A SUPPLEMENTARY MATERIAL

A.1 FaceXHuBERT Algorithm

Algorithm 1 depicts a high level pseudo code of the training procedure of FaceXHuBERT described in section Sec. 4. Given an audio waveform A, Subject Label, Emotion Label as inputs and corresponding 4D scan Y as output, the proposed network learns the mapping between the inputs and the output during training epochs. The network optimizes on Huber Loss function and uses Adam optimizer to update the weights and biases during backpropagation.

Algorithm 1 Network Training

Given input audio (A) and output 4D-Scan (Y):
$A = AudioWaveform$ ▷ Raw Audio Waveform
$Y = [T, V]$ -D Matrix ▷ Ground-Truth 4D Data

$S \leftarrow SubjectEmbedding(SubjectOneHot)$
$E \leftarrow EmotionEmbedding(EmotionOneHot)$
$B \leftarrow AudioEncoder(A)$
$X \leftarrow InputRepresentationAdjustment(B)$
$H \leftarrow GRU(X)$
$\tilde{H} \leftarrow H \odot S \odot E$
$\hat{Y} \leftarrow FullyConnected(\tilde{H})$
$\mathcal{L} \leftarrow HuberLoss(Y, \hat{Y})$
$AdamOptimizer()$

A.2 Input Representation Adjustment

This module adjusts the input representation and output representations and do not contain any trainable parameters. This is devised to ensure the one-to-one frame level relationship between input X and output Y such that $T_X = T_Y = T$. This function is generically devised in such a way that it can handle any input-output frequency pair given, $f_o \leq f_i$, where f_i is the frequency of encoded discrete representation of the input audio and f_o is the frequency of the face scan data. If $\frac{f_i}{f_o} = k \in Z^+$ where Z^+ denotes the set of positive integers, then the adjustment is a straightforward reshape function such that input dimension $(T_X, B) = (kT_Y, B)$ becomes a (T_Y, kB) dimensional data. If $k \notin Z^+$, we resample the input representation using linear interpolation so that input dimension (T_X, B) becomes $(\lceil k \rceil T_Y, B)$ before reshaping the embedding into $(T_Y, \lceil k \rceil B)$ dimensional data to ensure $T_X = T_Y = T$. Here $\lceil k \rceil$ denotes the ceiling of the decimal representation, in other words we take the next positive integer. In our implementation, we ensure that the value of $k = 2$ for it to be coherent with the implemented network architecture. In case of other datasets, where $k \neq 2$, the model definition needs to be slightly modified in terms of input dimension of the GRU in the decoder.

A.3 Data Pre-process

Algorithm 2 represents the dataset pre-processing procedure. This step is a prerequisite to train the proposed model effectively. The vertex data in BIWI dataset is not normalized. Without scaling the data to a certain uniform range, the network does not train well. Although this is not the only way to normalize the data, we highly recommend normalizing the dataset so that all three coordinates (i.e. 3D coordinates- X, Y, Z) have the same range of values (e.g. [-0.5,0.5], [-1,1] or [0,1]) before starting training. Appropriate step-by-step guide together with code to prepare and pre-process the dataset are available in the project's GitHub repository provided with the supplementary material to facilitate reproducibility of our work.

Algorithm 2 Data Pre-process

$Neutral = [ArrayOfSubjectsWithNeutralFaces]$
$Templates = [EmptyArray]$
while There is $Subject$ to process **do**
  $S \leftarrow Neutral[Subject]$
  $X \leftarrow S[:, 0]$; $Y \leftarrow S[:, 1]$; $Z \leftarrow S[:, 2]$
  $S[:, 0] \leftarrow (X - mean(X)) \div (max(X) - min(X))$
  $S[:, 1] \leftarrow (Y - mean(Y)) \div (max(Y) - min(Y))$
  $S[:, 2] \leftarrow (Z - mean(Z)) \div (max(Z) - min(Z))$
  $Templates.append(S)$
  while There is $sequence$ to process **do**
    while There is $frame$ to process **do**
      $\tilde{X} \leftarrow \frac{(sequence[frame,:,0] - mean(X))}{(max(X) - min(X))}$
      $\tilde{Y} \leftarrow \frac{(sequence[frame,:,1] - mean(Y))}{(max(Y) - min(Y))}$
      $\tilde{Z} \leftarrow \frac{(sequence[frame,:,2] - mean(Z))}{(max(Z) - min(Z))}$
      $sequence[frame, :, 0] \leftarrow \tilde{X}$
      $sequence[frame, :, 1] \leftarrow \tilde{Y}$
      $sequence[frame, :, 2] \leftarrow \tilde{Z}$
    end while
    $SaveProcessedSequence(sequence)$
  end while
end while

A.4 User studies

The user studies for the presented work were conducted similarly with A/B testing. For stage 1 user study explained in Sec. 5.2, we adopt the exact same strategy as done in [68]. We were able to recruit 52 participants for the survey through Prolific. For each row in Tab. 2, the confidence intervals (CI) for stage 1 user study experiment are (from left to right in the table)- (i) 3.17, 3.28, 3.66, 3.43, (ii) 4.56, 4.46, 5.33, 4.93, (iii) 5.05, 3.96, 5.08, 5.33, (iv) 2.98, 3.50, 4.94, 4.48 at 95% confidence level, which demonstrates that our user study results are reliable.

For perceptual evaluation described in section Sec. 5.3.2 for stage 2 phase, three similar user study experiments were conducted. In all three experiments, the participants were randomly shown 12 pairs (to ensure a good duration of the study) of facial animation videos and asked to choose the one that is realistic and more expressive than the other in accordance with the audio. Fig. 6 is the introduction message that the participants were shown. Fig. 7 shows the user interface of the survey related to the user study experiments. In total, we were able to recruit 147 participants from different demographic backgrounds for the experiments ensuring a good sample representation of the population. Among this 147 participants, 51 participated in Experiment I, 31 in Experiment II and 65 in

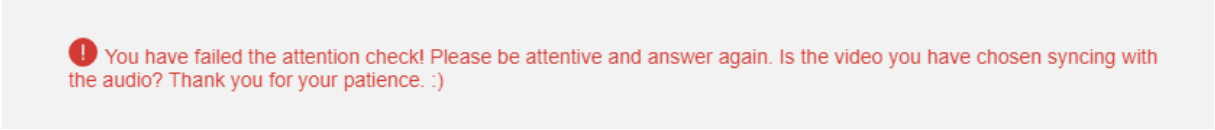

Figure 5: Attention check warning message during user studies.

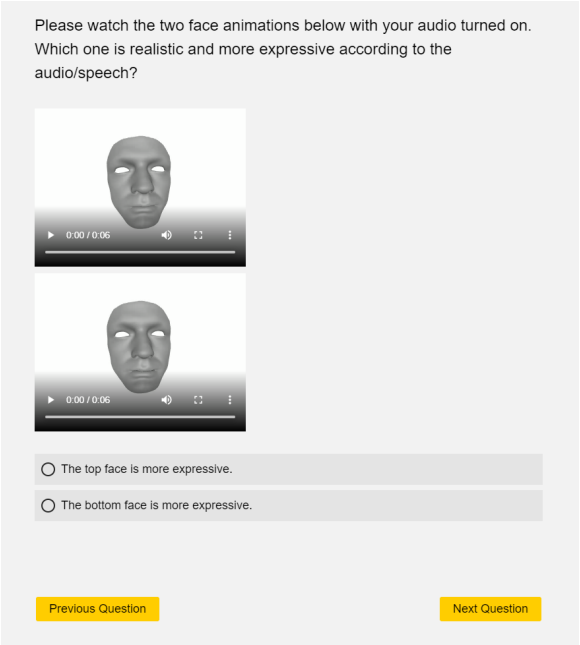

Figure 7: User interface of user study surveys.

Experiment III. The confidence intervals (CI) for our experiments are 5.12, 6.04, 4.35 respectively at 95% confidence level which shows that our user study results are reliable.

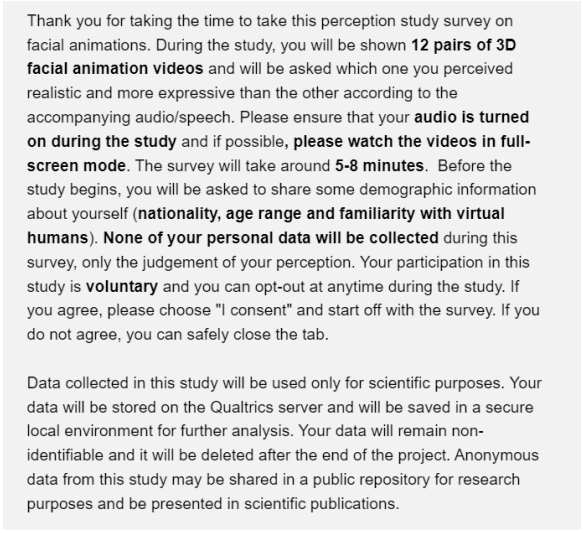

Figure 6: Introduction page of the user study surveys.

Additionally, the participants were shown attention check question items, where one of the videos was facial animation that was generated by a model trained on MFCC (Mel-frequency cepstral coefficients) features instead of FaceXHuBERT Encoder and the other is either ground-truth or generated by our approach. The animations generated by MFCC based model do not produce coherent animations where the lip-sync is incongruous to the accompanied audio. If the participants had chosen the animation video generated by MFCC based model, there were shown the warning message depicted in Fig. 5. Those participants' responses are then manually reviewed for inconsistency in the user study data.