# LipSync3D: Data-Efficient Learning of Personalized 3D Talking Faces from Video using Pose and Lighting Normalization (Supplementary)

Avisek Lahiri[1,2*†] Vivek Kwatra[1*] Christian Frueh[1*] John Lewis[1] Chris Bregler[1]
[1]Google Research [2]Indian Institute of Technology Kharagpur
{avisek,kwatra,frueh,jplewis,bregler}@google.com

## 1. User Study Analysis

We conducted a user study to quantitatively compare our lip-sync and perceptual quality against the state-of-the-art audio-driven frameworks of Wav2Lip [9], NVP [11], IJCV'19 [13] and TBE [3]. In the study, N=35 raters were each shown a total 29 sample clips consisting of synthetic and real videos. For competing methods, we used their released videos (NVP, TBE) or generated results with their pre-trained models (IJCV'19, Wav2Lip). The raters were primarily drawn from a pool of subjects without research expertise, supplemented with a minority (N=14) who were researchers. The subgroup of researchers included some having familiarity with computer vision topics but none were expert on speech-driven animation. The raters were asked three questions: **Q1:** *Is the video real or fake?* **Q2:** *Rate the quality of the lip-sync, i.e. how well does the motion of the lips match the audio, on a 3-point (discrete) scale (poor, acceptable, great).* **Q3:** *Rate the picture quality, e.g. naturalness, resolution, and consistency of the video, on a discrete 5-point scale from 1-5 (poor-great).*

Figure 2 shows, for each question, the percentage of raters who selected each rating. We report the Mean Opinion Scores (MOS) of the questions in Table 1. As is evident, among the competing methods, our method receives the most favorable user ratings.

We performed a statistical analysis to confirm the significance of these ratings. For Q1 (only) we excluded the text-to-speech results from consideration, since it was straightforward to judge the videos as "fake" due to the computer-generated speech. Questions Q2 and Q3 are still relevant in the text-to-speech case, however, since it is possible to rate the quality of lip-sync and overall image naturalness even when the voice is clearly synthetic.

The statistical analysis confirms that the differences in real-fake ratings on Q1 are significant (Kruskal-Wallis test, $\chi^2 = 158$, p<1e-04), and our method outperforms the other methods after adjusting for multiple comparisons (Tukey's Honest Significant Differences (HSD) IJCV p adj.= .003, Wav2Lip p adj.<1e-04, NVP p adj.<1e-04). For Q2 the differences in ratings are significant (Kruskal-Wallis $\chi^2 = 279$, p<1e-04), and our method outperforms most competing methods with statistical significance after adjusting for the multiple tests (Tukey's HSD TBE p adj.= .035, Wav2Lip p adj.<1e-04, NVP p adj.<1e-04), however the difference versus IJCV'19 is not significant. For Q3 (Kruskal-Wallis $\chi^2 = 248$, p<1e-04) our method outperforms most competing methods with statistical significance after adjustment for multiple comparison (HSD IJCV p adj.<1e-04, Wav2lip p adj.<1e-04, NVP p adj.= 0.04 however the comparison with TBE is not significant.

| Method | Is Real ? (% Yes) | [no TTS] Is Real ? (% Yes) | Lip-Sync (1-3) ) | Visual Quality (1-5) |
|---|---|---|---|---|
| Real | 97.6 | 97.6 | 2.95±0.03 | 4.55±0.13 |
| IJCV'19 | 60.7 | 60.7 | 2.45±0.11 | 2.49±0.17 |
| Wav2Lip | 34.4 | 37.6 | 1.72±0.12 | 3.35±0.20 |
| NVP | 44.4 | 50.0 | 1.80±0.13 | 3.75±0.19 |
| TBE | 50.7 | n/a | 2.17±0.17 | 4.07±0.26 |
| **Proposed** | **71.6** | **77.25** | **2.46±0.10** | **4.10±0.15** |

Table 1: User study analysis. Column 1: percentage of "real" ratings by category. Column 2: percentage of "real" ratings with text-to-speech driven results removed. Column 3: mean opinion score of lip-sync quality. Column 4: mean opinion score of picture quality.

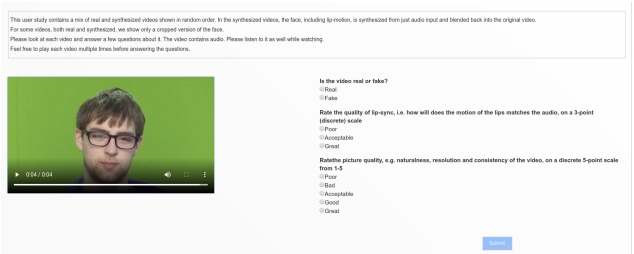

Figure 1: Screen shot from our user study.

These significance results for Q2 and Q3 (and in particular the lack of significance for IJCV and TBE respectively) are plausible given cursory examination of the videos. The results of IJCV'19 show qood quality lip-sync but the over-

*Equal contribution
†Work done while an intern at Google

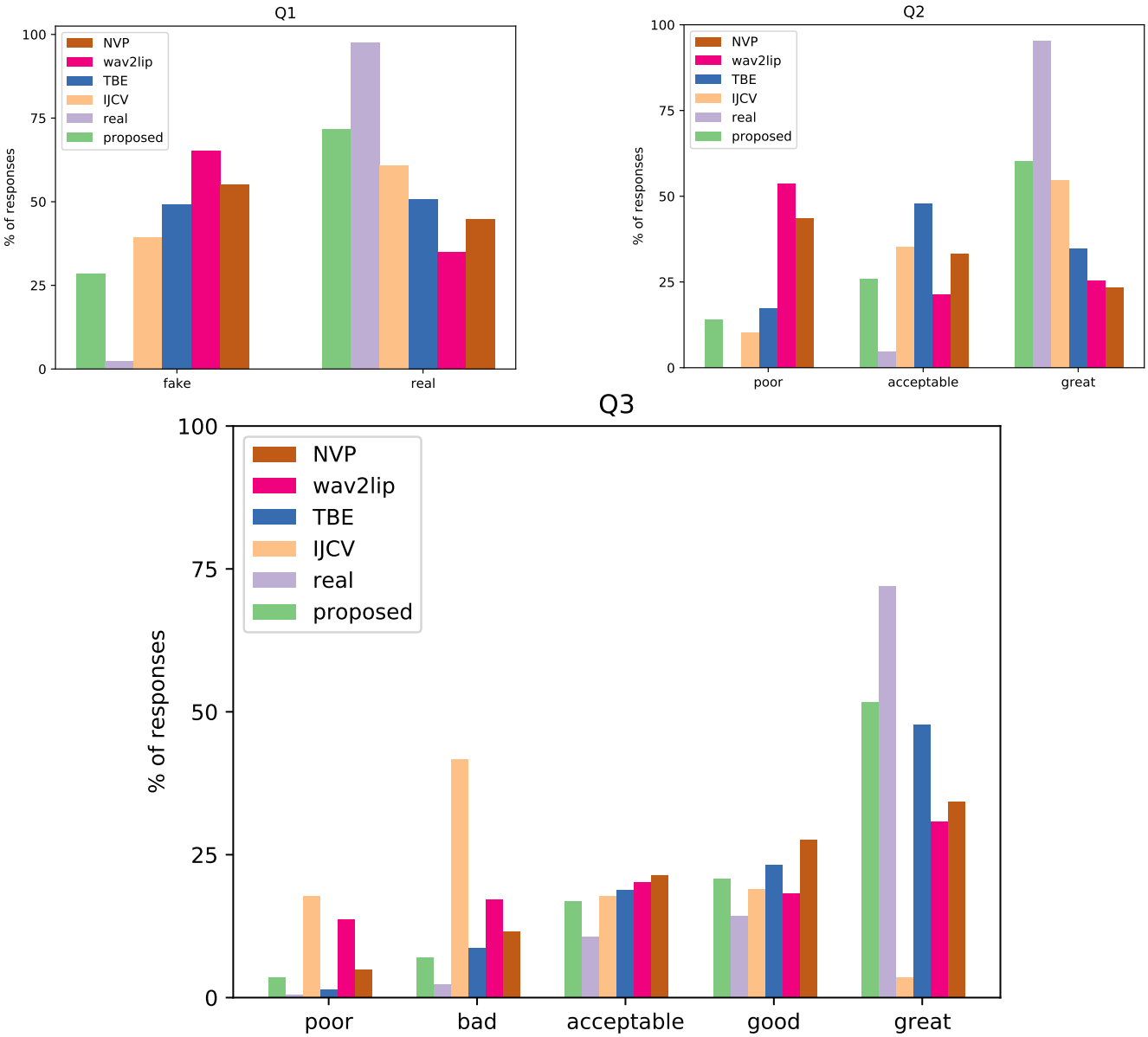


Figure 2: Raw user study results. **Q1**: Percentage of real/fake ratings for each of the six video categories. Viewers believe our synthetic videos (proposed) are real roughly two-thirds of the time. **Q2**: Ratings of lip-sync quality for the six video categories, expressed as percentages. Our synthetic videos (proposed) are perceived as at least comparable to those of IJCV while being clearly superior to other competing methods in lip-sync quality. **Q3**: Ratings of picture quality for the six video categories, expressed as percentages. Here our method greatly outperforms IJCV (and NVP and wav2lip) while being at least comparable to TBE. It can be seen that our method is the best performer across the three questions.

all image quality is limited, thus explaining its good performance on Q2 but poor performance on Q3. TBE operates in part by re-mixing input video frames so it results in high picture quality by definition (Q3), but its lip-sync quality is poorer than our method and that of IJCV'19.

## 2. Comparison Notes on Text-based-Editing [3]

According to the user study, among the 3D model based methods, Text-based-Editing (TBE) is the second-best method (following our method). However, our framework has some distinct training and inference time advantages over TBE:

- TBE was trained on a training corpus of more than 1

hour of video recording. Our model was trained on ∼7 minutes of data in this case.

- TBE assumes an accurate text transcript and uses phoneme based alignment tools to align the text with audio. In contrast, our model only requires a speech signal as input.
- The average training time of TBE is 42 hours. Our typical training time is somewhere in between 3-5 hours.
- The inference speed of TBE is significantly slower. This is mainly attributed to the costly viseme search (∼5 minutes for 3 words). Our method executes within a few tens of milliseconds.
- TBE also relies on neural rendering for learning to generate facial texture based on the illumination in the training sequence. It is thus not apt for operating under new ambient lighting without further retraining. Our framework is capable of seamlessly adapting to novel lighting conditions during inference.

## 3. Restrictions on comparison with Suwajanakorn *et al.* [10]

The work by Suwajanakorn *et al.* also involves training a personalized talking face model. However, [10] only synthesized results for a single person (the former U.S. president Barak Obama), using hours of training video. While our model is perfectly capable of similar synthesis, due to our organization's ethical/legal constraints we cannot train on any living political personality. On this ground, we are unable to directly compare our work with that of Suwajanakorn *et al.*.

Although we cannot visually compare the two methods, our framework offers the following advantages:

- Suwajanakorn *et al.* trained on 14 hours of weekly President addresses recorded between 2009-2016. In contrast, our framework just requires ∼5 minutes of training video.
- Suwajanakorn *et al.* demonstrate their outputs only under the specific studio lighting setup of the President's office. Their texture generation network is not designed with the goal of handling diverse ambient lighting. In contrast, our network seamlessly disentangles and normalizes the effects of illumination, thereby enabling inference under diverse lighting conditions.
- Typical training data pre-processing time of Suwajanakorn *et al.* is around 2 weeks on 10 cluster nodes of Intel Xeon E5530. In contrast our combined preprocessing and training takes only about 3-5 hours on a single system equipped with a single NVIDIA P1000 GPU.

## 4. Discussion on LSE metrics

We have used the official code release [1] by the authors of Wav2Lip for evaluating the automated LSE metrics, LSE-D (lower is better) and LSE-C (higher is better). We faced two issues while using this metric:

**(a:)** Even though user study suggest that the lip-sync quality of Wav2Lip is usually inferior to our model, the LSE metrics are always better for Wav2Lip. We observed this pattern over a range of different videos. LSE metrics are computed from paired audio-visual representation coming from a SyncNet [1] network. Wav2Lip also leverages a SyncNet architecture and audio-visual representations as a lip-sync loss during training. We hypothesize that since a similar architecture is used both as a training loss and for scoring, Wav2Lip may be biased to do particularly well on this metric. A short communication with the authors also revealed that they share a similar hypothesis. Thus, we refrained from reporting LSE metrics for Wav2Lip. However, for other frameworks, the metric yields numbers consistent with human evaluations of lip-sync.

**(b:)** The code sometimes fails to detect faces even under normal illumination and thereby does not give LSE metrics. So, we could not report LSE metrics on all comparing videos.

## 5. Leaving out Wav2Lip for Self-reenactment Comparisons

In main paper, while comparing (Figure 10 ) methods for self-reenactment tasks, we did not include Wav2Lip among the competing methods. Along with the current audio, Wav2Lip also feeds in the sequence of target frames with the lip region unmasked. Since this is a self-reenactment setting, the input target frames are the final expected output from the network. So, Wav2Lip has an unfair advantage over our method since our framework is entirely audio driven. The masked (around lip region) target frames are just utilized for pasting back synthesized lip region. Thereby, we do not compare Wav2Lip for self-reenactment. A similar advantage is also available to LipGAN [8] which is a precursor to the framework of Wav2Lip.

## 6. Selecting Code Length for Audio and Previous Atlas

In this section we report studies to determine the code length for encoding the current time step's spectrogram and previous time step's predicted atlas. Since we wish to automatically determine acceptable settings of these parameters, the study was conducted for a self-reenactment task in which we have access to ground truth facial information.

[1] https://github.com/Rudrabha/Wav2Lip

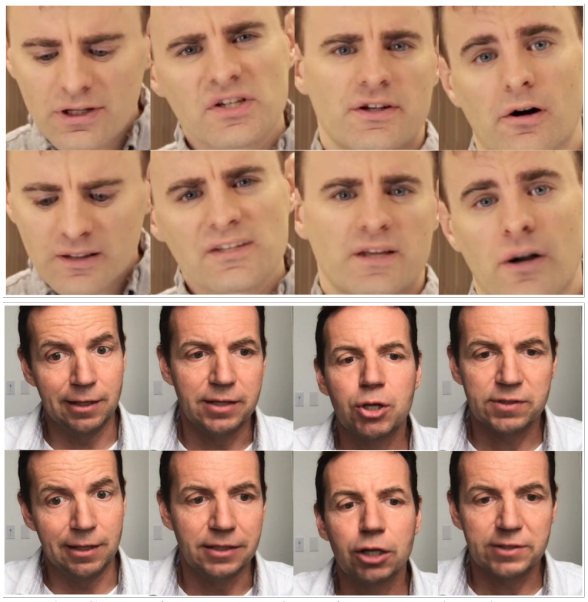

Figure 3: Comparing our result against ground-truth. For each subject, top row is the original sequence of frames, while the bottom row is the resynthesized sequence.

For this study, we curated a custom dataset of subjects selected from YouTube instructional videos, webcam recordings, and studio conversations. The custom dataset had around 10,000 audio-synchronized frames. Sample frames and corresponding reconstructions from two such subjects are shown in Figure 3.

### 6.1. Selection of Previous Atlas Code Length, $N_a$

We conducted an ablation study to determine the length $N_a$ of the latent code $L^A_{t-1}$. The code length governs the contribution of the previous visual state to the current frame. With increasing code length, the model starts to incorrectly neglect the current audio input, instead basing its output mostly on the previous state. In Table 2 we report the average metrics over different $N_a$. Note that $N_a = 0$ signifies a model trained without auto-regression. Both SSIM and LMD improve when using auto-regression initially but deteriorate as we increase $N_a$, with $N_a = 2$ giving the best results.

### 6.2. Selection of Audio Code Length, $N_S$

As mentioned earlier, we used a 32-dimensional vector for encoding the audio spectrogram. This number was chosen by performing a parameter sweep over $N_S \in \{8, 16, 32, 64, 128\}$ on our custom dataset. In Table 3 we compare the SSIM and LMD metrics, averaged over the subjects in the dataset. We observe that $N_S = 32$ yields the most favorable performance. Hence, we use it as our default choice when encoding the spectrogram.

| Metric | ← AR Code Length: $N_a$ → | | | |
|---|---|---|---|---|
| | 0 | 2 | 8 | 16 |
| SSIM | 0.92 | **0.93** | 0.901 | 0.889 |
| LMD | 2.00 | **1.88** | 2.91 | 2.16 |

Table 2: Parameter sweep for selecting latent code lengths, $N_a$, for encoding previous time step atlas based on LMD (lower is better) and SSIM (higher is better) metrics. Best results are marked in bold.

| | ← Audio Code Length: $N_S$ → | | | | |
|---|---|---|---|---|---|
| Metric | 8 | 16 | 32 | 64 | 128 |
| LMD | 1.83 | 1.87 | **1.77** | 1.81 | 1.82 |
| SSIM | 0.907 | 0.914 | **0.917** | 0.917 | 0.910 |

Table 3: Parameter sweep for selecting audio latent code length based on LMD and SSIM metrics. Best results are marked in bold.

## 7. Network Architectures

In the main paper, we describe the architectures of the audio encoder (which computes the latent code from audio spectrograms) and the geometry decoder (that computes the 3D vertices from the audio latent code). Here, we describe the additional network components of our model.

### 7.1. Texture Decoder Architecture

In Table 4 we present details of the texture decoder. The input to the decoder is either a 32D vector (conditioned on only audio latent code), or 34D (conditioned on audio latent code + previous time step atlas latent code). This is followed by a series of convolution and upsampling layers.

### 7.2. Auto-regressive Encoder Architecture

In our auto-regressive architecture, the previous atlas at the previous time step is encoded as an additional latent vector (along with the audio encoded vector). The output is a latent vector of length $N_a = 2$. The encoder architecture for the autoregressive model is shown in Table 5.

**Training by "Teacher Forcing":** As stated in the main paper, during training we do not provide the actual previous predicted atlas as an additional input to the auto-regressive model, since this would entail a recursion in the network. Instead we follow the *Teacher Forcing* [14] paradigm of training the network with ground truth previous atlas.

In our initial experiments, we implemented a recursive network and fed in the actual predicted previous atlas to the model during training. The reconstruction quality of that approach was worse than using ground truth atlases during training (*i.e.* Teacher Forcing), however. Note that

during inference, the *predicted* previous atlas is fed to the model, because the ground truth atlas is not known at that time. Also, for predicting the first frame, we provide an *'all-zeros'* image as a proxy for previous frame because there is no previous frame to start with. To handle this case, we train the model by feeding it with *'all-zeros'* for the previous atlas with a probability of 20%. This trains the model to reconstruct the atlas both with and without the knowledge of previous time step's visual state.

## 8. Subject Details: GRID, CREMA-D and TCD-TIMIT

For self-reenactment studies we performed experiments on GRID [2], TCD TMIT [4] and CREMA-D [6] datasets. Following the exact setting in [13], we select the same set of 10 subjects (see Table 6) from each of the datasets).

## 9. Sharpness of Synthesized Lip Region

In main paper, we mentioned that our method is capable of generating high quality lip-sync, and we objectively established this with the commonly used CPBD metric. The metric was evaluated on the entire face. While it is true that the sharpness of the final composite full face is a primary factor of photo-realism, we also acknowledge that our method benefits from copying the texture from upper part of face from target frames.

Here we focus on determining the sharpness of only the lower half of the face (below the nostrils). In Table 7, we compare against Wav2Lip, LipGAN, NVP and TBE on the user study videos. Even on the lower mouth region, our method manifests better CPBD score across all competing methods.

## 10. Applications

Our approach of generating textured 3D geometry enables us to address a broader variety of applications than purely image-based or 3D-only techniques, as discussed here. Sample screenshots from some of these applications are shown in Figure 4.

**3D Talking Avatars:** 3D avatars can make multiplayer online games and Virtual Reality (VR) environments more social and engaging. They may also be employed for audio/video chat applications and virtual visual assistants. While such avatars can be driven by a video feed from a web-cam or head-mounted camera, the ability to generate a 3D talking face from just audio obviates the need for any auxiliary camera device, and also helps preserve privacy, while reducing bandwidth requirements at the same time. Our technique supports generating both 3D textured faces as well as CGI avatars for these applications.

**Video creation and editing:** Our approach can be used for editing videos, *e.g.* to insert new content in an online course, or to correct an error without the cumbersome and sometimes impossible procedure of re-shooting the whole video under original conditions. Instead, a new audio transcript may be recorded for the edited portion, followed by applying our synthesis technique to modify the corresponding video segment. Such a speech-to-video or text-to-video system may be useful in multiple domains such as education, advertising and entertainment. We can also generate cartoon renderings for these videos, which may be preferred in some applications, *e.g.* sketch videos, stylized animations, or assistive technologies such as pronunciation visualization.

**Video translation and dubbing:** Even though we train our models on videos in a single language, they are surprisingly robust to both different languages as well as text-to-speech (TTS) audio at inference time. Using available transcripts or a speech recognition system to obtain captions, and subsequently a text-to-speech system to generate audio, we can automatically translate and lip-sync existing videos into different languages. In conjunction with appropriate video re-timing and voice-cloning [5], the resulting videos look fairly convincing. We have employed our approach for translating and dubbing videos from English to Spanish or Mandarin, and vice-versa. Notably, in contrast to narrator-driven techniques [7, 12], our approach for video dubbing does not require a human actor in the loop, and is therefore more scalable across languages.

## References


[1] J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016. 3

[2] Martin Cooke, Jon Barker, Stuart Cunningham, and Xu Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 120(5):2421–2424, November 2006. 5

[3] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 1, 2

[4] Naomi Harte and Eoin Gillen. TCD-TIMIT: An audio-visual corpus of continuous speech. *IEEE Transactions on Multimedia*, 17(5):603–615, 2015. 5

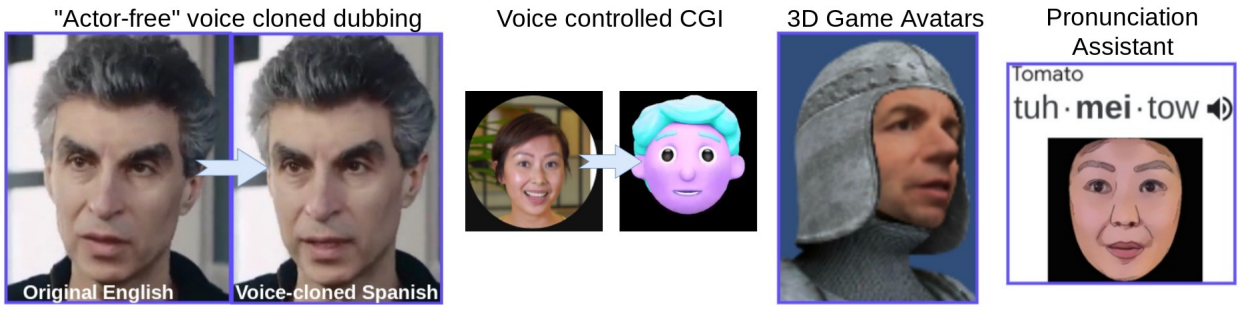

Figure 4: Sample applications enabled by our 3D talking face generation pipeline.


[5] Wei-Ning Hsu, Yu Zhang, Ron Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, Patrick Nguyen, and Ruoming Pang. Hierarchical generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*, 2019. 5

[6] Michael K Keutmann, Samantha L Moore, Adam Savitt, and Ruben C Gur. Generating an item pool for translational social cognition research: methodology and initial validation. *Behavior research methods*, 47(1):228–234, 2015. 5

[7] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, N. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. Deep Video Portraits. *ACM Transactions on Graphics 2018 (TOG)*, 2018. 5

[8] Prajwal KR, Rudrabha Mukhopadhyay, Jerin Philip, Abhishek Jha, Vinay Namboodiri, and CV Jawahar. Towards automatic face-to-face translation. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1428–1436, 2019. 3

[9] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 484–492, 2020. 1

[10] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing Obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017. 3

[11] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *ECCV*, pages 716–731. Springer, 2020. 1

[12] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *CVPR*, pages 2387–2395, 2016. 5

[13] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with GANs. *IJCV*, pages 1–16, 2019. 1, 5

[14] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989. 4

| Input | Type | Kernel | Stride | Channels | Outputs |
|---|---|---|---|---|---|
| Input (Latent Vector): | | | | | |
| = 32D (only audio) | | | | | |
| = 34D (Audio + AutoRegressive) | | | | | |
| Latent Vector | FC | - | - | - | 16384 |
| Reshape: 4×4×1024 | | | | | |
| 2× Bilinear Upsample | | | | | |
| 8×8×1024 | Conv | 3×3 | 1×1 | 512 | 8×8×512 |
| 2× Bilinear Upsample | | | | | |
| 16×16×512 | Conv | 3×3 | 1×1 | 256 | 16×16×256 |
| 2× Bilinear Upsample | | | | | |
| 32×32×256 | Conv | 3×3 | 1×1 | 128 | 32×32×128 |
| 2× Bilinear Upsample | | | | | |
| 64×64×128 | Conv | 3×3 | 1×1 | 64 | 64×64×64 |
| 2× Bilinear Upsample | | | | | |
| 128×128×64 | Conv | 5×5 | 1×1 | 3 | 128×128×3 |

Table 4: Architecture of the texture decoder. Each fully connected (FC) and convolution layer is followed by a ReLU non-linearity, while only the last convolution layer is followed by a tanh non-linearity. The length of the input latent vector depends on the mode of the experiment.

| Input | Type | Kernel | Stride | Channels | Outputs |
|---|---|---|---|---|---|
| Input (RGB): | | | | | |
| =128×128×3 | | | | | |
| Input | Conv | 5×5 | 2×2 | 128 | 64×64×128 |
| 64×64×128 | Conv | 5×5 | 2×2 | 256 | 32×32×256 |
| 32×32×256 | Conv | 5×5 | 2×2 | 512 | 16×16×512 |
| 16×16×512 | Conv | 5×5 | 2×2 | 1024 | 8×8×1024 |
| 8×8×1024 | Conv | 5×5 | 2×2 | 2048 | 4×4×2048 |
| 4×4×2048 | FC | - | - | - | 2 |

Table 5: Architecture of the encoder for the previous atlas in auto-regressive mode. The encoder input is an RGB image and output is a latent vector. Each convolution layer is followed by a ReLU non-linearity. The last fully-connected (FC) layer is followed by a tanh non-linearity.

| Dataset | Test Subject ID |
|---|---|
| GRID | 2, 4, 11, 13, 15, 18, 19, 25, 31, 33 |
| TCD-TMIT | 8, 9, 15, 18, 25, 28, 33, 41, 55, 56 |
| CREMA-D | 15, 20, 21, 30, 33, 52, 62, 81, 82, 89 |

Table 6: IDs of subjects used for self-reenactment experiment.

| LipGAN | Proposed | Wav2Lip | Proposed | TBE | Proposed | NVP | Proposed |
|---|---|---|---|---|---|---|---|
| 0.07 | 0.14 | 0.06 | 0.13 | 0.12 | 0.18 | 0.10 | 0.18 |

Table 7: Comparing CPBD metrics (on lower half of faces) of competing methods against our proposed method. For each method, we select common pairs of videos for the competing and our method from the pool of user study videos.