Here we provide further information on some sections of the paper. First, we include an extended table containing a more complete comparison of available dyadic interaction datasets. Then, we describe the rationale behind using 32 frames per video chunk and the procedure used to crop the face-only videos, as part of the proposed methodology. Finally, we detail the training strategy, the algorithm used to define the data splits (such that a balance was kept on the participants and sessions features) and report the resulting distribution of OCEAN values among them.

S1. Face-to-face dyadic datasets comparison (Sec. 2)

For the sake of completeness, Table S1 contains an extended review of publicly available face-to-face dyadic interactions datasets that contain at least audiovisual data. Most of the datasets are tailor-made for too specific purposes or limited in the number of participants, recordings, views, context annotations or language. Hence, there is no big enough general purpose database in the literature that could allow for an integral analysis of both, the interaction and the participants.

S2. Size of video chunks (Sec. 4.1)

The original Video Action Transformer [13] uses an I3D backbone pretrained on Kinetics-400 [6] for spatiotemporal feature extraction. Such backbone uses 64 frames per chunk, which is equivalent to around 3 seconds of video. Instead, we opted for the R(2+1)D backbone [25] pretrained on IG-65M dataset, which has shown to provide significant performance gains [12]. This backbone uses 32 frames per chunk, so by using a stride of 2 we manage to encode approximately the same time window as the original method with half the number of frames while reducing the memory load. This is equivalent to downsampling the original videos from 25 fps to 12.5, that is, 1 frame every 0.08 seconds. Although not frequent, there is a chance to miss some fast-paced facial and body micro-actions in such downsampling process. However, there is also the trade-off we try to balance between losing some of these fast micro-actions and being able to include a larger, and also important, temporal context.

S3. Face detection and tracking (Sec. 4.1)

As described in the main paper, we use a face chunk video as one of the inputs of the model, which is used together with the participants’ metadata to form the query of the transformer model. In order to detect the faces we use MobileNet-SSD [14], deployed using Tensorflow Object Detection API [15] and pretrained on the Wider Face Dataset [27]. As we consider only frontal cameras, the detection task is not very challenging, therefore, on more than 95% of the videos the detection ratio is higher than 75%. In case the gap between consecutive detections is lower than 25 frames (1 second), we linearly interpolate the coordinates of the boxes. Since there are frames in which the frontal cameras capture both participants, we need to identify the target person before computing the face chunks. In order to do so, we employ a basic tracking algorithm based on the following 2 steps: (1) identify target person’s face: given a video, the face of the target person is considered the first detection that has a mean intersection over union (IoU) score higher than 0.2 with respect to all the other faces in the video; (2) track target person face throughout the video based on the IoU.
S4. Training strategy (Sec. 4.2)

The proposed model was trained using Adam optimizer with \( \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e^{-8} \) and a learning rate of 1e\(-5\). We used a batch size of 2 and the Mean Squared Error as the loss function. We compute the validation error approximately 30 times per epoch and select the model that gives the best results considering the mean with its previous and next evaluation scores. The final results, detailed in Sec. 4.3 of the main paper, were obtained by freezing the layers of the R(2+1)D backbones, as strategies such as finetuning end-to-end or only the last block of the feature extractors led to fast overfitting.

S5. Personality trait (OCEAN) values over splits (Sec. 4.2)

In this section, we briefly describe the procedure used to define the data splits used during the experiments described in the experimental section.

In order to split the data among training, validation and test subsets, some sessions needed to be removed so that no participants were repeated in any of the subsets. The final split was selected using a greedy optimization method that iteratively removed and added sessions based on their importance until a valid split ratio was found. Such importance was determined by the groups distribution and the importance until a valid split ratio was found. Such importance was determined by the groups distribution and the importance until a valid split ratio was found.

In order to define the data splits used during the experiments described in the experimental section.

In order to split the data among training, validation and test subsets, some sessions needed to be removed so that no participants were repeated in any of the subsets. The final split was selected using a greedy optimization method that iteratively removed and added sessions based on their importance until a valid split ratio was found. Such importance was determined by the groups distribution and the number of remaining sessions per participant. In particular, the method tried to minimize a set of costs to: (1) ensure that distributions among splits were not different by means of a Kolmogorov-Smirnov significance test [17]; (2) ensure that Pearson’s correlation of gender, age and per-
sonality values among splits did not differ by a large margin; (3) attempt to have a uniform distribution in validation and test with respect to age and gender to correct selection bias; (4) attempt to have a close-to-uniform distribution of group combinations; and (5) try to maximize the number of sessions without losing participants, while considering also the train/validation/test ratio. The resulting distribution of OCEAN values among splits can be seen in Fig. S1.

References


Figure S1. Distribution of the self-reported personality trait (OCEAN) values across train, validation and test splits used to evaluate the proposed personality inference method. X axis refers to z scores for each personality trait. Y axis refers to number of participants.


