epochs from channel j.

$$\overline{f_j} = \frac{1}{n} \sum_{i=1}^{n} f_{ij}$$

The averaged trail of features $\overline{F}^{fb}$ for each frequency band $\delta, \theta, \alpha, \beta$ and ω is than flattened to a one dimensional vector and represents one training sample.

II. Further SHAP Analysis

For our top-performing model, XGBoost, the SHAP values indicated that higher slope values of the log-log PSD regression line, specifically in the omega band (0.5-30Hz), were positively correlated (refer to Fig. B.3 f)). This suggests a shift: as aging occurs, there's a decrease in lower frequencies and an increase in higher frequencies. The R2 coefficient of the log-log PSD regression fit exhibited significant model impact, as depicted in Fig. B.3 c). However, translating these results into actionable insights demands deep domain expertise. As highlighted in the primary section of the paper, aging led to a convergence between the upper 75% and lower 25% quantiles of theta activity. This trend is evident in Fig. B.3 a) and b). In the beta band, a reduced kurtosis with

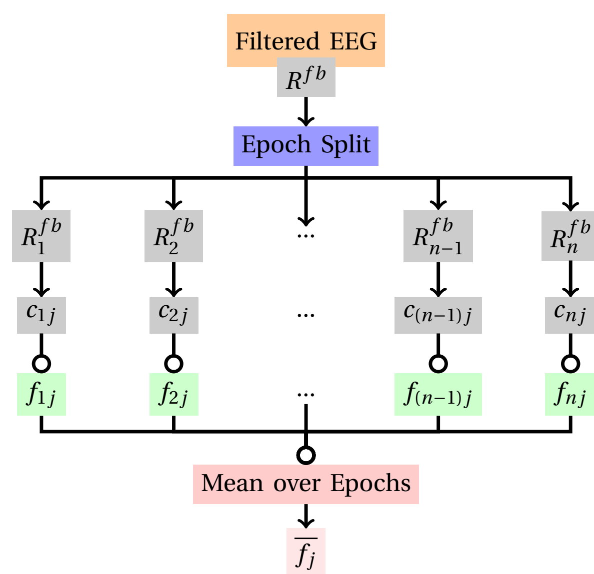

Fig. A.2: Illustration on how a filtered EEG signal is split into a set of n epochs. The feature vector f_{ij} is extracted from the epoch channel c_{ij}. By averaging over the feature vectors f_{ij} of an epoch i, we obtain $\overline{fj}$.

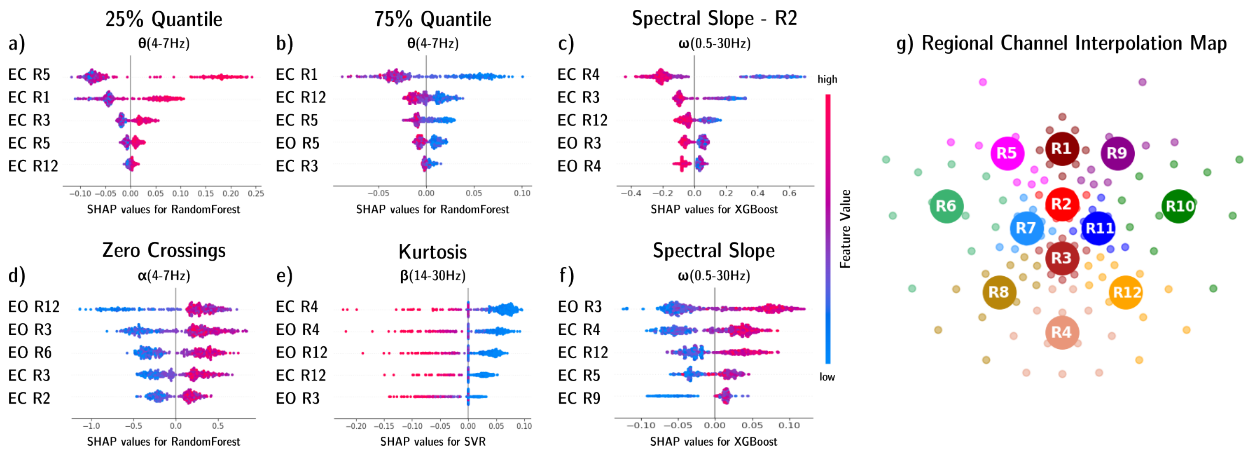

Fig. B.3: SHAP values (impact of a feature on the model's prediction) computed for RandomForest, XGBoost and SVR on 12-All. Each plot is specific to a feature in a frequency band: a) 25% and b) 75% Quantile in theta, c) R2 coefficient of the log-log PSD regression line fit, d) Zero Crossings in alpha, e) Kurtosis in beta, f) Slope of the log-log PSD regression line in omega. Figure g) shows a topological map of the sensory locations (subject facing up), where Small dots show electrodes and big dots the regional electrode. Labels on the y-axis indicate the resting state (EC: eyes closed, EO: eyes open) and region. The color coding represents the feature value, while the x-axis depicts its impact on the model's prediction. Each point corresponds to a testing sample.

higher age predictions hints at a wider signal distribution around the mean (see Fig. B.3 e). Meanwhile, elevated Zero Crossing values in the alpha band, as shown in Fig. B.3 d), suggest heightened neuronal activity in this frequency range with aging.

III. Feature Imitation Network

To train our Feature Imitation Network (FIN) models, we incorporated 400 non-labeled subjects from our dataset, originally designated for the brain age prediction challenge [4]. As these subjects couldn't be used for training age prediction models, they were, however, invaluable for training our FINs, which aim to predict underlying signal features. This approach expanded our training set to 3085 recordings, post-elimination of 115 faulty entries, a process detailed in the main body of the paper. We subjected all EEG recordings to the identical preprocessing routine previously established in our feature extraction pipeline, ensuring consistency across our dataset. Additionally, to simplify the FIN ensemble's overall complexity, we took a strategic step to further reduce channel dimensionality. We grouped the electrodes into five distinct regions, as depicted in Figure C.4, and performed signal averaging to interpolate the channels. Each recording was segmented into 2-second epochs (500 time points) after band-pass filtering to highlight our targeted five frequency bands $(\delta, \theta, \alpha, \delta, \omega)$. These filtered epochs formed individual samples in our training set. Using the *mne_features* library, we extracted the 75% and 25% quantiles from these signals [3], which became the target metrics for our FIN models.

We trained FIN models on EEG signals from five frequency bands $(\delta, \theta, \alpha, \delta, \omega)$ to predict the 75% and 25% quantiles. Inputs for each FIN model were one-dimensional vectors, each representing five regionally interpolated channels from an epoch, with a total of 2,500 features. The outputs were 10-dimensional vectors, based on the two quantiles forecasted for each channel. These models were trained on data specifically filtered for each frequency band, resulting in five unique FIN models. Refer to Figure C.5 for a visual representation of a FIN model. After every layer, except the output, we incorporated batch normalization, applied an activation function, and introduced dropout. For optimal hyper-parameters, we conducted a randomized grid search, experimenting with aspects like the number and size of hidden layers, learning rate, batch size, activation function, dropout rate, weight decay, and optimizer momentum. Additionally, we employed the *MinMaxScaler* from the scikit-learn package [5] to scale the target data during training. In Table I you can see the over all performance results for each of our five fin models. You can find the scatter plots of the actual and predicted values, as well as the error distribution plot in Figure C.7 and Figure C.8.

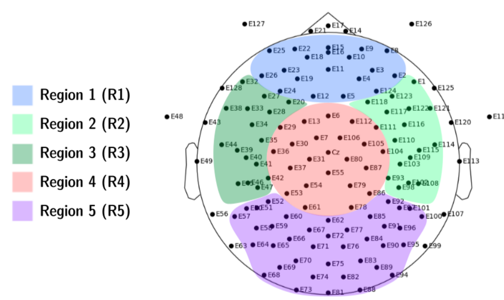

Fig. C.4: Topological map of the electrodes grouped into 5 regions. Each color represents a group of electrodes which were combined through signal averaging in effort of reducing the channel dimension.

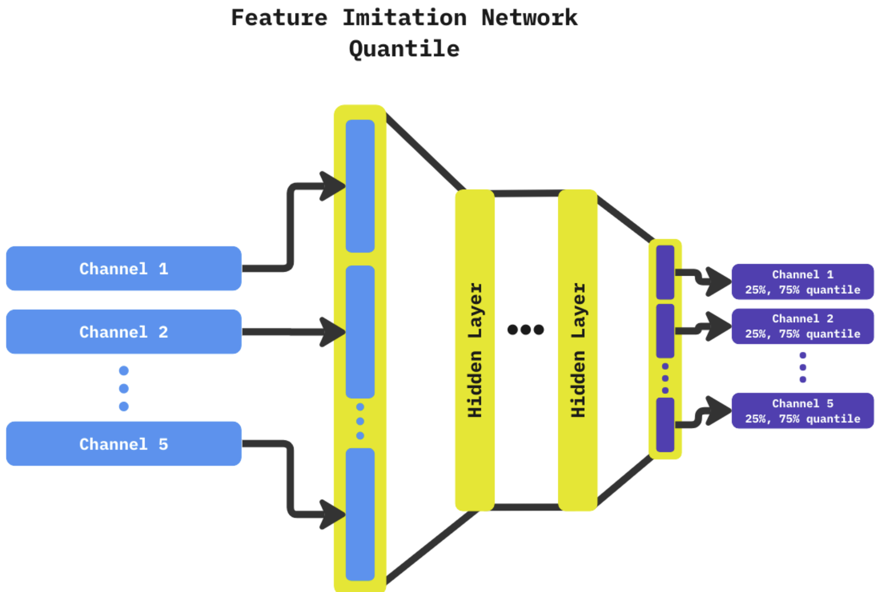

Fig. C.5: The figure illustrated one FIN model for predicting the upper 75% and lower 25% quantile for all five input channels specifically filtered into a distinct frequency band.

TABLE I: The table shows the R^2 regression score each model trained for a specific frequency band achieved.

FIN Model	R^2
quantile (delta)	0.863
quantile (theta)	0.761
quantile (alpha)	0.930
quantile (beta)	0.946
quantile (omega)	0.842

We assembled the FIN ensemble by removing each model's output layer and directly connecting its final layer to the meta-learner's input, as illustrated in Figure C.6. Initially, the FIN models were frozen, with only the meta-learner's weights adjustable during training. Once this phase concluded, we unfroze the FIN models for end-to-end training to refine the entire ensemble. Hyper-parameter tuning was executed for the meta-learner, which shared the same architectural blueprint as our FIN models. This tuning process mirrored the adjustments we made during the FIN training phase.

References

[1] O. Al Zoubi, C. Ki Wong, R. T. Kuplicki, H.-w. Yeh, A. Mayeli, H. Refai, M. Paulus, J. Bodurka, Predicting age from brain eeg signals—a machine learning approach, Frontiers in Aging Neuroscience (2018).

[2] D. Engemann, A. Mellot, R. Höchenberger, H. Banville, D. Sabbagh, L. Gemein, T. Ball, A. Gramfort, A reusable benchmark of brain-age prediction from m/eeg resting-state signals, NeuroImage (2022).

[3] J.-B. Schiratti, J.-E. Le Douget, M. Le van Quyen, S. Essid, A. Gramfort, An ensemble learning approach to detect epileptic seizures from long intracranial eeg recordings, in: ICASSP 2018 - 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP, 2018.

[4] NeuroTechX, INRIA, TAILOR, Brain age prediction challenge from eeg, https://tailor-network.eu/brain-age-prediction-challenge/ (11 2022).

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. e. a. Grisel, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

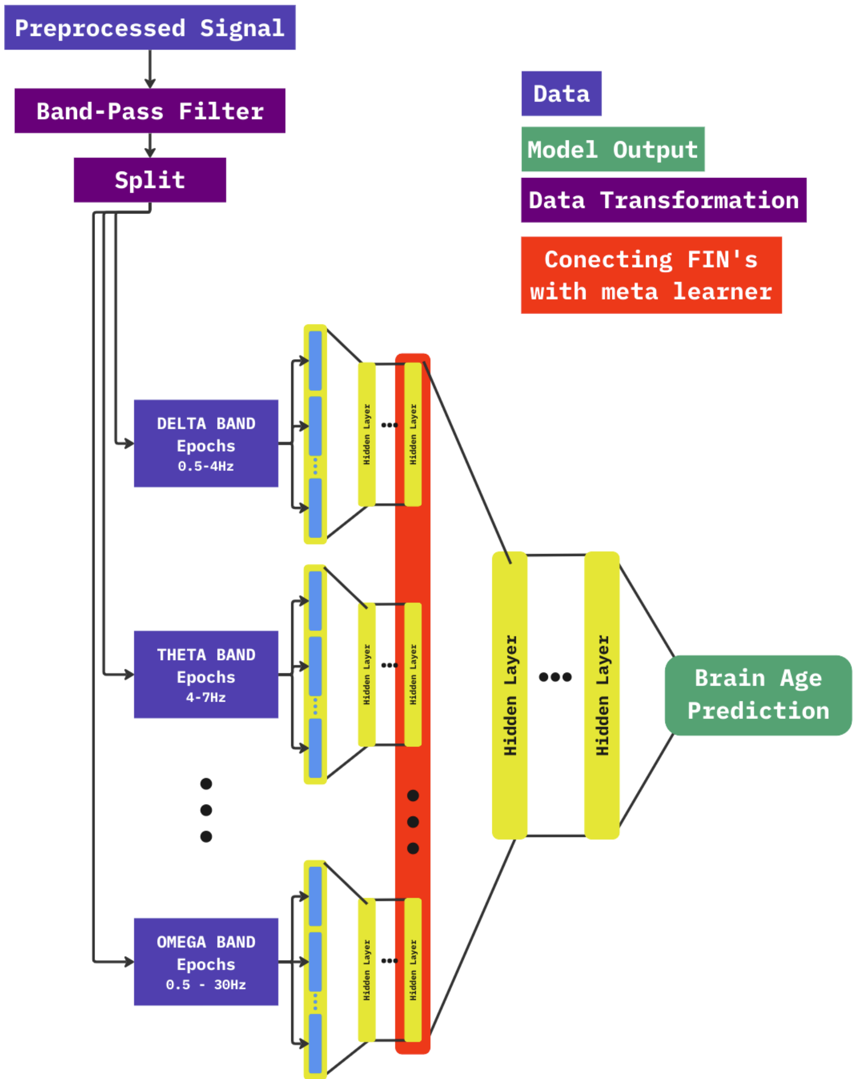

Fig. C.6: The figure illustrates the ensemble of FIN models all connected to the meta-learner predicting the age. For each FIN model, the output layer is removed. The last layer of the FIN models is then connected to the input layer of the meta-learner.

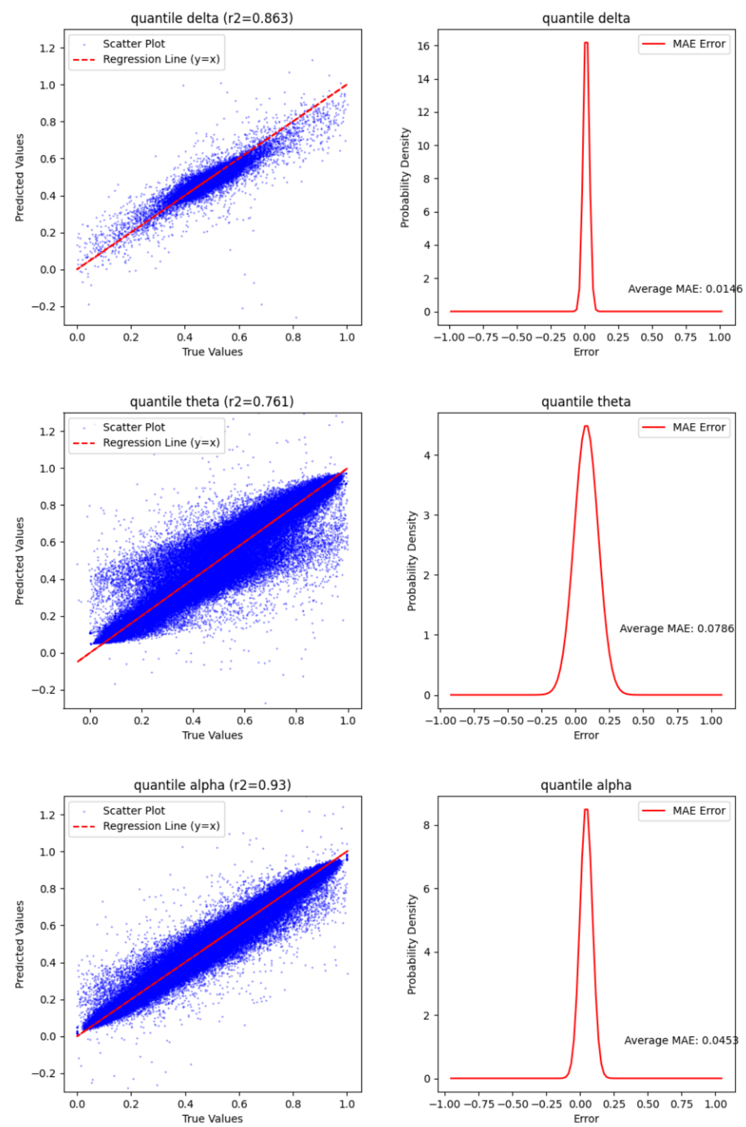

Fig. C.7: The figure shows scatter plots for the actual quantile values and the predicted values for the FIN trained on delta (top), theta (middle) and alpha (bottom) filtered data.

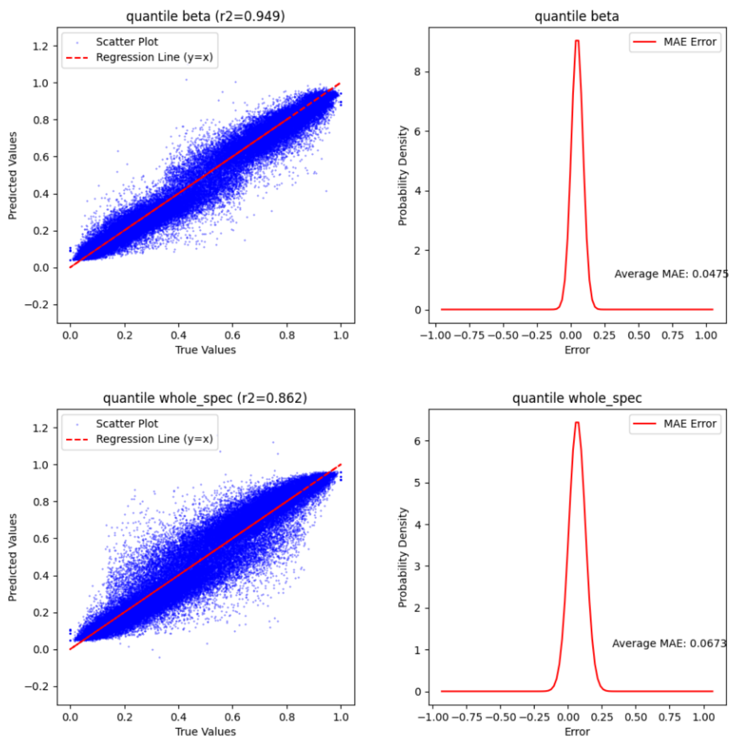

Fig. C.8: The figure shows scatter plots for the actual quantile values and the predicted values for the FIN trained on beta (top) and omega (bottom) filtered data.