Expressive Neural Voice Cloning

Paarth Neekhara* PNEEKHAR@ENG.UCSD.EDU
Shehzeen Hussain* SSH028@ENG.UCSD.EDU
Shlomo Dubnov SDUBNOV@UCSD.EDU
Farinaz Koushanfar FKOUSHANFAR@ENG.UCSD.EDU
Julian McAuley JMCAULEY@ENG.UCSD.EDU
University of California, San Diego, 9500 Gilman Dr, La Jolla, CA 92093
** Denotes Equal Contribution*

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Voice cloning is the task of learning to synthesize the voice of an unseen speaker from a few samples. While current voice cloning methods achieve promising results in Text-to-Speech (TTS) synthesis for a new voice, these approaches lack the ability to control the expressiveness of synthesized audio. In this work, we propose a controllable voice cloning method that allows fine-grained control over various style aspects of the synthesized speech for an unseen speaker. We achieve this by explicitly conditioning the speech synthesis model on a speaker encoding, pitch contour and latent style tokens during training. Through both quantitative and qualitative evaluations, we show that our framework can be used for various expressive voice cloning tasks using only a few transcribed or untranscribed speech samples for a new speaker. These cloning tasks include style transfer from a reference speech, synthesizing speech directly from text, and fine-grained style control by manipulating the style conditioning variables during inference. [1]

Keywords: TTS, Voice Cloning, Expressive TTS, Deep Learning

1. Introduction

Recent research efforts in voice cloning have focused on synthesizing a person's voice from only a few reference audio samples. While such a system can generate speech from text for a new speaker, it leaves out control over various style aspects of speech. Explicit control over the style aspects of cloned speech is desirable for several applications, such as: voice-overs in animated films, synthesizing realistic and expressive speech for DeepFake videos, translating speech from one language to another while preserving speaking style and speaker identity, advertisement campaigns with expressive speech in multiple voices and languages (etc.). Expressive voice cloning systems can also help create personalized speech interfaces with voice assistants in smartphones, cars, and home assistants. Since speech serves as a primary communication interface between machine learning agents and humans, the ability to speak *expressively* is a very desirable quality for voice cloning systems. Furthermore, such systems can potentially empower individuals who have lost their ability to speak.

1. Audio examples: https://expressivecloning.github.io/
Interactive Demo: https://expressivecloning.github.io/app.html

The goal of voice cloning is commonly formulated as learning to synthesize the voice of an unseen speaker using only a few seconds of transcribed or untranscribed speech. This is typically done by embedding speaker-dependent information from the available speech samples of the new speaker, and conditioning a trained multi-speaker Text-to-Speech (TTS) model on the derived speaker embedding Arik et al. (2018); Jia et al. (2018). While such a system can achieve promising results in closely retaining speaker-specific characteristics in the cloned speech, it does not offer control over other aspects of speech that are not contained in the text or the speaker-specific embedding. These aspects include variation in tone, speaking rate, emphasis and emotions.

Several past works have focused on the problem of expressive TTS synthesis by learning latent variables for controlling the style aspects of speech synthesized for a given text Wang et al. (2018); Skerry-Ryan et al. (2018). Such models are usually trained on a single-speaker expressive speech dataset to learn meaningful latent codes for various style aspects of the speech. Recent works Stanton et al. (2018); Valle et al. (2020), have extended the idea of learning style representations to a multi-speaker setting by conditioning the TTS synthesis model on both speaker identity and style encodings. Such techniques show promise in disentangling style and speaker specific information, and generate different style variants of synthesized speech for the same text and speaker. However, these methods are limited by the speakers used in the training set and cannot be directly used for synthesizing voices of speakers not seen during training.

Adapting multi-speaker TTS models for voice cloning requires scaling up model training to a large multi-speaker TTS dataset, containing several minutes of transcribed speech from thousands of speakers. High speaker diversity in the training data is important to achieve generalization on unseen speakers Arik et al. (2018); Jia et al. (2018). *The goal of our work is to perform TTS synthesis for an unseen speaker with control over the style aspects of generated speech.* As a first step in this direction, we train a TTS model conditioned on speaker encodings and latent style tokens Wang et al. (2018) on a large multi-speaker dataset. While this model is able to generate voices for unseen speakers, we find that the results fall short in terms of speech naturalness and style control during synthesis. Our results suggest that learning meaningful latent style aspects is difficult when training on a large multi-speaker dataset containing speech with mostly neutral style and expressions.

To address problem of disentangling style and speaker characteristics on a large multi-speaker dataset containing mostly style-neutral speech, we propose a voice cloning model that is conditioned on both latent and heuristically derived style information. Specifically, we condition our TTS synthesis model on (i) text, (ii) speaker encoding (iii) pitch contour of the target speech and (iv) latent style tokens Wang et al. (2018). By conditioning synthesis on various style aspects and speaker embeddings derived from the target speech, we are able to train a model that offers fine-grained style control for synthesized speech. To adapt inference for an unseen speaker, we can either perform zero-shot inference or fine-tune the synthesis model on the limited text and speech pairs for the new speaker. Through both quantitative and qualitative evaluations, we demonstrate that our proposed model can make a new voice express, emote, sing or copy the style of a given reference speech.

The main **contributions** of this study are as follows:

- We introduce the problem of expressive voice cloning — Synthesizing a person's voice from a few audio samples and allowing control over the style aspects of the synthesized

speech. While past works have studied the problem of expressive TTS, they have not investigated the problem of cloning a new speaker's voice in an expressive manner.

- We propose an expressive voice cloning framework by training a controllable TTS model on a large multi-speaker dataset with mostly style-neutral speech. By allowing explicit control over the speaker encoding, pitch and rhythm of the synthesized speech, we are able to generate expressive speech for a new speaker that generalizes beyond the distribution of the training data.
- We develop three benchmark tasks and define metrics to evaluate expressive voice cloning systems in terms of speaker similarity, style similarity and naturalness of synthesized speech. We demonstrate that our proposed framework significantly outperforms baseline models that do not use explicit pitch contours for training.

2. Background and Related Work

Neural TTS: State-of-the-art neural approaches for natural TTS synthesis Ping et al. (2018a); Shen et al. (2018) typically decompose the waveform synthesis pipeline into two steps: (1) Synthesizing perceptually informed mel-spectrograms from language using an attention based sequence-to-sequence model like Tacotron Wang et al. (2017) or Tacotron 2 Shen et al. (2018). (2) Vocoding the synthesized spectrograms to audible waveforms using a neural vocoder van den Oord et al. (2016); Prenger et al. (2018); Neekhara et al. (2019) or heuristic methods like the Griffin-Lim Griffin et al. (1984) algorithm. Multi-speaker TTS models Gibiansky et al. (2017); Ping et al. (2018b) extend this line of work by additionally conditioning the spectrogram synthesis model on speaker embeddings, which are trained end-to-end using the speaker labels in the TTS dataset. While these approaches achieve promising results in synthesizing speech for multiple speakers for a given text, they cannot be directly used to synthesize voices of speakers not seen during training.

Voice Cloning: Voice cloning focuses on generative modeling of speech conditioned on a speaker encoding derived from a few reference speaker audio samples. While speech synthesis models exist van den Oord et al. (2016); Wang et al. (2017), it has been a challenge to adapt these voice models to new speakers with limited data. Recent efforts have been made in designing systems that can learn to synthesize a person's voice from only a few audio samples Arik et al. (2018); Chen et al. (2019); Cooper et al. (2020); Huang et al. (2020); Jia et al. (2018). They train a separate speaker encoding network to condition a multi-speaker TTS model on speaker dependent information. Since the speaker encoding network operates on waveforms, it can be used for zero-shot voice cloning from untranscribed utterances of a target speaker. Additionally, the authors of Arik et al. (2018) demonstrate that the synthesis model can be fine-tuned on limited text and audio pairs of a new speaker to improve the speaker similarity of the cloned speech.

Expressive Speech Synthesis: Prior works Wang et al. (2018); Stanton et al. (2018); Skerry-Ryan et al. (2018) on expressive speech synthesis focus on models that can be conditioned on text and a latent embedding for style or prosody. During training, the style embeddings are derived using a learnable module called *Global Style Tokens (GST)*, that operates on the target speech for a given phrase and derives a style embedding through attention over a dictionary of learnable vectors. During inference, the synthesizer can be

conditioned on different reference audios to produce style variants of speech for the same text. Manipulating these latent style variables during inference offers some coarse control over the style of the synthesized speech. Recently proposed Mellotron model Valle et al. (2020) uses a combination of explicit and latent style variables to offer more fine-grained control over the expressive characteristics of synthesized speech. Specifically, Mellotron conditions the spectrogram synthesis network on pitch contour, GSTs Wang et al. (2018) and speaker ID during training. During inference, the synthesizer can be conditioned on the melodic information—pitch and rhythm of a reference speech and synthesize speech in the voice of a given speaker in the training set. The authors demonstrate that explicit conditioning on pitch contour during training phase, makes it possible to generalize the inference on various melodic pitch contours. While Mellotron allows expressive TTS for speakers in the training dataset, since it uses a fixed size speaker embedding matrix for speaker conditioning, it cannot be used to generate speech for new speakers.

3. Methodology

Our expressive voice cloning framework is a multi-speaker TTS model that is conditioned on speaker encodings and style aspects of speech. Style conditioning in expressive TTS models is popularly done by learning a dictionary of latent style vectors called Global Style Tokens (GST) Wang et al. (2018). While GSTs can learn meaningful latent codes when trained on a dataset with high variation in expressions, we empirically find that it offers limited style control when trained on a large multi-speaker dataset with mostly neutral prosody.

Signal processing heuristics like the Yin algorithm De Cheveigné and Kawahara (2002) can derive the fundamental frequency contour (pitch contour) and voicing decisions from speech, which can be useful for expressive speech synthesis. We find that using a combination of latent and heuristically derived style information in the TTS model not only provides fine-grained control over the style aspects of synthesized speech, but also scales up to a large multi-speaker dataset to produce more natural sounding audio for an unseen speaker. A high level overview of our expressive voice cloning framework is shown in Figure 1. Similar to past works on voice cloning Arik et al. (2018); Jia et al. (2018), the three main components *Speaker Encoder*, *Mel Spectrogram Synthesizer* and *Vocoder* are all trained separately. We describe the individual components of our framework and their training objectives in the following sections.

3.1. Speaker Encoder

Speaker conditioning in multi-speaker TTS models is usually done using a lookup in the speaker embedding matrix which is randomly initialized and trained end-to-end with the synthesizer. While such a framework learns speaker-specific information via the embedding vectors, synthesis cannot be generalized to unseen speakers. To adapt the multi-speaker TTS model for the goal of voice cloning, the speaker embedding layer can be replaced with a speaker encoder that derives speaker specific information from the target waveform. In this setting, the speaker encoder can obtain embeddings for speakers not seen during training using a few reference speech samples. To obtain meaningful embeddings, the speaker encoder should be trained to discriminate between different speakers for the task of speaker verification Wan et al. (2017).

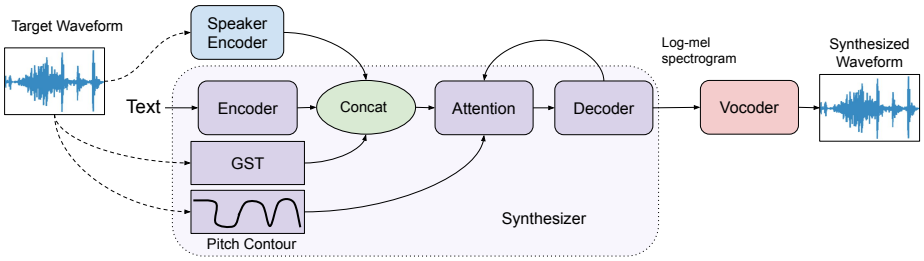

Figure 1: Expressive Voice Cloning Model: Tacotron-2 TTS model conditioned on speaker and style characteristics derived from the target audio of a given text. At inference time, the model can be provided independent references for style and speaker encodings to achieve expressive voice cloning.

We follow the speaker encoder architecture described in Wan et al. (2017); Louppe (2019b). The network is a stack of 3 LSTM layers with 256 cells in each layer that operate on mel-spectrograms with 40 channels. The final speaker embedding is obtained by projecting the LSTM output at the last layer to 256 dimensions followed by L_2 normalization. Note that ours is a smaller model than that used in Jia et al. (2018) which had 768 cells in each LSTM layer. The speaker encoder is trained to optimize a generalized end-to-end speaker verification loss Wan et al. (2017), that encourages high cosine similarity between embeddings from same speaker and low similarity between different speaker embeddings. During inference, each utterance is broken into smaller segments of 1,600 ms with 1,000 ms overlap between consecutive segments. The final embedding is estimated by averaging the embedding of each individual segment.

3.2. Mel-Spectrogram Synthesizer

The goal of our synthesis model is to disentangle the style and speaker-specific information in speech by conditioning our TTS synthesis model on the speaker encoding and various style aspects. To this end, we adapt the synthesis model used in Mellotron Valle et al. (2020) for the task of voice cloning. Mellotron is a multi-speaker TTS model that extends Tacotron 2 GST Wang et al. (2018) by additional conditioning on pitch contours and speaker embeddings. To adapt Mellotron for voice cloning, we remove the speaker embedding layer and replace it with the speaker encoder network described in Section 3.1.

At its core, our synthesis model based on Tacotron 2 Shen et al. (2018), is an LSTM based sequence-to-sequence model composed of an encoder that operates on a sequence of characters and a decoder that generates the individual frames of the mel spectrogram while attending over the encoded representations. Along with the encoded representation for text, we concatenate the speaker encoding (obtained from the speaker encoder) and the GST embedding at each time-step. The GST embedding is obtained by querying a dictionary of latent style vectors with the target mel-spectrogram using a multi-headed attention mechanism described in Wang et al. (2018). Decoding occurs in an autoregressive manner where we synthesize one mel spectrogram frame at a time by providing the fundamental frequency (from the pitch contour) and the mel spectrogram of the previous frame as the

input to the decoder. The pitch contours are derived from the target speech using the Yin algorithm with harmonicity thresholds between 0.1 and 0.25.

In this way, we can factor mel-spectrogram synthesis into the following variables: *text (t)*, *speaker encoding (s)*, *pitch contour (f_0)* and *latent style embedding obtained from GST (z)*. Formally, our synthesizer is a generative model $g(t, s, f_0, z; W)$ that is parameterized by trainable weights W, trained to optimize a loss function L that penalizes the differences between the generated and ground truth mel spectrogram. That is,

$$\min_{W} \ \mathbb{E}_{(t_i,a_i)\sim D} \left\{ L(g(t_i, s_i, f_{0i}, z_i; W), mel_i) \right\} \tag{1}$$

where D is the dataset containing text and audio pairs (t_i, a_i). The variables $(s_i, f_{0i}, z_i, mel_i)$ are all derived from the target waveform a_i. For the loss function L, we use the L2 loss between the generated and ground truth mel spectrograms.

During training, the synthesizer learns another latent variable: the attention map between the encoder and decoder states which captures the alignment between text and audio. Following the notation used in Valle et al. (2020), we call this latent variable *rhythm*, since it controls the timing aspects of synthesized speech. Note that unlike other style aspects which can be obtained directly from a_i, deriving *rhythm* requires both text and audio (t_i, a_i). In our experiments, we obtain the *rhythm* by using our synthesizer as a forced-aligner. That is, for a given text and audio pair, we derive the attention map between the encoder and decoder states by doing a forward pass through our model using teacher forcing. Therefore, during inference, our synthesizer g can be explicitly conditioned on rhythm r derived from some text and audio pair: $g(t, s, f_0, z, \boldsymbol{r}; W)$.

While the style aspects are obtained from the target waveform of the same speaker during training, we can use a different reference audio and text pair during inference. For example, we can transfer the pitch contour and rhythm of a style reference audio S from a different speaker to the voice of a given target speaker T as follows:

$$mel = g(t_S, s_T, f_{0S}, z_T, r_S; W) \tag{2}$$

The output *mel* should have the same pitch and rhythm as the style reference S and should retain the latent style aspects and voice of the target speaker T. In our work we focus on three different cloning tasks with different sources of style conditioning information which we discuss in Section 4.3.

Additionally, to assess the importance of pitch contours during training, we train another TTS model that is conditioned only on the latent style aspects obtained using GST. We use the same Tacotron2 architecture and GST module as our proposed model. Formally, this alternative synthesizer $g(t, s, z; W)$ is trained to optimize the same objective as Equation 1:

$$\min_{W} \ \mathbb{E}_{(t_i,a_i)\sim D} \left\{ L(g(t_i, s_i, z_i; W), mel_i) \right\} \tag{3}$$

We refer to this alternative model as *Tacotron2 + GST* in our experiments. Similar to our proposed system, this model can also be additionally conditioned on rhythm. Since we are not explicitly conditioning the model on pitch contours, we expect the pitch variation in speech to be captured as part of the latent style tokens. We empirically demonstrate that using only latent style representation on a large multi-speaker dataset with neutral prosody offers limited style control and audio naturalness.

Vocoder: For decoding the synthesized mel-spectrograms into listenable waveforms, we use the WaveGlow Prenger et al. (2018) model trained on the single speaker Sally dataset Valle et al. (2020). An advantage of WaveGlow over WaveNet van den Oord et al. (2016) is that it allows real-time inference, while being competitive in terms of audio naturalness. The same vocoder model is used across all experiments and datasets. We find that the vocoder model trained on a single speaker generalizes well across all speakers in our datasets.

3.3. Cloning Techniques: Zero-Shot and Model Adaptation

We adopt the following two approaches for cloning the voice of a new speaker from a few transcribed or untranscribed speech samples:

Zero-Shot: For zero-shot voice cloning, we derive the speaker embedding by taking the mean followed by L-2 normalization of the speaker encodings of the individual samples of the target speaker. Since speaker encodings are obtained directly from the waveforms, we do not require audio transcriptions of the new speaker for zero-shot voice cloning.

Model Adaptation: When transcribed samples of a new speaker are available, we can fine-tune our synthesis model using the text and audio pairs. As shown in Neural Voice Cloning Arik et al. (2018), fine-tuning can significantly improve the speaker similarity metrics of the cloned speech. Also, the authors of Arik et al. (2018) observe that fine-tuning the whole synthesis model is faster and more effective than fine-tuning only the speaker embedding layer since more degrees of freedom are allowed in the whole model adaptation. Our preliminary experiments on model adaptation suggested the same. We hypothesize the reason for this is that fine-tuning the last-few layers of the synthesis model is essential, if not sufficient, to adapt the synthesizer to the speaker-specific speech characteristics. Therefore, we study the following two model adaptation techniques: **Adaptation whole -** Fine-tune all the parameters of the synthesis model on the text and audio pairs of the new speaker. **Adaptation decoder -** Fine-tune only the decoder parameters of the synthesis model. The advantage of only adapting the decoder parameters is that it requires fewer speaker-specific model parameters and a shared encoder can be used across all speakers in a real-world deployment setting. In both of the above adaptation settings, we fine-tune our model for 100 to 200 iterations using Adam optimizer with a learning rate of 1e-4. Model adaptation takes up to 6 minutes for fine-tuning on 1 to 20 samples of the target speaker on a single Nvidia Titan 1080 GPU.

4. Experiments

4.1. Datasets and Training

We train our mel-spectrogram synthesis model on the clean subset of the publicly available Libri-TTS Zen et al. (2019) dataset—*train-clean-100* and *train-clean-360*. This clean subset contains around 245 hours of speech across 1151 speakers sampled at 24 kHz. Past works on voice cloning Wan et al. (2017); Arik et al. (2018) trained their synthesis models on the LibriSpeech dataset Panayotov et al. (2015) and empirically demonstrated the importance of a speaker-diverse training dataset for the task of voice cloning. We filter out utterances longer than 10 seconds and resample waveforms to 22050 Hz.

For training the synthesizer, we warm start our model using the pre-trained Mellotron checkpoint which is trained on a subset of LibriTTS containing 123 speakers. The speaker embedding layer is replaced with our speaker encoding network which is kept frozen during training. We use a validation set with 250 examples and train the model using a batch size of 32 and an initial learning rate of 5e-4. We use an Adam optimizer Kingma and Ba (2015) to update the weights and anneal the learning rate to half its value every 50k mini-batch iterations. We include details of our model architecture and hyper-parameters in Section 4.2 and point to our codebase [2] for precise model implementation.

For the *Tacotron 2 + GST* model, we use the same Tacotron 2 architecture and GST hyper-parameters as our proposed model. Training for the proposed model and the *Tacotron 2 + GST* model converged in 210,000 and 185,000 mini-batch iterations respectively and took around 4 seconds per iteration on a single Nvidia Titan 1080 GPU. The Resemblyzer speaker encoder Louppe (2019b,a) used in our experiments is trained on the VoxCeleb Nagrani et al. (2019), VoxCeleb2 Chung et al. (2018) and LibriSpeech-other Panayotov et al. (2015) datasets containing a total of 8.4k speakers. The authors of Louppe (2019a) report a 4.5% Equal Error Rate (EER) for the task of speaker verification using this speaker encoder on their internal test set.

4.2. Model architecture and hyper-parameter details

Our spectrogram synthesizer is closely follows the Tacotron-2 architecture Shen et al. (2018). The model is composed of an encoder and decoder with attention. The encoder processes the input sequence using a stack of 3 1-d convolution layers followed by a bi-directional LSTM. The input tokens are embedded using a 512 dimensional embedding layer. Each convolutional layer has 512 filters with a kernel size of 5 followed by batch normalization and ReLU activation functions. The single bidirectional LSTM consists of 256 hidden units in each direction leading to a 512 dimensional embedding at each time-step. At each time-step of the encoder output, we concatenate the speaker encoding and the GST embedding. The GST module we use follows the same architecture as that proposed in Wang et al. (2018) except that we use 8 attention heads instead of 4.

The attention module follows the location sensitive attention procedure and hyper-parameters described in Shen et al. (2018). The decoder network is an autoregressive network that takes in as input the spectrogam frame and fundamental frequency of the previous time-step. The spectrogram frame goes through a *pre-net* module which consists of 2 fully connected layers with 256 hidden units and ReLU activation. The *pre-net* module acts as an information bottleneck which is essential for learning attention Shen et al. (2018). The decoder is a stack of 2 convolutional layers with 1024 units in each layer. The final predicted mel-spectrogram is passed through a 5 layer convolutional *post-net* which predicts a residual that can be added to improve overall reconstruction error. Each post-net layer uses 512 filters with kernel width of 5 followed by batch normalization and *tanh* activation function.

We use Adam Kingma and Ba (2015) optimizer with hyper-parameters with an initial learning rate 1e-4 and $\beta_1 = 0.9, \beta_2 = 0.999$. During training, we use a mini-batch size of 32 and anneal learning rate to half its value every 50k mini-batch iterations.

2. https://expressivecloning.github.io/

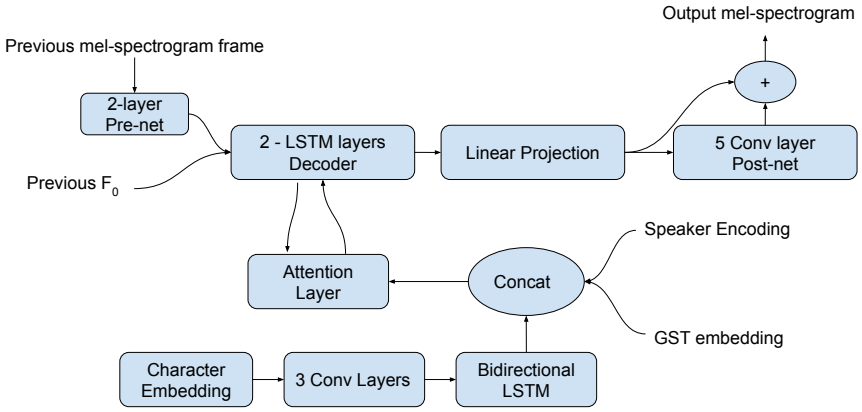

Figure 2: Mel-spectrogram synthesizer model architecture

4.3. Cloning Tasks

In this section, we discuss the three main tasks for which we evaluate our voice cloning methods. When cloning the voice of a new speaker, we require a few audio samples of the speaker to obtain the speaker encoding. We refer to these samples as *target speaker samples*. We perform voice cloning for the speakers in the VCTK dataset Veaux et al. (2017). The VCTK dataset contains speech sampled at 48 KHz from 108 native English speakers, the majority of which have British accents. We down-sampled the audio to 22,050 KHz to make it consistent with our training data. To synthesize the speech for a given speaker encoding and text, our synthesis model additionally requires various style conditioning variables described in Section 3.2. While the latent GST embedding can be obtained from the *target speaker samples*, pitch contour and rhythm information needs to be derived from a *style reference audio* that corresponds to the given text. In case we do not have a style reference audio available, we can synthesize one using a single speaker TTS system. To evaluate our cloning techniques objectively in terms of style and speaker disentanglement, and also assess their usefulness in real world settings, we perform the following cloning tasks:

1. Text *Cloning speech directly from text:* For cloning speech directly from text, we first synthesize speech for the given text using a single speaker TTS model: Tacotron 2 + WaveGlow trained on the LJ Speech Ito (2017) dataset. We then derive the pitch contour of the synthetic speech using the Yin algorithm De Cheveigné and Kawahara (2002) and scale the pitch contour linearly to have the same mean pitch as that of the *target speaker samples*. For deriving rhythm, we use our proposed synthesis model as a forced aligner between the text and Tacotron2-synthesized speech. We use the *target speaker samples* for obtaining the GST embedding for both our proposed model and the baseline Tacotron2 + GST model.

2. Imitation - *Reconstruct a sample from the target speaker:* In this setup, we use a text and audio pair of the target speaker (not contained in the *target speaker samples*), and try to reconstruct the audio from its factorized representation using our synthesis model. All of the style conditioning variables - pitch, rhythm and GST embedding are derived from the

speech sample we are trying to imitate. The imitation task is a toy experiment that allows quantitative evaluation of style similarity metrics between the synthesized speech and style reference.

3. Style Transfer - *Transfer the pitch and rhythm of speech from a different expressive speaker:* The goal of this task is to transfer the pitch and rhythm from some expressive speech to the cloned speech for the target speaker. For this task, we use examples from the single speaker Blizzard 2013 dataset King and Karaiskos (2013) as style references. This dataset contains expressive audio book readings from a single speaker with high variation in emotion and pitch. For our proposed model, we use this *style reference audio* to extract the pitch and rhythm. Similar to the Text task, we scale the pitch contour to have the same mean as that of the *target speaker samples*. In-order to retain speaker-specific latent style aspects, we use *target speaker samples* to extract the GST embedding. For the Tacotron2 + GST model, which does not have explicit pitch conditioning, we use the *style reference audio* for obtaining the GST embedding and the rhythm.

4.4. Results

For the above described cloning tasks, we evaluate three aspects of the cloned speech: i) speaker similarity to the target speaker, ii) style similarity to the reference style and iii) speech naturalness. We encourage the readers to listen to our audio examples referenced in the footnote of the first page to contextualize the following results.

Speaker Classification Accuracy: We train a speaker classifier on the VCTK dataset to classify a given utterance as one of the 108 speakers. The speaker classifier is a two layer neural network with 256 hidden units that takes as input the speaker encoding obtained through our pre-trained speaker encoder network. Similar to Arik et al. (2018), our speaker classifier achieves 100% accuracy on a hold out set containing 200 examples from the VCTK dataset. However, since our classification model and training dataset for the synthesizer are not the same as Arik et al. (2018) (1,151 speakers in ours vs. 2,481 speakers in Arik et al. (2018)), we do not make direct comparisons with their work. We conduct our speaker classification evaluations on all 108 speakers of the VCTK dataset. We clone 25 speech samples per speaker for each task described in Section 4.3. Figure 3 (left) shows the speaker classification accuracy curves for all cloning tasks and techniques with respect to the number of target speaker samples. Our results are consistent with the following findings of Arik et al. (2018)—Model adaptation significantly outperforms the zero-shot voice cloning technique since it allows the model to adjust to the speaker characteristics of the new speaker. More target speaker samples helps improve speaker classification accuracy, although in the zero-shot scenario we do not observe much improvement after 10 target speaker samples.

For zero-shot voice cloning, both Tacotron2-GST and our proposed model achieve similar speaker classification accuracy for *Text* and *Style Transfer* cloning tasks. The accuracy of our proposed model is slightly higher for the imitation task as compared to other tasks for both model adaptation and zero-shot voice cloning. This implies that conditioning on the actual pitch contour of the target speaker improves speaker specific characteristics of the cloned speech. While linear scaling of a reference style pitch contour works well, our findings motivate future research on predicting speaker-specific pitch contours from text and speaker encodings.

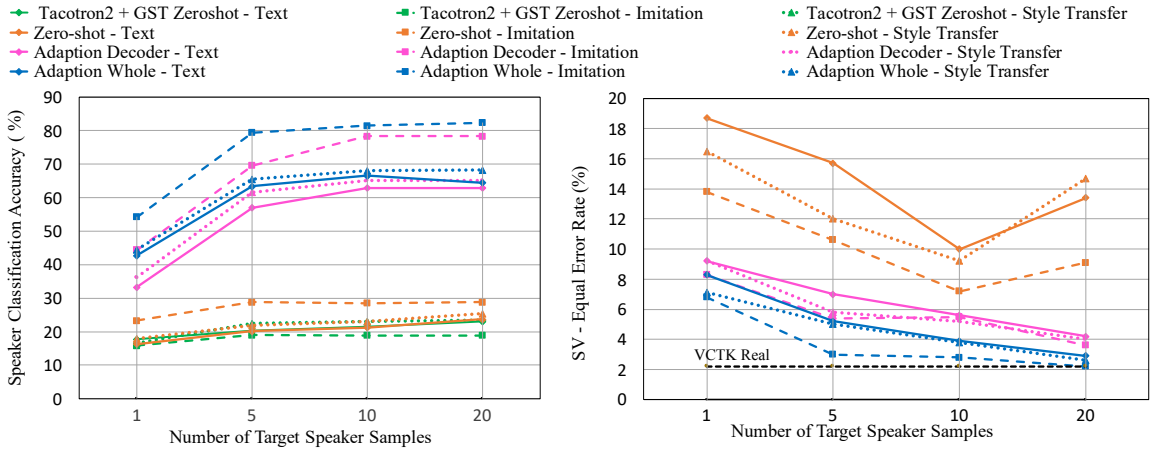

Figure 3: Speaker similarity evaluation of each cloning technique for different voice cloning tasks in terms of Speaker Classification Accuracy and Speaker Verification Equal Error Rate (SV-EER).

Speaker verification Equal Error Rate (SV-EER): SV-EER is another objective metric used to evaluate speaker similarity between the cloned audio and the ground-truth reference audio. We use a speaker verification system that scores the speaker similarity between two utterances based on the cosine similarity of the encodings obtained using the speaker encoder described in Section 3.1. Equal Error Rate (EER) is the point when the false acceptance rate and false rejection rate of the speaker verification system are equal.

We perform speaker verification evaluations on randomly selected 20 speakers in the VCTK dataset. We enroll 5 speech samples per speaker in the speaker verification system and synthesize 50 speech samples per speaker for each cloning task. EERs are estimated by pairing each sample of the same speaker with another sample from a different speaker. Figure 3 shows the plots of SV-EER for different cloning techniques and tasks using our proposed model and also the those estimated using real data. Our observations on the SV-EER metric are similar to those on the accuracy metric. Model adaptation outperforms zero-shot cloning techniques and with more than 10 cloning samples achieves similar EER as the real data. Additionally, we include human evaluation scores on speaker similarity in our supplementary material.

Style Similarity: In order to evaluate the similarity between the style of synthesized and reference audio, we perform quantitative evaluation on the Imitation task described in Section 4.3. We use the following error metrics: Gross Pitch Error (GPE) Nakatani et al. (2008), Voicing Decision Error (VDE) Nakatani et al. (2008) and F0 Frame Error (FFE) Chu and Alwan (2009). Results are presented in Table 1 in which we compare the error values for different approaches when using 10 target speaker samples for cloning. We synthesize 25 speech samples per speaker for all speakers in the VCTK dataset to estimate the reported error values. Our proposed models significantly outperform the Tacotron 2 + GST baseline, clearly indicating the importance of pitch contour conditioning for accurate style transfer.

	Imitation			*Style Transfer*
Approach	GPE	VDE	FFE	Style-MOS
Tacotron2 + GST - Zero-shot	20.37%	26.39%	29.47%	2.69 ± 0.11
Proposed Model - Zero-shot	3.72%	10.65%	11.74%	3.15 ± 0.11
Proposed Model - Adaptation Whole	2.97%	12.58%	13.60%	3.40 ± 0.10
Proposed Model - Adaptation Decoder	2.39%	11.60%	12.51%	3.29 ± 0.10

Table 1: Style similarity evaluations for the imitation and style transfer tasks. We use three objective error metrics (lower values are better). For the style transfer task we present the mean opinion scores on style similarity (Style-MOS) with 95% confidence interval.

We also conduct a crowd-sourced listening test on Amazon Mechanical Turk (AMT) for the style transfer task in which we ask the listeners to rate the style similarity between the ground truth style reference and synthesized audio on a 5 point scale (interface for this study is included in the supplementary material). For each cloning technique (using 10 target speaker samples), we synthesize 25 audio samples per speaker for 20 speakers in the VCTK dataset leading to 500 evaluations of each technique. We present the style similarity Mean Opinion Scores (Style-MOS) in Table 1. It can be seen that our proposed models significantly outperform the Tacotron 2 + GST model. Model adaptation techniques perform slightly better than zero-shot method suggesting that fine-tuning improves the model predictions for an unseen speaker encoding.

Naturalness: To assess speech naturalness, we conducted a crowd-sourced listening test on AMT and asked listeners to rate each audio utterance on a 5-point naturalness scale to collect Mean Opinion Scores (MOS). Similar to the above mentioned user study, we use 10 target speaker samples for each cloning technique. All evaluations are conducted on randomly selected 20 VCTK speakers with 25 audio samples synthesized per speaker. Each sample is rated independently by a single listener leading to 500 evaluations for each technique per cloning task. We report the MOS of Real data and audio synthesized using different cloning techniques in Table 2. Our proposed model significantly outperforms the baseline Tacotron2 + GST model for both zero-shot and model adaptation techniques. This suggests that pitch contour conditioning in a multi-speaker setting helps improve speech naturalness in addition to providing higher style similarity. It can be seen that the naturalness is even further improved with model adaptation techniques since it allows the generative model to adjust for the unseen speaker encodings.

5. Broader Impact

Speech interfaces enable hands-free operation and can assist users who are visually or physically impaired. Research into machine generation of speech is driven by the prospect of offering services where humans interact solely with machines, thereby eliminating the cost of live agents and significantly reducing the cost of providing services. Since speech serves as a primary communication interface between machine learning agents and humans, the ability to speak expressively in a new voice can help create more personalized machine assis-

	Cloning Task		
Approach	Text	Imitation	Style Transfer
Real data VCTK		4.11 ± 0.08	
Real data Blizzard		4.07 ± 0.08	
Tacotron2 + GST - Zero-shot	2.67 ± 0.10	2.51 ± 0.10	3.02 ± 0.09
Proposed Model - Zero-shot	3.56 ± 0.09	3.54 ± 0.10	3.53 ± 0.10
Proposed Model - Adaptation Whole	3.75 ± 0.09	3.71 ± 0.09	3.60 ± 0.09
Proposed Model - Adaptation Decoder	3.61 ± 0.09	3.57 ± 0.09	3.45 ± 0.09

Table 2: Mean Opinion Score (MOS) for speech naturalness with 95% confidence intervals.

tants. Furthermore, such systems can also empower individuals who have lost their ability to speak.

Explicit control over the style aspects of cloned speech is also desirable for several multimedia applications. These include: voice overs in animated films, synthesizing realistic and expressive speech for videos, translating speech from one language to another while preserving the speaking style and speaker identity, advertisement and political campaigns with expressive speech in multiple voices or languages, etc.

Our intent for generating expressive speech is to advance the research of synthetic audio generation, such that it can aid in the accessibility of speech interfaces and support users with speech impairments, as well as contribute to mainstream use in movies, digital storytelling and modern-day streaming services. This work provides us with an opportunity to collaborate with researchers for advancing multi-disciplinary investigation of AI techniques. However, any emerging technology can also be abused. Realistic voice cloning technology can be used to create voice-overs for subjects of DeepFake videos, and has the potential to be used maliciously to spread disinformation or for creating inappropriate content. Also, the technology can be abused for circumventing speech based user authentication systems in smart devices. We seek to discourage the unethical use of our technology. Upon the release of public access to our voice cloning app, we plan to incorporate techniques to watermark the speech generated by our model. This will allow us to thwart the misuse of our technology and curb any spread of misinformation using our platform. It is our intention to develop Expressive Voice Cloning in a way that its potential for abuse is minimized and maximise its use as a tool for learning, education and experimentation.

6. Conclusion and Future Work

In this work we introduce an expressive voice cloning and define three benchmark tasks to evaluate such systems. We empirically find that learning only latent style tokens is insufficient to capture expressiveness in speech when training the synthesis model on a speaker-diverse dataset with mostly neutral prosody. Our proposed model uses a combination of heuristically derived and latent style information, which not only offers fine-grained control over style aspects but also improves speech naturalness. We demonstrate that our proposed model can successfully extract and transfer style and speaker characteristics from

unseen audio references to the synthesized speech. We recommend future works on models for predicting speaker specific pitch contours directly from style labels (like *happy*, *sad*, *neutral* etc) and text to allow control over expressions of the synthesized speech when a style reference audio is not available.

References

Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *NeurIPS*. 2018.

Mengnan Chen, Minchuan Chen, Shuang Liang, Jun Ma, Lei Chen, Shaojun Wang, and Jing Xiao. Cross-lingual, multi-speaker text-to-speech synthesis using neural speaker embedding. In *INTERSPEECH*, 2019.

Wei Chu and Abeer Alwan. Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend. In *ICASSP*. IEEE, 2009.

J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

E. Cooper, C. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *ICASSP*, 2020.

Alain De Cheveigné and Hideki Kawahara. Yin, a fundamental frequency estimator for speech and music. 2002.

Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. In *NIPS*, 2017.

Daniel W. Griffin, Jae, S. Lim, and Senior Member. Signal estimation from modified short-time Fourier transform. *IEEE Trans. Acoustics, Speech and Sig. Proc*, 1984.

Y. Huang, L. He, W. Wei, W. Gale, J. Li, and Y. Gong. Using personalized speech synthesis and neural language generator for rapid speaker adaptation. In *ICASSP*, 2020.

Keith Ito. The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/, 2017.

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

Simon J. King and Vasilis Karaiskos. The blizzard challenge 2013. In *In Blizzard Challenge Workshop*, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Gilles Louppe. Master thesis : Automatic multispeaker voice cloning. 2019a.

Gilles Louppe. *Resemblyzer - https://github.com/resemble-ai/Resemblyzer/*, 2019b. URL https://github.com/resemble-ai/Resemblyzer/.

Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019.

Tomohiro Nakatani, Shigeaki Amano, Toshio Irino, Kentaro Ishizuka, and Tadahisa Kondo. A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments. *Speech Communication*, 2008.

Paarth Neekhara, Chris Donahue, Miller Puckette, Shlomo Dubnov, and Julian McAuley. Expediting TTS synthesis with adversarial vocoding. In *INTERSPEECH*, 2019.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*. IEEE, 2015.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep Voice 3: Scaling text-to-speech with convolutional sequence learning. In *ICLR*, 2018a.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ö. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John L. Miller. Deep voice 3: 2000-speaker neural text-to-speech. In *ICLR*, 2018b.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: A flow-based generative network for speech synthesis. In *ICASSP*, 2018.

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 *ICASSP*, 2018.

R. J. Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J. Weiss, Rob Clark, and Rif A. Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *arXiv:1803.09047*, 2018.

Daisy Stanton, Yuxuan Wang, and R. J. Skerry-Ryan. Predicting expressive speaking style from text in end-to-end speech synthesis. *arXiv:1803.09017*, 2018.

Rafael Valle, Jason Li, Ryan Prenger, and Bryan Catanzaro. Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. *ICASSP*, 2020.

Aäron 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:1609.03499*, 2016.

Christophe Veaux, Junichi Yamagishi, and Kirsten Macdonald. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2017.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. *arXiv:1710.10467*, 2017.

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. In *INTERSPEECH*, 2017.

Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A. Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv:1803.09017*, 2018.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *INTERSPEECH*, 2019. doi: 10.21437/Interspeech.2019-2441.