More than Words: In-the-Wild Visually-Driven Prosody for Text-to-Speech

Michael Hassid
Google Research
hassid@google.com

Michelle Tadmor Ramanovich
Google Research
tadmor@google.com

Brendan Shillingford
DeepMind
shillingford@deepmind.com

Miaosen Wang
DeepMind
miaosen@deepmind.com

Ye Jia
Google Research
jiaye@google.com

Tal Remez
Google Research
talremez@google.com

Abstract

*In this paper we present VDTTS, a **V**isually-**D**riven **T**ext-to-**S**peech model. Motivated by dubbing, VDTTS takes advantage of video frames as an additional input alongside text, and generates speech that matches the video signal. We demonstrate how this allows VDTTS to, unlike plain TTS models, generate speech that not only has prosodic variations like natural pauses and pitch, but is also synchronized to the input video.*

Experimentally, we show our model produces well-synchronized outputs, approaching the video-speech synchronization quality of the ground-truth, on several challenging benchmarks including "in-the-wild" content from VoxCeleb2. Supplementary demo videos demonstrating video-speech synchronization, robustness to speaker ID swapping, and prosody, presented at the project page.[1]

1. Introduction

Post-sync, or dubbing (in the film industry), is the process of re-recording dialogue by the original actor in a controlled environment after the filming process to improve audio quality. Sometimes, a replacement actor is used instead of the original actor when a different voice is desired such as Darth Vader's character in Star Wars [1].

Work in the area of automatic audio-visual dubbing often approaches the problem of generating content with synchronized video and speech by (1) applying a text-to-speech (TTS) system to produce audio from text, then (2) modifying the frames so that the face matches the audio [2]. The second part of this approach is particularly difficult, as it requires generation of photorealistic video across arbitrary filming conditions.

[1]Project page:
http://google-research.github.io/lingvo-lab/vdtts

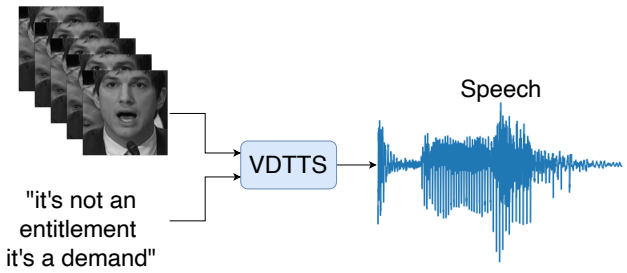

Figure 1. Given a text and video frames of a speaker, VDTTS generates speech with prosody that matches the video signal.

In contrast, we extend the TTS setting to input not only text, but also facial video frames, producing speech that matches the facial movements of the input video. The result is audio that is not only synchronized to the video but also retains the original prosody, including pauses and pitch changes that can be inferred from the video signal, providing a key piece in producing high-quality dubbed videos.

In this work, we present VDTTS, a visually-driven TTS model. Given text and corresponding video frames of a speaker speaking, our model is trained to generate the corresponding speech (see Fig. 1). As opposed to standard visual speech recognition models, which focus on the mouth region [3], we provide the full face to avoid potentially excluding information pertinent to the speaker's delivery. This gives the model enough information to generate speech which not only matches the video but also recovers aspects of prosody, such as timing and emotion. Despite not being explicitly trained to generate speech that is synchronized to the input video, the learned model still does so.

Our model is comprised of four main components. Text and video encoders process the inputs, followed by a multi-source attention mechanism that connects these to a decoder that produces mel-spectrograms. A vocoder then produces waveforms from the mel-spectrograms.

We evaluate the performance of our method on GRID [4]

as well as on challenging in-the-wild videos from VoxCeleb2 [5]. To validate our design choices and training process, we also present an ablation study of key components of our method, model architecture, and training procedure.

Demo videos are available on the project page,[1] demonstrating video-speech synchronization, robustness to speaker ID swapping, and prosody. We encourage readers to take a look.

Our main contributions are that we:

- present and evaluate a novel visual TTS model, trained on a wide variety of open-domain YouTube videos;
- show it achieves state-of-the-art video-speech synchronization on GRID and VoxCeleb2 when presented with arbitrary unseen speakers; and
- demonstrate that our method recovers aspects of prosody such as pauses and pitch while producing natural, human-like speech.

2. Related work

Text-to-speech (TTS) engines which generate natural sounding speech from text, have seen dazzling progress in recent years. Methods have shifted from parametric models towards increasingly end-to-end neural networks [6, 7]. This shift enabled TTS models to generate speech that sounds as natural as professional human speech [8]. Most approaches consist of three main components: an encoder that converts the input text into a sequence of hidden representations, a decoder that produces acoustic representations like mel-spectrograms from these, and finally a vocoder that constructs waveforms from the acoustic representations.

Some methods including Tacotron and Tacotron 2 use an attention-based autoregressive approach [7, 9, 10]; followup work such as FastSpeech [11, 12], Non-Attentive Tacotron (NAT) [8, 13] and Parallel Tacotron [14, 15], often replace recurrent neural networks with transformers.

Extensive research has been conducted on how to invert mel-spectrograms back into waveforms; since the former is a compressed audio representation, it is not generally invertible. For example, the seminal work of Griffin and Lim [16] proposes a simple least-squares approach, while modern approaches train models to learn task-specific mappings that can capture more of the audio signal, including the approaches of WaveNet as applied to Tacotron 2 [9], MelGAN [6, 17], or more recent work like WaveGlow [18] which trains a flow-based conditional generative model, DiffWave [19] which propose a probabilistic model for conditional and unconditional waveform generation, or WaveGrad [20] that make use of data density gradients to generate waveforms. In our work, we use the fully-convolutional SoundStream vocoder [21].

TTS prosody control Skerry-Ryan et al. [22] define prosody as "the variation in speech signals excluding phonetics, speaker identity, and channel effects." Standard TTS approaches tend to be trained to produce neutral speech, due the difficulty of modeling prosody.

Great efforts have been made towards transferring or controlling the prosody of TTS audio. Wang et al. [23] created a style embedding by using a multi-headed attention module between the encoded input audio sequence and the global style tokens (GSTs). They trained a model jointly with the Tacotron model using the reconstruction loss of the mel-spectrograms. At inference time, they construct the style embedding from the text to enable style control, or from other audio for style transfer.

A Variational Auto-Encoder (VAE) latent representation of speaking style was used by [24]. During inference time, they alter speaking style by manipulating the latent embedding, or by obtained it from a reference audio. Hsu et al. [25] used a VAE to create two levels of hierarchical latent variables, the first representing attribute groups, and the second representing more specific attribute configurations. This setup allows fine-grained control of the generated audio prosody including accent, speaking rate, etc.

Silent-video-to-speech In this setup, a silent video is presented to a model that tries to generate speech consistent with the mouth movements, without providing text. Vid2Speech [26] uses a convolutional neural network (CNN) that generates an acoustic feature for each frame of a silent video. Lipper [27] use a closeup video of lips and produces text and speech, while [28] directly generated the speech without a vocoder. Prajwal et al. [29] propose a speaker specific lip-reading model.

Datasets For our task, we require triplets consisting of: a facial video, the corresponding speech audio, and a text transcript. The video and text are used as model inputs,

	Utts.	Hrs.	Vocab.	Speakers	ID	Source
GRID [4]	34K	43	51	34	✓	Studio
LRS2 [30]	47K	29	18K	-	✗	BBC
LRS3 [31]	32K	30	17K	3.8K	✗	TED/TEDx
VoxCeleb2 [5]	1M	2442	~35K*	6.1K	✓	YouTube
LSVSR [2, 3]	3M	3130	127K	~464K	✗	YouTube

Table 1. Audio-visual speech dataset size comparison in terms of number of utterances, hours, and vocabulary. Numbers are shown before processing; the resulting number of utterances we use for VoxCeleb2 and LSVSR are smaller. In Yang et al. [2], LSVSR is called MLVD. (*VoxCeleb2 lacks transcripts, so we use an English-only automated transcription model [32] to produce transcripts for training purposes, also used for vocabulary size measurement in this table.)

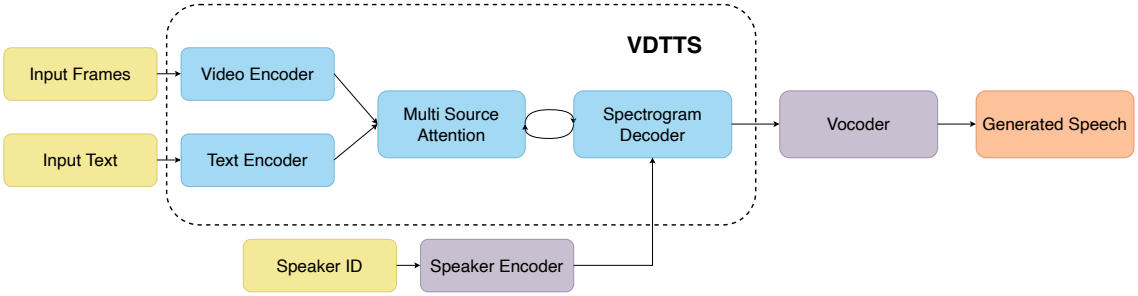

Figure 2. The overall architecture of our model. Colors: inputs: yellow, trainable: blue, frozen: purple, output: orange.

whereas the speech audio is used as ground-truth for metrics and loss computation.

GRID is a standard dataset filmed under consistent conditions [4]. LRW [33] and LRS2 [30] are based on high-quality BBC television content, and LRS3 [31] is based on TED talks; however, these datasets are restricted to academic use only. VoxCeleb2 [5] and LSVSR [2, 3], being based on open-domain YouTube data, contain the widest range of people, types of content, and words. A comparison of dataset size appears in Table 1.

In this work, we adopt GRID as a standard benchmark, and VoxCeleb2 and LSVSR due to their greater difficulty.

Automated dubbing A common approach to automated dubbing is to generate or modify the video frames to match a given clip of audio speech [2, 34, 35, 36, 37, 38, 39, 40, 41]. This wide and active area of research uses approaches that vary from conditional video generation, to retrieval, to 3D models. Unlike this line of work, we start from a fixed video and generate audio instead.

Recent visual TTS work uses both text and video frames to train a TTS model, much like our approach. Concurrent work to ours [42, 43] take this approach, the former using GRID and the latter using just LRS2. Unlike our work, these approaches explicitly constrain output signal length and attention weights to encourage synchronization.

3. Method

In this section, we describe the architecture of the proposed model and depict its components. Full architectural and training details are given in Appendix A and Appendix B respectively.

Overview Fig. 2 illustrates the overall architecture of the VTTS model. As shown, and similarly to [44], the architecture consists of (1) a video encoder, (2) a text encoder, (3) a speaker encoder, (4) an autoregressive decoder with a multi-source attention mechanism, and (5) a vocoder. The method follows [44] using the combined $L_1 + L_2$ loss.

Let T_x and T_y be the length of input video frame and phoneme sequences respectively. Let D_w, D_h and D_c be the width, height and the number of channels of the frames, D_e the dimension of the phoneme embeddings, and $\mathcal{P}$ the set of phonemes.

We begin with an input pair composed of a source video frame sequence $x \in \mathbb{R}^{T_x \times D_w \times D_h \times D_c}$ and a sequence of phonemes $y \in \mathcal{P}^{T_y}$.

The video encoder receives a frame sequence as input, produces a hidden representation for each frame, and then concatenates these representations, i.e.,

$$h_x = \text{VideoEncoder}(x) \in \mathbb{R}^{T_x \times D_m}, \tag{1}$$

where D_m is the hidden dimension of the model.

Similarly, the text encoder receives the source phonemes and produces a hidden representation,

$$h_y = \text{TextEncoder}(y) \in \mathbb{R}^{T_y \times D_m}. \tag{2}$$

The speaker encoder maps a speaker to a 256-dimensional speaker embedding,

$$d_i = \text{SpeakerEncoder}(\text{speaker}_i) \in \mathbb{R}^{256}. \tag{3}$$

The autoregressive decoder receives as input the two hidden representations h_x and h_y, and the speaker embedding d_i, and predicts the mel-spectrogram of the synthesized speech using the attention context,

$$\hat{z}^t = \text{Decoder}(\hat{z}^{t-1}, h_x, h_y, d_i). \tag{4}$$

Finally, the predicted mel-spectrogram $[\hat{z}^1, \hat{z}^2, \ldots, \hat{z}^{T_z}]$ is transformed to a waveform using a frozen pretrained neural vocoder [21].

Video encoder Our video encoder is inspired by VGG3D as in [3]. However, unlike their work and similar lipreading work, we use a full face crop instead of a mouth-only crop to avoid potentially excluding information that could be pertinent to prosody, such as facial expressions.

Text encoder Our text encoder is derived from Tacotron 2's [9] text encoder. Each phoneme is first embedded in a D_e-dimensional embedding space. Then the sequence of phoneme embeddings is passed through convolution layers and a Bi-LSTM layer.

Speaker encoder In order to enable our model to handle a multi-speaker environment, we use a frozen, pretrained speaker embedding model [45] following [10]. When the speaker ID is provided in the dataset, as for GRID and VoxCeleb2, we generate embeddings per utterance and average over all utterances associated with the speaker, normalizing the result to unit norm. For LSVSR the speaker identity is unavailable, so we compute the embedding per-utterance. At test time, while we could use an arbitrary speaker embedding to make the voice match the speaker for comparison purposes, we use the average speaker embedding over the audio clips from this speaker. We encourage the reader to refer to the project page [1], in which example videos demonstrate how VDTTS preforms when speaker voice embeddings are swapped between different speakers.

Decoder Our RNN-based autoregressive decoder is similar to the one proposed by [9], and consists of four parts: a *pre-net*, a fully connected network reprojecting the previous decoder output onto a lower dimension before it is used as input for future time steps; an attention module, in our case *multi-source attention*, discussed later; an LSTM core; and a *post-net* which predicts the final mel-spectrogram output.

The decoder receives as input the output sequences of the: video encoder h_x, the text phoneme encoder h_y as well as the speaker embedding produced by the speaker encoder d_i, and generates a mel-spectrogram of the speech signal $\hat{z}^t$. In contrast to [9], which do not support speaker voice embeddings, we concatenate them to the output of the *pre-net* to enable our model to be used in a multi-speaker environment, i.e: the input for the *multi-source attention* at timestep t is

$$q^t = \text{concat}(\text{PreNet}(\hat{z}^{t-1}), d_i). \tag{5}$$

Multi-source attention A multi-source attention mechanism, similar to that of Textual Echo Cancellation [44], allows selecting which of the outputs of the encoder are passed to the decoder in each timestep.

The multi-source attention, as presented in Fig. 3, has an individual attention mechanism for each of the encoders, without weights sharing between them. At each timestep t, each attention module outputs an attention context,

$$c_x^t = \text{Att}_x(q^t, c_x^{t-1}, h_x);\ c_y^t = \text{Att}_y(q^t, c_y^{t-1}, h_y), \tag{6}$$

where q^t is the output of the *pre-net* layer of the decoder at timestep t.

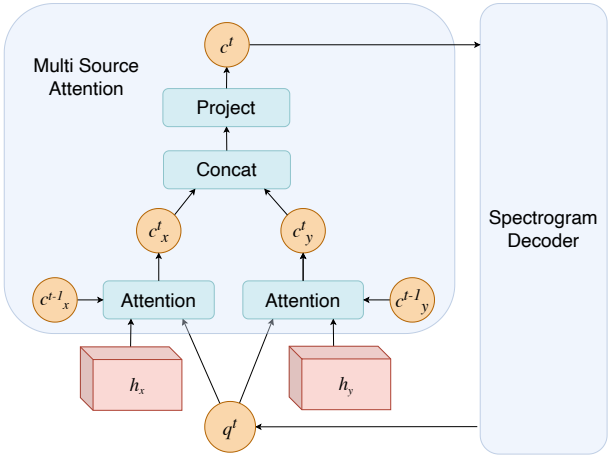

Figure 3. The Multi Source Attention Mechanism.

The input of the decoder at timestep t is the projection of the concatenation of the two contexts described above via a linear layer,

$$c^t = \text{Linear}([c_x^t, c_y^t]). \tag{7}$$

While [44] aggregated the context vectors using summation, we found that a concatenation and projection work better in our setting as shown in Sec. 4.6.

We use a Gaussian mixture attention mechanism [46] for both modalities (video and text), since it is a soft monotonic attention which is known to achieve better results for speech synthesis [47, 48, 49].

Full architectural details appear in Appendix A.

4. Experiments

To evaluate the performance of the proposed video enhanced TTS model we conducted experiments on two very different public datasets: GRID [4] and VoxCeleb2 [5]. GRID presents a controlled environment allowing us to test our method on high quality, studio captured videos with a small vocabulary, in which the same speakers appear in both the train and test sets. VoxCeleb2, however, is much more in-the-wild, therefore it is more diverse in terms of appearance (illumination, image quality, audio noise, face angles, etc.), and the set of speakers in the test set do not appear in the training set. This allows us to test the ability of the model to generalize to unseen speakers.

4.1. Evaluation Metrics

We objectively evaluate prosodic accuracy, video-speech synchronization, and word error rate (WER). We further evaluate synchronization subjectively with human ratings as described below.

Pitch (fundamental frequency, $F0$) and voicing contours are computed using the output of the YIN pitch tracking algorithm [50] with a 12.5ms frame shift. For cases in which

the predicted signal is too short we pad using a domain-appropriate padding up to the length of the reference. If it is too long we clip it shorter.

In the remainder of this section we define and provide intuition for the metrics in the experimental section.

4.1.1 Mel Cepstral Distortion (MCD$_K$) [51]

is a mel-spectrogram distance measure defined as:

$$\text{MCD} = \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{k=1}^{K}(f_{t,k}-\hat{f}_{t,k})^2}, \tag{8}$$

where $\hat{f}_{t,k}$ and $f_{t,k}$ are the k-th Mel-Frequency Cepstral Coefficient (MFCC) [52] of the t-th frame from the reference and the predicted audio respectively. We sum the squared differences over the first $K = 13$ MFCCs, skipping $f_{t,0}$ (overall energy). MFCCs are computed using a 25ms window and 10ms step size.

4.1.2 Pitch Metrics

We compute the following commonly used prosody metrics over the pitch and voicing sequences produced from the synthesized and the ground-truth waveforms [48, 53].

F0 Frame Error (FFE) [54] measures the percentage of frames that either contain a 20% pitch error or a voicing decision error.

$$\text{FFE} = \frac{\sum_t \mathbb{1}[|p_t - p'_t| > 0.2p_t]\mathbb{1}[v_t = v'_t] + \mathbb{1}[v_t \neq v'_t]}{T} \tag{9}$$

where p, p' are the pitch, and v, v' are the voicing contours computed over the predicted and ground-truth audio.

Gross Pitch Error (GPE) [55] measures the percentage of frames where pitch differed by more than 20% on frames and voice was present on both the predicted and reference audio.

$$\text{GPE} = \frac{\sum_t \mathbb{1}[|p_t - p'_t| > 0.2p_t]\mathbb{1}[v_t = v'_t]}{\sum_t \mathbb{1}[v_t = v'_t]} \tag{10}$$

where p, p' are the pitch, and v, v' are the voicing contours computed over the predicted and ground-truth audio.

Voice Decision Error (VDE) [55] measures the proportion of frames where the predicted audio is voiced differently than the ground-truth.

$$\text{VDE} = \frac{\sum_t \mathbb{1}[v_t \neq v'_t]}{T} \tag{11}$$

where v, v' are the voicing contours computed over the predicted and ground-truth audio.

4.1.3 Lip Sync Error

We use *Lip Sync Error - Confidence* (LSE-C) and *Lip Sync Error - Distance* (LSE-D) [56] to measure video-speech synchronization between the predicted audio and the video signal. The measurements are taken using a pretrained SyncNet model [57].

4.1.4 Word Error Rate (WER)

A TTS model is expected to produce an intelligible speech signal consistent with the input text. To measure this objectively, we measure WER as determined by an automatic speech recognition (ASR) model. To this end we use a state-of-the-art ASR model as proposed in [32], trained on the LibriSpeech [58] training set. The recognizer was not altered or fine-tuned.

Since LSVSR is open-ended content, and out-of-domain compared to the audiobooks in LibriSpeech, the ASR performance may result in a high WER even on ground-truth audio. Thus, we only use the WER metric for relative comparison. In Appendix C, we compute the WER on predictions from a text-only TTS model trained on several datasets to establish a range of reasonable WERs; we confirm that a rather high WER is to be expected.

4.1.5 Video-speech sync Mean Opinion Score (MOS)

We measured video-speech synchronization quality with a 3-point Likert scale with a granularity of 0.5. Each rater is required to watch a video at least twice before rating it and a rater cannot rate more than 18 videos; each video is rated by 3 raters. Each evaluation was conducted independently; different models were not compared pairwise. The (averaged) MOS ratings are shown as a 90% confidence interval.

In Sec. 4.4 we rate a total of 200 videos each containing a unique speaker, while in Sec. 4.3 we chose 5 clips per speaker resulting in a total of 165 videos.

4.2. Data preprocessing

Several preprocessing steps were conducted before training and evaluating our models, including audio filtering, face cropping and limiting example length.

We follow a similar methodology first proposed by [3] while creating the LSVSR dataset. We limit the duration of all examples to be in the range of 1 to 6 seconds, and transcripts are filtered through a language classifier [60] to include only English. We also remove utterances which have less than one word per second on average, since they do not contain enough spoken content. We filter blurry clips and use a neural network [61] to verify that the audio and video channels are aligned. Then, we apply a landmarker as in [62] and keep segments where the face yaw and pitch remain within $\pm 15^\circ$ and remove clips where an

	MOS ↑	LSE-C ↑	LSE-D ↓	WER ↓	MCD ↓	FFE ↓	GPE ↓	VDE ↓
GROUND-TRUTH [42]	-	7.68	6.87	-	-	-	-	-
VISUALTTS [42]	-	5.81	8.50	-	-	-	-	-
GROUND-TRUTH	2.68 ± 0.04	7.24	6.73	26%	-	-	-	-
TTS-TEXTONLY [59]	1.51 ± 0.05	3.39	10.44	**19%**	15.76	0.48	0.30	0.42
VDTTS-LSVSR	2.10 ± 0.06	5.85	7.93	55%	12.81	0.37	0.21	0.32
VDTTS-GRID	$\mathbf{2.55 \pm 0.05}$	**6.97**	**6.85**	26%	**7.89**	**0.14**	**0.07**	**0.11**

Table 2. **GRID evaluation.** This table shows our experiments on the GRID dataset. The top two rows present the numbers as they appear in VISUALTTS [42]. GROUND-TRUTH shows the metrics as evaluated on the original speech/video. TTS-TEXTONLY shows the performance of a vanilla text-only TTS model, while VDTTS-LSVSR and VDTTS-GRID are our model when trained on LSVSR and GRID respectively. While VDTTS-GRID archives the best overall performance, it is evident VDTTS-LSVSR generalizes well enough to the GRID dataset to outperform VISUALTTS [42]. See Sec. 4.1 for an explanation of metrics; arrows indicate if higher or lower is better.

	MOS ↑	LSE-C ↑	LSE-D ↓	WER ↓	MCD ↓	FFE ↓	GPE ↓	VDE ↓
GROUND-TRUTH	2.79 ± 0.03	7.00	7.51	-	-	-	-	-
TTS-TEXTONLY [59]	1.77 ± 0.05	1.82	12.44	**4%**	14.67	0.59	0.38	0.42
VDTTS-VOXCELEB2	$\mathbf{2.50 \pm 0.04}$	**5.99**	**8.22**	48%	**12.17**	**0.46**	0.31	**0.30**
VDTTS-LSVSR	2.45 ± 0.04	5.92	8.25	25%	12.23	**0.46**	**0.29**	0.31

Table 3. **VoxCeleb2 evaluation**. GROUND-TRUTH shows the synchronization quality of the original VoxCeleb2 speech and video. TTS-TEXTONLY represents a vanilla text-only TTS model, while VDTTS-VOXCELEB2 and VDTTS-LSVSR are our model when trained on VoxCeleb2 and LSVSR respectively. By looking at the WER, it is evident VDTTS-VOXCELEB2 generates unintelligible results, while VDTTS-LSVSR generalizes well to VoxCeleb2 data and produces better quality overall. See Sec. 4.1 for an explanation of metrics; arrows indicate if higher or lower is better.

eye-to-eye width of less than 80 pixels. Using the extracted and smoothed landmarks, we discard minor lip movements and nonspeaking faces using a threshold filter. The landmarks are used to compute and apply an affine transformation (without skew) to obtain canonicalized faces. Audio is filtered [63] to reduce non-speech noise.

We use this methodology to collect a similar dataset to LSVSR [3], which we use as our in-the-wild training set with $527,746$ examples, and also to preprocess our version of VoxCeleb2, only changing the maximal face angle to $30°$ to increase dataset size. Running the same processing as described above on VoxCeleb2 results in $71,772$ train, and $2,824$ test examples. As for GRID which we use as our controlled environment, we do not filter the data, and only use the face cropping part of the aforementioned pipeline to generate model inputs.

4.3. Controlled environment evaluation

In order to evaluate our method in a controlled environment we use the GRID dataset [4]. GRID is composed of studio video recordings of 33 speakers (originally 34, one is corrupt). There are 1000 videos of each speaker, and in each video a sentence is spoken with a predetermined "GRID" format. The vocabulary of the dataset is relatively small and all videos were captured in a controlled studio environment over a green screen with little head pose variation.

We compare VDTTS to the recent VisualTTS [42] method using the same methodology reported by the authors. To that end, we take 100 random videos from each speaker as a test set. We use the remainder 900 examples per speaker as training data, and also for generating a lookup-table containing the speaker embedding, averaged and normalized per speaker, as explained in Sec. 3. At test time we present our models with video frames alongside the transcript and the average speaker embedding.

We evaluate our method using the metrics mentioned in Sec. 4.1, and compare it to several baselines: (1) VISUALTTS [42]; (2) PnG NAT TTS zero-shot voice transferring model from [59], a state-of-the-art TTS model trained on the LibriTTS [64] dataset, denoted as TTS-TEXTONLY; (3) our model when trained over LSVSR (see Sec. 4.2); and (4) our model trained on the GRID training set.

Unfortunately, VisualTTS [42] did not provide their random train/test splits. Therefore, we report the original metrics as they appear in [42] alongside the numbers we found over our test set. Luckily, the two are comparable, as can be seen by the two rows in Table 2 named GROUND-TRUTH.

The results appear in Table 2. Observe that, when trained on GRID, our method outperforms all other methods over in all metrics except WER. Moreover, our model trained on LSVSR, as we will see in a later section, gets better video-speech synchronization results than VisualTTS, which was trained on GRID, showing that our "in-the-wild" model generalizes to new domains and unseen speakers.

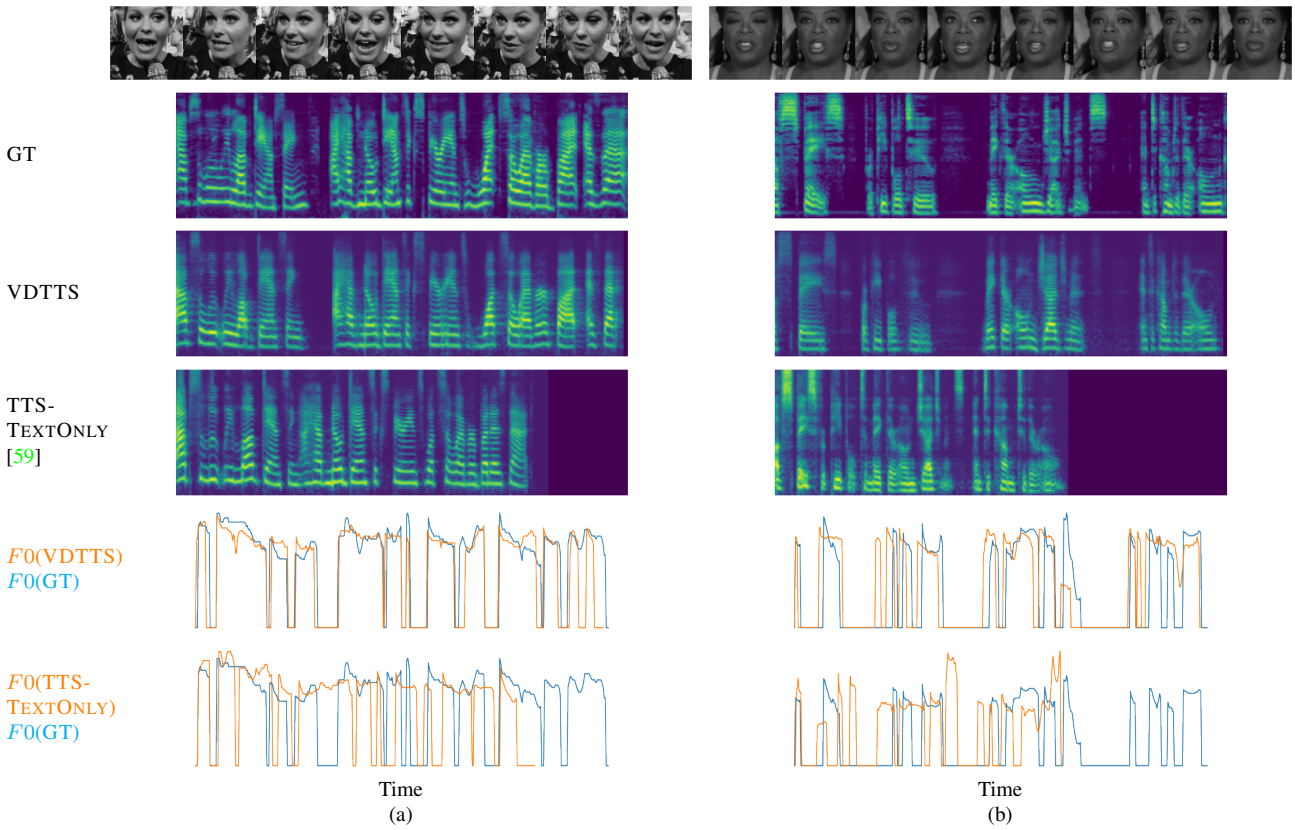

Figure 4. **Qualitative examples.** We present two examples (a) and (b) from the test set of VoxCeleb2 [5]. From top to bottom: input face images, ground-truth (GT) mel-spectrogram, mel-spectrogram output of VDTTS, mel-spectrogram output of a vanilla TTS model TTS-TEXTONLY, and two plots showing the normalized pitch $F0$ (normalized by mean nonzero pitch, i.e. mean is only over voiced periods) of VDTTS and TTS-TEXTONLY compared to the ground-truth signal. For actual videos we refer the reader to the project webpage.

4.4. In-the-wild evaluation

In this section we evaluate VDTTS on the in-the-wild data from the test set of VoxCeleb2 [5]. This is an open-source dataset made of in-the-wild examples of people speaking and is taken from YouTube. We preprocess the data as described in Sec. 4.2. Since this data is not transcribed, we augment the data with transcripts automatically generated using [32], yielding 2,824 high quality, automatically transcribed test videos. We create a speaker embedding lookup table by averaging and normalizing the speaker voice embeddings from all examples of the same speaker.

As a baseline we again use TTS-TEXTONLY, a text-only TTS model from [59]. The results are shown in Table 3.

Initially we trained our model on the train set of VoxCeleb2, called VDTTS-VOXCELEB2. Unfortunately, as can be seen by the high WER of 48%, the model produced difficult-to-comprehend audio. We hypothesized that noisy automated transcripts were the culprit, so trained the model on an alternative in-the-wild dataset with human generated transcripts, LSVSR, we denote this model by VDTTS-LSVSR. As hypothesized, this leads to a great improvement in WER and reduced the error to only 24% while leaving most other metrics comparable. For more details refer to Appendix C.

For qualitative examples of VDTTS-LSVSR we refer the reader to Sec. 4.5.

4.5. Prosody using video

We selected two inference examples from the test set of VoxCeleb2 to showcase the unique strength of VDTTS, which we present in Fig. 4. In both examples, the video frames provide clues about the prosody and word timing. Such visual information is not available to the text-only TTS model, TTS-TEXTONLY [59], to which we compare.

In the first example (see Fig. 4(a)), the speaker talks at a particular pace that results in periodic gaps in the ground-truth mel-spectrogram. The VDTTS model preserves this characteristic and generates mel-spectrograms that are much closer to the ground-truth than the ones generated by TTS-TEXTONLY without access to the video.

Similarly, in the second example (see Fig. 4(b)), the

	LSE-C ↑	LSE-D ↓	WER ↓	MCD ↓	FFE ↓	GPE ↓	VDE ↓
Full VDTTS	**5.92**	**8.25**	**25%**	12.23	**0.46**	**0.29**	**0.31**
VDTTS-no-sp-emb	1.49	12.14	27%	14.5	0.67	0.43	0.37
VDTTS-small	1.48	12.45	38%	14	0.6	0.4	0.43
VDTTS-sum-att	5.74	8.47	28%	**12.22**	**0.46**	**0.29**	**0.31**
VDTTS-no-text	5.90	8.28	98%	12.99	0.53	0.35	0.35
VDTTS-no-video	1.44	12.62	27%	14.36	0.58	0.34	0.47
VDTTS-video-len	1.58	12.37	28%	13.98	0.59	0.37	0.42
VDTTS-mouth	5.51	8.59	29%	12.24	0.52	0.41	**0.31**

Table 5. **Ablation study**, showing different variations of the VDTTS model and hence the contribution of these components to the performance of VDTTS. See Sec. 4.6 for a detailed explanation of the different models, and Sec. 4.1 for definitions of metrics. Arrows indicate if higher or lower is better.

speaker takes long pauses between some of the words. This can be observed by looking at the gaps in the ground-truth mel-spectrogram. These pauses are captured by VDTTS and are reflected in the predicted result below, whereas the mel-spectrogram of TTS-TEXTONLY does not capture this aspect of the speaker's rhythm.

We also plot $F0$ charts to compare the pitch generated by each model to the ground-truth pitch. In both examples, the $F0$ curve of VDTTS fits the ground-truth much better than the TTS-TEXTONLY curve, both in the alignment of speech and silence, and also in how the pitch changes over time.

To view the videos and other examples, we refer the reader to the project page[1].

4.6. Ablation

In this section we conduct an ablation study to better understand the contribution of our key design choices.

Results are presented in Table 5 using the following abbreviations for the models: (1) *VDTTS-no-sp-emb*: VDTTS without the use of a speaker embedding. Although unlikely, this version could possibly learn to compensate for the missing embedding using the person in the video. (2) *VDTTS-small*: VDTTS with smaller encoders and decoder, with $D_m = 512$ as in [9]. (3) *VDTTS-sum-att*: VDTTS using a summation (as in [44]) instead of concatenation in the Multi Source Attention mechanism. (4) *VDTTS-no-text*: VDTTS without text input, can be thought of as a silent-video-to-speech model. (5) *VDTTS-no-video*: VDTTS without video input, can be thought of as a TTS model. (6) *VDTTS-video-len*: VDTTS trained with empty frames, used as a baseline of a TTS model which is aware of the video length. (7) *VDTTS-mouth*: VDTTS which is trained and evaluated on the mouth region only (as in most speech recognition models).

VDTTS-no-sp-emb performs poorly on the video-speech synchronization metrics LSE-C and LSE-D, likely due to underfitting since the model is unable to infer the voice of the speaker using only the video.

Looking at *VDTTS-small*, makes it evident that increasing D_m beyond what was originally suggested by Ding et al. [44] is required.

Another interesting model is *VDTTS-no-text*, which has access only to the video frame input without any text. In terms of video-speech synchronization it is on par with the full model for LSE-C and LSE-D, but fails to produce words as can be seen by its high WER. Intriguingly, outputs from this model look clearly synchronized, but sounds like English babbling, as can be seen in the examples on the project page[1]. On one hand, this shows that the text input is necessary in order to produce intelligible content, and on the other hand it shows the video is sufficient for inferring synchronization and prosody without having access to the underlying text.

Although it seems that *VDTTS-video-len* shows similar results to the *VDTTS-no-video* model, the former produces speech signal which corresponds to the original scene length (as desired), which the latter does not.

Lastly, the *VDTTS-mouth* performs a bit worse than the full model, which suggests that the use of the full face crop is indeed beneficial to the model.

5. Discussion and Future Work

In this paper we presented VDTTS, a novel visually-driven TTS model that takes advantage of video frames as an input and generates speech with prosody that matches the video signal. Such a model can be used for post-sync or dubbing, producing speech synchronized to a sequence of video frames. Our method also naturally extends to other applications such as low-quality speech enhancement in videos and audio restoration in captioned videos.

VDTTS produces near ground-truth quality on the GRID dataset. On open-domain "in-the-wild" evaluations, it produces well-synchronized outputs approaching the video-speech synchronization quality of the ground-truth and performs favorably compared to alternate approaches.

Intriguingly, VDTTS is able to produce video-synchronized speech without any explicit losses or constraints to encourage this, suggesting complexities such as synchronization losses or explicit modeling are unnecessary. Furthermore, we demonstrated that the text and speaker embedding supply the speech content and voice, while the prosody is produced by the video signal. Our results also suggest that the "easiest" solution for the model to learn is to infer prosody visually, rather than modeling it from the text.

In the context of synthesis it is important to address the potential for misuse by generating convincingly false audio. Since VDTTS is trained using video and text pairs in which the speech pictured in the video corresponds to the text spoken, synthesis from arbitrary text is out-of-domain, making it unlikely to be misused.

References

[1] V. Canby, "Lucas returns with "The Jedi"," *The New York Times*, p. 24, May 1983. 1

[2] Y. Yang, B. Shillingford, Y. Assael, M. Wang, W. Liu, Y. Chen, Y. Zhang, E. Sezener, L. C. Cobo, M. Denil, Y. Aytar, and N. de Freitas, "Large-scale multilingual audio visual dubbing," *arXiv:2011.03530 [cs, eess]*, Nov. 2020. 1, 2, 3

[3] B. Shillingford, Y. Assael, M. W. Hoffman, T. Paine, C. Hughes, U. Prabhu, H. Liao, H. Sak, K. Rao, L. Bennett *et al.*, "Large-scale visual speech recognition," *arXiv preprint arXiv:1807.05162*, 2018. 1, 2, 3, 5, 6

[4] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006. 1, 2, 3, 4, 6

[5] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018. 2, 3, 4, 7

[6] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016. 2

[7] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017. 2

[8] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "PnG BERT: Augmented BERT on phonemes and graphemes for neural TTS," in *INTERSPEECH*, 2021. 2

[9] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *ICASSP*, 2018. 2, 4, 8

[10] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *NeurIPS*, 2018. 2, 4

[11] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *NeurIPS*, 2019. 2

[12] Y. Ren, C. Hu, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text-to-speech," in *ICLR*, 2021. 2

[13] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-Attentive Tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling," *arXiv preprint arXiv:2010.04301*, 2020. 2

[14] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Weiss, and Y. Wu, "Parallel Tacotron: Non-autoregressive and controllable TTS," in *ICASSP*, 2021. 2

[15] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Skerry-Ryan, and Y. Wu, "Parallel Tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling," in *INTERSPEECH*, 2021. 2

[16] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984. 2

[17] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *arXiv preprint arXiv:1910.06711*, 2019. 2

[18] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621. 2

[19] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," *arXiv preprint arXiv:2009.09761*, 2020. 2

[20] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "Wavegrad: Estimating gradients for waveform generation," *arXiv preprint arXiv:2009.00713*, 2020. 2

[21] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *arXiv preprint arXiv:2107.03312*, 2021. 2, 3

[22] R. J. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron," *arXiv:1803.09047 [cs, eess]*, Mar. 2018. 2

[23] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis," *arXiv:1803.09017 [cs, eess]*, Mar. 2018. 2

[24] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," *arXiv:1812.04342 [cs, eess]*, Feb. 2019. 2

[25] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, "Hierarchical Generative Modeling for Controllable Speech Synthesis," *arXiv e-prints*, Oct. 2018. 2

[26] A. Ephrat and S. Peleg, "Vid2speech: speech reconstruction from silent video," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5095–5099. 2

[27] Y. Kumar, R. Jain, K. M. Salik, R. R. Shah, Y. Yin, and R. Zimmermann, "Lipper: Synthesizing thy speech using multi-view lipreading," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 2588–2595. 2

[28] R. Mira, K. Vougioukas, P. Ma, S. Petridis, B. W. Schuller, and M. Pantic, "End-to-end video-to-speech synthesis using generative adversarial networks," *arXiv preprint arXiv:2104.13332*, 2021. 2

[29] K. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, "Learning individual speaking styles for accurate lip to speech synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 796–13 805. 2

[30] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3444–3453. 2, 3

[31] T. Afouras, J. S. Chung, and A. Zisserman, "Lrs3-ted: a large-scale dataset for visual speech recognition," *arXiv preprint arXiv:1809.00496*, 2018. 2, 3

[32] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *arXiv preprint arXiv:2005.09629*, 2020. 2, 5, 7

[33] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 87–103. 3

[34] A. Lahiri, V. Kwatra, C. Frueh, J. Lewis, and C. Bregler, "Lipsync3d: Data-efficient learning of personalized 3d talking faces from video using pose and lighting normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2755–2764. 3

[35] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: learning lip sync from audio," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 95, 2017. 3

[36] R. Kumar, J. Sotelo, K. Kumar, A. de Brébisson, and Y. Bengio, "Obamanet: Photo-realistic lip-sync from text," *arXiv preprint arXiv:1801.01442*, 2017. 3

[37] P. KR, R. Mukhopadhyay, J. Philip, A. Jha, V. Namboodiri, and C. Jawahar, "Towards automatic face-to-face translation," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1428–1436. 3

[38] L. Song, W. Wu, C. Qian, R. He, and C. C. Loy, "Everybody's talkin': Let me talk as you want," *arXiv preprint arXiv:2001.05201*, 2020. 3

[39] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala, "Text-based editing of talking-head video," *arXiv preprint arXiv:1906.01524*, 2019. 3

[40] H. Kim, M. Elgharib, M. Zollhöfer, H.-P. Seidel, T. Beeler, C. Richardt, and C. Theobalt, "Neural style-preserving visual dubbing," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–13, 2019. 3

[41] A. Jha, V. Voleti, V. Namboodiri, and C. Jawahar, "Cross-language speech dependent lip-synchronization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7140–7144. 3

[42] J. Lu, B. Sisman, R. Liu, M. Zhang, and H. Li, "Visualtts: Tts with accurate lip-speech synchronization for automatic voice over," *arXiv preprint arXiv:2110.03342*, 2021. 3, 6

[43] C. Hu, Q. Tian, T. Li, Y. Wang, Y. Wang, and H. Zhao, "Neural dubber: Dubbing for silent videos according to scripts," *arXiv preprint arXiv:2110.08243*, 2021. 3

[44] S. Ding, Y. Jia, K. Hu, and Q. Wang, "Textual echo cancellation," *arXiv preprint arXiv:2008.06006*, 2020. 3, 4, 8

[45] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883. 4

[46] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013. 4

[47] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural tts," *arXiv preprint arXiv:1906.00672*, 2019. 4

[48] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702. 4, 5

[49] A. Polyak and L. Wolf, "Attention-based wavenet autoencoder for universal voice conversion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 6800–6804. 4

[50] A. De Cheveigne and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America, 111(4): 1917–1930*, 2002. 4

[51] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," pp. 125–128, 1993. 5

[52] V. Tiwari, "Mfcc and its applications in speaker recognition," *International journal on emerging technologies*, vol. 1, no. 1, pp. 19–22, 2010. 5

[53] B. Sisman, J. Yamagishi, S. King, and H. Li, "An overview of voice conversion and its challenges: From statistical modeling to deep learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020. 5

[54] W. Chu and A. Alwan, "Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3969–3972. 5

[55] A. S. I. T. I. K. Nakatani, Tomohiro and T. Kondo, "A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments," *Speech Communication, 50(3)*, pp. 203–214, 2008. 5

[56] V. P. N. KR Prajwal, Rudrabha Mukhopadhyay and C. Jawahar, "A lip sync expert is all you need for speech to lip generation in the wild," in *28th ACM International Conference on Multimedia*, 2020, pp. 484–492. 5

[57] J. S. Chung and A. Zisserman, "Out of time: automated lip sync in the wild," in *PWorkshop on Multi-view Lip-reading (ACCV)*, 2016. 5

[58] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210. 5

[59] Y. Jia, M. T. Ramanovich, T. Remez, and R. Pomerantz, "Translatotron 2: Robust direct speech-to-speech translation," *arXiv preprint arXiv:2107.08661*, 2021. 6, 7

[60] A. Salcianu, A. Golding, A. Bakalov, C. Alberti, D. Andor, D. Weiss, E. Pitler, G. Coppola, J. Riesa, K. Ganchev *et al.*, "Compact language detector v3," 2018. 5

[61] J. S. Chung and A. Zisserman, "Lip reading in profile," 2017. 5

[62] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823. 5

[63] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, and J. R. Hershey, "Universal sound separation," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019, pp. 175–179. 6

[64] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019. 6, 13