# VisemeNet: Audio-Driven Animator-Centric Speech Animation

YANG ZHOU, University of Massachusetts Amherst
ZHAN XU, University of Massachusetts Amherst
CHRIS LANDRETH, University of Toronto
EVANGELOS KALOGERAKIS, University of Massachusetts Amherst
SUBHRANSU MAJI, University of Massachusetts Amherst
KARAN SINGH, University of Toronto

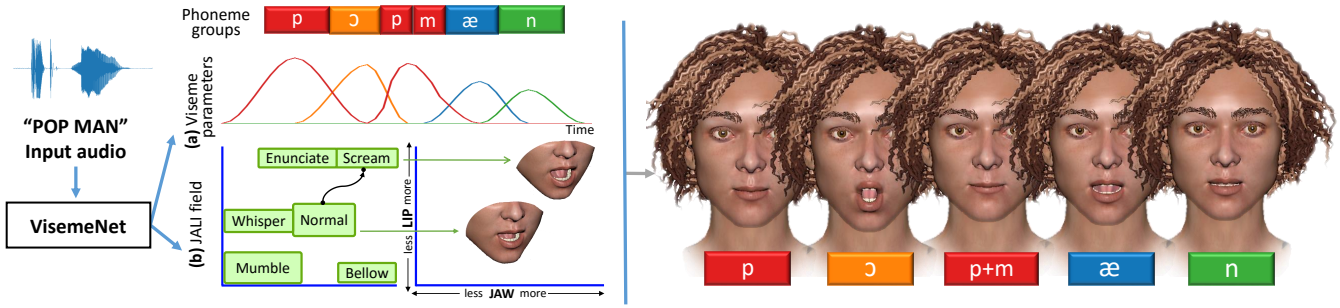



Fig. 1. VisemeNet is a deep-learning approach that uses a 3-stage LSTM network, to predict compact animator-centric viseme curves with proper co-articulation, and speech style parameters, directly from speech audio in near real-time (120ms lag).

We present a novel deep-learning based approach to producing animator-centric speech motion curves that drive a JALI or standard FACS-based production face-rig, directly from input audio. Our three-stage Long Short-Term Memory (LSTM) network architecture is motivated by psycho-linguistic insights: segmenting speech audio into a stream of phonetic-groups is sufficient for viseme construction; speech styles like mumbling or shouting are strongly co-related to the motion of facial landmarks; and animator style is encoded in viseme motion curve profiles. Our contribution is an automatic real-time lip-synchronization from audio solution that integrates seamlessly into existing animation pipelines. We evaluate our results by: cross-validation to ground-truth data; animator critique and edits; visual comparison to recent deep-learning lip-synchronization solutions; and showing our approach to be resilient to diversity in speaker and language.

CCS Concepts: • **Computing methodologies** → **Animation**; *Machine learning algorithms*;

Additional Key Words and Phrases: facial animation, neural networks

**ACM Reference Format:**
Yang Zhou, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. 2018. VisemeNet: Audio-Driven Animator-Centric Speech Animation. *ACM Trans. Graph.* 37, 4, Article 161 (August 2018), 10 pages. https://doi.org/10.1145/3197517.3201292

Authors' addresses: Yang Zhou, University of Massachusetts Amherst, yangzhou@cs.umass.edu; Zhan Xu, University of Massachusetts Amherst, zhanxu@cs.umass.edu; Chris Landreth, University of Toronto, c.landreth@rogers.com; Evangelos Kalogerakis, University of Massachusetts Amherst, kalo@cs.umass.edu; Subhransu Maji, University of Massachusetts Amherst, smaji@cs.umass.edu; Karan Singh, University of Toronto, karan@dgp.toronto.edu.


0730-0301/2018/8-ART161 $15.00
https://doi.org/10.1145/3197517.3201292

## 1 INTRODUCTION

The importance of realistic computer generated human faces cannot be understated. A diverse set of applications ranging from entertainment (movies and games), medicine (facial therapy and prosthetics) and education (language/speech training and cyber-assistants) are all empowered by the ability to realistically model, simulate and animate human faces. Imperfect emulation of even subtle facial nuance can plunge an animated character into the Uncanny Valley, where the audience loses trust and empathy with the character. Paradoxically, the greater the rendered realism of the character, the less tolerant we are of flaws in its animation [MacDorman et al. 2009]. The expressive animation of speech, unsurprisingly, is a critical component of facial animation that has been the subject of research for almost half-a-century [Bailly et al. 2012].

Typically, keyframing or performance capture are used for high-end commercial animation. Keyframing by professional animators is both expressive and editable, but laborious and prohibitively expensive in time and effort. Performance capture solutions are the opposite; recording a vocal performance is relatively easy, but hard to further edit or refine, and voice actors often have visually inexpressive faces.

Recent approaches however, have shown that there is enough detail in the audio signal itself to produce realistic speech animation. JALI [Edwards et al. 2016] presented a FACS [Ekman and Friesen 1978] and psycho-linguistically inspired face-rig capable of animating a range of speech styles, and an animator-centric procedural lip-synchronization solution using an audio performance and speech transcript as input. Following JALI, a number of conncurrent approaches have successfully shown the use of deep learning to capture a mapping between input audio and an animated face [Karras et al. 2017; Suwajanakorn et al. 2017; Taylor et al. 2017]. These approaches however, produce results that, unlike JALI, do not integrate well into a typical animator workflow. This paper addresses the

problem of producing animator-centric speech animation directly from input audio.

Arguably, existing deep learning based solutions can be used to first generate a text transcript [Graves and Jaitly 2014] from audio input, and then align it with the audio [McAuliffe et al. 2017], resulting in the input required by JALI. As shown in the evaluation (Section 6), dissecting the problem into these three independent modules is error prone, and further any solution involving a speech transcript becomes strongly language and lexicon specific. In contrast, Karras et al. [Karras et al. 2017] show that an audio signal has the potential to predict an animated face, agnostic of language.

Our approach is inspired by the following psycho-linguistic observations:

- While classification of a precise phonetic stream from audio can be difficult due to categorical perception [Fugate 2013], and its dependence on cultural and linguistic context, our problem of predicting a stream of phoneme-groups is simpler. Aurally individual phonemes within a phoneme-group are often hard to distinguish (eg. *pa* and *ba*) [Liberman et al. 1957], but unnecessary for speech animation as they map to near identical visemes [Fisher 1968].
- The Jaw and Lip parameters in the JALI model that capture speech style as a combination of the contribution of the tongue-jaw and face-muscles to the produced speech, are visually manifested by the motion of facial landmarks on the nose, jaw and lips.
- The profile of speech motion curves (attributes like onset, apex, sustain and decay) capture speaker or animator style in professionally keyframed animation. Learning these curves from training data allow us to encode aspects of animator or speaker style.

We thus propose a three-stage network architecture: one that learns to predict a sequence of phoneme-groups from audio; another that learns to predict the geometric location of important facial landmarks from audio; and a final stage that learns to use phoneme-groups and facial landmarks to produce JA-LI parameter values and sparse speech motion curves, that animate the face.

The contribution of this paper is thus a deep-learning based architecture to produce state-of-the-art animator-centric speech animation directly from an audio signal in near real-time (120ms lag). Our evaluation is fivefold: we evaluate our results quantitatively by cross-validation to ground-truth data; we also provide a qualitative critique of our results by a professional animator; as well as the ability to further edit and refine the animated output; we also show our results to be comparable with recent non-animator-centric audio to lip-synchronization solutions; finally we show that with speech training data that is reasonably diverse in speaker, gender and language, our architecture can provide a truly language agnostic solution to speech animation from audio.

## 2 RELATED WORK

The large body of research on audiovisual speech animation [Bailly et al. 2012] can be broadly classified into *procedural*, *performance-capture*, *data-driven*, and more specifically the very recent *deep-learning* based techniques. We focus on techniques based on deep-learning, that are of greatest relevance to our research, after a brief overview of the other three approaches (refering the reader to [Edwards et al. 2016] for a more detailed review).

*Procedural.* Procedural speech animation segments speech into a sequence of phonemes, which are then mapped by rules to visemes. A *viseme* or *visible phoneme* [Fisher 1968] refers to the shape of the mouth at the apex of a given phoneme (see Figure 2). The three problems that a procedural approach must solve are: *mapping* a given phoneme to a viseme (in general a many-many mapping based on the spoken context [Taylor et al. 2012]); *co-articulation*, or the overlap in time between successive visemes, resulting from the fluid and energy efficient nature of human speech, often addressed using Cohen and Massaro's [1993] seminal *dominance model*; *viseme profile*, or the curve shape that defines the attack, apex, sustain and decay of a viseme over time [Bailly 1997] . JALI [Edwards et al. 2016] defines the state of the art in procedural speech animation, producing compact animator-friendly motion curves that correspond directly to the input phonetic stream.

*Performance-capture.* Performance-capture based speech animation transfers motion data captured from a human performer onto a digital face [Williams 1990]. Performance capture has become increasingly powerful and mainstream with the widespread adoption of cameras, depth sensors, and reconstruction techniques, that can produce a 3D face model from a single image [Hu et al. 2017]. Real-time performance-based facial animation research [Li et al. 2013; Weise et al. 2011] and products like *Faceware* (*faceware.com*), are able to create high quality general facial animation, including speech, and can be further complemented by speech analysis [Weise et al. 2011]. The disadvantage of performance capture is that is visually limited by the actor's performance and is difficult for an animator to edit or refine.

*Data-driven.* These approaches smoothly stitch pieces of facial animation data from a large corpus, to match an input speech track [Bregler et al. 1997], using morphable [Ezzat et al. 2002], hidden Markov [Wang et al. 2012], and active appearance models (AAM) [Anderson et al. 2013; Taylor et al. 2012]. These data-driven methods tend to be limited in scope to the data available, and the output, like performance-capture, is not animator-centric.

*Deep learning-based speech animation.* Recent research has shown the potential of deep learning to provide a compelling solution to automatic lip-synchronization simply using an audio signal with a text transcript [Taylor et al. 2017], or even without it [Karras et al. 2017].

Taylor et al.'s approach [2017] requires an input text transcript, which even if automated [Graves and Jaitly 2014], introduces more complexity and language dependence than we believe is necessary for animating speech. The predicted output is an 8-dimensional face-space, that is then re-targeted to any animation rig. While the rig can be post-edited by an animator, there is no compact mapping from audio to animator-centric viseme curves or facial action units. As a result while it is possible to overlay the speech output with expression, it is harder to edit or refine the animation of the spoken content or its style. In contrast, our network is trained to map audio to viseme animation curves, which are both sparse (i.e., very few viseme controls are active at a time, have non-zero value only for short periods), and are low-dimensional (the number of visemes is small). Our viseme curves are more directly related to speech, compared to the viseme-agnostic mouth pose parameterization used in [Taylor et al. 2017].

Karras et al.'s approach [2017] removes any dependence on a phonetic transcript, but is designed to directly output the vertex positions of a 3D mesh. While they produce impressive results from a small training set, their output is not well suited to current animation practice. Arguably, vertex positions or facial landmarks could be re-targeted to any facial animation rig [Ribera et al. 2017], yet the re-targeted result is not animator-centric. In addition, breaking the rig prediction into separate steps is problematic for a number of reasons. First, face rigs have lots of redundancy, thus a per frame best fit to rig space can have temporal discontinuities. Second, rig animation curves are often correlated. Independent editing or sparsification of these curves can break perceptual constraints (e.g., a closed mouth on bilabial sounds). Third, accumulated errors in vertex positions can be exacerbated after subsequent fitting steps. In contrast, our method does not break rig prediction into separate, independently optimized steps. Our whole network with all its stages is trained as a single pipeline optimized to minimize animation error.

Suwajanakorn et al. [2017] present a deep network to synthesize an output video of Barack Obama, combining an input audio and target video of him speaking. While their problem statement is quite different from ours, we are encouraged by their successful use of a Long-Short Term Memory (LSTM) network for lip-synchronization, and adopt it for the stages of our overall network.

From a technical point of view, our proposed network architecture differs from the above deep learning approaches in several key design aspects. It employs a three-stage network architecture to (a) segment speech audio into streams of phonetic groups related to viseme construction, (b) predict facial landmarks capturing speech style cues, (c) then uses the extracted phoneme groups, landmarks and audio features to produce viseme motion curves. All stages are necessary to achieve high performance, as discussed in our experiments section (Section 6). Furthermore, we combine different sources of audiovisual data for training. Our training data include audio clips with ground-truth phonemes, video clips with tracked landmarks, and 3D facial animations with ground-truth visemes. All stages are jointly trained using multi-task learning to minimize errors in the prediction of phoneme groups, landmarks, visemes, co-articulation, and speech style parameters.

## 3 ALGORITHM DESIGN

Our approach is designed to achieve high-quality, animator-editable, style-aware, language agnostic, real-time speech animation from audio.

Following the current animation practice of FACS-like face rigs, and state-of-the-art animator-centric speech [Edwards et al. 2016], our network uses an input audio signal to predict a sparse and compact set of viseme values (see Figure 2), and jaw and lip parameters, over time. Our neural network architecture (see Figure 3) is designed to exploit psycho-linguistic insights and make the most effective use of our training data.

*Phoneme group prediction.* A large part of our network, the "phoneme group stage" in Figure 3 (top left box), is dedicated to map audio to phonemes groups corresponding to visemes. For example, the two labio-dental phonemes /$f$ and $v$/ form a group that maps to a single, near-identical viseme [Edwards et al. 2016], where the lower lip is pressed against the upper teeth in Figure 2 (last row, right). We identified 20 such visual groups of phonemes expressed in the International Phonetic Alphabet (IPA) in Figure 2. Our network is trained to recognize these phoneme groups from audio without any text or phonetic transcript. By predicting only phonetic groups relevant to animation, our task is simpler and less sensitive to linguistic context. The network can also be trained with less data than most speech processing pipelines. As demonstrated in the evaluation (Section 6), using off-the-shelf audio-to-text techniques to extract individual phonemes that subsequently predict visemes, leads to lower performance than our architecture.

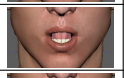
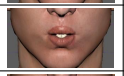
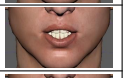
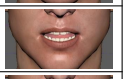
| Viseme | Phoneme | Output | Viseme | Phoneme | Output |
|---|---|---|---|---|---|
| Ah | ɑ, ɔ, a |  | LNTD | l, n, t, d, ʃ, ʟ, ɾ |  |
| Aa | æ |  | GK | g, k, ŋ, q, ɢ |  |
| Eh | e, ɛ |  | MBP | b, m, p |  |
| Ee | i |  | R | ɹ |  |
| Ih | ɪ |  | WA_PED AL | w, ʊ, ʍ |  |
| Oh | o, ɒ |  | JY | j, dʒ, c, ɟ |  |
| Uh | ʊ, ʌ, ɞ, ɐ, ɶ, ö, ɚ, ɨ |  | S | s, z, ɣ |  |
| U | u |  | ShChZh | ʃ, tʃ, ʒ, ɬ, ɮ, |  |
| Eu | œ, y, ɯ, ø, ɵ |  | Th | θ, ð |  |
| Schwa | ə, ɘ |  | FV | f, v, ɱ |  |

Fig. 2. List of visemes along with groups of phonemes (in International Phonetic Alphabet format) and corresponding lower face rig outputs that our architecture produces.

*Speech style prediction.* Phoneme groups alone have no information of vocal delivery and cannot predict visemes, specially for expressive emotional speech. The same phoneme /æ/ (see Figure 1b) might be pronounced conversationally, or screamed. These style attributes of the audio performance can be captured using jaw and lip parameters [Edwards et al. 2016]. These parameters are also strongly correlated to the (2D frontal) jaw and lip landmarks in a visual capture of the vocal performance. The "landmark stage" of our network in Figure 3 (bottom-left box) is designed to predict a set of jaw and lip landmark positions over time given input audio.

*Viseme prediction.* The last part of our network, the "viseme stage" in Figure 3(right-box), combines the intermediate predictions of phoneme groups, jaw and lip parameters, as well as the audio signal itself to produce visemes. By training our architecture on a combination of data sources containing audio, 2D video, and 3D animation of human speech, we are able to predict visemes accurately. We represent visemes based on the JALI model [Edwards et al. 2016], comprising a set of intensity values for 20 visemes and 9 co-articulation rules, and **JA**W and **LI**P parameters that capture

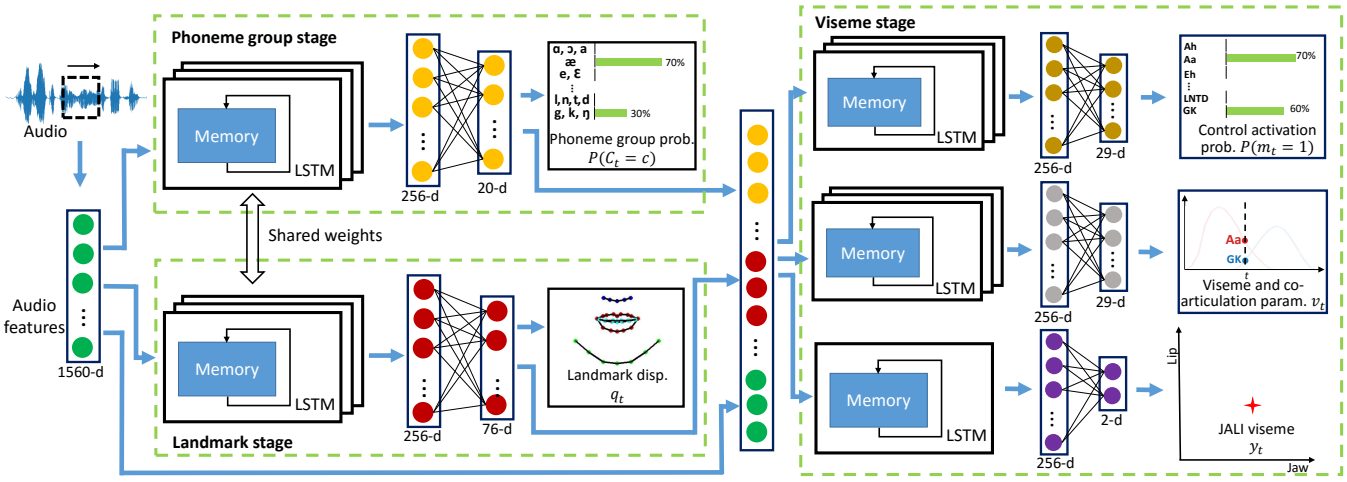



Fig. 3. Our architecture processes an audio signal (left) to predict JALI-based viseme representations: viseme and co-articulation control activations (top right), viseme and co-articulation rig parameters (middle right), and 2D JALI viseme field parameters (bottom right). Viseme prediction is performed in three LSTM-based stages: the input audio is first processed through the phoneme group stage (top left) and landmark stage (bottom left), then the predicted phoneme group and landmark representations along with audio features are processed through the viseme stage.

speaking styles. The viseme and co-articulation values over time can animate standard FACS-based production rigs, with the JA-LI parameters for a rig adding control over speech style. Notably, these are precisely the controls professional animators keyframe, and are thus directly amenable to editing and refinement.

*Transfer learning from audiovisual datasets.* We need reliable sources of diverse training data with compelling variation in terms of different speakers, speech styles, and emotional content, to train a network that generalizes well. One potential source of training data would be audio clips with corresponding streams of 3D facial rig parameters. Such a large coherent dataset with enough variability is not easily available, and would be too expensive to create with professional animators. On the other hand, there is a wealth of publicly available audiovisual corpora (2D audio+video+text transcript clips), such as BIWI [Fanelli et al. 2010], SAVEE [Haq and Jackson 2009], and GRID [Cooke et al. 2006]. Although these corpora do not contain any facial rig parameters, they are nonetheless valuable, in the context of our network architecture:

- The faces in the video clips can be accurately detected and annotated with landmarks through modern computer vision. The extracted facial landmarks corresponding to the speech audio are then useful to train the landmark stage of our network.
- The text transcripts can be automatically aligned with the audio clip [McAuliffe et al. 2017], to provide training phonemes for the phoneme group stage of our network.

To make use of the large amounts of data in these audiovisual datasets, we employ a *transfer learning* procedure to train our network. We first "pre-train" the phoneme group and landmark stages of our network based on training phoneme groups and landmarks extracted from the above audiovisual datasets. We then initialize these two stages according to their pre-trained parameters, and then jointly train the whole network. To perform this joint training, we still need a dataset of audio clips with associated streams of facial rig parameters. Mapping phoneme groups and facial landmarks to visemes however, is significantly easier than mapping general speech audio to phonemes and landmarks. Further, the phoneme groups are strongly correlated to visemes, while the landmarks are strongly correlated to jaw and lip parameters. Thus, to train the viseme stage, a much smaller dataset of example 3D animations with face rig parameters is required for sufficient generalization. We empirically observed that using our transfer learning procedure results in a better generalization performance than simply training the entire network on a small dataset of audio clips with rig parameters. We also found that adapting a Multi-Task Learning (MTL) procedure to train the network simultaneously according to multiple objectives involving phoneme group, landmark, viseme and other rig parameter prediction was also important to achieve high performance.

*Memory-enabled networks.* We adapt a memory-enabled neural network architecture based on Long Short-Term Memory units (LSTMs) [Hochreiter and Schmidhuber 1997; Olah 2015] for all stages of our network. We believe that memory-based networks are important to correctly capture co-articulation and speech context from input audio, which even for a human listener, is challenging from isolated audio fragments. Memory-enabled networks explicitly store and represent a large amount of context in the input signal that is useful to reliably infer the spoken context. Finally, another advantage of our architecture is that it can predict viseme motion curves in near real-time (120ms or 12 frame lag), given the input audio on modern GPUs. In the following sections, we discuss the network architecture and training procedure in more detail.

## 4 NETWORK ARCHITECTURE

Our network architecture takes an audio signal as input and outputs viseme representations based on JALI. As discussed in the previous section and shown in Figure 3, our network has a three stage-architecture: the input audio is first processed through the phoneme group and landmark stages, then the predicted phoneme

group and landmark representations along with audio features are processed through the viseme prediction stage. All branches are based on a combination of memory-based LSTM units, which encode context in the input signal, and fully connected layers, which decode the memory of the units into time-varying predictions. Below we discuss our input audio representation and the three stages of our network in more detail.

*Input audio representation.* Given an audio signal as input, we extract a feature vector for each frame encoding various power spectrum and audio frequency signal characteristics. Our feature vector concatenates 13 Mel Frequency Cepstral Coefficients (MFCCs) [Davis and Mermelstein 1980] that have been widely used for speech recognition, 26 raw Mel Filter Bank (MFB) features that have been shown to be particularly useful for emotion discrimination in audio [Busso et al. 2007], and finally 26 Spectral Subband Centroid features that often result in better speech recognition accuracy when they are used in conjuction with MFCCs [Paliwal 1998]. The resulting 65-dimensional feature vector is passed as input to all three stages of our network for further processing. The features are extracted every 10 ms, or in other words feature extraction is performed at a 100 FPS rate. The frequency analysis is performed within windows of size 25 ms in the input audio.

*Phoneme group stage.* The phoneme group stage takes as input the audio features concatenated from 12 frames before the current frame, and also the audio features of the current frame plus 11 frames after the current one. This means that given real-time audio inputs, the network will infer visemes with a lag of 120 ms, plus the required time to extract audio features and perform viseme inference given the audio features per frame (feature extraction and network inference take 1 ms per frame measured on a TitanX GPU, allowing real-time inference with the abovementioned lag). The concatenation produces a 1560-dimensional sharfeature vector $\mathbf{x}_t$ per frame $t$ ( 65 features x 24 frames) covering audio signal information in a window of 240$ms$. The rationale behind using this window is that it approximately matches the average duration of a phoneme in normal speech. We also found that that such window size represented a good trade-off between fast processing time and high phoneme group prediction accuracy.

The feature vector $\mathbf{x}_t$ passes through three layers of unidirectional LSTM units that hierarchically update their internal memory state (encoded with a 256-dimensional vector in our implementation) based on the input information in the audio features. We found that at least three layers (i.e., a deep network) were necessary to achieve sufficient generalization. The LSTM units can choose to either store in their memory cells representations of the incoming features, or alternatively erase representations from their memory cells. The choices of erasing or storing, as well as the transformations of the input features are controlled through non-linear functions (sigmoid and hyperbolic tangent functions) with learnable parameters (for their exact form, we refer to the popular tutorial [Olah 2015] and [Hochreiter and Schmidhuber 1997]).

At each frame, the memory state of the last LSTM layer is decoded towards probabilistic predictions of phoneme groups. The decoding is performed through two non-linear transformations. The first transformation involves a fully connected layer that takes as input the representation of the uppermost LSTM layer, applies a linear transformation on it to produce a 256-dimensional output, which is further processed through the commonly used REctified Linear Unit (RELU) non-linearity. The second layer takes the resulting vector, applies another linear transformation on it producing a 20-dimensional vector $\mathbf{z}_t$, which is then passed through a softmax function to output a per-frame probability for each phoneme group listed in Figure 2.

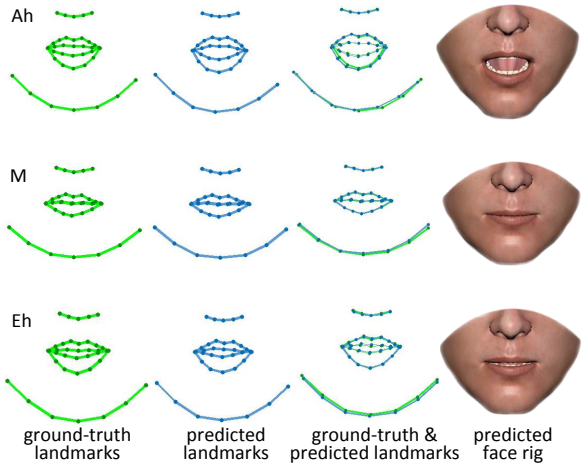



Fig. 4. Landmark output predictions for a few test frames. The spoken phoneme is shown on the top left. The first column shows ground-truth landmarks. The second column shows landmark predictions from the landmark stage of our network for the same frames. In the third column the predictions are overlaid on the ground-truth to show differences. The last column shows the corresponding predicted rig outputs.

Overall, this network stage can be seen as a non-linear function $f$ that considers audio features up to the current frame $\mathbf{x}_{1:t}$ (by exploiting the LSTM recurrent connections) to output phoneme group probabilities: $P(C_t = c) = f(\mathbf{x}_{1:t}, \boldsymbol{\theta}, \boldsymbol{\phi})$ where $C_t$ is a discrete random variable whose possible values $c$ are our phoneme groups, $\boldsymbol{\theta}$ are the LSTM parameters, and $\boldsymbol{\phi}$ are the decoder parameters. We note that we also experimented with a Bidirectional LSTM, as proposed in [Graves and Schmidhuber 2005], yet we did not perceive any noticeable differences in the output predictions. We suspect this is because we consider audio features from a large window containing both past and future frames.

*Landmark stage.* This stage of our network takes as input the 1560-dimensional feature vector $\mathbf{x}_t$ per frame (same input as in the phoneme group part), passes it through three LSTM layers and decodes the uppermost layer memory into a sparse set of 38 2D facial landmarks, representing the jaw, lip, and nose configuration per frame. Ground-truth and predicted landmarks are visualized in Figure 4. The landmarks do not aim at capturing the morphology of a particular face, but instead approximately capture the shape of the lips, positions of jaw and nose of an average face. We found that predicting these visual cues are particularly useful to infer correct visemes that reflect speech style (e.g., mumbling, screaming). In particular, the advantage of using these visual cues is that we can exploit audio and landmarks extracted from video available in large, public audiovisual datasets as additional supervisory signal to train the LSTM layers, as described in the next section.

Since phonetic groups and lower face shape are correlated, the three LSTM layers are shared between the phoneme group and landmark stages. Sharing representations is a common strategy in multi-task learning [Caruana 1997], which helps generalization when tasks are correlated. The decoder of the landmark stage is specific to landmark predictions and has its own learned parameters. It is composed of two transformations, implemented as a fully connected layer followed by RELUs, and a second fully connected layer that outputs landmark displacements per frame. The displacements are expressed relative to landmarks representing an average lower face in neutral expression. The displacements are stored in a 76-dimensional vector $\mathbf{q}_t$ per frame $t$, which simply concatenates displacement coordinates of all the landmarks. Given the neutral face landmarks $\mathbf{b}$, the animated landmark positions can be computed as $\mathbf{b} + \mathbf{q}_t$ per frame. Overall, this part of our network can be seen as another non-linear function $h$ that considers audio features up to the current frame $\mathbf{x}_{1:t}$ and outputs landmark displacements per frame: $\mathbf{q}_t = h(\mathbf{x}_{1:t}, \boldsymbol{\theta}, \boldsymbol{\omega})$ where $\boldsymbol{\theta}$ are the shared LSTM parameters, and $\boldsymbol{\omega}$ are the decoder parameters of this stage.

*Viseme stage.* The viseme stage takes as input the produced phoneme group representations $\mathbf{z}_t$ (i.e., phoneme group activations before applying softmax), landmark displacements $\mathbf{q}_t$, and also the audio features $\mathbf{x}_t$ and outputs JALI-based rig parameters and controls that determine the visemes per frame. Here, we use the audio features as additional input since phoneme groups and landmarks might not entirely capture all speech style-related information existing in audio (for example, fast breathing due to a nervous style of speech will not be captured in landmarks or phonemes, yet will manifest in audio features).

The viseme stage produces the following outputs per frame $t$ (see also Figure 3, right):

(a) 29 continuously-valued viseme animation and co-articulation parameters present in JALI (we refer to [Edwards et al. 2016] for more details). The rig parameters are represented by a 29-dimensional continuous vector $\mathbf{v}_t$.

(b) 29 binary random variables, represented as a vector $\mathbf{m}_t$, that indicate whether each of the above viseme and co-articulation control parameters is active per frame. The underlying reason for using these binary variables is that the activations of viseme and co-articulation rig parameters are largely sparse (Figure 1a), since at a given frame only one viseme is dominantly active. If we train the network to match the viseme and co-articulation parameters in the training data, without considering these binary indicator variables, then the predicted values of these action units are often biased towards low or zero values since most of the time their corresponding training values are zero.

(c) an output 2-dimensional vector $\mathbf{y}_t$ representing the 2D JALI viseme field values capturing speech style per frame $t$.

Given the inputs $\{\mathbf{z}_t, \mathbf{q}_t, \mathbf{x}_t\}$ concatenated as a single vector, the viseme stage uses a three-layer LSTM-based architecture to produce the above outputs. Here, we found that using separate LSTM layers (i.e., without shared parameters) for each output type offers the best performance, probably due to the fact that the JALI viseme field is designed to be independently controllable from the rest of rig parameters [Edwards et al. 2016], and also because the continuous rig parameter values vary widely given their activation state. Each

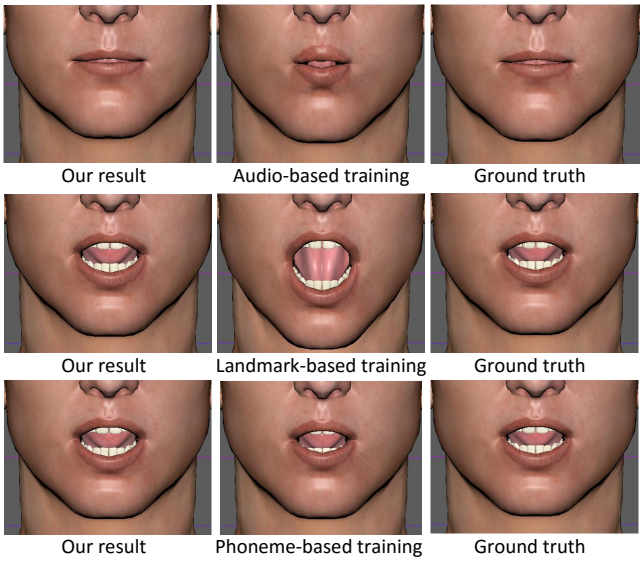



Fig. 5. Characteristic outputs of alternative architectures versus our method for a frame from our test splits. **(top)** Comparison with using audio features alone ("audio-based" training) **(middle)** Comparison with using landmarks and audio features alone ("landmark-based" training) **(bottom)** Comparison with using phoneme groups and audio features alone ("phoneme-based" training). Ground-truth for the corresponding test frame is shown on the right column. We refer to the video for the full clip.

LSTM layer uses units with a 256-dimensional memory state. The rig parameters $\mathbf{v}_t$ and JALI viseme field values $\mathbf{y}_t$ are computed by decoding their corresponding uppermost LSTM layer memory through two dedicated fully connected layers with a RELU non-linearity in-between. The binary indicator variables are predicted by decoding their corresponding uppermost LSTM layer memory, followed by two dedicated fully connected layers with a RELU non-linearity in-between, and finally a sigmoid function that produces the probability of each rig parameter to be active or not. At test time, we first predict these activation probabilities. Then we produce the continuous values of the corresponding viseme and co-articulation rig parameters whose activation probability is above a threshold, which we automatically set during training.

Overall, this part of our network can be seen as a set of three non-linear functions $g_1, g_2, g_3$ that considers phoneme group predictions $\mathbf{z}_{1:t}$, landmark predictions $\mathbf{q}_{1:t}$, and audio features $\mathbf{x}_{1:t}$ up to the current frame and output probabilities of rig parameter activations $P(\mathbf{m}_t) = g_1(\mathbf{z}_{1:t}, \mathbf{q}_{1:t}, \mathbf{x}_{1:t}, \xi_1)$, rig parameter values $\mathbf{v}_t = g_2(\mathbf{z}_{1:t}, \mathbf{q}_{1:t}, \mathbf{x}_{1:t}, \xi_2)$, and viseme values $\mathbf{y}_t = g_3(\mathbf{z}_{1:t}, \mathbf{q}_{1:t}, \mathbf{x}_{1:t}, \xi_3)$, where $\xi = \{\xi_1, \xi_2, \xi_3\}$ are learned parameters of each corresponding set of LSTM layer and decoder. In the following section, we describe how all the parameters of our network are learned.

## 5 TRAINING

We follow a two-step, transfer learning procedure to train our network. Motivated by the observation that there are large valuable amounts of audiovisual data with text transcripts available in public repositories, we first "pre-train" the phoneme group and landmark stages of our network based on 15 hours of recorded audio along with ground-truth phoneme groups and tracked facial landmarks

from video as supervisory signal. After this pre-training step, we fine-tune the whole network jointly to accurately predict visemes based on a smaller, painstakingly created dataset consisting of one hour of audio with exemplar streams of JALI rig parameter values. Below we explain our datasets and pre-processing, then we discuss pre-training and joint training procedures. Training and test splits are discussed in the results section.

*Audiovisual dataset.* This dataset contains audiovisual data from three repositories. First, we use the GRID dataset [Cooke et al. 2006], which contains transcripts, audio and video recordings of 1000 sentences spoken by 34 speakers (18 male, 16 females) in a neutral style of speech with total time duration of about 14 hours. The sentences are deliberately chosen to cover common phonemes in English. Second, we use the SAVEE dataset [Wang 2010], which contains transcripts, audio and video recordings of 480 sentences spoken by 4 male speakers expressing different emotions, including anger, disgust, fear, sadness and surprise. The total clip duration is 30 minutes. Third, we used the BIWI 3D audiovisual corpus dataset [Fanelli et al. 2010], which contains transcripts, audio, video and RGBD recordings of 1109 sentences spoken by 14 speakers (6 males and 8 females) in various emotional and neutral styles of speech with total time duration of about 1 hour. We pre-processed the videos of these datasets to extract facial landmarks involving lips, jaw and nose (Figure 4, left column) through DLib [King 2009] and FaceWare Analyzer [Faceware 2017]. The landmarks were aligned with our average face template in neutral poses and normalized to be invariant to face location, orientation and size. Then we extracted training landmark displacements for each frame relative to the average face. Given the provided text transcripts in these datasets, we used the Montreal Forced Aligner (MFA) [McAuliffe et al. 2017] to align audio with text and extract phonemes along with corresponding phoneme groups.

*JALI-annotated dataset.* An experienced animator created rigs according to the JALI template for the BIWI dataset (total $1h$ of rig motion curves with corresponding audio involving the 14 BIWI speakers).

*Pre-training.* Given $N$ audio clips with sequences of landmark displacements $\hat{\mathbf{q}}_{1:t_n}$ (where $t_n$ is number of frames of the $n^{th}$ audio clip, $n = 1...N$), and corresponding phoneme groups $\hat{c}_{1:t_n}$ extracted per frame from the training audiovisual datasets, the goal of the pre-training step is to estimate the decoder parameters $\boldsymbol{\phi}$ of the phoneme group stage, the decoder parameters $\boldsymbol{\omega}$ of the landmark stage, and the parameters $\boldsymbol{\theta}$ of their shared LSTM layers. The parameters are learned such that the predicted phoneme groups match the training ones as much as possible, the predicted landmark coordinates are as close as possible to the training ones, and also the predicted landmarks do not change over time abruptly. The goals can be expressed with a combination of a classification loss $L_c(\boldsymbol{\theta}, \boldsymbol{\phi})$ for phoneme groups, a regression loss $L_q(\boldsymbol{\theta}, \boldsymbol{\omega})$ for landmarks, and another loss $L'_q(\boldsymbol{\theta}, \boldsymbol{\omega})$ that promotes smoothness in the predicted landmark movement. This combination is expressed as the following multi-task loss:

$$L_1(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) = w_c\ L_c(\boldsymbol{\theta}, \boldsymbol{\phi}) + w_q L_q(\boldsymbol{\theta}, \boldsymbol{\omega}) + w'_q L'_q(\boldsymbol{\theta}, \boldsymbol{\omega}) \quad (1)$$

where weights of the three losses are set as $w_c = 0.75$, $w_q = 0.25$, $w'_q = 0.1$ in all our experiments, computed via hold-out validation. The classification loss $L_c(\boldsymbol{\theta}, \boldsymbol{\phi})$ favors parameters that maximize the probability of the training phoneme groups, or equivalently minimize their negative log-probability. It can be expressed as the popular cross-entropy multi-class loss:

$$L_c(\boldsymbol{\theta}, \boldsymbol{\phi}) = -\frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{t_n}\sum_{t=1}^{t_n}\log P(C_t = \hat{c}_t)\right) \quad (2)$$

The regression loss $L_q(\boldsymbol{\theta}, \boldsymbol{\omega})$ is expressed as the absolute differences (i.e., $L_1$-norm loss) between the training and predicted landmark coordinates (their total number is $M = 76$ coordinates from 38 2D landmarks):

$$L_q(\boldsymbol{\theta}, \boldsymbol{\omega}) = \frac{1}{N}\frac{1}{M}\sum_{n=1}^{N}\left(\frac{1}{t_n}\sum_{t=1}^{t_n}||\mathbf{q}_t - \hat{\mathbf{q}}_t||_1\right) \quad (3)$$

The smoothness loss $L'_q(\boldsymbol{\theta}, \boldsymbol{\omega})$ penalizes large absolute values of landmark motion derivatives with respect to time:

$$L'_q(\boldsymbol{\theta}, \boldsymbol{\omega}) = \frac{1}{N}\frac{1}{M}\sum_{n=1}^{N}\left(\frac{1}{t_n}\sum_{t=1}^{t_n}||\dot{\mathbf{q}}_t||_1\right) \quad (4)$$

where $\dot{\mathbf{q}}_t$ represents the derivative of the predicted landmark displacements over time. The derivative is computed through central finite differences in our implementation.

Minimizing the above multi-task loss function is done through batch gradient descent with batch size 256, learning rate 0.00001, momentum set to 0.9, over $2M$ iterations.

*Joint training.* For joint training, we initialize the parameters $\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}$ of the phoneme group and landmark stages to their pre-trained values, which are already expected to be close to a desired local minimum. Then we estimate all the parameters of the whole network jointly, including the parameters $\boldsymbol{\xi} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3\}$ of the viseme prediction branch, such that based on the JALI-annotated dataset, we satisfy the above goals: (a) the predicted viseme and co-articulation parameter activations match the ground-truth ones through a binary classifications loss $L_a(\boldsymbol{\xi}_1)$, (b) the predicted viseme and co-articulation parameters are as close as possible to the ground-truth ones when these units are active through a regression loss $L_v(\boldsymbol{\xi}_2)$ modified to consider these activations, (c) the predicted 2D JALI viseme field values are also as close as possible to the ground-truth ones through a regression loss $L_j(\boldsymbol{\xi}_3)$, (d) the rig parameters and JALI field values do not change abruptly over time through two smoothness losses $L'_v(\boldsymbol{\xi}_2)$ and $L'_j(\boldsymbol{\xi}_3)$, and finally (e) the predicted phoneme groups and landmarks remain close to the ground-truth ones (as done in the pre-training step). This goal can be expressed again as a multi-task loss:

$$\begin{aligned} L_2(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}, \boldsymbol{\xi}) = L_1(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega}) + w_a L_a(\boldsymbol{\xi}_1) + w_v L_v(\boldsymbol{\xi}_2) \\ + w_j L_j(\boldsymbol{\xi}_3) + w'_v L'_v(\boldsymbol{\xi}_2) + w'_j L'_j(\boldsymbol{\xi}_3) \end{aligned} \quad (5)$$

where $L_1(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\omega})$ is the loss of Eq. 1 (same as pre-training, but now evaluated in the JALI-annotated dataset). The loss weights are set in all our experiments via hold-out validation as follows: $w_a = 0.1, w_v = 0.2, w_j = 0.2, w'_v = 0.15$, and $w'_j = 0.15$. Note that this loss function is not decomposable because the predictions (and in turn, the losses) associated with the viseme branch depend on the predicted phonemes and landmarks of the other two stages during training. Below we describe the individual loss functions in detail.

The loss function $L_a(\xi_1)$ penalizes disagreements between predicted parameter activations $\mathbf{m}_t$ and ground-truth parameter activations $\hat{\mathbf{m}}_t$ for each training frame $t$. Since multiple rig parameters can be active at a given time, this loss function attempts to maximize the probability of correct, individual activations per parameter (or equivalently minimize their negative log-probability). It can be expressed as a sum of $A = 29$ binary cross-entropy losses, one per rig parameter:

$$L_a(\xi_1) = -\frac{1}{N}\frac{1}{A}\sum_{n=1}^{N}\sum_{a=1}^{N}\left(\frac{1}{t_n}\sum_{t=1}^{t_n}[\hat{m}_{a,t}=1]\log P(m_{a,t}=1) \right. \\ \left. \frac{1}{t_n}\sum_{t=1}^{t_n}[\hat{m}_{a,t}=0]\log P(m_{a,t}=0)\right) \quad (6)$$

where $[\hat{m}_{a,t}=1], [\hat{m}_{a,t}=0]$ are binary functions indicating whether the rig parameter $a$ is active or not at frame $t$.

The loss function $L_v(\xi_2)$ measures absolute differences between the training values $\hat{v}_{a,t}$ and predicted values $v_{a,t}$ of each viseme and co-articulation rig parameter $a$ when these are active according to the ground-truth binary activity indicator functions:

$$L_v(\xi_2) = \frac{1}{N}\frac{1}{A}\sum_{n=1}^{N}\sum_{a=1}^{A}\left(\frac{1}{t_{n,a}}\sum_{t=1}^{t_n}[\hat{m}_{a,t}=1]\cdot|v_{a,t}-\hat{v}_{a,t}|\right) \quad (7)$$

where $t_{n,a}$ is the number of frames where the rig parameter $a$ is active per clip $n$ in the ground-truth (i.e., $t_{n,a} = \sum_t[\hat{m}_{a,t}=1]$) An alternative approach would be to evaluate rig parameter differences when these are inactive (i.e., pushing the predictions towards 0 in these cases). However, we found that this degrades the prediction quality of the rig parameters because the network over-focuses on making correct predictions in periods where visemes are inactive. The smoothness loss $L'_v(\xi_2)$ penalizes large derivatives of predicted viseme and co-articulation parameters over time:

$$L'_v(\xi_2) = \frac{1}{N}\frac{1}{A}\sum_{n=1}^{N}\sum_{a=1}^{A}\left(\frac{1}{t_{n,a}}\sum_{t=1}^{t_n}[\hat{m}_{a,t}=1]\cdot|\dot{v}_{a,t}|\right) \quad (8)$$

Finally, the loss function $L_j(\xi_3)$ measures absolute differences between the training values $\hat{\mathbf{y}}_t$ and predicted values $\mathbf{y}_t$ of the 2D JALI viseme field, while $L'_j(\xi_3)$ penalizes large changes in the viseme field values over time:

$$L_j(\xi_3) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{t_n}\sum_{t=1}^{t_n}||\mathbf{y}_t-\hat{\mathbf{y}}_t||_1\right) \quad (9)$$

$$L'_j(\xi_3) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{1}{t_n}\sum_{t=1}^{t_n}||\dot{\mathbf{y}}_t||_1\right) \quad (10)$$

Minimizing the above multi-task loss function is done through batch gradient descent with with batch size 256, learning rate 0.00001, momentum set to 0.9, over 200K iterations.

*Thresholding activation probabilities and hold-out validation.* In the training stage, we also compute a threshold $thr_a$ for each rig parameter $a$ that determines when to activate it based on the rig control activation probability produced by our network, i.e., check $P(m_{a,t} > thr_a)$. One potential choice could be to simply set $thr_a = 0.5$. Yet, we found that optimizing the threshold per rig parameter through a dense grid search in a small hold-out validation dataset (10% our training dataset clips are used for hold-out validation) and selecting the value yielding the best precision and recall in rig activations in that dataset offered better performance. The same hold-out validation set and procedure are used to set the weights of the loss terms in Equations 1 and 5.

*Implementation and running times.* Our network is implemented in Tensorflow. Pre-training takes $30h$ and joint training takes $15h$ in our training datasets measured on a TitanX GPU. At test time, audio feature extraction and network inference (forward propagation) is performed at 100 FPS (10ms per frame) with a lag of 120 ms relative to the current frame (see Section 4, phoneme branch paragraph). Our code for training and testing the network, trained models, datasets, and results are available on our project web page. [1]

## 6 EVALUATION

We evaluated our method and alternatives both quantitatively and qualitatively. In this section, we primarily focus on quantitative evaluation. We refer the reader to the video for qualitative results and comparisons.

*Methodology.* We focused on the BIWI 3D audiovisual dataset to validate our method and alternatives. As mentioned in the previous section, exemplar JALI-based motion curves were provided by an artist for the whole dataset, thus we can compare predicted rig parameters to ground-truth. In addition, each of the 14 speakers of the BIWI dataset speaks the same sentence in both neutral and expressive styles, conveying emotions such as anger, sadness, fear, nervousness, and excitement. Thus, we can compare our method and alternatives on how well they handle different styles of speech.

Because this JALI-annotated dataset has only 14 speakers, we perform the evaluation through a leave-one-out approach: for each of the 14 BIWI speakers, we perform pre-training on our audiovisual dataset (including GRID, SAVEE, and BIWI but excluding that BIWI speaker), and then perform joint training on the JALI-annotated BIWI dataset using the other 13 speakers. As a result, we form 14 training and test splits, where the test splits always involve a speaker not observed during training. Since we aim at learning a speaker-independent, generic model, we believe that this is a more compelling and practically useful generalization scenario, compared to training and testing on the same speaker.

*Quantitative evaluation measures.* The first evaluation measure we use is the rig parameter *activation precision*, which measures how often we activate the right viseme and co-articulation rig parameters based on the ground-truth. Specifically, given a binary variable $\hat{m}_{a,t}$ indicating whether the rig parameter $a$ is activated or not at frame $t$ in the ground-truth, and given our predicted binary variable $m_{a,t}$ for that parameter and frame, the precision is calculated as the number of times we correctly predict activations for the rig parameters (i.e., $\sum_{a,t}[\hat{m}_{a,t}=1 \,\&\, m_{a,t}=1]$, or in other words, the number of true positives) normalized by the total number of predicted activations (i.e., $\sum_{a,t}[m_{a,t}=1]$). We also evaluate the rig parameter *activation recall*, which measures out of all the ground-truth rig parameter activations, what fraction of them we predict correctly. The recall is calculated as the number of times we correctly predict activations for the rig parameters (again, number of true positives) divided by

[1] http://people.umass.edu/~yangzhou/visemenet

Table 1. Precision and recall for activation of rig controls for our full method and degraded versions of it for neutral and expressive speech styles, averaged over all test splits (higher precision and recall are better). We also report the average standard deviation (SD) of precision and recall for each variant.

| | neutral | | | expressive | | |
|---|---|---|---|---|---|---|
| | precision (%) | recall (%) | SD | precision (%) | recall (%) | SD |
| full method | **89.5** | **92.2** | 1.9 | **90.1** | **92.3** | 2.2 |
| landmark-based | 73.6 | 82.0 | 5.1 | 74.4 | 82.7 | 4.9 |
| phoneme-based | 87.8 | 91.5 | 1.8 | 88.2 | 91.6 | 2.0 |
| audio-based | 68.7 | 81.3 | 5.2 | 69.3 | 82.2 | 4.9 |
| no transfer learning | 85.8 | 89.5 | 2.3 | 86.1 | 89.6 | 2.2 |
| no shared weights (LP) | 88.5 | 91.5 | 1.0 | 88.8 | 91.6 | 1.9 |
| shared weights (V) | 89.0 | 91.9 | 1.7 | 89.3 | 91.9 | 1.9 |
| ASR-based | 87.6 | 89.2 | 1.6 | 88.4 | 89.8 | 1.4 |
| GRU-based | 87.4 | 91.2 | 2.1 | 87.8 | 91.1 | 2.3 |
| sliding window-based | 78.1 | 77.8 | 1.5 | 78.6 | 78.2 | 1.5 |

Table 2. Percentage difference of motion curves (including viseme, co-articulation, and JALI field parameters) for our method and degraded versions of our architecture for neutral and expressive styles of speech, averaged over all test splits (lower difference is better). We also report the standard deviation (SD) of the percentage differences for each variant.

| | neutral | | expressive | |
|---|---|---|---|---|
| | motion curve differences (%) | SD | motion curve differences (%) | SD |
| full method | **7.8** | 0.8 | **7.6** | 0.9 |
| landmark-based | 13.6 | 1.8 | 13.2 | 1.5 |
| phoneme-based | 9.0 | 0.7 | 8.8 | 0.7 |
| audio-based | 14.5 | 1.8 | 14.2 | 1.6 |
| no transfer learning | 9.7 | 0.9 | 9.6 | 0.8 |
| no shared weights (LP) | 8.8 | 0.6 | 8.7 | 0.7 |
| shared weights (V) | 9.5 | 0.7 | 9.2 | 0.7 |
| ASR-based | 9.1 | 0.6 | 8.8 | 0.6 |
| GRU-based | 9.1 | 0.7 | 9.0 | 0.8 |
| sliding window-based | 15.4 | 0.3 | 15.2 | 0.4 |

the total number of ground-truth activations $\sum_{a,t}[\hat{m}_{a,t} = 1]$. In the ideal scenario, precision and recall should be both 100%.

We also measure the *motion curve differences*, which evaluates the absolute differences of our predicted rig parameter values (viseme, co-articulation, and JALI field parameters) compared to their ground-truth values averaged over the test frames where the corresponding rig parameters are active either in the ground-truth or in the predictions. We note that we do not consider inactive parameters in the evaluation of motion curve differences because the motion curves are very sparse; zero-values would dominate the measure otherwise. Since all the rig parameters are normalized between $[0, 1]$ during training and testing, these differences can be treated as percentages.

*Quantitative Comparisons.* We compare our network with the following alternative architectures: (a) **landmark-based:** we eliminate the phoneme group stage of our network i.e., visemes are predicted based on landmarks and audio features only, (b) **phoneme-based:** we eliminate the landmark stage of our network i.e., visemes are predicted based on phoneme groups and audio features only, (c) **audio-based:** we eliminate both the landmark and phoneme group stages i.e., visemes are predicted directly from audio features alone, which also implies that there is no pre-training since no training phoneme groups or landmarks can be used in this condition (d) **no transfer learning:** we keep all the stages of our network, yet we train only on the JALI-annotated BIWI training split and not on the rest of the audiovidual datasets, (e) **no shared weights (LP):** we disable weight sharing between the landmark and phoneme stages i.e., the LSTM layers of these two stages have independent parameters in this variant, (f) **shared weights (V):** we force weight sharing between the layers of the three LSTM modules used for predicting rig control activations, viseme/co-articulation parameters, and JALI parameters in the viseme stage (this is in contrast to our proposed architecture that uses LSTMs without shared weights in this stage). (g) **ASR-based:** instead of using our phoneme group prediction part, we pass the input audio through the popular Google Cloud automatic speech recognition engine [Google 2017] to extract text, then extract phonemes based on the MFA forced aligner [McAuliffe et al. 2017], form the phoneme groups of Figure 2, encode them into a binary vector with 1s corresponding to present phoneme groups and 0s for the non-present ones per frame, and pass this vector to our viseme branch (instead of our phoneme group representations). This alternative architecture tests the condition where phonemes are acquired automatically in separate stages through existing off-the-shelf tools, (h) **GRU-based:** we use Gated Recurrent Units (GRUs) [Cho et al. 2014] instead of LSTMs as memory modules, (i) **sliding window-based:** instead of using LSTMs in our three stages, we experimented with the memory-less neural network modules based on three fully connected hidden layers operating on sliding windows, as proposed in [Taylor et al. 2017]. We note that we further benefited this approach by using the same type of inputs in each stage (i.e., the viseme stage receives audio, landmarks, and phonemes as input instead of using phonemes alone as done in [Taylor et al. 2017]) and also using the same pre-training and transfer learning procedure as in our approach (without these enhancements, the performance was worse). We also increased the number of hidden nodes per layer so that the number of learnable parameters is comparable to the one in our architecture (using the original number of hidden nodes also resulted in worse performance).

We trained these alternative architectures based on the same corresponding loss functions in the same training sets as our method, performed the same hyper-parameter tuning procedure as in our method, and evaluated them in the same test splits. Table 1 and Table 2 report the abovementioned evaluation measures for our method and the alternatives for neutral and expressive styles of speech. Our full method offers the best performance in terms of all evaluation measures and different styles of speech. Based on the results, if one attempts to skip our intermediate phoneme group and landmark stages, and predict visemes from audio features directly ("audio-based" condition), then the performance degrades a lot (see also accompanying video and Figure 5 for qualitative comparisons). Skipping the phoneme group stage ("landmark-based" condition) also results in a large performance drop, which indicates that recognizing phoneme groups is crucial for predicting correct visemes, as the psycho-linguistic literature indicates. Skipping landmarks ("phoneme-based" condition) also results in a noticeable drop in performance for both neutral and expressive styles of speech. Using off-the-shelf tools for viseme recognition ("ASR-based" condition) also results in worse performance than our approach. Note also

that this ASR-based approach is language-dependent and requires an input language-based phoneme model specification, while our approach is language-agnostic since our phoneme groups are based on the International Phonetic Alphabet (IPA). Furthermore, we observed that transfer learning and weight sharing in the landmark and phoneme stages improve performance. Using GRUs results in worse performance compared to LSTMs. Finally, replacing the LSTMs with fully connected network modules operating on sliding windows causes a large drop in performance.

*Qualitative comparisons.* Our video shows facial animation results produced by our method and degraded versions of our architecture as well as comparisons with previous works [Karras et al. 2017], [Suwajanakorn et al. 2017], and [Taylor et al. 2017]. Quantitative comparisons with these previous works are not possible because their used test rigs are not FACS-enabled, and their implementation is not publicly available. In contrast to our approach, none of these previous methods produce editable, animator-centric viseme curves or facial action units. We also demonstrate generalization to speech animation involving different languages.

## 7 CONCLUSION

We presented an animator-centric, deep learning approach that maps audio to speech motion curves. There are various avenues for future work. Our implementation currently uses hand-engineered audio features. Replacing them with learned features, similarly to what is done in image and shape processing pipelines, could help improving performance. Another interesting extension would be to incorporate a discriminator network that would attempt to infer the quality of the generated animations, and use its predictions to boost the performance of our viseme generator network, as done in cGAN-based approaches [Isola et al. 2016] for image and shape synthesis. Finally, our method is able to drive only the lower part of the face. Learning to control the upper face e.g., eyes, without explicit supervisory signals would also be a fruitful direction.

The marriage between animator-centric techniques and deep-learning has the potential to fundamentally alter current facial animation practice in film and game studios, leaving animators free to focus on the creative and nuanced aspects of character expression. We believe our solution is a significant step in this direction.

## ACKNOWLEDGEMENTS

We acknowledge support from NSERC and NSF (CHS-1422441, CHS-1617333, IIS-1617917). We thank Pif Edwards and anonymous reviewers for their valuable feedback. Our experiments were performed in the UMass GPU cluster obtained under the Collaborative R&D Fund managed by the Massachusetts Technology Collaborative.

## REFERENCES

Robert Anderson, Björn Stenger, Vincent Wan, and Roberto Cipolla. 2013. Expressive Visual Text-to-Speech Using Active Appearance Models. In *Proc. CVPR*.

Gérard Bailly. 1997. Learning to speak. Sensori-motor control of speech movements. *Speech Communication* 22, 2-3 (1997).

Gérard Bailly, Pascal Perrier, and Eric Vatikiotis-Bateson. 2012. *Audiovisual Speech Processing*. Cambridge University Press.

Christoph Bregler, Michele Covell, and Malcolm Slaney. 1997. Video Rewrite: Driving Visual Speech with Audio. In *Proc. SIGGRAPH*.

Carlos Busso, Sungbok Lee, and Shrikanth S. Narayanan. 2007. Using neutral speech models for emotional speech analysis. In *Proc. InterSpeech*.

Rich Caruana. 1997. Multi-task Learning. *Machine Learning* 28, 1 (1997).

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Y. Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. EMNLP*.

Michael M Cohen and Dominic W Massaro. 1993. Modeling Coarticulation in Synthetic Visual Speech. *Models and Techniques in Computer Animation* 92 (1993).

Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. 2006. An audio-visual corpus for speech perception and automatic speech recognition. *J. Acoustical Society of America* 120, 5 (2006).

Steven B. Davis and Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoustics, Speech and Signal Processing* 28, 4 (1980).

Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. 2016. JALI: An Animator-centric Viseme Model for Expressive Lip Synchronization. *ACM Trans. Graphics.* 35, 4 (2016).

Paul Ekman and Wallace V Friesen. 1978. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press.

Tony Ezzat, Gadi Geiger, and Tomaso Poggio. 2002. Trainable Videorealistic Speech Animation. In *Proc. SIGGRAPH*.

Faceware. 2017. Analyzer. http://facewaretech.com/products/software/analyzer. (2017).

G. Fanelli, J. Gall, H. Romsdorfer, T. Weise, and L. Van Gool. 2010. A 3-D Audio-Visual Corpus of Affective Communication. *IEEE Trans. Multimedia* 12, 6 (2010).

Cletus G Fisher. 1968. Confusions among visually perceived consonants. *J. Speech, Language, and Hearing Research* 11, 4 (1968).

Jennifer MB Fugate. 2013. Categorical perception for emotional faces. *Emotion Review* 5, 1 (2013).

Google. 2017. Google Cloud Voice. https://cloud.google.com/speech. (2017).

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. ICML*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5 (2005).

S. Haq and P.J.B. Jackson. 2009. Speaker-dependent audio-visual emotion recognition. In *Proc. AVSP*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997).

Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017. Avatar Digitization from a Single Image for Real-time Rendering. *ACM Trans. Graphics.* 36, 6 (2017).

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2016. Image-to-Image Translation with Conditional Adversarial Networks. *arxiv* abs/1611.07004 (2016).

Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. 2017. Audio-driven Facial Animation by Joint End-to-end Learning of Pose and Emotion. *ACM Trans. Graphics.* 36, 4 (2017).

Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *J. of Machine Learning Research* 10, Jul (2009).

Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. 2013. Realtime Facial Animation with on-the-Fly Correctives. *ACM Trans. Graphics* 32, 4 (2013).

Alvin M Liberman, Katherine Safford Harris, Howard S Hoffman, and Belver C Griffith. 1957. The discrimination of speech sounds within and across phoneme boundaries. *J. Experimental Psychology* 54, 5 (1957).

Karl F. MacDorman, Robert D. Green, Chin-Chang Ho, and Clinton T. Koch. 2009. Too real for comfort? Uncanny responses to computer generated faces. *Computers in Human Behavior* 25, 3 (2009).

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal Forced Aligner: trainable text-speech alignment using Kaldi. In *Proc. Interspeech*.

Christopher Olah. 2015. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs. (2015).

Kuldip K Paliwal. 1998. Spectral subband centroid features for speech recognition. In *Proc. ICASSP*.

Roger Blanco i Ribera, Eduard Zell, J. P. Lewis, Junyong Noh, and Mario Botsch. 2017. Facial Retargeting with Automatic Range of Motion Alignment. *ACM Trans. Graphics.* 36, 4 (2017).

Supasorn Suwajanakorn, Steven M. Seitz, and I. Kemelmacher-Shlizerman. 2017. Synthesizing Obama: Learning Lip Sync from Audio. *ACM Trans. Graphics* 36, 4 (2017).

Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. 2017. A Deep Learning Approach for Generalized Speech Animation. *ACM Trans. Graphics.* 36, 4 (2017).

Sarah L. Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. 2012. Dynamic Units of Visual Speech. In *Proc. SCA*.

Lijuan Wang, Wei Han, and Frank K Soong. 2012. High Quality Lip-Sync Animation for 3D Photo-Realistic Talking Head. In *Proc. ICASSP*.

Wenwu Wang. 2010. *Machine Audition: Principles, Algorithms and Systems: Principles, Algorithms and Systems.* IGI Global.

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. 2011. Realtime performance-based facial animation. In *ACM Trans. Graphics*, Vol. 30.

Lance Williams. 1990. Performance-driven Facial Animation. In *Proc. SIGGRAPH*.