

# Comparative Analysis of Generalization and Harmonization Methods for 3D Brain fMRI Images: A Case Study on OpenBHB Dataset

## Supplementary Material

### 7. Rationale

#### 7.1. Model

In this work, We adopt the architecture proposed in the paper "Improved Brain Age Estimation with Slice-based Set Networks" [8]. This model introduces a novel approach for making predictions from 3D MRIs by encoding each 2D slice with a deep 2D-CNN model. The information from these 2D-slice encodings is combined using set networks or permutation invariant layers, such as mean aggregation. The model is designed to preserve spatial information about the ordering of slices.

Our brain age prediction model is an adaptation of a framework initially designed for brain cancer classification [8]. In this modification, we transform the model into a regression task for predicting the age of the brain.

The model architecture consists of six main layers. Layer 1 is a 3D Convolutional Layer that takes one input and produces 32 output channels. Subsequent layers, following this pattern, progressively increase the number of channels, with Layer 2 producing 64, Layer 3 producing 128, and Layer 4 producing 256 channels. Layer 5 which is another 3D Convolutional Layer consolidates learned features by accepting 256 channels and outputting 64 channels.

Two additional layers contribute to feature synthesis. Layer 6, a 3D Convolutional Layer processes the 64 channels and produces the final feature representation for age prediction. The Output Layer consists of a single 3D Convolutional Layer, taking 64 channels and generating the predicted brain age.

#### 7.2. Training Parameters

The model is trained for 50 epochs with a learning rate of 0.01 and weight decay of 0.001. We use the Mean Squared Error (MSE) loss function during training and employ the Adam optimizer. For DANN and DANNCE, we set penalty weight as 0.1, featurizer learning rate as  $1e^{-4}$ , classifier learning rate as  $1e^{-3}$  and discriminator learning rate as  $1e^{-3}$ . For deepCORAL we set the weight penalty equal to DANN and DANNCE which is 0.1.