Making Virtual Humans Talk

Daniel Lee [*1,2] **Akshana Dassanaike-Perera** [*1,3]

1. Introduction

With the rise of Large Language Model-powered chatbots and the metaverse, it is becoming increasingly important to simulate humans in the virtual world. As such, an important sub-problem is modeling human speech.

Instead of arbitrarily moving their faces when speaking, characters with faces that match the sentence they are speaking will be recognized as more realistic. The application of machine learning to predict how a character's or robot's face will move given an audio input will enable the creation of characters that more closely match real life. Other potential applications include sign language animations, in which algorithms can generate the correct sign language given a sentence, which could more easily translate media into sign language, greatly improving internet accessibility.

We build upon previous models like FaceFormers [5] and Image Avatar [2] that frame facial animation as a sequence-to-sequence learning problem between audio and facial movements. We attempt to utilize a similar framework and build upon their findings. Our goal is to explore if the introduction of an auxiliary 2D reconstruction task during training can help refine our model's 3D predicted animations.

2. Related Work

Many facial animation approaches attempt to leverage specific features to extract context from audio inputs. Models like JALI [4] implement a viseme-based (speech transcript-based) approach that relies on phonetic alignment computed from audio and speech transcript data to procedurally animate phonemes, which are combined with audio signal features to animate a 3D character. While viseme-based models still predict 3D animations, they require additional processing due to their reliance on speech transcripts. Suwajanakorn et al. [10] propose a mouth-shape-based model that constructs a 3D mouth shape based on captured video and uses input audio to predict how a mouth moves using neural networks. While relatively lightweight with respect to both the model and training data, it fails to predict how the rest of the face outside of the mouth, such as the eyes, may move. Our model aims to address this issue by using the entire 3D face data in training.

In contrast, Karras et al. [7] introduced an emotion-based approach by leveraging both audio and emotions during training and inference that has seen promising results in Nvidia's Omniverse Audio2Face App Beta[1]. Karras et al. propose that by allowing the model to learn latent emotional descriptors, models are able to integrate how a subject feels while speaking into the predicted face movements, allowing for finer predictions that capture varying emotions in facial movements.

Cao et al. [2] introduce a more general approach that attempts to reconstruct a 2D image-based representation of facial movements. Cao et al. utilize face tracking and image-based rendering to generate their representations. In contrast, Fan et al. [5] proposes a transformer-based approach that directly leverages 3D face movements by using self-attention layers to create contextual audio embeddings and cross-modal attention between audio embeddings and 3D facial movements.

3. Dataset and Features

We utilize the VOCASET dataset from VOCA [3] to train and test our model. VOCASET contains 480 sequences of audio and corresponding 3D facial movements captured from 12 different subjects. Each sequence was sampled at 60 FPS and is 3-4 seconds long. Our training set consists of 320 sequences captured across 8 subjects, our validation set consists of 80 sequences across 2 subjects, and our test set consists of 80 sequences across 2 subjects. The subjects for each dataset split are unique to that split; the 8 subjects within our training set do not appear in our test set or validation, and so on.

For pre-processing, we follow the same steps as Fan et al. [5]. We have a pre-trained version of wav2vec 2.0 [1] as a standalone acoustic model to pre-process our audio files from .wav files into vectorized representations. To pre-process 3D facial animations, we aggregate spatial information for all vertices for each timestep and normalize

[*]Equal contribution [1]Stanford University [2]SUNet ID: jaeylee, jaeylee@stanford.edu [3]SUNet ID: akshana, akshana@stanford.edu.

[1]https://www.nvidia.com/en-us/omniverse/apps/audio2face/

them to align with our template facial mesh. However, in contrast to Fan et al., we introduce an additional medium for each example in our dataset by rendering our 3D facial animations into a video sampled at 5 FPS.

4. Methods

4.1. Baseline

As a baseline model, we implement our own version of FaceFormers [5], Cao et al.'s transformer-based approach to facial animation prediction. FaceFormers treats facial animation as a sequence-to-sequence learning problem and leverages both audio and facial movements during training time to refine its animation predictions.

To encode audio features, FaceFormers uses a pre-trained model of wav2vec 2.0 [1] to encode audio information. wav2vec 2.0 uses a combination of multi-headed self-attention layers along with a feed-forward neural network to generate contextual audio embeddings.

Separately, FaceFormers encodes facial motions by adding a periodic positional encoding of periodic cosine and sine terms alternating between facial motion vertices at each timestep to encode temporal order within the facial motion features:

$$\begin{aligned} PPE_{t,2i} &= \sin((t\%p)/10000^{2i/d})) \\ PPE_{t,2i+1} &= \cos((t\%p)/10000^{2i/d})) \end{aligned} \tag{1}$$

Afterward, it uses a biased multi-headed self-attention layer to create contextual facial motion embeddings using temporal bias from ALiBi [9]. The ALiBi bias is added to the result of the query-key matrix multiplication within the self-attention layer. Just as segments of audio have an important context within a speech phrase, so do facial animations. A facial position in one time step depends on its previous position and influences its next position.

To encode both audio- and facial motion-embeddings together, FaceFormers uses biased cross-modal multi-head attention. To align both modalities, they add an alignment bias term to each feature channel $(1 \leq t \leq t, 1 \leq j \leq kT$:

$$B^A(i,j) = \begin{cases} 0 & ki \leq j < k(i+1) \\ -\infty & \textit{otherwise} \end{cases} \tag{2}$$

The cross-modal attention transforms tokens within the contextual audio embeddings A_{kT} into key and value matrices K^A and V^A. The facial motion embeddings F_t are transformed into a query matrix Q^F. The attention is computed as:

$$Att(K^A, V^A, Q^F, B^A) = \textit{softmax}(\frac{Q^F(K^A)^T}{\sqrt{d_k}} + B^A)V^A \tag{3}$$

Finally, the output from the biased cross-modal attention layer is fed through a feed-forward layer to decode joint embeddings into predicted facial animations.

4.2. wav2vec Improvements

Although FaceFormers [5] achieves top-of-the-line performance, one drawback is its large number of parameters and its corresponding slowness during inference time.

To address this, we focus on the model architecture. In the audio encoder, we remove SpecAugment [8] and its related masking within the wav2vec 2.0 forward step. Although SpecAugment has seen promising results within the domain of speech recognition, we remove it to reduce model parameters and to avoid masking blocks of frequency information — features that can be valuable in capturing speaker emotion and tone.

4.3. Fast-Transformers

Furthermore, to improve model speed during inference time, we implement the fast-transformers architecture from Vyas et al. [12] that computes clustered attention instead of the transformers architecture introduced by Vaswani et al. [11] that computes "vanilla" attention.

Instead of solely finding a transformation from an input into a query matrix Q, Vyas et al. proposes partitioning Q into non-overlapping clusters $S \in \{0,1\}^{NxC}$ and computing the centroids for each cluster j and the newly computed Q^C in lieu of Q:

$$\begin{aligned} Q_j^C &= \frac{\sum_{i=1}^N S_{ij} Q_i}{\sum_{i=1}^N S_{ij}} \\ V^C &= \textit{softmax}(\frac{Q^C K^T}{\sqrt{d_k}})V \end{aligned} \tag{4}$$

The final value of i-th query becomes the value of the nearest centroid to it:

$$V = \sum_{j=1}^{C} S_{ij} V_j^C \tag{5}$$

As such, cluster attention allows us to approximate attention while only needing to update attention weights once per cluster, which we suspect will lead to faster predictions during inference.

4.4. Dropout Rate

In order to make sure that the model was not overfitting, we introduced a dropout rate. Dropout is a regularization method that randomly "drops" some number of layer outputs. This has the effect of reducing variance in the performance and leads to less overfitting of the training data. We use a dropout rate of 0.5, which means that the chance a layer

output is going to be dropped is modeled as $Bernoulli(p = 0.5)$. We chose a dropout rate of 0.5 because it achieves maximum regularization, which should help ensure that our model does not overfit.

4.5. 2D Reconstruction Task

FaceFormers architecture is designed to leverage both audio and 3D facial motion features during training, omitting other relevant modalities to facial animation predictions that models like Image Avatar [2] are predicated upon. As such, we introduce an auxiliary 2D reconstruction task during training to help refine our model's 3D facial animation predictions.

Specifically, using input video rendered from the 3D facial motion features, we add a video encoding and decoding architecture inspired by Kabra et al.[6] that computes the cross-modal attention between audio, 3D facial movement, and 2D video features.

To encode video information, we use two 2D convolutional layers to extract features within images across timesteps. Then, we use multi-headed self-attention across timesteps to generate contextual embeddings for each frame. Afterward, we use cross-modal multi-headed attention between frame embeddings and joint sound and motion embeddings from our baseline model to capture feature information across each medium: audio, spatial, and visual. Finally, we use a multilayer perceptron for feature-extraction and a multilayer perceptron for frame-reconstruction across timesteps. In this modification, we use "vanilla attention" from Vaswani et al. [11]. Adding the video reconstruction task adds an additional computationally intensive objective during training, so the marginal speed increases from implementing clustered-attention become less impactful.

5. Experiments/Results/Discussion

5.1. Evaluation

Since our model attempts to predict human-like facial animations, we aim to minimize mean L2 error between predicted facial vertices and ground truth facial vertices averaged across all t timesteps for each example:

$$L_{MSE} = \frac{1}{t}\sum_{i=0}^{i=t}\sum_{j=1}^{j=n}\|gt_{i,j} - \hat{y}_{i,j}\|_2^2 \tag{6}$$

When evaluating our model with the auxiliary 2D reconstruction task, we add an additional term to our loss calculation of the L2 error between predicted video pixels and the original

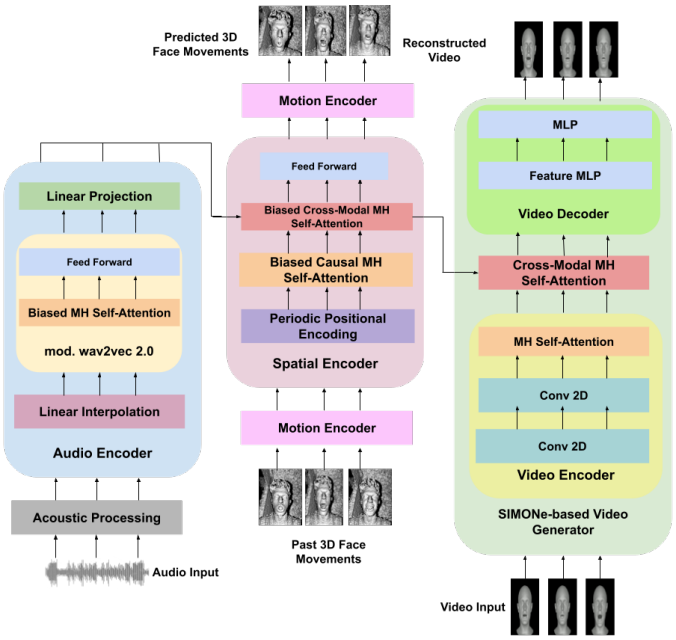

Figure 1. Model Architecture with Auxiliary 2D Reconstruction Task

video pixels, weighted by a tuning parameter λ:

$$L_{MSE} = \frac{1}{t}\sum_{i=0}^{t}\sum_{j=1}^{n}\|gt_{i,j} - \hat{y}_{i,j}\|_2^2 + \frac{\lambda}{t_v}\sum_{i=0}^{t_v}\sum_{j=1}^{n}\|gt_{i,j} - p_{i,j}\|_2^2 \tag{7}$$

5.2. Parameters

Across all models, we used a learning rate of 0.0001, as it yielded the highest accuracy (lowest test error) without compromising on processing time. We trained each variation of our model for 25 epochs, as we empirically found that there was negligible improvement above 25 epochs across all models. For the 2D reconstruction model, we chose a tuning parameter of $\lambda = 10^{-10}$, as that roughly balances the loss contribution of the facial animation prediction and video reconstruction tasks during early epochs.

5.3. Results

Key: "Truth" refers to renders of the ground truth facial motion data, "Base" refers to predictions from FaceFormers. "Wav2Vec" refers to the model with the wav2vec modifications, "Fast Transformers" refers to the model using fast transformers, "Dropout" refers to the model with a dropout rate of 0.5, and "Video Integration" refers to the model with the 2D reconstruction task.

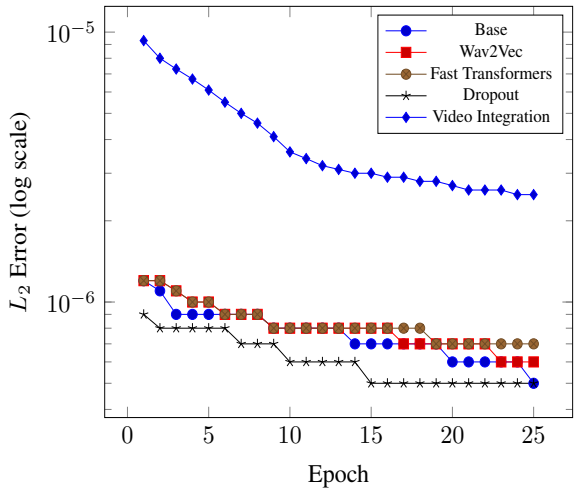

Figure 2. L_2 error per epoch for different models

	Testing time (s)	Test loss
Base	2.256	5.0×10^{-7}
Wav2Vec	2.249	6.0×10^{-7}
Fast Transformers	2.059	7.0×10^{-7}
Dropout Rate	2.251	5.0×10^{-7}
Video Integration	2.252	2.5×10^{-6}

Figure 3. Testing time per sentence for different models and final test loss

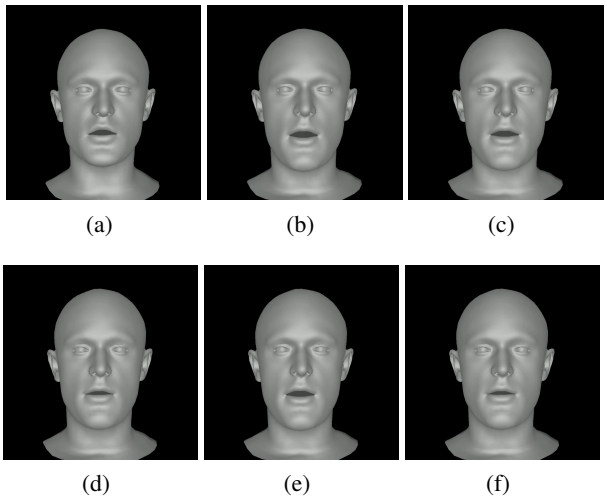

Figure 4. (a) Truth (b) Base (c) Wav2Vec (d) Fast Transformers (e) Dropout (f) Video Integration
Comparison of different models at frame 60 speaking the same sentence

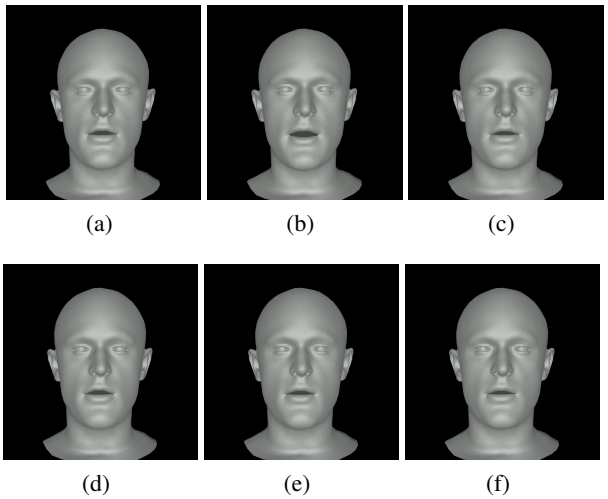

Figure 5. (a) Truth (b) Base (c) Wav2Vec (d) Fast Transformers (e) Dropout (f) Video Integration
Comparison of different models at frame 61 speaking the same sentence

5.4. Discussions

5.4.1. Quantitative

With respect to the accuracy, both the wav2vec modifications and fast transformers implementations performed slightly worse than the baseline. The dropout model was able to match the baseline model's accuracy, but the 2D reconstruction model was significantly worse than the baseline. Although the 2D reconstruction model had a higher loss, its loss continued to decrease over epochs, indicating that the model was still able to learn.

During training, L_2 error's variation across epochs was not significant in the wav2vec and fast transformers modifications. Additionally, the dropout model was able to converge to the same loss roughly 10 epochs faster than the baseline model.

In terms of inference time per test example, the wav2vec modification model, the dropout model, and the video integration model did not significantly speed up relative to the base model. This makes sense as neither modification made significant changes to the speed bottlenecks of the model. However, we note that the testing time for the fast transformers modification decreased by approximately 10% compared to the base model, confirming our suspicion that computing clustered attention would allow us to speed up predictions.

5.4.2. Qualitative

Although accuracy (L2-error) is an important measurement of the quality of our results, speech animations are highly

subjective domains that also require qualitative results regarding their rendered output. To measure the qualitative performance of each model, we rendered the predictions on 8 test-set sentences and compared multiple characteristics between the predicted animations.

The wav2vec modification resulted in a far less "jittery" animation (meaning that from one frame to the next, there were fewer sudden moves of vertices) than both the base model and the ground truth output. However, it caused the upper part of the face to move very slightly compared to the base model. The fast transformers model was visually very similar to the base model; we could not identify any significant differences between both models' predictions. While the dropout model was similar to the base model, it resulted in a slight decrease in the upper face movement. The video integration model was similar to the wav2vec modification in that it resulted in a far less jittery animation than both the base model and the truth output while causing the upper part of the face to move very slightly compared to the base model. However, it differed from the wav2vec modification in that it also made the mouth region move less than the base model and the truth output. While it was possible to match what the model was saying with actual audio, it was more difficult than the base model.

6. Conclusion/Future Work

6.1. Conclusion/Analysis

We conclude that despite having worse accuracy compared to the base model, our modification of the wav2vec encoder that removed SpecAugment and masking resulted in a less jittery mouth movement and less overall movement in the upper face section. In applications where smooth mouth-movements are needed without emotionally-expressive faces, such as characters wearing a mask, our method can be used.

The model implementing fast-transformers resulted in approximately 10% faster predictions with comparable qualitative performance to the baseline model. In responsive, interactive applications that require quick predictions, the fast-transformers method could be used.

Further, we found that using a dropout rate can significantly decrease the time until convergence during training, as the dropout model converges in 40% fewer epochs with respect to the baseline model. This is most likely due to the dropout rate reducing model overfitting, which leads to better performance on the test set that it has never seen before. However, due to the bias-variance tradeoff, we note that the dropout model may result in higher losses on other new examples compared to the base model.

Finally, we conclude that implementing an auxiliary 2D reconstruction task during training led to less jittery movement across the entire face with significantly less movement in the upper part of the face, even less than the model with wav2vec modifications. Therefore, the introduction of 2D video in training led to suppressed facial animations across the entire face. The introduction of the auxiliary task could have resulted in the model giving less focus to the predicted facial animations. To combat this, we hypothesize that further reducing the tuning parameter or implementing the auxiliary task on a pre-trained model could allow the model to better focus on facial animation predictions before using video to refine model embeddings. We also note that the higher test loss of this model may be attributed to the double-descent phenomenon, in which the test loss might increase when more parameters are added until a certain point when it decreases again. While training, we ran into multiple memory issues, causing us to trim the size of the auxiliary task. If more memory was available, this method may be able to yield better performance than the base model.

6.2. Future Work

In the future, we hope to expand our model to result in less jittery motions around the mouth while maintaining upper-face movements and expressions. We also hope to use other types of fast transformers that are compatible with our dataset's features to decrease the testing time. We hope to adjust the hyperparameters more to achieve even lower test loss in a shorter amount of time and explore other factors that can be added to the training layer to help achieve lower loss/faster processing time.

7. Acknowledgements

We are grateful for the mentorship of Demi Guo during the start of this project.

8. Contributions

Daniel worked on getting the base model running, rendering the output video frames, working with faster transformers, and altering the dropout rate. Akshana worked on modifying the wav2vec process, implementing the 2D reconstruction task, and testing the final result. Both members contributed equally to come up with ideas, write the report, and construct a poster.

References

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[2] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Trans. Graph.*, 35(4), jul 2016.

[3] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J. Black. Capture, learning, and synthesis of 3d speaking styles, 2019.

[4] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: An animator-centric viseme model for expressive lip synchronization. *ACM Trans. Graph.*, 35(4), jul 2016.

[5] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Faceformer: Speech-driven 3d facial animation with transformers, 2021.

[6] Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition, 2021.

[7] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Trans. Graph.*, 36(4), jul 2017.

[8] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech 2019*. ISCA, sep 2019.

[9] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2021.

[10] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph.*, 36(4), jul 2017.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[12] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention, 2020.