Speech Driven Tongue Animation

Salvador Medina[1, 2], Denis Tome[2], Carsten Stoll[2], Mark Tiede[3], Kevin Munhall[4]
Alex Hauptmann[1], Iain Matthews[2]
[1]Carnegie Mellon University, [2]Epic Games, [3]Haskins Laboratories, [4]Queens University
{salvadom, alex}@cs.cmu.edu, tiede@haskins.yale.edu, munhallk@queensu.ca
{denis.tome, carsten.stoll, iain.matthews}@epicgames.com

Abstract

Advances in speech driven animation techniques allow the creation of convincing animations for virtual characters solely from audio data. Many existing approaches focus on facial and lip motion and they often do not provide realistic animation of the inner mouth. This paper addresses the problem of speech-driven inner mouth animation.

Obtaining performance capture data of the tongue and jaw from video alone is difficult because the inner mouth is only partially observable during speech. In this work, we introduce a large-scale speech and mocap dataset that focuses on capturing tongue, jaw, and lip motion. This dataset enables research using data-driven techniques to generate realistic inner mouth animation from speech.

We then propose a deep-learning based method for accurate and generalizable speech to tongue and jaw animation, and evaluate several encoder-decoder network architectures and audio feature encoders. We find that recent self-supervised deep learning based audio feature encoders are robust, generalize well to unseen speakers and content, and work best for our task.

To demonstrate the practical application of our approach, we show animations on high-quality parametric 3D face models driven by the landmarks generated from our speech-to-tongue animation method.

1. Introduction

Virtual human characters with strikingly realistic facial animation are possible through advances in facial performance capture and speech-driven animation. However, virtual characters often lack realistic representation and motion of the inner mouth, in particular for the tongue. Tongue animation is often subdued and unnatural, breaking the illusion of realism and contributing to an uncanny valley experience.

Accurately animating the tongue is a challenging task. Typical optical performance capture approaches fail because the inner mouth articulators are only partially observable even when the mouth is open. Manually animating the tongue and jaw requires a skilled artist familiar with speech articulation, and is time consuming due to the rapid and complex motions required for speech production. Animating manually is clearly not an option for any real-time or interactive application. In all cases, a low-latency and real-time automatic animation solution is preferred.

In practice, inner mouth animations for movies and video games often make use of rule-based or procedural animation approaches. The result is to broadly match the utterance of specific sounds, such as dental consonants, open mouth vowels, or articulation that moves the tongue from the mouth floor to the palate and vice-versa. In many cases, the inner mouth region is simply intentionally poorly lit while speech articulation is approximated by placing the tongue in a neutral position.

In this paper we consider the problem of automatic speech-driven tongue and jaw animation using a data-driven sequence to sequence approach. Sequence to sequence models have shown impressive results on a diverse set of regression and forecasting problems, and they have been applied in a wide variety of research areas. In our task, the input is streaming speech audio waveform, and the output is the temporally corresponding set of 3D landmark locations of motion captured speech articulators. We recorded and release a new dataset, comprising over 2.5 hours of labelled speech, that includes *ground truth* landmarks tracked using an Electromagnetic Articulograph (EMA) [5], a specialized speech motion capture system. The dataset is publicly available for further research[1].

We leverage recent work in deep-learning based speech audio feature representations and compare ML-based approaches with traditional features based on phonetic or frequency based representations. Our experiments show that deep learning speech representations greatly improve generalization and resiliency to noise over traditional features.

[1] https://salmedina.github.io/tongue-anim

The landmark locations predicted by our model may be used to drive *any* facial animation rig. We demonstrate rig solving using a professional FACS-inspired [11] MetaHuman facial rig based on the capture subject. General retargeting is shown on further MetaHuman characters that can be customized using the *MetaHuman Creator Tool* [12]. To animate a rig, we perform an optimization that minimizes the distance between the predicted landmarks and its corresponding locations on the facial mesh and solve for the parametric representation of the animation. This approach means the results can be readily used in game engines or any digital content creation (DCC) software.

In summary, our main contributions in this work are as follows: (1) We introduce a framework for speech-driven tongue animation that trains a high-quality speech-to-animation model for the tongue and jaw. (2) We thoroughly analyze and compare a diverse set of audio representations by introducing self-supervised deep learning audio features for the task of speech-to-animation. (3) We present a pipeline approach that drives a high-quality parametric 3D face model from a few 3D landmark constraints through a fast optimization method. (4) We release code and a novel large-scale speech-to-tongue mocap dataset to train tongue and jaw speech animation models.

2. Related Work

Many approaches in the literature focus on lips and facial deformations. Animating the mouth interior has often been neglected as this particular task is challenging due to the lack of data and generalized subject-independent models. Recent vision based generative animation approaches have shown compelling results using generative adversarial networks (GANs) [41, 51, 56], image-to-image translations [57], or neural rendering [20, 21, 46]. However, none of these methods output 3D animation directly but implicitly generate 2D image frames of speaking faces. Our application is to learn to synthesize 3D speech and tongue motion that can be used in existing 3D computer animation pipelines.

Tongue modeling dates back to [35] which modeled a two-dimensional surface projection of the tongue in the sagittal plane that ignores the intrinsic structure of the tongue and only accounts for geometric surface deformations. The bio-mechanical model by [52] models soft tissue deformations and non-linear geometric effects. The 2D physiological model proposed in [15] unifies the tongue, jaw, and laryngeal structures using a 2D finite-element simulation from MRI data of a single subject. The parameterized model in [22] describes the tongue's surface through B-splines by forming a grid of bi-cubic patches over 60 control points found on the top and under the tongue to produce realistic tongue shapes as described in [45]. The phone shapes were matched by manually setting the parameters of the tongue model and achieving a tongue animation by blending the shapes between phonemes.

Since then, representing the audio through symbolic sound units such as phonemes onto shape parameters is a general approach for speech animation [8, 24, 29, 30, 40, 48]. For instance, JALI [10] is a procedural method to generate viseme units of mouth shapes from phonemes. The viseme sequences are blended into co-articulated motions to animate a FACS-based face rig model [33]. The generated animation from these approaches can be transferred to different characters if they share a common rigging system.

Different approaches have also explored tongue animation from different input modalities. For instance, in [44] a tongue 3D model is animated directly from electromagnetic articulography (EMA) data. This approach does not include any audio processing as the animation is synchronized with the recorded audio from the EMA capture session through their open-sourced framework [43].

Tongue animation has also been achieved from ultrasound images. In [13], the authors explore animating the tongue from low-resolution imagery represented by Eigen-Tongue [18] features, and mapped into control parameters through a Gaussian mixture model. Later in [6] more realistic animations are obtained from ultrasound images using a snake contour extraction algorithm and driving a finite element model of the tongue, achieving animations at 21 FPS.

A multimodal end-to-end hidden Markov model (HMM) proposed by [42] is capable of synthesizing audio and generating tongue motion. Unlike previous work, their method replaces the midsagittal EMA data with tongue model parameters as the target articulatory representation. A follow-up multimodal approach [54] replaces the HMM with a bottleneck long-term recurrent convolutional network (BTR-CNN). The network is trained on text and audio to predict EMA positions as a proxy to the tongue movement while considering embedded articulatory features while training the model.

Similar to our approach, other work also considers only speech audio as input. In [27] the input speech is represented as a sequence of phonemes which is mapped into EMA sensor positions through a HMM. The predicted articulatory movements control the deformations of a 3D tongue model. Similarly, in [28] a stacked restricted Boltzman machine predicts EMA sensor positions from audio represented as mel frequency cepstral coefficients (MFCC). The predicted positions are fit to a volume-preserving model through a finite element method to generate animations. Zhu et. al. [59] also use MFCC as input features to solve articulatory inversion on EMA positions using a 2-layered bidirectional LSTM preceded by a linear projection of the audio features into the RNN. This model achieves state-of-the-art results on the MNGU0 dataset [37]. However, in [4] they demonstrate that gated recurrent unit (GRU) networks have a slight performance improvement over LSTM architectures since the GRU layers have fewer parameters making them less prone to over-fitting.

In this work, we move beyond linguistically motivated features such as phonemes or MFCCs by exploring robust and continuous audio feature representations that recent deep learning models provide. These features enable generalization across speakers even on out-of-domain utterances. We also investigate deep-learning architectures [19, 47, 58] to map the audio feature representations into EMA sensor positions from an articulatory inversion perspective.

3. Tongue Mocap Data

Tongue motion capture is a common practice in speech pathology. A popular method for capturing tongue motion is electromagnetic articulography (EMA) [17]. Traditionally, EMA is captured in a midsagittal fashion as shown in [49] and [53].

We collected a new tongue motion capture dataset for the speech animation task with additional parasagittal sensors. The linguistic analysis in [32] demonstrates the importance of adding lateral sensors to describe richer tongue motion dynamics. The data was captured using a Carstens AG501 EMA device [5] following the ethical and health guidelines approved by the Institutional Review Board (IRB). A configuration of ten sensors is used to acquire the motion of the tongue, jaw, and the lips. The sensors are attached to the surface using medical grade cyanoacrylate glue. While not painful or permanent, it is an invasive process. The actor sits below nine RF transmitters creating an electromagnetic field that energises coils in the sensors whose currents are processed to recover five degrees of freedom for each sensor: three for position (x, y, z) and two for rotation (azimuth and elevation). EMA sensors were sampled at 250 Hz and mono audio was synchronously recorded at a sampling rate of 48 kHz.

Five sensors were positioned on the tongue: the midsagittal dorsum, blade, and tip, as well as both left and right parasagittal sensors on the blade. The tongue tip sensor was positioned 5 mm behind the apex to avoid any damage to the actor's teeth. Two sensors are located on the lower jaw: one on the gingival margin at the medial incisors, and one in a parasagittal location between the canine and first premolar. Two more are placed in a midsagittal manner on the upper and lower lips at the vermilion border. The final sensor was placed at the right lip corner apex. To enable stabilization of the speech articulator landmarks with respect to rigid head position three additional sensors are positioned: one on the upper medial incisor and one each on the left and right mastoid process. The stabilization sensors capture rigid skull position and rotation over six degrees of freedom. A visualization of the sensor placements is shown in Figure 1 with the naming convention summarized in Table 1.

The data was recorded in a single 8-hour session by the actor reading a total of 2160 sentences. A subset of 720 sentences from the Harvard set [38] was repeated at both a regular and a fast pace. The remaining 1440 sentences come from the TIMIT dataset [14]. In our experiments, we used a subset of 1902 cropped samples which exclude reading errors and non-verbal gestures, giving a total of 2.55 hours of articulator mocap sequences paired with audio samples.

We also captured an HD reference video from two cameras synchronized to the EMA capture, enabling future visual-based analysis since the actor was prepared with visible markers.

Table 1. EMA sensors position on the tongue, lips, and jaw. The placement is either Midsagittal (M) or Parasagittal (P).

EMA Sensor	Position	Placement
TD	Tongue Dorsum	M
TB	Tongue Blade	M
BR	Tongue Blade Right	P
BL	Tongue Blade Left	P
TT	Tongue Tip	M
UL	Upper Lip	M
LC	Center Lip, Right Corner	P
LL	Lower Lip	M
LI	Jaw, Medial Incisors	M
LJ	Jaw, Canine & First Premolar	P

4. Methodology

Our proposed learning-based prediction pipeline consists of an encoder-decoder model followed by an optional rig-solving animation step. First the input audio is encoded into a compressed latent feature representation by an *audio encoder*. Then a sequence of sparse landmark positions are predicted by an *articulation decoder*. Finally these sparse points become the constraints of a *rig optimizer* module that identifies the optimal animation parameters to match corresponding mesh locations of the tongue, lips, and jaw on a rigged 3D model.

Formally, our dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ is defined as the set of pairs where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}, \mathbf{x}_i \in \mathcal{X}$ denotes the set of audio input samples and $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}, \mathbf{y}_i \in \mathcal{Y}$ is the corresponding sequence of EMA landmark positions. Each input audio $\mathbf{x}_i \in \mathbb{R}^T$ represents a one-dimension waveform consisting of T_i samples according to the duration of the audio and the sampling rate under which it was captured, while $\mathbf{y}_i \in \mathbb{R}^{S_i \times L \times 3}$ contains a series of S_i continuous frames of $L = 10$ 3D landmark positions.

As a first approach, we focus on finding the best model $E : \mathcal{X} \rightarrow \mathcal{Z}$ that encodes the input audio signal into a latent audio feature space $\mathcal{Z} \in \mathbb{R}^a$, where a is the dimensionality

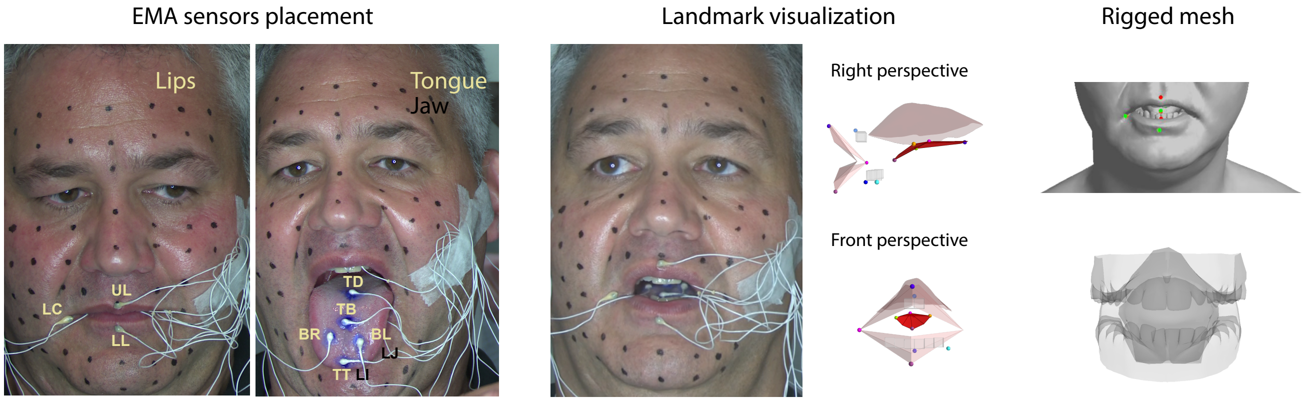


Figure 1. Sensor configuration for capturing the tongue, lips, and jaw mocap dataset. Notice how in the landmark visualization we reflect the position of LC and LJ using symmetry. The lips, teeth and palate are presented for reference of the placement of the tongue.

of the audio feature representation. The audio embeddings $\mathbf{z}_i \in \mathbb{R}^{S_i \times a}$ are later decoded by the articulation decoder $D : \mathcal{Z} \rightarrow \mathcal{Y}$ to predict a sequence of landmark positions $\mathbf{y}_i \in \mathbb{R}^{S_i \times L \times 3}$, expressed in the EMA stabilized pose space. Finally, the predicted landmark positions $\hat{\mathbf{y}}_i = D(E(\mathbf{x}_i))$ are mapped into the face mesh pose space $\mathcal{M}$ by applying a similarity transformation $\mathcal{A} : \mathcal{Y} \rightarrow \mathcal{M}$ resulting in the sequence of mesh constraints $m_i \in \mathbb{R}^{S_i \times L \times 3}$. A summary of the combinations of different encoders and decoders considered is shown in Figure 2.

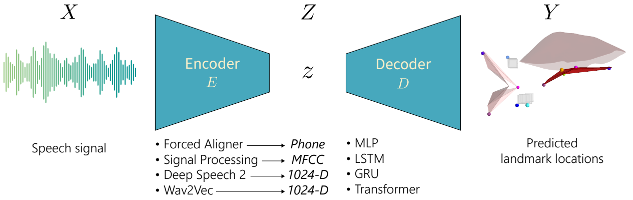


Figure 2. Visualization of the configuration space of encoders E, latent representations Z, and decoders D explored in this work. The latent space Z is defined by the choice of encoder. The architecture of the decoder can be chosen independently of the encoder.

4.1. Audio Encoding

For the encoding stage, we explore five different audio feature representations, ranging from traditional methods to recent neural network-based audio features.

Phoneme: We perform a phonetic segmentation of the speech signal as used in [10, 25, 58]. We specifically use allophonic representations that include the 39 phone representation from ARPAbet with the lexical stress variants of the actor's diction that were found in the dataset, yielding 79 allophonic representations.

MFCC: We also employ the common mel-frequency cepstral coefficients [36], which are widely used for speech processing tasks and convert the audio frequencies into a perceptually based logarithmic mel scale which is useful for characterizing human speech.

DeepSpeech2: We extract intermediate representations from the neural ASR model DeepSpeech2 (DS2) [3]. Specifically, we selected the output embedding from the Bi-LSTM layers as the latent audio feature representation to obtain embeddings with a higher generalization, and avoid a bias towards the English character distribution learned at the pre-FC layer.

Wav2Vec-Z and -C: Wav2Vec [39] takes the raw audio waveform as input which is directly processed by two causal convolutional networks (CCNN). It was trained for the task of learning a general representation of speech audio for any downstream application, rather than a specific task such as ASR. The input audio is fed into a CCNN that predicts a latent representation of the audio in the *z-features* (W2V-Z). A sequence of z-features in a larger window are then fed to the second CCNN to compute the contextual *c-features* (W2V-C). We experiment using both features.

4.2. Articulation Decoding

The decoding stage maps input speech features into 3D speech articulator landmark positions. Different neural network architectures were evaluated for this task, from simple methods like a Multilayer Perceptron (MLP) [47] to models with higher complexity like Recurrent Neural Networks (RNNs) [16] and a Transformer architecture [50].

MLP: We implemented a simple sliding overlapping input and output window MLP as proposed by [47] with a slight modification. We enforce causal prediction by outputing the most recent value rather than predicting the middle frame. This avoids looking at information in the future and reduces the prediction latency of the model.

RNNs: The variations of LSTM and GRU tested in our experiments are: *a)* unidirectional, *b)* bidirectional, and *c)* their corresponding multi-staged variants with one, two and five layers.

Transformer: Our Transformer model [50] follows the work from [9] and [55] by neglecting the decoder layers and using only stacked encoder layers with Multi-head Self-Attention (MSA). In our approach, the Transformer is trained to predict the 3D landmark positions by projecting the last Transformer encoder layer into the output space through a final linear layer. Finally, we calculate the mean of the output of overlapping sliding predictions similar to our MLP configuration.

5. Experiments

We perform an extensive evaluation of the audio feature encoding and articulation decoding architectures by training all combinations of encoding and decoding architectures.

Audio Encoding: In all of our experiments we downsampled the audio to 16 kHz since DeepSpeech2 and Wav2Vec are trained for that specific sampling rate. DeepSpeech2 outputs an audio feature representation for every 20 ms in the audio. Therefore, we consider 20 ms as the common frame duration for the encoding of the input signal for all audio encoding methods. This frame duration is also preserved in the articulation decoder networks and results in animation predictions generated at 50 frames per second.

We use the Montreal Forced Aligner [31] to obtain phone labels by aligning the transcripts from the recording session with the recorded audio. The resulting 72 class allophone labels are sampled every 20 ms followed by a one-hot encoding to represent the feature.

Following [2] we compute MFCC features by separating the mel frequency spectrum into 27 bins with a Fast Fourier Transform on a 2080 Hz window, resulting in a sequence of 27-D feature vectors.

From DeepSpeech2, we extracted the 1024-D output from the 5-layered Bidirectional-LSTM (Bi-LSTM) and discard the final English character classification layer.

In contrast, Wav2Vec's z- and c-features represent 10 ms of audio via a 512-D feature vector. To match the common frame duration we concatenated two sequential feature vectors to represent 20 ms of audio in a 1024-D feature.

Articulation Decoding: All network architectures in this work were implemented using PyTorch [34]. The models were trained and tested by splitting the dataset of 1900 utterances into two groups: train and test, in an 80/20 ratio.

We fine-tune the hyper-parameters of the models using the Hyperband algorithm [26] with Optuna [1]. In our experiments, we search for the optimal hidden layer size, learning rate, dropout rate, and number of hidden layers. The search space for the different parameters is the following: hidden layer size $[128 \ldots 2048]$, learning rate $[10^{-10} \ldots 10^{-1}]$, and dropout rate $[0.01 \ldots 0.99]$. For the MLP network, we search the number of hidden layers in the range $[1 \ldots 4]$.

We utilize the mean squared error (MSE) over the predicted 3D landmark positions as the loss function for training. Model weights were optimized using the Adam optimizer [23] using $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The initial learning rate was 10^{-5}, with a dropout rate equal to 0.25, and a batch size of 32. These hyper-parameters were also chosen using an Optuna search.

The MLP consists of a fully-connected layer with a ReLU activation function, followed by a second fully connected layer. Each hidden layer has 512 units.

For the RNN models both LSTM and GRU architectures were evaluated in both unidirectional (forward in time) and bidirectional configurations. In all cases we tested models with 1, 2, and 5 layers, and each of the RNN models output a prediction for each 20 ms input feature.

We explored different configurations of Transformer depth and width, and found the best setup to be 4 encoder layers with 8 heads. The Transformer model is trained using Adam at an initial learning rate of 5×10^{-8}. A warm-up procedure using an L_1 loss was found to improve stability in the initial training stage. We then switched to an L_2 loss conditioned by an empirically determined threshold $\delta = 3$ to reach convergence.

The RNN and MLP models were trained for 40 epochs, while the Transformer models were trained for up to 1000 epochs with an early stop criteria of 100 epochs.

All training samples were formed as windows of 15 audio feature representations in length (300 ms duration) sampled randomly from the training data. Audio input features were aligned with their corresponding EMA output landmark locations which were nearest-neighbor downsampled from 250 Hz to 50 Hz.

6. Evaluation

In this section, we evaluate all the different configurations of audio feature encoders and articulation decoders. We define the sample error $e^{(i)}_{sample}$ as the L2-Norm of the estimated landmark positions with respect to the ground truth over the full sequence S_i, as shown in Eq. 1.

$$e^{(i)}_{sample} = \frac{1}{S_i} \sum_{s=1}^{S_i} \frac{1}{L} \sum_{l=1}^{L} \| \hat{y}^{(i)}_{s,l} - y^{(i)}_{s,l} \|_2, \forall i. \qquad (1)$$

Where $y^{(i)}_{s,l}$ denotes the position of the l^{th} landmark at time s for the i^{th} audio sample with duration of S_i audio frames, and $\hat{y}^{(i)}_{s,l}$ is the l^{th} landmark's position predicted by the articulation decoder at time s.

Table 2. Model architecture evaluation using different audio feature representations: Phonemes (Phone), MFCC, DeepSpeech2 (DS2), Wav2Vec c- (W2V-C) and z- (W2V-Z) features. Models were trained with 300 ms input windows of audio. The error is the temporal mean L2-norm in mm calculated through the test split. The number of parameters reported is the amount of trainable parameters per architecture design. The inference time is the mean time over the test split measured as ms per second of audio input.

Decoder \ Feature	Phone	MFCC	DS2	W2V-C	W2V-Z	Num. Parameters	Inference [ms]	Latency [ms]
MLP 15:5	2.445	2.075	2.393	1.959	1.937	6.62×10^7	0.232	300
LSTM-1L	4.207	2.344	2.269	2.047	2.140	3.17×10^6	1.150	20
LSTM-2L	4.209	2.178	4.206	1.990	4.212	5.27×10^6	2.238	20
LSTM-5L	2.656	2.037	2.264	1.999	1.960	1.16×10^7	5.432	20
Bi-LSTM-1L	3.664	2.346	2.375	2.373	3.481	6.33×10^6	2.229	300
Bi-LSTM-2L	4.577	2.109	2.844	2.188	3.874	1.26×10^7	4.512	300
Bi-LSTM-5L	4.365	**1.912**	2.218	1.927	2.929	3.15×10^7	11.000	300
GRU-1L	4.150	2.290	2.250	1.949	2.071	2.38×10^6	1.144	20
GRU-2L	2.623	2.117	2.179	1.897	1.980	3.95×10^6	2.193	20
GRU-5L	2.661	2.006	2.184	1.916	1.954	8.68×10^6	5.339	20
Bi-GRU-1L	4.405	2.368	2.529	2.055	2.613	4.76×10^6	2.290	300
Bi-GRU-2L	3.143	1.953	2.947	1.932	2.513	9.48×10^6	4.439	300
Bi-GRU-5L	**2.341**	1.973	**2.058**	**1.757**	**1.784**	2.37×10^7	10.955	300
Transformer	2.368	2.283	2.168	1.935	1.942	5.045×10^7	3.515	300

The overall performance of each model is measured by the mean sample error over the entire test set. The results from these experiments are shown in Table 2.

6.1. Articulation Decoder

Architecture: Table 2 summarizes the performance evaluation of the different articulation decoders (rows) when different audio feature representations (columns) are used.

The values reported in the table represent the mean sample error, in millimeters, evaluated over the test set. Analyzing the results, we see an improvement in the performance of the MLP architecture by widening the context of the input window as well as the output window. This architecture version is comparable to a single-layer GRU and LSTM network. However, the number of network parameters required for the MLP is greater when compared to RNN-based counterparts. The GRU architecture shows a slight improvement over the LSTM architecture, as seen in [4], due in part to the smaller amount of parameters in each layer making it less susceptible to over-fitting.

Based on these results, we can appreciate how all the articulation decoders we presented are capable of learning how to predict the pose to a reasonably low error. Notably, the LSTM and GRU models' performance improves as we increase their complexity by increasing the number of layers. Furthermore, our results also show that bidirectional GRU and LSTM models learning capability improves as they are able to look ahead in the sequence.

Audio Encoding: Deep-learning based audio features and MFCCs perform better than phone-based features for all our architecture choices. The MFCC audio features show better quantitative evaluation results compared to DS2. However, DS2 and W2V features show far better qualitative performance when *generalizing* to input speech from out of domain speakers. This is demonstrated in the supplementary video which shows side-by-side predicted landmark positions for all feature types.

Both Wav2Vec feature variants show similar behavior, although the layered RNN architectures take more advantage of the c-features. Furthermore, there is a substantial improvement when moving from a 2-layer to a 5-layer version of the architecture in both the unidirectional and bidirectional versions of the RNNs. The best architecture from a test-set perspective consists of encoding the audio with Wav2Vec c-features and estimating the landmark positions using a bidirectional 5-layered GRU.

The same set of experiments were replicated with a training input window of 1000 ms. The results are consistent with the results described in Table 2. The overall performance slightly improved across all the models at the cost of a longer inference time, latency, and number of parameters. Further details are included in the supplementary material.

6.2. Qualitative Evaluation

To visually verify the results described in Table 2, we invite the reader to watch the supplementary video for a visual comparison of the animations from the MLP 15:5, 5-layer Bi-LSTM, 5-layer Bi-GRU, and Transformer models. All the landmark visualizations are visible in a sin-

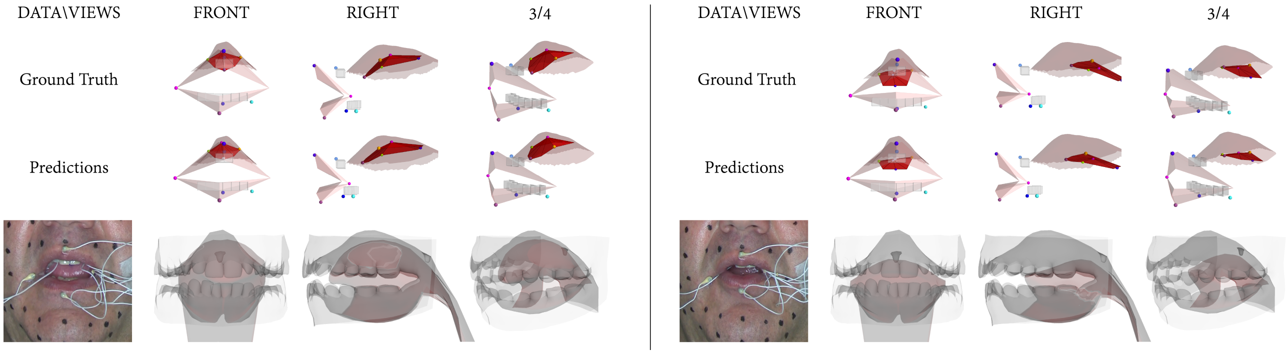


Figure 3. Visualization of two frames from test samples. On the first row we see the ground truth landmark locations. Second row displays the predicted landmark locations. The third row shows the corresponding frame from the video reference and solved animation frame.

gle frame from a lateral view. We found that all models have convincing performance while predicting *in-domain* audio from our dataset for both training and test samples. However, phone-based features did not perform as well as the other models with *out-of-domain* data such as audio from other actors while they were speaking or even singing, as they require an ASR step and forced-alignment which is dependent on the language and prosody. In general, the DeepSpeech2 and Wav2Vec-based models had a similar performance when the decoder was a multi-staged RNN or a Transformer model. We observe the architecture with Wav2Vec z-feature encoding and an MLP 15:5 decoder shows compelling tongue motion results, but exhibits temporal jitter for the jaw prediction which is not present in the LSTM and GRU models.

We visualized the errors described in Table 2 by tracing predicted landmark motions over time against the ground truth. The results can be found in the supplementary material. We found that the animated lips open and close correctly, which is remarkable considering that only three landmarks drive the mesh. No major anomalies were observed in the motion of the jaw and tongue. We also compared the ground truth 3D landmark visualization against predicted landmark positions and ground-truth video. Frames from the generated animation are shown in Figure 3.

Finally, we conducted a user-study to evaulate perceptual performance. Animations generated from our predicted tongue, lips, and jaw positions were preferred over mismatched or null motion sequences of the tongue. Further details on this study can be found in the supplementary material.

7. Parametric Face Model Optimization

To demonstrate retargeting to a final animation output rig we use a high-quality artist designed FACS-based [7,11] 3D face model. The model and animation rig conforms to the MetaHuman standard [12] and closely resembles the actor. The output is represented as a triangle mesh $M = (V, F)$ defined by a set of vertices V that build the mesh faces F. The face model is controlled through a *P*-dimensional control parameter vector θ that deforms the mesh in a differentiable manner for any given frame in the animation. In our model $P = 173$ for the whole face, of which 9 parameters control the tongue and 12 move the jaw. We define $M(\theta_t)$ as the mesh posed by these control parameters for frame t. To estimate the pose of the mesh based on the predicted landmarks of the tongue, jaw, and lips we first manually predefine a correspondence between the transformed predicted landmark positions on the mesh coordinate system $\mathbf{v}_{\hat{\mathbf{y}}}^{(i)}$ to a set of points on mesh M as $C_t = \{f_l, b_l\}_{l=1}^{L}$, where $f_l \in F$ is a triangle index and b_l a barycentric coordinate defining a point on the triangle for the l^{th} landmark. In our case, $L = 10$ since we have five sensors on the tongue, two on the jaw, and three on the lips. These locations are shown in Figure 4.

We perform an initial alignment of the data from the EMA sensors to the mesh in a neutral pose by calculating the best similarity transform A that maps the points $\mathbf{v}_{\hat{\mathbf{y}}}^{(N)}$ into correspondences $C(\theta_N)$ for the neutral pose N described by parameters θ_N. While the geometry of our 3D face model is based on a 3D scan of the actor, the face geometry is not a perfect reconstruction. The teeth and tongue are adapted by an artist from a generic model and do not precisely align. To account for these small differences, we calculate relative offsets $\delta_l = A(\mathbf{v}_{\hat{\mathbf{y}}}^{(l)} - \mathbf{v}_N^{(l)})$ between each landmark and the surface of the neutral 3D model.

The energy between the predicted landmarks at frame t and its corresponding point on the mesh is defined as:

$$e_{pose}(\theta_t) = \sum_{l=1}^{L} \|C(\theta_t)_l - (C(\theta_N)_l + \delta_l)\|^2. \quad (2)$$

Our input data is sparse and asymmetric. There are three markers for the lips that capture the right side of the face, and there are two markers for the jaw that cover the left side. For this reason, we enforce symmetry on both sides of the face on our parameter vector. In addition, we add an L1 regularization to our solver to ensure sparse activation for the model parameters:

$$e_{prior}(\theta_t) = \sum_{l=1}^{L} |\theta_l^{(t)}|. \quad (3)$$

Resulting in the following combined energy function:

$$e(\boldsymbol{\theta}) = e_{pose}(\boldsymbol{\theta}) + \alpha \; e_{prior}(\boldsymbol{\theta}). \quad (4)$$

We minimize $e(\boldsymbol{\theta})$ using an L-BFGS optimizer and initialize the parameters θ_t for frame t using the parameters from the previous frame θ_{t-1} for all T frames in the animation. The prior weight is $\alpha = 0.01$ for all results shown in this paper. Finally, no additional temporal smoothness priors were included in the optimization.

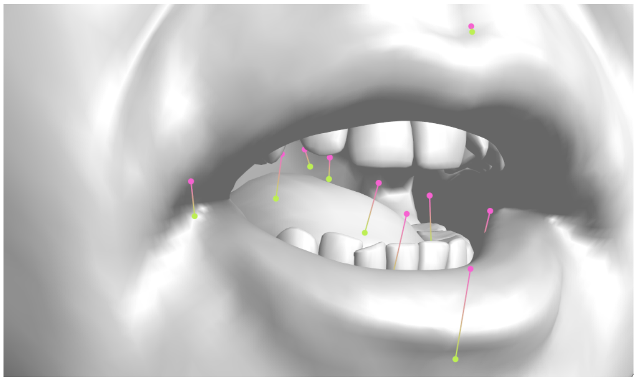


Figure 4. FACS-based face model landmark correspondence visualization before optimization. Green: mesh surface landmarks $C(\theta_t)_l$; Pink: target constraints $C(N)_l + \delta_l$ generated by the articulation decoder.

8. Animation Results

The animation results shown in this paper and the supplementary video were generated using the pipeline proposed in this paper. In the video we include animations created using *out-of-domain* voices that were not heard during training time, uttering *out-of-domain* sentences that are very different to the training corpus. We also present prediction results of the capture subject uttering sentences from the test data to allow the reader to make a side-by-side subjective comparison versus known ground truth EMA sensor landmark positions and so verify the correctness of the predicted motion. In the video we demonstrate our approach is speaker-independent, generalizing across content, speaking style, gender, and language.

9. Summary

Our large scale 3D articulatory dataset enables the training of deep learning models for obtaining realistic inner mouth animations. Our results demonstrate that recent deep learning based audio feature representations outperform traditional speech feature encoding methods for speech-to-animation, regardless of the articulation decoding architecture. To the best of our knowledge this paper is the first to demonstrate these modern features are preferred to drive animation. DL audio features also enable robust generalization to both new speakers and new speech content.

Our best results combine Wav2Vec-C features with a bidirectional 5-layered GRU. We demonstrate practical application by showing convincing speech animation on a high-quality parametric 3D mouth rig driven by a few landmarks generated from our articulation decoder model.

Our approach enables greater realism and animation quality for both audio-driven animation, and performance capture based pipelines.

9.1. Limitations and Future Work

Recording EMA mocap data is invasive and requires expert assistance. Our dataset is limited in speech variability and expressiveness which could be solved by diversifying the data with more than one speaker.

Solving for the parametric 3D face model currently requires manually specifying initial landmark to mesh correspondences (one-time per model). Generating a more accurate inner mouth animation model and automating the alignment will simplify capturing a wider variety of performers.

While we can animate the lips and mouth using the landmarks captured with our dataset, the expressiveness of these regions is limited by the sparse location of the landmarks. Future work may use additional correspondences extracted from the simultaneous video capture to increase the fidelity of the reconstructions and further constrain the 3D model during the rig optimization step.

10. Acknowledgements

We greatly appreciate the motivation, contributions, and assistance of our colleagues in producing the work shown in this paper: Eric Vatikiotis-Bateson, Rohan Bali, Weirong Chen, David Corral, Gareth Edwards, Pablo Garrido, Ginés Hidalgo, Jaekoo Kang, Boram Kim, Kim Libreri, Pascal von Lieshout, Relja Lubobratović, Vladimir Mastilovic, Philip Rubin, Beata Sobkow, Nenad Šunjka, Nick Whiting, Thibaut Weise, and Stephan Veen.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. 5

[2] Simon Alexanderson, Gustav Henter, Taras Kucherenko, and Jonas Beskow. Style controllable speech-driven gesture synthesis using normalising flows. *Computer Graphics Forum*, 39(2):487–496, 05 2020. 5

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony Han, Lappi Vaino Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 173–182, New York, NY, USA, 2016. JMLR.org. 4

[4] Théo Biasutto-Lervat and Slim Ouni. Phoneme-to-articulatory mapping using bidirectional gated rnn. In *INTERSPEECH*, 2018. 2, 6

[5] Carstens Medizinelektronik GmbH. 3D electromagnetic articulograph. `https://www.articulograph.de`. 1, 3

[6] Shicheng Chen, Yifeng Zheng, Chengrui Wu, Guorui Sheng, Pierre Roussel, and Bruce Denby. Direct, near real time animation of a 3D tongue model using non-invasive ultrasound images. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4994–4998, Calgary, AB, Canada, 2018. IEEE, IEEE. 2

[7] Jeffrey F. Cohn, Zara Ambadar, and Paul Ekman. Observer-based measurement of facial expression with the facial action coding system. *The handbook of emotion elicitation and assessment*, 1(3):203–221, 2007. 7

[8] Zhigang Deng and Ulrich Neumann. efase: Expressive facial animation synthesis and editing with phoneme-isomap controls. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 251–260, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929:1–12, 2021. 5

[10] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 2, 4

[11] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial Action Coding System: The Manual*. Paul Ekman Group, San Francisco, CA, USA, 2002. 2, 7

[12] Epic Games. MetaHuman Creator. `https://www.unrealengine.com/en-US/metahuman-creator`. 2, 7

[13] Diandra Fabre, Thomas Hueber, and Pierre Badin. Automatic animation of an articulatory tongue model from ultrasound images using gaussian mixture regression. In *Fifteenth Annual Conference of the International Speech Communication Association*, pages 2293–2297, Singapore, 2014. ISCA. 2

[14] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, and David S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *STIN*, 93:27403, 1993. 3

[15] H. Hirai. A physiological model of speech organs incorporating tongue-larynx interaction. *J Acoust. Soc. Jpn.(J)*, 52:918, 1995. 2

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[17] Philip Hoole and Andreas Zierdt. Five-dimensional articulography. In *Speech Motor Control: New developments in basic and applied research*, pages 331–349. Oxford Scholarship Online, Toronto, Canada, 2010. 3

[18] Thomas Hueber, Guido Aversano, Gérard Chollet, Bruce Denby, Gérard Dreyfus, Yacine Oussar, Pierre Roussel, and Maureen Stone. Eigentongue feature extraction for an ultrasound-based silent speech interface. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 1, pages I–1245. IEEE, 2007. 2

[19] 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 Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 3

[20] Hyeongwoo Kim, Mohamed Elgharib, Hans-Peter Zollöfer, Michael Seidel, Thabo Beeler, Christian Richardt, and Christian Theobalt. Neural style-preserving visual dubbing. *ACM Transactions on Graphics (TOG)*, 38(6):178:1–13, 2019. 2

[21] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 2

[22] Scott A. King and Richard E. Parent. A 3D parametric tongue model for animated speech. *The Journal of Visualization and Computer Animation*, 12(3):107–115, 2001. 2

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, abs/1412.6980:11, 2014. 5

[24] Sumedha Kshirsagar and Nadia Magnenat-Thalmann. Visyllable based speech animation. *Comput. Graph. Forum*, 22(3):632–640, 2003. 2

[25] Felix Kuhnke and Jörn Ostermann. Visual speech synthesis from 3D mesh sequences driven by combined speech features. In *2017 IEEE International Conference on Multime-*

dia and Expo (ICME), pages 1075–1080, Hong Kong, China, 2017. IEEE. 4

[26] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017. 5

[27] Changwei Luo, Jun Yu, Xian Li, and Leilei Zhang. Hmm based speech-driven 3D tongue animation. In *2017 IEEE International Conference On Image Processing (ICIP)*, pages 4377–4381, Beijing, China, 2017. IEEE, IEEE. 2

[28] Ran Luo, Qiang Fang, Jianguo Wei, Wenhuan Lu, Weiwei Xu, and Yin Yang. Acoustic vr in the mouth: A real-time speech-driven visual tongue system. In *2017 IEEE Virtual Reality (VR)*, pages 112–121. IEEE, 2017. 2

[29] Dominic W. Massaro, Jonas Beskow, Michael M. Cohen, Christopher L. Fry, and Tony Rodgriguez. Picture my voice: Audio to visual speech synthesis using artificial neural networks. In *Auditory-Visual Speech Processing*, pages 133–138, Santa Cruz, California, USA, August 1999. ISCA. 2

[30] Dominic W. Massaro, Ying Liu, Trevor H. Chen, and Charles Perfetti. A multilingual embodied conversational agent for tutoring speech and language learning. In *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing*, pages 825–828, Pittsburgh, PA, USA, 2006. ISCA. 2

[31] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502, Stockholm, Sweden, 2017. ISCA. 5

[32] Salvador Medina, Sarah L. Taylor, Mark Tiede, Alexander Hauptmann, and Iain Matthews. Importance of parasagittal sensor information in tongue motion capture through a diphonic analysis. *Interspeech 2021*, pages 3340–3344, 2021. 3

[33] Frederic I. Parke and Keith Waters. *Computer Facial Animation*. A K Peters/CRC Press, New York, USA, 1996. 2

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, volume 32, pages 8024–8035. Curran Associates, Inc., Red Hook, NY, USA, 2019. 5

[35] Joseph S Perkell. *A physiologically-oriented model of tongue activity in speech production.* PhD thesis, Massachusetts Institute of Technology, 1974. 2

[36] Lawrence Rabiner. *Fundamentals of speech recognition.* PTR Prentice Hall, Englewood Cliffs, N.J., 1993. 4

[37] Korin Richmond, Phil Hoole, and Simon King. Announcing the electromagnetic articulography (day 1) subset of the mngu0 articulatory corpus. In *INTERSPEECH*, 2011. 2

[38] EH Rothauser. Ieee recommended practice for speech quality measurements. *IEEE Trans. on Audio and Electroacoustics*, 17:225–246, 1969. 3

[39] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In *INTERSPEECH*, pages 1–9, Graz, Austria, 2019. ISCA. 4

[40] Eftychios Sifakis, Andrew Selle, Avram Robinson-Mosher, and Ronald Fedkiw. Simulating speech with a physics-based facial muscle model. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 261–270, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. 2

[41] Yang Song, Jingwen Zhu, Dawei Li, Andy Wang, and Hairong Qi. Talking face generation by conditional recurrent adversarial network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 919–925, Macao, China, 7 2019. International Joint Conferences on Artificial Intelligence Organization. 2

[42] Ingmar Steiner, Sébastien Le Maguer, and Alexander Hewer. Synthesis of tongue motion and acoustics from text using a multimodal articulatory database. *IEEE ACM Trans. Audio Speech Lang. Process.*, 25(12):2351–2361, 2017. 2

[43] Ingmar Steiner and Slim Ouni. Artimate: an articulatory animation framework for audiovisual speech synthesis. *ArXiv*, abs/1203.3574:1–4, 2012. 2

[44] Ingmar Steiner and Slim Ouni. Progress in animation of an ema-controlled tongue model for acoustic-visual speech synthesis. *ArXiv*, abs/1201.4080:1–8, 2012. 2

[45] Maureen Stone and Andrew Lundberg. Three-dimensional tongue surface shapes of english consonants and vowels. *The Journal of the Acoustical Society of America*, 99(6):3728–3737, 1996. 2

[46] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 2

[47] Sarah L. Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017. 3, 4

[48] Sarah L. Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. Dynamic units of visual speech. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, pages 275–284, Postfach 2926, Goslar, Germany, 2012. Eurographics Association. 2

[49] Mark Tiede, Carol Y. Espy-Wilson, Dolly Goldenberg, Vikramjit Mitra, Hosung Nam, and Ganesh Sivaraman. Quantifying kinematic aspects of reduction in a contrasting rate production task. *The Journal of the Acoustical Society of America*, 141(5):3580–3580, 2017. 3

[50] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762:1–12, 2017. 4, 5

[51] Konstantinos Vougioukas, S. Petridis, and M. Pantic. End-to-end speech-driven facial animation with temporal gans. In *BMVC*, pages 1–12, Newcastle, UK, 2018. Springer. 2

[52] Reiner Wilhelms-Tricarico. Physiological modeling of speech production: Methods for modeling soft-tissue articulators. *The Journal of the Acoustical Society of America*, 97(5):3085–3098, 1995. 2

[53] Alan A. Wrench. A multichannel articulatory database and its application for automatic speech recognition. In *In Proceedings 5th Seminar of Speech Production*, pages 305–308, Kloster Seeon, Bavaria, Germany, 2000. Phonetik, University Munich. 3

[54] Lingyun Yu, Jun Yu, and Qiang Ling. Bltrcnn-based 3-d articulatory movement prediction: Learning articulatory synchronicity from both text and audio inputs. *IEEE Transactions on Multimedia*, 21:1621–1632, 2019. 2

[55] Ce Zheng, Sijie Zhu, Mat'ias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3D human pose estimation with spatial and temporal transformers. *ArXiv*, abs/2103.10455:1–10, 2021. 5

[56] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9299–9306, Honolulu, Hawaii, USA, 2019. AAI. 2

[57] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. Makelttalk: speaker-aware talking-head animation. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 2

[58] Yang Zhou, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. Visemenet: Audio-driven animator-centric speech animation. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018. 3, 4

[59] Pengcheng Zhu, Lei Xie, and Yunlin Chen. Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings. In *INTERSPEECH*, 2015. 2

Speech Driven Tongue Animation - Supplementary Material

Salvador Medina[1, 2], Denis Tome[2], Carsten Stoll[2], Mark Tiede[3], Kevin Munhall[4]
Alex Hauptmann[1], Iain Matthews[2]
[1]Carnegie Mellon University, [2]Epic Games, [3]Haskins Laboratories, [4]Queens University
{salvadom, alex}@cs.cmu.edu, tiede@haskins.yale.edu, munhallk@queensu.ca
{denis.tome, carsten.stoll, iain.matthews}@epicgames.com

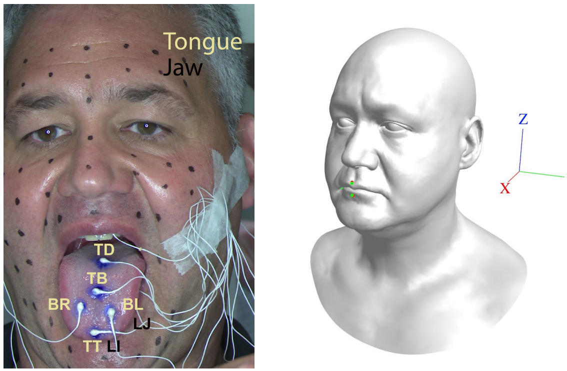
1. Further Experiments

We repeated the same set of experiments as described on Section 5, but with a larger input window of 1000 ms instead of 300 ms. As we can see in Table 1, the results across all the models and features follow the same pattern with an overall improvement, at the cost of an increase in the number of parameters and inference time per model. The inference time of these models make them **non-practical to use in real-time applications**, such as interactive avatars in video games or telecommunications.

We also found in both experimental setups, training the models with an input of 300 ms and 1000 ms that traditional audio feature representations such as phonemes and MFCC did not generalize as well as deep-learning based representations on out-of-domain speech audio. A comparison of the resulting animations is shown in the supplementary video.

2. Jaw Motion Analysis

In Figure 2 we visualized all the samples of both jaw sensors LI (midsagittal) and LJ (parasagittal) in a 3D scatter plot from three different views. As in can be seen, our captured data follows the Posselt's Envelope of Motion (PEM) [1]. The PEM describes the range under which the jaw can move due to the physiological constraints based on the bones, muscles and tendons that form the jaw. For context, we can see in Figure 1b that the reference frame of our captured data is the following: the X-axis describes the anteroposterior direction, the Y-axis the mediolateral direction, and the Z-axis the vertical direction. As carefully described in [2], the jaw motion from a frontal view usually follows a shield appearance as seen in Figure 2a. From a sagittal or lateral view, it follows a prolonged fang shape as shown in Figure 2b. While from a top view, the jaw motion follows a diamond shape as the one displayed on Figure 2c. Our visualization shows that our data follows such shapes with undefined edges. The main reason being that the actor did not do any extreme articulation during the capture session, as he only uttered regular sentences at a normal pace and fast pace in a neutral emotion. This visualization supports that it would be appropriate to increase the variability of the gesticulations during a capture session in future work.



(a) Sensor Placement

(b) Axis reference

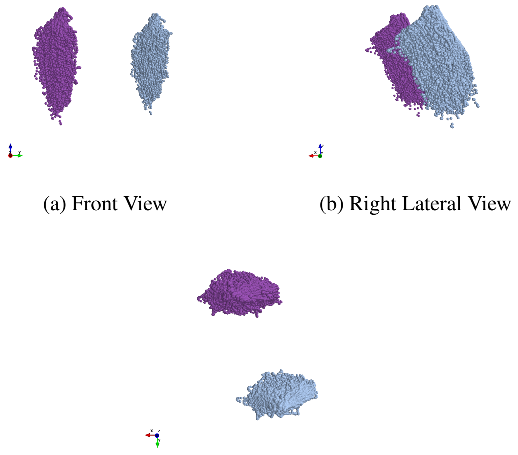
Figure 1. Reference images. (a) Sensor placement on tongue, lips and jaw. (b) Head-axis reference: X-axis describes the anteroposterior direction (back to front), Y-axis describes the mediolateral direction (right to left), and Z-axis describes the vertical direction (bottom to top).

3. Predicted Tongue Motion Analysis

We traced the predicted motion and compared it against the ground truth on test data samples for a better understanding of our best model's behavior beyond the temporal mean sample error. The trace visualization in Figure 3a shows how the long motions are close to the ground truth. However, the sensor's positions have an overall shift to the front of the mouth, as these depend on the predicted initial position of the tongue at the beginning of the sequence. We also found that the articulatory decoder network learned to generate shortcuts on fast and complex motions which take place within a small space. In Figure 3b, we can see a clear shortcut on the motions of the *Blade Left* and *Tongue Tip* sensors. These results are reasonable since the models were trained only on a Mean Square Error loss leading the pre-

Table 1. Model architecture evaluation using different audio feature representations. Models were trained with 1 s input windows of audio. The error is the temporal MSE in mm calculated through the validation split. The number of parameters reported is the amount of trainable parameters per architecture design. The inference time is the mean time over the validation split measured as ms per second of audio input.

Decoder \ Feature	Phone	MFCC	DeepSpeech2	W2V-C	W2V-Z	Num. Parameters	Inference [ms]
MLP 50:15	2.315	2.157	2.619	1.957	1.928	3.29×10^{8}	0.309
LSTM-1L	2.657	2.299	2.350	2.048	4.219	3.17×10^{6}	1.150
LSTM-2L	4.216	2.219	2.342	2.016	2.026	5.27×10^{6}	2.238
LSTM-5L	2.609	2.133	2.331	2.014	1.994	1.16×10^{7}	5.432
Bi-LSTM-1L	3.355	2.074	2.204	1.977	2.272	6.33×10^{6}	2.229
Bi-LSTM-2L	2.268	1.874	2.096	1.825	1.781	1.26×10^{7}	4.512
Bi-LSTM-5L	2.247	1.732	1.987	1.754	1.708	3.15×10^{7}	11.000
GRU-1L	4.195	2.213	2.283	1.943	2.001	2.38×10^{6}	1.144
GRU-2L	2.559	2.098	2.248	1.905	1.945	3.95×10^{6}	2.193
GRU-5L	2.570	2.003	2.235	1.908	1.943	8.68×10^{6}	5.339
Bi-GRU-1L	4.304	1.964	2.094	1.828	1.910	4.76×10^{6}	2.290
Bi-GRU-2L	2.206	1.784	2.091	1.744	1.714	9.48×10^{6}	4.439
Bi-GRU-5L	**2.179**	**1.660**	**1.935**	**1.684**	**1.648**	2.37×10^{7}	10.955
Transformer	2.349	2.393	2.139	1.926	2.044	5.049×10^{7}	3.552



(a) Front View

(b) Right Lateral View

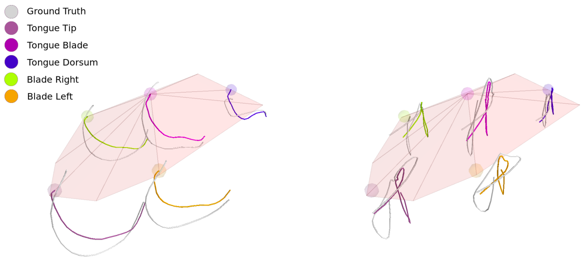
(c) Top View

Figure 2. Different views for Posselt's Envelope of Motion for both jaw sensors LI and LJ: midsagittal (purple) and parasagittal (blue) respectively.

dicted sensor position to a local minimum. Future work should address these issues by adding more constraints to the learning loss.

3.1. Landmark Pose Prediction Error Analysis

We analyzed the landmark position error of the model predictions across all audio representations on the and summarize the results in Figure 4.



(a) Long Motion

(b) Prediction Shortcut

Figure 3. Visualization of 300 [ms] of predicted motion on a test sample vs. ground truth. Ground truth is in gray and the predicted motion is in color according to the sensor. (a) Shows an example of a predicted long motion. (b) Shows an example of motion shortcuts on prediction visible on *Blade Left* and *Tongue Tip* sensors.

Overall, the model trained with phonetic representation shows the highest error followed by the model trained with DeepSpeech2 audio features. The models trained using MFCC and both Wav2Vec features follow a similar pattern across all the landmark predictions.

The most accurate landmark prediction is the upper lip (UL), with a mean error of 1.16 mm across all audio features. The least accurate landmark is the tongue tip (TT) with a mean error of 2.26 mm. Most of the errors occur while predicting landmark positions for the tongue and lower lip (LL) as these points have the most significant motion during an utterance. Midsagittal and parasagittal jaw landmark (LI, LJ) predictions have lower error, which

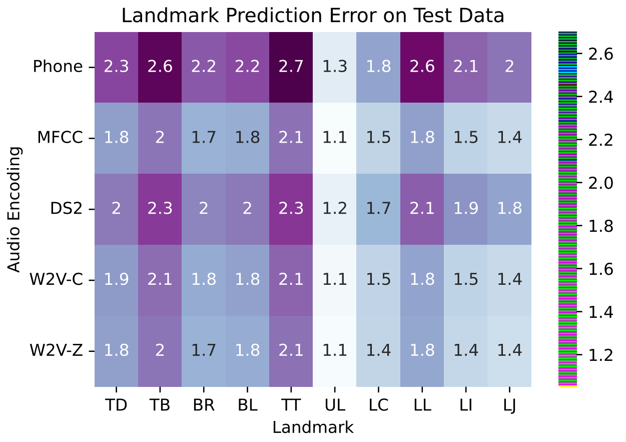


Figure 4. Landmark Prediction Error on the test set of the bidirectional 5-layered GRU over all audio encodings. The error values shown are the mean sample error in mm of each landmark.

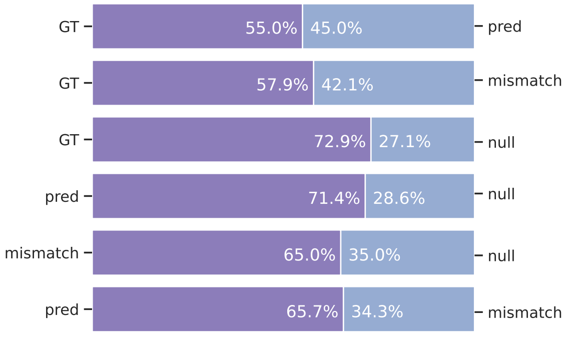


Figure 5. Pairwise preference perception user-study results. *GT*: Ground-truth animation; *pred*: predicted animation; *mismatch*: tongue mismatched animation; *null*: nullified tongue animation

seems reasonable since the jaw moves slower than the tongue and lower lip in general.

4. Perceptual User Study

We conducted a pairwise perceptual user-study to evaluate our generated animations using ten test sentences. The range of jaw and tongue motion go from tongue visible tongue motions to almost closed teeth. For each test sentence, we generated four animations using: *a)* ground truth (*GT*), *b)* prediction of our best model (*pred*), *c)* an animation taking the lips from *GT* and injecting the tongue animation from another sample (*mismatch*), and *d)* by removing the tongue motion from *GT* (*null*). We generated six pairwise comparison videos per sample, presenting a total of 60 videos to 15 users and asked them to select the *most realistic animation* from each pair.

The results shown in Figure 5 present the percentage of users that preferred one animation over the other. As we can see, most users prefer *GT* over the others (rows 1-3). Moreover, an interesting finding from this user study reveals that users prefer an animated tongue over a nullified tongue motion, even when the tongue motion is not matching the spoken sentence (rows 3-5). Finally, the fact that users prefer *GT* over *mismatch* and *pred* over *mismatch* (row 2, 6) proves the consistency of our estimations as most of the users prefer *pred* over *mismatch* (row 6).

5. External Asset License Details

In this work, we used public tools and pre-trained models to encode the speech audio. We used the Montreal Forced Aligner[1] to extract the phoneme representations by using the English acoustic model, which has an MIT license. To obtain the MFCC features, we used the Python package Librosa[2] released under the ISC license. For DeepSpeech2 features, we used publicly available code[3] and its LibriSpeech pre-trained model[4] released under the MIT license. Finally, we used the the *large* Wav2Vec pre-trained model[5] from the Fairseq[6] package to encode the input audio, both the package and the model are released under an MIT license. As a final remark, LibriSpeech is a corpus which comprises 1000 hours of diverse speech audio, and is released under the CC BY 4.0 license.

The high quality animations shown in the supplementary video were created using the MetaHuman Creator[7] tool and rendered in Unreal Engine with permission from Epic Games.

References

[1] Ulf Posselt et al. Range of movement of the mandiblew. *The Journal of the American Dental Association*, 56(1):10–13, 1958. 1

[2] Gaspard Zoss, Derek Bradley, Pascal Bérard, and Thabo Beeler. An empirical rig for jaw animation. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 1

[1] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner

[2] https://github.com/librosa/librosa

[3] https://github.com/SeanNaren/deepspeech.pytorch

[4] https://github.com/SeanNaren/deepspeech.pytorch/releases

[5] https://github.com/pytorch/fairseq/tree/main/examples/wav2vec

[6] https://github.com/pytorch/fairseq

[7] https://www.unrealengine.com/en-US/digital-humans

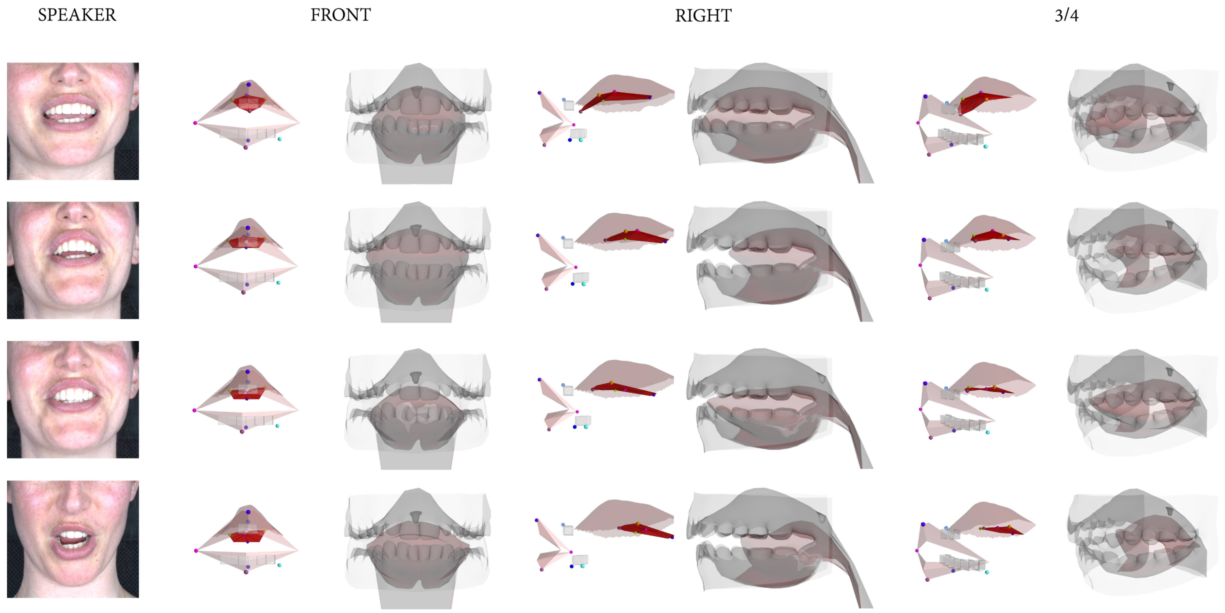


Figure 6. Frames from the generated animation at inference time for a speech input incoming from a speaker not seen during training. For each camera view, both predicted landmark locations and the solved animation rig outputs are shown.

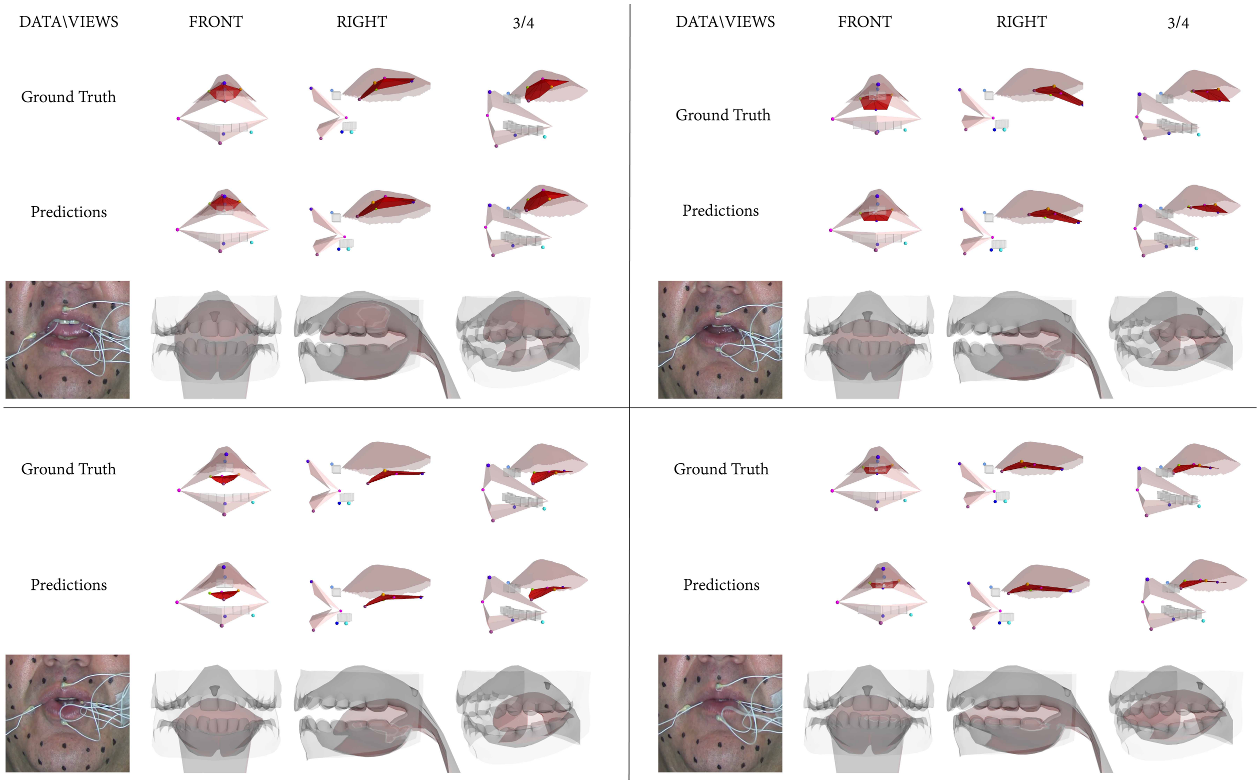


Figure 7. Visualization of a few samples from the training speaker on test speech samples not in the training set. For each camera view, both predicted landmark locations and the solved animation rig outputs are provided. As this is test data we can also show comparison to the ground truth EMA landmark locations.