Unsupervised Any-to-Many Audiovisual Synthesis via Exemplar Autoencoders

Kangle Deng*[1,2] Aayush Bansal[1] Deva Ramanan[1]

[1] Carnegie Mellon University [2] Peking University

Abstract

We present an unsupervised approach that enables us to convert the speech input of any one individual to an output set of potentially-infinitely many speakers, i.e., one can stand in front of a mic and be able to make their favorite celebrity say the same words. Our approach builds on simple autoencoders that project out-of-sample data to the distribution of the training set (motivated by PCA/linear autoencoders). We use an exemplar autoencoder to learn the voice and specific style (emotions and ambiance) of a target speaker. In contrast to existing methods, the proposed approach can be easily extended to an arbitrarily large number of speakers in a very little time using only two-three minutes of audio data from a speaker. We also exhibit the usefulness of our approach for generating video from audio signals and vice-versa. We suggest the reader to check out our project webpage for various synthesized examples: https://dunbar12138.github.io/projectpage/Audiovisual.

1. Introduction

We tackle *any*-to-*many* audiovisual translation that enables anyone to generate the voice and image stream of a known speaker. We focus primarily on audio synthesis, but also present results for joint audio-video generation. Importantly, our approach allows the synthesized data to capture subtle properties of the target speaker including: (1) the scene-context, such as the ambient appearance and acoustics of the environment (e.g., conference room, seminar hall, or large public conventions); and (2) stylistic prosody of the particular speech (e.g., a "happy" vs "angry" delivery). Surprisingly, we show that one can obtain state-of-the-art results with purely data-driven unsupervised methods based on *exemplar auto-encoders*.

This technology enables a wide range of applications in the entertainment industry. We can now create movies and documentaries about historical figures in their voice. We can

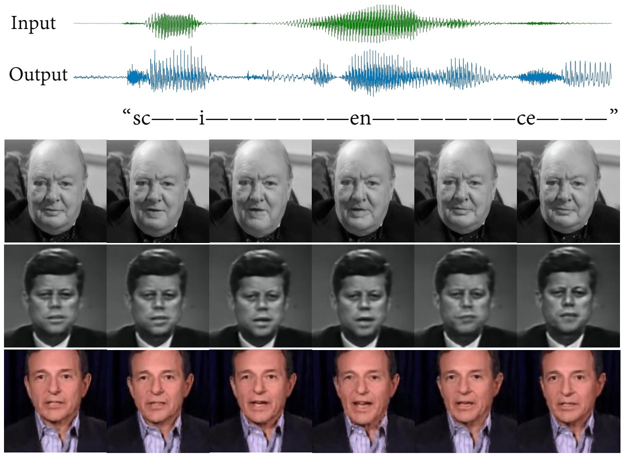


Figure 1. Our approach translates the input audio from *any speaker* to that of a list of *many known* speakers. We show an example of audio translation in the **top** half. With little modification, our approach can also generate video alongside audio from an input audio signal. The **second** half shows a variety of facial examples created using our method for audio-video generation. The spoken word is also presented to show the consistency between the generated lip motion and the speech content.

generate the sound of old actors who are no longer able to perform in contemporary film. This work can also be useful to help those who have lost their voice [19, 27], or provide sound to the ones who never had [16]. It also enables us to create interactive lessons, and personalized voice mails to deliver a voice message. Figure 1 shows example results of audio-video synthesis using our approach.

Audio translation: Earlier works [7, 28, 42, 45] require *parallel* audio data to learn translation between two people. This setup can convert only one known voice to another known voice. Recent work like CycleGAN-VC [22] and StarGan-VC [23] have started to explore learning from *non-parallel* audio data. These approaches (including [8, 10, 20, 21, 22, 23, 39]) restrict themselves to known inputs at test time. More recently, Qian et al. [36] proposed Auto-VC as a zero-shot audio translation system capable of translating any source speaker to any target speaker. Unlike

* Work done as an intern at Carnegie Mellon University.

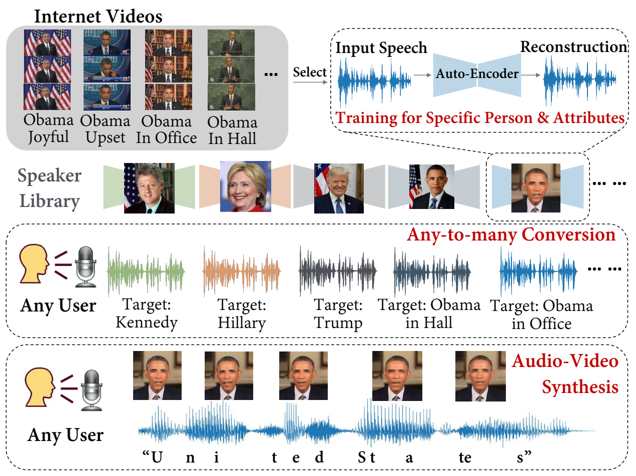


Figure 2. **Overview:** Our approach builds on simple auto-encoders. We train person-specific models using *in-the-wild* web audio. At test time, a user can input speech from anyone (represented using mic), and be able to generate the audio output from a list of many-known speakers (shown in top-row). With the same design principles, our approach can also generate video alongside audio from a speech signal.

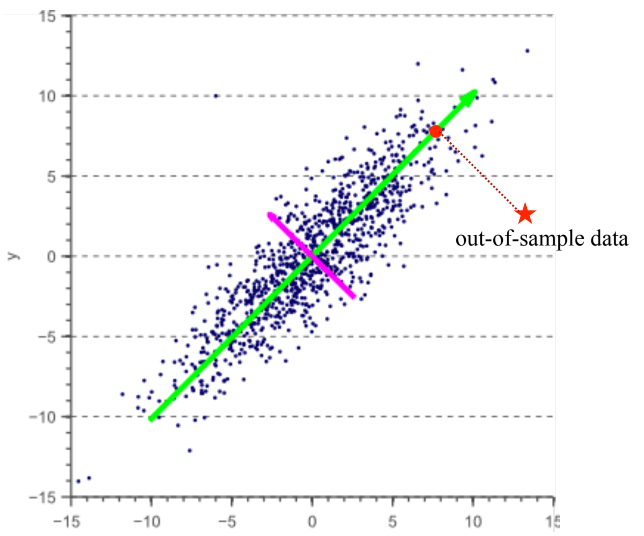


Figure 3. **Linear (exemplar) autoencoders:** We provide visual intuition of why linear autoencoders with *sufficiently small bottlenecks* should produce reasonable reconstructions (the red dot) when applied to out-of-sample data (the red star). Our analysis stems from the well-known result that linear autoencoders can be learned with principle component analysis (PCA) [1]. The above PCA visualization demonstrates that weights of the linear autoencoder – visualized as eigenvectors drawn in green – capture the dataset-specific *style* (properties common to all samples from a particular speech dataset). On the other hand, bottleneck activations – visualize as the projected coordinates of each sample – capture the sample-specific *content* (properties that capture individual differences between samples). It is straightforward to prove that a linear autocoder is guaranteed to produce the minimum error reconstruction of an out-of-sample datapoint given the style subspace spanned by the input dataset.

our approach, AutoVC learns a supervised audio embedding from training data of K individuals labeled with their identity. We experimentally find that such supervised representations struggle to operate on in-the-wild examples that differ from the training set, and struggle to capture subtle stylistic and ambient speaker information.

Audio-to-video translation: We also examine audio to video translation. Prior work [13, 41, 44] has primarily relied on intermediate human keypoints [6]. In this work, we generate both audio and videos directly from audio input with a generic, unsupervised learning-based framework (that does not make use of supervised intermediates like keypoints).

Approach: Our core approach is remarkably simple: given a target speaker with style and ambient environment represented as an audiovisual stream, we learn an *autoencoder specific to that target stream*. Given any input stream, we translate it into the target simply by passing it through the target audioencoder. Because we learn autoencoders specific to each individual target stream, we deem our algorithmic approach as "exemplar autoencoders". We find best results by first learning an autoencoder over the target audio stream, and then learning a video decoder on the learned audio embeddings. Figure 2 provides an overview of our approach.

Why does it work? At first glance, the philosophy behind our approach may seem quite strange. Why should a model trained exclusively on one specific person (and even one specific style) produce reasonable output when applied to a very different input? Indeed, the ability of a model to generalize to different kinds of input remains one of the fundamental and elusive goals in all of machine learning. We provide two intuitions that rely on rather remarkable but under-appreciated properties of autoencoders and their ability to generalize. First, in the linear case, it is well known that linear autoencoders are equivalent to PCA [1]. In this setting, it is straightforward to prove that linear autoencoders will produce the best reconstruction of any input datapoint, in terms of squared error from the subspace spanned by the training dataset [4] (Figure 3). This makes them particularly helpful for translation tasks where one wishes to mimic the style common to a particular target dataset. Secondly, in the specific case of audio autoencoders, we exploit the fact that linguistic phonemes tend to cluster quite well (Figure 4). This suggests that code books learned by autoencoders tend to capture linguistic content (or words) that generalize across different speakers and styles. In the supplemental, we provide a formal proof.

Contributions: (1) We demonstrate exemplar autoencoders for any-to-many audio synthesis. Our approach can be used as an off-the-shelf plug and play tool for target-

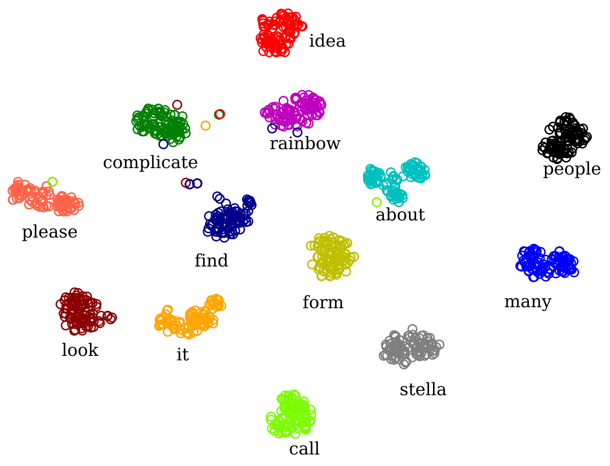


Figure 4. **Our insights for a person-specific auto-encoder:** We observe acoustic features (MEL spectrogram) of the words in the two selected sentences spoken by 100 different speakers in VCTK dataset [47]. We find that this representation enables us to easily cluster the words/content irrespective of who said it. Given a spoken word w by anyone, the closest point in the target speech space should be the same word but in the target-specific style.

specific voice conversion. A live public demo is available on our project webpage. (2) We move beyond well-curated datasets and work with in-the-wild web audios in this paper. (3) Finally, we demonstrate that person-specific audio embeddings are also useful in audiovisual synthesis.

2. Background

A tremendous interest in audio-video generation for health-care, quality-of-life improvement, educational, and entertainment purposes has influenced a wide variety of work in audio, natural language processing, computer vision, and graphics literature. In this work, we seek to explore a standard representation for a user-controllable any-to-many audio translation that can be easily extended to visual tasks.

Speech Synthesis & Voice Conversion: Earlier works [15, 50] in speech synthesis use text inputs to create Text-to-Speech (TTS) systems. Sequence-to-sequence (Seq2seq) structures [43] have led to significant advancements in TTS systems [30, 40, 48]. Recent works [18] have extended these models to incorporate multiple speakers. These approaches enable audio conversion by generating text from the input via a speech-to-text (STT), and use a TTS for target audio. Despite enormous progress in building the TTS systems, it is not trivial to embody perfect emotion and prosody due to the limited expressiveness of bare text [34]. In this work, we seek the problem of voice conversion from an input speech directly to enable a user to capture various nuances.

Audio-to-Audio Conversion: The problem on audio-to-audio conversion has largely been confined to a one-to-one translation, be it using a paired [7, 28, 42, 45] or unpaired [21, 22, 23, 39] data setup. Recent works [35, 36] have begun to explore any-to-any translation with a goal to input any arbitrary voice and any target speaker. These approaches use an autoencoder along with a speaker embedding to generate voice of a target speaker. Our work builds on their observation. In this work, we observe that representation of a target speaker via a low-dimensional embedding is not able to capture stylistic attributes of speech, such as joyful, or sad, in a press conference, or a telephone. We can, however, capture these subtle but important aspects via an exemplar autoencoder that is trained for a specific target. Importantly, a general autoencoder cannot be easily extended to videos due to a highly unstructured space. Exemplar autoencoders, on the other hand, can be easily extended to videos and other modalities.

Audio-Video Synthesis: There is a growing interest [9, 26, 29] in computer vision community to jointly study audio and video for better recognition [24], localizing sound [12, 38], or learning better visual representation [32, 33]. Closely related to ours is the work [13, 41, 44, 49] on synthesizing videos (talking-heads) from an audio signal. These works use human keypoints [6] as an intermediate to generate realistic video outputs. In this work, our goal is to synthesize both audio and video from an input audio signal without an intermediate. Our analysis on audio also finds application in generating audio from videos. We are also extending audio support to the problems in video retargeting [2], thereby making it unsupervised audio-video retargeting. These different applications could only be possible due to exemplar autoencoders.

User-Controllable Content Creation: Our design decisions have chiefly been influenced by user-perspective. The use of *many* exemplar auto-encoders provide flexibility to a user to select the target speaker and a scenario. Not only this, our system can be easily extended in few minutes to new examples using only two-three minutes of audio and a few seconds of video sequence for a new person or scenario. Our work can also be useful in video editing systems directly from audio [3, 11]. We release a live public web-demo with this work that enables anyone to input audio using a mic and generate audio-video of their favorite person.

3. Exemplar Autoencoders

There is an enormous space of stylistic information ranging from prosody, pitch, emotions, and environment. It is challenging for a single *large* model to learn different things. However, many *small* models can easily capture the various nuances. In this work, we seek the problem of any-to-many voice-conversion via exemplar auto-encoders (explicitly trained for a speaker and scenario).

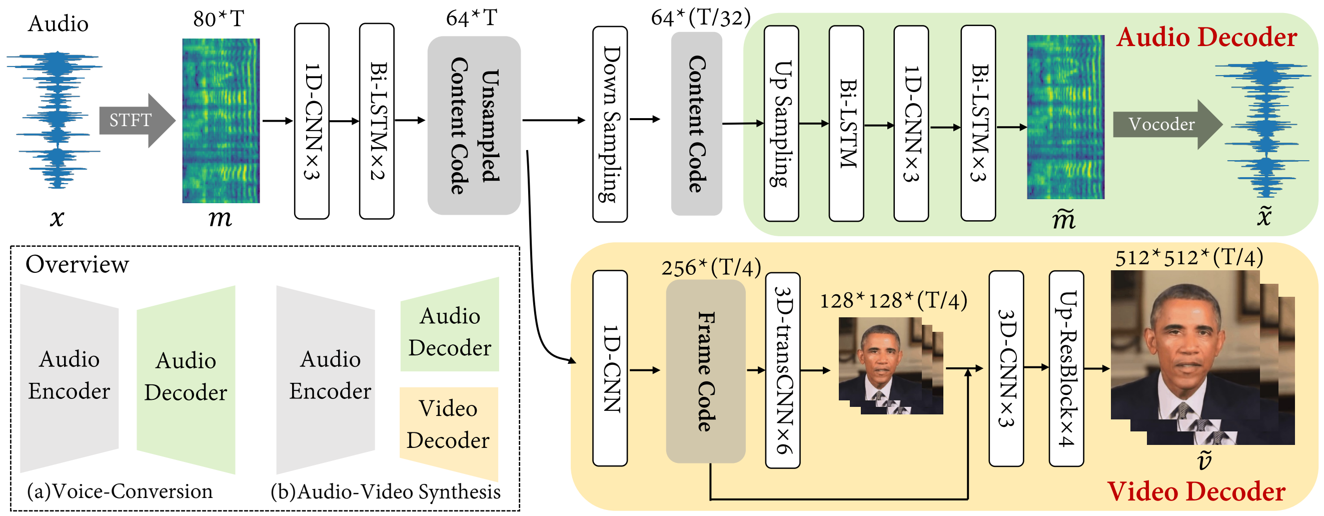


Figure 5. **Network Architecture:** (a) The voice-conversion network consists of a content encoder, and an audio decoder (denoted as green). This network serves as a person & attributes-specific auto-encoder at training time, but is able to convert speech from anyone to personalized audio for the target speaker at inference time. (b) The audio-video synthesis network incorporates a video decoder (denoted as yellow) into the voice-conversion system. The video decoder also regards the content encoder as its front-end, but takes the unsampled content code as input due to time alignment. The video architecture is mainly borrowed from StackGAN [37, 51], which synthesizes the video through 2 resolution-based stages.

3.1. Voice Conversion

To motivate our exemplar autoencoder approach, we revisit linear autoencoders. It is widely known that linear autoencoders can be learned with principle component analysis (PCA) [1]. Specifically, given a dataset $X_{n\times d}$, one can learn a k-dimensional linear projection $B_{d\times k}$ that mininizes reconstruction error as follows:

$$\min_B ||X - XBB^T||^2$$

where XBB^T is the best rank-k approximation of original dataset.

Let $S = \{xBB^T | x \in \mathbb{R}^d\}$ denotes the subspace given by the PCA of a dataset. It is also straightforward to show that given an out-of-sample datapoint, a linear autoencoder run on data point will produce an output that minimizes the reconstruction error of the input to subspace S:

$$\hat{x}BB^T = \arg\min_{s\in S} ||s - \hat{x}||^2$$

where $\hat{x}$ is an out-of-sample datapoint. Hence we can view the autoencoder as a *projection* of the datapoint $\hat{x}$ into the training dataset X.

We can extend PCA to a nonlinear autoencoder by replacing matrix B with an encoder function E, and B^T with a decoder function D. In this work, we train an exemplar autoencoder for a speaker and scenario by minimizing the reconstruction error:

$$\min_{E,D} \text{Error}(X, D(E(X)))$$

The encoder E compresses the content in bottleneck features, which can then be reconstructed using decoder D. Let us define the set $S = \{D(E(x)) | x \in \mathbb{R}^d\}$ of possible reconstructions for a given autoencoder. Qian et al. [36] pointed out that a tight enough bottleneck can extract content information from speech, which is speaker-independent. Our content encoder, therefore, should be able to remove speaker-dependent details and retrieve the content. On the other hand, our audio decoder should be able to add back the speaker-dependent style to the content and thus recover the speech from the content information. In this way, we can constrain the subspace S to exactly the exemplar speech space. We conjecture (with empirical verification in Table 3) the same reprojection property holds for out-of-sample data:

$$D(E(\hat{x})) \approx \arg\min_{s\in S} \text{Error}(s, \hat{x}) \qquad (1)$$

where $\hat{x}$ is an out-of-sample datapoint and “Error” is the reconstruction error used to train the autoencoder. Concretely, if we set $\hat{x}$ to be a particular word w spoken by anyone, the output will (1) be a datapoint from S, and (2) have roughly the minimum distance from $\hat{x}$. As depicted in Fig 4, the spoken words cluster regardless of who said it. Therefore, the output should be the word w but spoken by the exemplar speaker. We give a more formal proof in Section 6.

Network Architecture: An auto-encoder consists of two modules, a content encoder, E, that extracts the content information from speech Mel-spectrograms, and an audio decoder, D, that recovers the Mel-spectrograms from the content information. We transform the speech signal, x, into

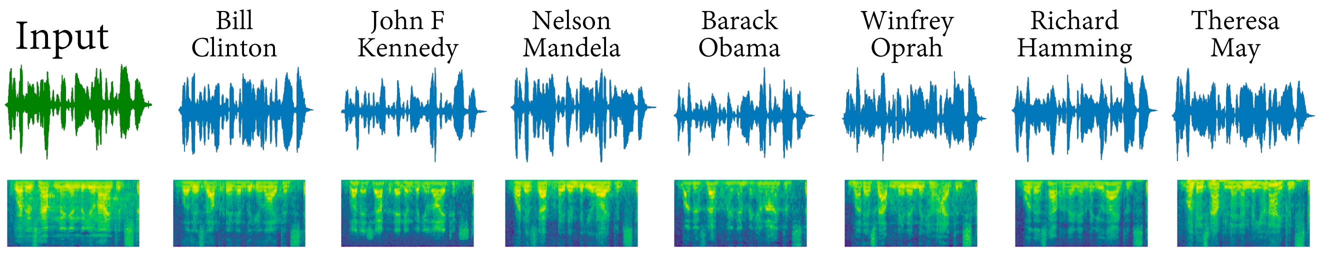


Figure 6. **Any-to-Many Audio Conversion:** Our approach enables a user to select from a list of speakers. Given an input audio on left, we show the generated outputs for various public figures and celebrities. The top row shows the waveform of input and the various outputs. We observe similar pattern for utterances and pauses across different waveforms. Varying amplitude of waveforms arises from the exemplar model trained for a particular scenario and ambience. We show MEL spectrograms in second row. Similar positions of the main formants (i.e. maximum values in MEL spectrograms) show the content consistency of voice conversion while variance of other minor formants shows the characteristics of each speaker.

80-channel speech Mel-spectrograms m using short-time Fourier Transform [31]. We use a WaveNet vocoder [30] to convert the generated Mel-spectrograms $\tilde{m}$ back to speech signal $\tilde{x}$.

Content Encoder: The input to the content encoder is the $80 \times T$ Mel-spectrogram, which is a 1D 80-channel signal (shown in Figure 5). This input is feed-forward to three layers of 1D convolutional layers with a kernel size of 5, each followed by batch normalization [17] and ReLU activation [25]. The channel of these convolutions is 512. The stride is one. There is no time down-sampling up till this step. The output is then fed into two layers of bidirectional LSTM [14] layers with both the forward and backward cell dimensions of 32. We then perform a different down-sampling for the forward and backward paths with a factor of 32 following [36]. The result content embedding is a matrix with a size of $64 \times (T/32)$.

Audio Decoder: The content embedding is up-sampled to the original time resolution of T. The up-sampled embedding is sequentially input to a 512-channel LSTM layer and three layers of 512-channel 1D convolutional layers with a kernel size of 5. Each step accompanies batch normalization and ReLU activation. Finally, the output is fed into two 1024-channel LSTM layers and a fully connected layer to project into 80 channels. The projection output is regarded as the generated Mel-spectrogram $\tilde{m}$.

Optimization & Inference: We do not assume the availability of any other data. We use only self-reconstruction of the target speaker's speech during training. Our loss function considers the reconstruction of both the Mel-spectrograms and the audio signal to jointly train the content encoder, the audio decoder, and the vocoder. The formulation is as follows:

$$\text{Error}_{audio} = \mathbb{E}\|x - \tilde{x}\|_1 + \mathbb{E}\|m - \tilde{m}\|_1 \tag{2}$$

During inference time, a target speaker is one of our pre-trained auto-encoders. Importantly, the source speaker can be anyone. Figure 6 and Figure 7 shows the audio translation outputs from the proposed approach.

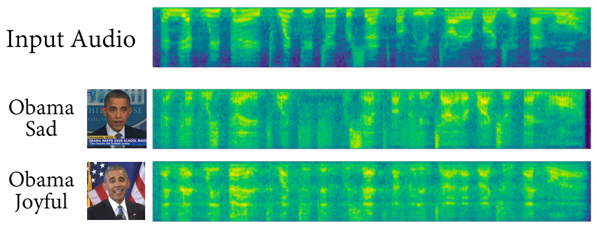
Training Details: Our model is trained at a learning rate of 0.001 and a batch size of 8. To train a model from scratch, it needs about 30 minutes of the target speaker's speech data and around 10k iterations to converge. Although our main structure is straightforward, the vocoder is usually a large and complicated network, which needs another 50k iterations to train. However, transfer learning can be beneficial in reducing the number of iterations and necessary data for training purposes. When fine-tuning a new speaker's autoencoder from a pre-trained model, we only need about 3 minutes of speech from a new speaker. The entire model, including the vocoder, converges around 10k iterations.

Mel-spectrogram: The speech data is sampled at 16 kHz. We clip the training speech into clips of 1.6s in length in order to stabilize the training process. The Mel-spectrograms during the training time are 80×128 in size. However, that doesn't restrict the flexibility of our framework since we can input arbitrarily long speech during the inference time.

3.2. Audio-Video Synthesis

We extend our framework to audio-video synthesis, which still takes the speech as input, but generates the target speaker's talking head video alongside the audio. As observed, there is a high correlation between lip movements and spoken words. It is, therefore, reasonable to train a sub-network to infer a "talking-head" video from an audio input

Network Architecture: We keep the voice-conversion framework unchanged and enhance it with an additional audio-to-video decoder. In the voice-conversion network, we have a content encoder that extracts content embedding from speech, and an audio decoder that generates audio output from that embedding. To include video synthesis, we add a video decoder which also takes the content embedding as input, but generates video output instead. As shown in the second part of Figure 5, we then have an audio-to-audio-



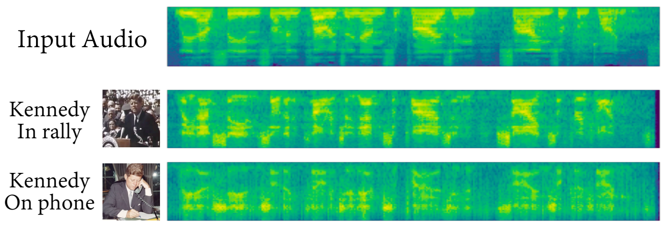


Figure 7. **Personalized Audio Outputs in MEL Spectrogram:** Human beings are multi-modal. A speech, therefore, is not just about voice, but also about emotional aspects intervened with ambiance of a place. Exemplar autoencoders can be easily trained to learn such emotional and ambient aspects from an audio. We show examples of generation with different characteristics such as sad Obama and joyful Obama, Kennedy in a rally vs. on phone-call. We observe same patterns of MEL spectrograms of both Obama and Kennedy. This similarity suggests that content is consistent. However, variance of minor formants depict different characterized details in each generation.

video pipeline.

Video Decoder: We borrow architecture of the video decoder from [37, 51]. We adapt this image synthesis network by generating the video frame by frame, as well as replacing the 2D-convolutions with 3D-convolutions to enhance temporal coherence. The video decoder takes the unsampled content codes as input. This step is used to align the time resolution with 20-fps videos in our experiments. We down-sample it with a 1D convolutional layer. This step helps smooth the adjacent video frames. The output is then fed into the synthesis network to get the video result $\tilde{v}$.

Optimization & Inference: We only use the target speaker's speech video during training. The loss function formulation is as follows:

$$\text{Error}_{audiovisual} = \mathbb{E}\|x - \tilde{x}\|_1 + \mathbb{E}\|m - \tilde{m}\|_1 + \mathbb{E}\|v - \tilde{v}\|_1 \quad (3)$$

We reconstruct the speech as well as infer the talking-head video from the audio input at test time. Importantly, we can input any speech and get the audio and video for the target speaker. Figure 8 shows the audio-video synthesis results using this formulation.

3.3. Other Applications

Predicting Audio from Video: Exemplar autoencoder provides an audio code space that can also be used to predict audio from video as shown Figure 9. We fix the pre-trained audio autoencoder, and train a video encoder that transforms the talking-head video to the bottleneck features of corresponding audio. At test time, we use the video encoder to predict bottleneck features, and predict audio using the pre-trained audio decoder.

Unsupervised audio-video retargeting: We extend audio support to video retargeting [2] by training an exemplar autoencoder for the target identity. During inference, we translate the input speech to the target.

4. Experiments

We now quantitatively evaluate the proposed method for any-to-many audio conversion and audio-video synthesis.

VCTK [47]	Zero-Shot	Extra-Data	SCA (%) (Voice Similarity)	MCD (Content Consistency)
StarGAN-VC [23]	✗	✓	69.5	582.1
VQ-VAE [46]	✗	✓	69.9	663.4
Chou et al. [8]	✗	✓	98.9	**406.2**
Blow [39]	✗	✓	87.4	444.3
Auto-VC [36]	✓	✓	98.5	408.8
Ours	✓	✗	**99.6**	420.3

Table 1. **Objective Evaluation for Audio Translation:** We do objective evaluation using VCTK dataset that provides paired data. The speaker-classification accuracy (SCA) criterion enables us to study the naturalness of generated audio samples and similarity to the target speaker, where **higher is better**. On the other hand, Mel-Cepstral distortion (MCD) assesses content preservation, where **lower is better**. Our approach achieves competitive performance to prior state-of-the-art without requiring any extra-data and yet be zero-shot. We do a more comprehensive human studies in Table 2 using CelebAudio-20 dataset to study the influence of data.

4.1. Audio Translation

Dataset: We use the publicly available VCTK dataset [47] and introduce a new CelebAudio dataset for in-the-wild audio translation setup to inspire future work in this direction.

VCTK Dataset: VCTK corpus [47] contains 44 hours of utterances from 109 speakers. Each speaker reads a different set of sentences, except for two paragraphs. While the conversion setting is non-parallel, there exists a small amount of parallel data enables us to conduct objective evaluation.

CelebAudio Dataset: We introduce a new *in-the-wild* dataset for audio translation to validate the effectiveness as well as the robustness of various approaches. This dataset consists of speeches (average 30 minutes) of various public figures collected from YouTube. The content of these speeches is entirely different from one another, thereby forcing the future methods to be non-parallel and unsupervised. There are 100 different speeches. We use 20 celebrities for

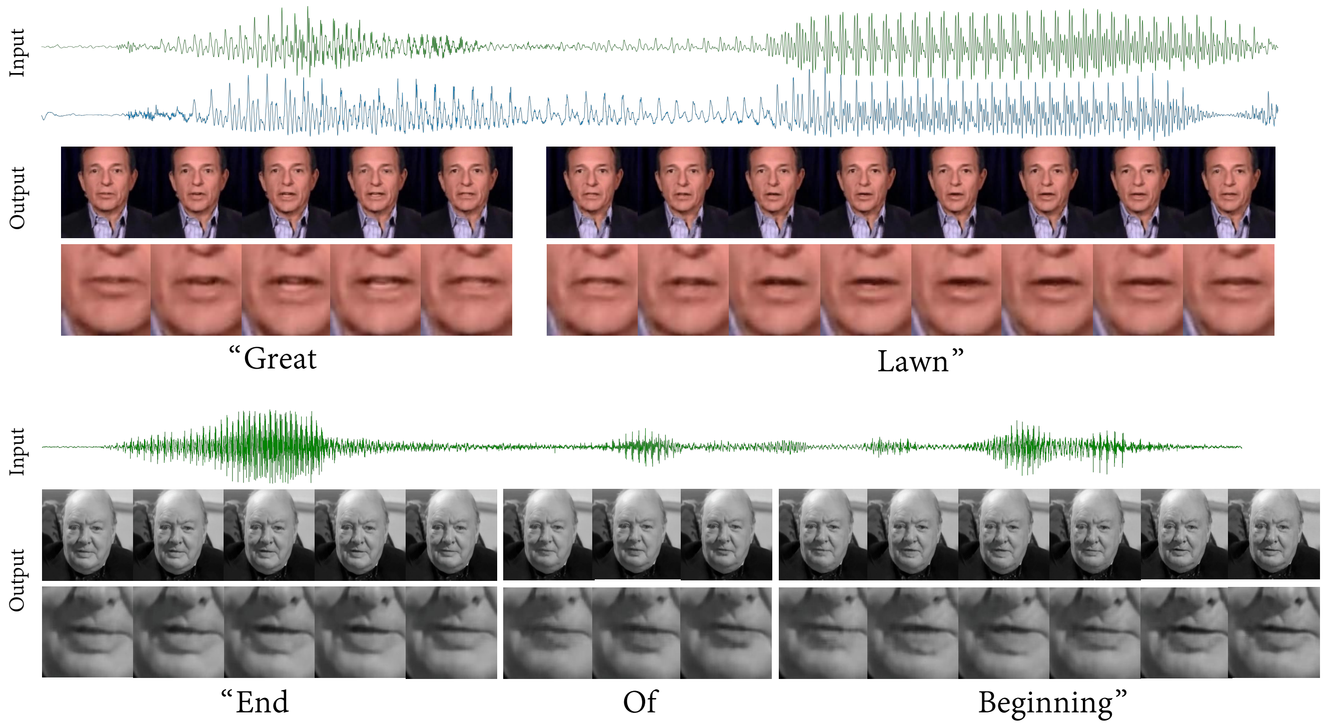


Figure 8. **Audio-Video Synthesis:** We use exemplar autoencoders to generate both audio and video outputs of a known target speaker for an audio input (Shown in top half). We also use our approach to generate videos of several celebrities from audio inputs for some interesting incidents of their lives, such as Winston Churchill (Shown in second half). A zoom-in view for mouth part is presented to show the lip motion is consistent with the spoken words. Additionally, we suggest the reader to see supplementary material for more audio-video results.

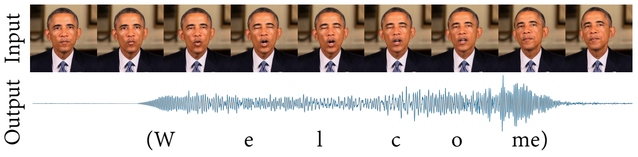


Figure 9. **Predicting Audio from Video:** Our approach used to predict audio from video. We use a pre-trained *Obama* auto-encoder, and regress the video input to the bottleneck features of the auto-encoder. At test time, we predict the audio bottleneck features from the input video and use the decoder of audio autoencoder to generate the corresponding audio.

human-studies, and call that subset **CelebAudio-20**.

Quantitative Evaluation: We conducted user studies for evaluation on CelebAudio-20 dataset, and do objective evaluation using VCTK dataset.

Speaker Classification Accuracy: We use the speaker-classification accuracy (SCA) criterion to study voice conversion for different approaches. We train a speaker classifier following Serra et al. [39]. We compute the percentage of times a translation is able to classify correctly. **Higher it is, better it is**.

Mel-Cepstral Distance: Following prior works [22, 23], we use Mel-Cepstral distortion (MCD) to another objective evaluation criterion to assess content consistency. This metric assesses the distance between the synthesized audio and ground-truth. **Lower it is, better it is**.

Human Studies: We extensively conducted human studies on Amazon Mechanical Turk (AMT) using the generated audio samples from CelebAudio-20 dataset. The goal of this study is to (1). assess the quality of generated samples; (2). the ability of an approach to produce voice similar to target speaker; and (3). to ensure the consistency of content during the translation process. We, therefore, conducted our studies in three different phases. In the first phase, we presented an audio sample to a user on AMT and asked: *How natural is this recording?*. The results of this phase enables us to study **naturalness** of generated content. In the next phase, we presented two audio samples (one real, and another generated) to a user and asked: *Are these two audios from the same speaker?*. We instructed users to pay attention to the voice only and ignore the speech content. This phase allows us to study the notion of **voice-similarity**. Finally, we once again presented two audio samples (one real, and another generated) to a user and asked: *Are these two people saying the same thing?*. Here, we instructed users to pay attention to the content and not voice. The results of this phase allows us to study how much content is preserved during audio translation, i.e., **content-consistency**. The users were asked to rate on a scale of 1-5, i.e., bad to excellent. For all these

CelebAudio-20	Extra Data	Naturalness ↑	Voice Similarity ↑ (VS)	Content Consistency ↑ (CC)	Normalized Area ↑ under VS-CC curve
Auto-VC [36]					
off-the-shelf	-	1.21 ± 0.45	1.31 ± 0.67	1.60 ± 0.78	0.084
fine-tuned	✓	2.35 ± 0.95	1.97 ± 1.14	$\mathbf{4.28 \pm 0.84}$	0.337
scratch	✗	2.28 ± 0.94	1.90 ± 1.01	4.05 ± 0.96	0.307
Ours	✗	$\mathbf{2.78 \pm 1.12}$	$\mathbf{3.32 \pm 1.34}$	4.00 ± 1.22	**0.531**

Table 2. **Human Studies for Audio**: We extensively conducted human studies on Amazon Mechanical Turk. We report Mean Opinion Score (MOS) to assess (1). naturalness; (2). voice similarity; and (3). content preservation. We also report area under voice-similarity and content-consistency curve to study audio translation. **Higher the better**. Auto-VC is an any-to-any audio conversion approach. We, therefore, use an off-the-shelf model for evaluation. We observe poor performance. We then fine-tuned the existing model using CelebAudio-20 dataset. We observe significant performance improvement in Auto-VC when restricting it to the same set of examples as ours. To make it more similar to ours (scratch), we even trained the models from scratch. The performance slightly dropped. Finally, area under voice similarity vs content consistency curve shows that our approach can generate significantly better audio outputs that sounds more like a target speaker while still preserving the original content without using extra data. The performance improvement is specifically due to the exemplar autoencoder approach that enables the use of larger parametric models to capture a set of speaker styles.

criteria: **Higher it is, better it is**.

We also compute the normalized area under the voice-similarity and content-consistency curve. This criterion is useful to study the joint notion of content and style for audio translation. Given a (source, target) pair, we always have two simple ways to generate the output : (1) output the source audio - 0 in voice similarity (VS) but 5 in content consistency (CC); (2) output a random audio from target speaker - 5 in VS but 0 in CC. However, good results should be in between with the source's content but the target's voice. E.g., a $(2.5, 2.5)$ result should be better than either $(5, 0)$ or $(0, 5)$. The normalized area under VS-CC enables us to study it effectively. **Higher it is, better it is**.

We select 10 speakers as target speakers from CelebAudio-20 and randomly choose 5 utterances from the other speakers for test. We then produce $5 \times 10 = 50$ conversions by converting one test utterance to each of the selected 10 speakers' voice. There are a total of 150 HITs for testing naturalness, voice similarity, and content consistency. Each HIT is assigned to 10 users. All the users of AMT were chosen to have Master Qualification (HIT approval rate more than 98% for more than $1,000$ HITs). We also restricted the users to be from United States to ensure English-speaking audience. Our setup and dataset are available on project page for public-use.

Baselines: We study the various aspects of our methods in contrast with several existing voice conversion systems, such as StarGAN-VC [23], VQ-VAE [46], Blow [39], and Auto-VC [36]. While possible, StarGAN-VC [23], VQ-VAE [46], Chou et al. [8], and Blow [39] does not claim zero-shot voice conversion. Therefore, we train these models on 20 speakers from VCTK dataset and perform traditional voice conversion between speakers within the training set. Shown in Table 1 , we observe that our approach outperforms these approaches for voice similarity and yet competitive for content consistency.

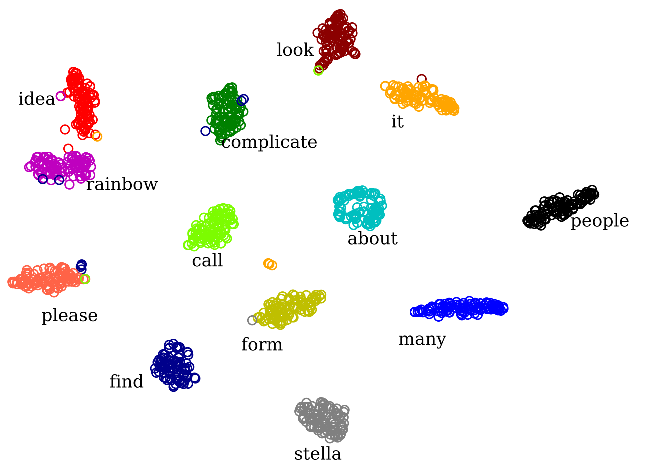
Auto-VC [36]: Auto-VC claims zero-shot conversion. Shown in Table 1, both the approaches achieve competitive performance for both S.C.A and M.C.D. We, therefore, extensively study two methods via human studies on AMT in Table 2. Since Auto-VC claims any-to-any audio conversion, we first use an **off-the-shelf** model[1] for evaluation. We observe poor performance, both quantitatively and qualitatively. We then **fine-tuned** the existing model using the audio data from 20 speakers, thereby making it any-to-many audio translation approach (similar to ours). We observe significant performance improvement in Auto-VC when restricting it to the same set of examples as ours. To make it more similar to ours, we even trained the models from **scratch** using exactly same data and settings as ours. The performance on all three criterion dropped with lesser data. On the other hand, our approach can generate significantly better audio outputs that sounds more like a target speaker while still preserving the original content. Importantly, our models are trained from scratch and does not require hundreds of hours of speech data for training.

4.2. Audio-Video Synthesis

Baseline: We adopt Nearest Neighbour as our baseline for audio-to-video synthesis. Given a corpus of training videos with audio, we find the closest audio frame in the training data for an input audio signal. We use the corresponding video frame as nearest-neighbor result.

We select talking-head videos from our CelebAudio-20

[1]Auto-VC inputs a reference audio (roughly 20 seconds long) to obtain a speaker embedding.



T-SNE for similarities of bottleneck features

Figure 10. We visualize bottleneck features learned by exemplar audio autoencoders for different speakers with nonlinear multi-dimensional scaling (MDS) [5]. We represent a particular word by the cosine similarity of its bottleneck code (from two different encoders) to other words uttered by the same speaker. We then apply T-SNE on this similarity representation, which demonstrates that similar words cluster together, regardless of the speaker. This suggests that person-specific exemplar autoencoders learn codebooks that can generalize across individuals.

and VoxCeleb[9] dataset for training, and design 2 tasks for evaluation: (1) same-speaker: input audio and output video are from the same speaker; (2) cross-speaker: input audio and output video are from different speakers. A human subject is shown the output of baseline and our method, and is asked to choose one with better visual quality. 97% of times human subjects picked the outputs from our approach. We observe the temporal consistency in the outputs as primary reason for better approval rate of our method.

5. Discussion

In this paper, we propose a plug-and-play system of any-to-many voice conversion based on exemplar autoencoders. Our approach takes advantage of the structured audio space, and is the first to employ projections of out-of-sample data in style transfer problems. Our approach also moves beyond well-curated datasets, and proves effectiveness on in-the-wild web audios. Finally, we take the lead to consider the problem of audio-to-video generation (without any intermediates) that has not got much attention so far, and find our exemplar autoencoders also useful in audio-video synthesis.

6. Formal Proof

Speech contains two types of information: (i) **speaker information** that describes the speaker-specific voice; (ii) **content information** that refers to the content being said, which is speaker-independent. It is natural to assume that speech is generated by the following process. First, a speaker identity s is drawn from the speaker space S. Then a content code w is drawn from the content space W. Finally, given the speaker identity s and the content code w, $x = f(s, w)$ denotes the speech of the content w spoken by speaker s, where f is the generating function of speech. Then the task of target-specific voice conversion can be described as: Given a specific target speaker s_{target}, find a function $h_{conversion}$ such that $\forall w \in W, \forall s \in S$:

$$h_{conversion}(f(s,w)) = f(s_{target}, w). \tag{4}$$

Qian et al. [36] pointed that given several assumptions, a tight enough bottleneck can extract content information from speech, which is speaker-independent. In our framework, we only have one specific speaker during the training time. As a result, redundant information about the specific speaker is shared among all the training data. Similar to [36], a tight enough bottleneck is found to extract the information that is the least to distinguish one word from another in one's speech. In other words, when given input from the specific speaker, our content encoder is able to remove any speaker-dependent information and extract the content. On the other hand, our audio decoder adds back the speaker-dependent style to the content and thus recovers the speech from the content information. We can formulate these properties as follow:

$$\forall x \in \{f(s_1, w) : w \in W\}, E(f(s_1, w)) = w \tag{5}$$

$$\forall w \in W, D(w) = f(s_1, w) \tag{6}$$

where s_1 denotes the specific speaker our autoencoder is trained on. Without loss of generality, here we redefine the content space W as the bottleneck feature space of s_1 autoencoder. Although the bottleneck feature space varies from one exemplar to another, we assume the training data contains roughly the whole complexity of different words and thus each trained space has roughly the same amount of information as the real content space.

Now that we have a content embedding space spanned by the bottleneck features, and a content encoder that maps the specific speaker's speech to this content embedding space, we present an important characteristic of the speech space and explain why it enables the exemplar autoencoder to conduct voice conversion.

Structured Speech Space: In human acoustics, one uses different shapes of his vocal tract[2] to pronounce different words with his same voice. Interestingly, different people use similar, or ideally the same, shapes of vocal tract to

[2]The vocal tract is the cavity in human beings where the sound produced at the sound source is filtered. The shape of the vocal tract is mainly determined by the positions and shapes of the tongue, throat and mouth.

pronounce the same words. For this reason, in the speech space we can find a built-in structure that the acoustic features of the same words by different speakers are very close (also shown in Figure 4). This property is crucial to the generalization of the content encoder.

$$diam(\{f(s, w) : s \in S\}) \leq \epsilon \quad (7)$$

where $diam(A)$ means the least upper bound of the distance between every 2 points in set A.

Since the same words by different speakers are well clustered, a function that encodes one speaker's speech into content can also extract correct content information from other speakers. Formally, if the content encoder, trained on one specific speaker s_1, is K-lipschitz continuous, then the error of the result by inputting other's speech can be bounded as:

$$\|E(f(s, w)) - E(f(s_1, w))\| \leq K\epsilon, \forall s \in S, \forall w \in W \quad (8)$$

In Eq.8, given any word w, $E(f(s_1, w)) = w$ is the content information extracted from w spoken by the target speaker s_1. $E(f(s, w))$ is the result if we input the same word spoken by others into the content encoder. Eq.8 guarantees that the content information extracted from other speakers should be very close to the ground-truth if $K\epsilon$ is very small (function E is smooth enough). That means our content encoder, though trained only on speaker s_1, can be applied to any other speaker and get almost correct results (observation from Figure 4 and Figure 10):

$$E(f(s, w)) \approx w, \forall s \in S, \forall w \in W \quad (9)$$

In this case, the content codes from other speakers can be further fed into the s_1-specific decoder and converted to s_1's voice:

$$D(E(f(s, w))) \approx f(s_1, w), \forall s \in S, \forall w \in W \quad (10)$$

w_B	$\|\|w_{B\to A} - w_B\|\|$	$\min_{w \in s_A} \|\|w - w_B\|\|$	$\arg\min_{w \in s_A} \|\|w - w_B\|\|$	Likelihood(%)
"many"	**18.27**	28.33	"many"	100
"complicate"	**16.15**	20.05	"complicate"	59.4
"idea"	**18.54**	26.57	"idea"	100
"about"	**14.20**	26.51	"about"	100
"rainbow"	**15.68**	27.63	"rainbow"	100
"form"	**13.08**	27.46	"form"	69.5
"people"	**11.08**	18.94	"people"	71.1
"look"	**11.37**	32.41	"look"	67.5
"find"	**15.88**	32.87	"find"	100
"it"	**7.54**	29.11	"it"	100
"please"	**17.46**	23.11	"please"	100
"call"	13.93	**13.25**	"call"	97.6
"stella"	**14.45**	15.77	"stella"	82.4
Average	**14.43**	24.77		

Table 3. **Verification of the reprojection property:** To verify the reprojection property that Eq. 1 describes, we randomly sample 13 words spoken by 2 speakers (A and B) in VCTK dataset. For the autoencoder trained by A, all the words spoken by B are out-of-sample data. Then for each word w_B, we generate $w_{B\to A}$ which is the projection to A's subspace. We need to verify 2 properties - (a) The output $w_{B\to A}$ lies in A's subspace; (b) The autoencoder minimizes the input-output error. To verify (a), we train a speaker classification network on speaker A and B, and predict the speaker of $w_{B\to A}$. We report the softmax output to show how much likely $w_{B\to A}$ is to be classified as A's speech (Likelihood in the table). To verify (b), we calculate (1) distance from w_B to $w_{B\to A}$; (2) minimum distance from w_B to any sampled words by A. It is clear in the table that the input w_B is much closer (sometimes as close as) to the projection $w_{B\to A}$ than any other sampled point in A's subspace. Furthermore, such minimum is reached when the content is kept the same. This empirically verifies the conjecture that the autoencoder minimizes the reconstruction error of even the out-of-sample input.

References

[1] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 1989. 2, 4

[2] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *ECCV*, 2018. 3, 6

[3] Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. Tools for placing cuts and transitions in interview video. *ACM Trans. Graph.*, 2012. 3

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006. 2

[5] Andreas Buja, Deborah F Swayne, Michael L Littman, Nathaniel Dean, Heike Hofmann, and Lisha Chen. Data visualization with multidimensional scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472, 2008. 9

[6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018. 2, 3

[7] Ling-Hui Chen, Zhen-Hua Ling, Li-Juan Liu, and Li-Rong Dai. Voice conversion using deep neural networks with layer-wise generative training. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 2014. 1, 3

[8] Ju-Chieh Chou, Cheng-Chieh Yeh, Hung-Yi Lee, and Lin-Shan Lee. Multi-target Voice Conversion without Parallel Data by Adversarially Learning Disentangled Audio Representations. *Proc. Interspeech*, 2018. 1, 6, 8

[9] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 3, 9

[10] Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba. High-Quality Nonparallel Voice Conversion Based on Cycle-Consistent Adversarial Network. In *IEEE ICASSP*, 2018. 1

[11] 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 Trans. Graph.*, 2019. 3

[12] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018. 3

[13] S. Ginosar, A. Bar, G. Kohavi, C. Chan, A. Owens, and J. Malik. Learning individual styles of conversational gesture. In *CVPR*, 2019. 2, 3

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 5

[15] A. J. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *IEEE ICASSP*, 1996. 3

[16] Zeynep Inanoglu and Steve Young. Data-driven emotion conversion in spoken english. *Speech Communication*, 2009. 1

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[18] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *NeurIPS*, 2018. 3

[19] Alexander B Kain, John-Paul Hosom, Xiaochuan Niu, Jan PH Van Santen, Melanie Fried-Oken, and Janice Staehely. Improving the intelligibility of dysarthric speech. *Speech communication*, 2007. 1

[20] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo. Acvae-vc: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder. *arXiv preprint arXiv:1808.05092*, 2018. 1

[21] Takuhiro Kaneko and Hirokazu Kameoka. Parallel-data-free voice conversion using cycle-consistent adversarial networks. *arXiv preprint arXiv:1711.11293*, 2017. 1, 3

[22] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *IEEE ICASSP*, 2019. 1, 3, 7

[23] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. StarGAN-VC2: Rethinking Conditional Methods for StarGAN-Based Voice Conversion. *Proc. Interspeech*, 2019. 1, 3, 6, 7, 8

[24] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 3

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5

[26] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. Seeing voices and hearing faces: Cross-modal biometric matching. In *CVPR*, 2018. 3

[27] Keigo Nakamura, Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano. Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech. *Speech Commun.*, 2012. 1

[28] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki. High-order sequence modeling using speaker-dependent recurrent temporal restricted boltzmann machines for voice conversion. In *Proc. Interspeech*, 2014. 1, 3

[29] Tae-Hyun Oh, Tali Dekel, Changil Kim, Inbar Mosseri, William T. Freeman, Michael Rubinstein, and Wojciech Matusik. Speech2face: Learning the face behind a voice. In *CVPR*, 2019. 3

[30] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 3, 5

[31] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & Systems (Second Edition)*. Prentice-Hall, Inc., 1996. 5

[32] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. *ECCV*, 2018. 3

[33] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016. 3

[34] John F Pitrelli, Raimo Bakis, Ellen M Eide, Raul Fernandez, Wael Hamza, and Michael A Picheny. The ibm expressive text-to-speech synthesis system for american english. *IEEE Transactions on Audio, Speech, and Language Processing*, 2006. 3

[35] Adam Polyak and Lior Wolf. Attention-based wavenet autoencoder for universal voice conversion. In *IEEE ICASSP*, 2019. 3

[36] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *ICML*, 2019. 1, 3, 4, 5, 6, 8, 9

[37] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 4, 6

[38] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *CVPR*, 2018. 3

[39] Joan Serrà, Santiago Pascual, and Carlos Segura. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. In *NeurIPS*, 2019. 1, 3, 6, 7, 8

[40] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *IEEE ICASSP*, 2018. 3

[41] Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *CVPR*, 2018. 2, 3

[42] Lifa Sun, Shiyin Kang, Kun Li, and Helen Meng. Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In *IEEE ICASSP*, 2015. 1, 3

[43] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 3

[44] Supasorn Suwajanakorn, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: Learning lip sync from audio. *ACM Trans. Graph.*, 2017. 2, 3

[45] Tomoki Toda, Alan W Black, and Keiichi Tokuda. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 2007. 1, 3

[46] Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017. 6, 8

[47] Christophe Veaux, Junichi Yamagishi, and Kirsten Macdonald. Superseded - CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit. 2016. 3, 6

[48] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *Proc. Interspeech*, 2017. 3

[49] O. Wiles, A.S. Koepke, and A. Zisserman. X2face: A network for controlling face generation by using images, audio, and pose codes. In *ECCV*, 2018. 3

[50] Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *Speech Communication*, 2009. 3

[51] Han Zhang, Tao Xu, and Hongsheng Li. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 4, 6