Survey: Text-to-Speech Synthesis

Shyam Thombre
CFILT
IIT Bombay, Mumbai
shyam.thombre@iitb.ac.in

Pushpak Bhattacharyya
CFILT
IIT Bombay, Mumbai
pb@cse.iitb.ac.in

Preethi Jyothi
CSALT
IIT Bombay, Mumbai
pjyothi@cse.iitb.ac.in

Abstract

Speech synthesis is the task of generating a speech signal corresponding to a given input text. The output is expected to sound natural (prosody, intonation, etc., must be similar to that of a native speaker) and intelligible (the pronunciations must be correct). The state-of-the-art text-to-speech (TTS) architectures generally follow a three-stage process– transliterating the input graphemes to phonemes using a phonemizer, converting the phoneme sequences to time-frequency representation mel-spectrograms, and finally generating the raw speech waveforms using the mel-spectrograms. In this paper, we look at the various architectures that were developed for the text-to-mel and vocoder models. We start with the classical approaches– diphone-based and corpus-based speech synthesis– and proceed towards the recent deep learning based solutions.

1 Introduction

Language is the primary mode of communication for human beings. Although most animals communicate in their own way, humans are the only ones who have excelled at cognitive language communication. This has been a crucial aspect in the overall development of humanity through ages. Language allowed humans to pass down their experiences and learnings to the next generations, warning them of dangers as well as providing them with huge pool of wisdom. For thousands of years, humans have been trying to understand the world around them and sharing the acquired knowledge, which allowed them to progress at a rapid rate. The world right now owes to the application of this culminated knowledge for the betterment of society.

Speech and written text are at the heart of all communication. While speech is one of the easiest and quickest ways to communicate one's thoughts and ideas, written text retains information for a very long time (such as books), allowing many people to read after long intervals. As one would expect, the speech and written text are related. Thus, one mode of communication can be converted to another, retaining the content information. However, the human speech has overall more information than the corresponding text. This extra information is in the form of pitch, shimmer, jitter, stuttering, intonation, etc. This makes the conversion from written text to speech a difficult problem.

1.1 Problem Statement

Text-to-Speech synthesis is the task of generating a speech signal corresponding to a given input text. The ultimate goes of this task is to generate speech that is intelligible and natural sounding. Intelligibility implies that the utterance of each word in the input text is correct and can be understood by a native listener. Natural sounding emphasizes that the speech characteristics, like prosody, intonation, etc., must be similar to that of a native speaker. Furthermore, the audio quality of the generated speech should be good, that is, it should not have noise or any speech artifacts. This has been shown in Figure 1.

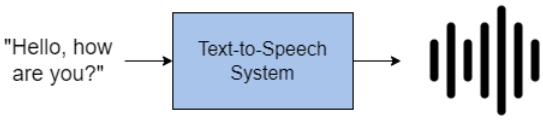

Figure 1: The Task of Text-to-Speech Synthesis

The conversion of text to corresponding speech is a one-to-many mapping, since there are multiple ways to utter the same sentence. So, including variations in the output speech for the same sentence is another task for text-to-speech synthesis. Also, most of the current TTS systems generate read-speech (the speech uttered by a person reading some text). Hence, generating good conversational speech that capture the disfluencies correctly is another milestone that is yet to be achieved.

1.2 Motivation

Speech generation or synthesis, the artificial production of human speech, is a problem that people have been trying to solve for a long time. In 1779, a German-Danish scientist modeled the vocal tract of humans and was able to produce the five long vowel sounds in English[1]. Later, people also worked on conditioning the speech synthesis on a given input text. This would enable people with visual impairments or reading disabilities to comprehend written pieces of text. It can also give voice to people with speaking disabilities, for example, people with damaged vocal tract. Further, with the development in the fields of automatic speech recognition and machine translation, a good text-to-speech system would enable the production of speech-to-speech machine translation systems, removing the most critical communication barrier among people speaking different languages.

It is important to empower everyone with the tools and ability of learning. Many of the developing nations face a variety of problem, one of them being low-literacy rates. For example, some people might not be able to read the script of a language, but can listen and understand properly. If we can make a system to automatically read out text, it would give everyone a chance at gaining knowledge through books. In such cases, technologies such as machine translation and text-to-speech conversion could prove immensely helpful.

2 Classical Approaches to Speech Synthesis

The earlier approaches to speech synthesis involved using a database of sound units. Multiple variations of all possible sounds that could be uttered in a specific language are recorded. The raw speech waveform are generated by concatenating these small speech units in appropriate order. Depending on the algorithm used to concatenate, these units are changed using digital signal processing techniques. The generated speech was intelligible but not natural sounding.

2.1 Diphone-based Speech Synthesis

A diphone is made up of two phones (simplest speech sounds) that are connected. There are many diphone-based approaches that use signal processing techniques like Pitch Synchronous Overlap-Add (PSOLA) (Charpentier and Moulines, 1989), Frequency Domain PSOLA (FD-PSOLA), Time Domain PSOLA (TD-PSOLA), and other derivatives (Khan and Chitode, 2016). In all of these, the speech is first decomposed into smaller segments (at the level of diphones) and then combined smartly to produce the expected output (overlapped addition). The character level breakdown of text (graphemes or converted to phonemes) can serve as the information for selecting the smaller speech unit. The methods used in above techniques for obtaining segmented signals are as follows:

PSOLA: Speech is divided into pitch-synchronous short-term waveforms which are then varied in time or spectral domain to obtain different synthetic versions of the same speech unit. Since the approach works directly on raw waveforms of speech signal, there is no information loss.

FD-PSOLA: First, the spectral envelope is computed using the linear predictive analysis. Next, the pitch of the segmented speech is modified via linear interpolation, to obtain synthetic speech units. Due to modifications of magnitudes in frequency domain (no phase considerations), there are unnatural discontinuities at the concatenation boundaries.

TD-PSOLA: The prosody of the speech waveforms can be manipulated using this approach. This results in the production of high quality time scale and pitch modifications. TD-PSOLA is computationally efficient, but requires a large dataset.

2.2 Corpus-based Speech Synthesis

Like diphone-based speech synthesis, corpus-based speech synthesis is a part of concatenative speech synthesis paradigm. In this approach, we select sound units from a large database and concatenate them to minimize a cost function. Here, the text can also have additional annotations containing prosodic and phonetic context information. The database is first transformed into a state transition network, with phonemes as states, as shown in Fig 2. The network is fully connected since hypothetically any sequence of phonemes is possible.

The cost function to be minimized is composed of two parts- target cost and concatenation cost. These costs are calculated as the weighted sum of difference between feature vectors of target and selected unit. The target cost measures the difference between the selected sound unit and the target unit. The feature vector used for this cost includes pitch, power, duration, voicing, vowel/consonant, consonant type, and point of articulation. The concatenation cost measures the quality of two connected

[1] From https://en.wikipedia.org/wiki/Speech_synthesis

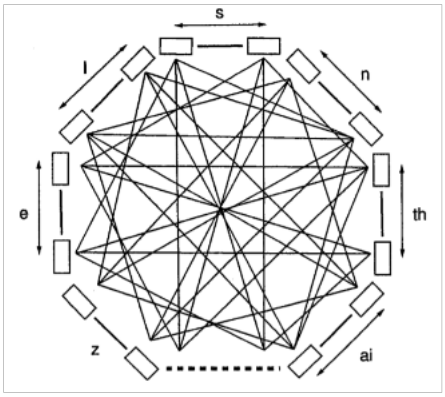

Figure 2: State Transition Network in Unit Selection Synthesis [2]

units. For this cost, the feature vector includes cepstral distance, difference in log power, and pitch. The weights for these cost functions are learned using weight search space or regression training.

Using the fully connected network and the cost functions we can use viterbi decoding for selecting final units in the appropriate order to generate the speech. Since the search space for viterbi decoding would be very large, it is pruned based on phonetic context, target cost, and concatenation cost.

3 Text to Mel spectrogram

Many diverse speech synthesis algorithms were developed by the researchers going from concatenative approaches to the deep learning models. Some of the prominent concatenative approaches were diphone-based signal processing methods and corpus-based data driven methods (Khan and Chitode, 2016). Later with the advancements in deep learning and availability of more data, better models were developed such as WaveNet, Tacotron, and more. Of the two approaches, deep learning provides more natural sounding speech with high quality audio.

3.1 WaveNet

WaveNet (Oord et al., 2016), developed by Google DeepMind, was among the first deep neural architectures that was trained to generate raw audio waveforms and produced amazing results. It is an autoregressive generative model, that is, the sample at each timestep is predicted based on the previously predicted samples. So, the joint probability of the speech waveform with samples $x = \{x_1, x_2, ..., x_T\}$ is written as,

$$p(x) = \prod_{t=1}^{T} p(x_t | x_1, x_2, ..., x_{t-1}) \quad (1)$$

This conditional probability is modeled using deep learning models, which are dilated causal convolutional layers (Van Oord et al., 2016) in the case of WaveNet. Due to the dilation, the receptive field of the convolution layers increases drastically, allowing the model to consider larger context at a time. Figure 3 shows the causal nature of the convolutional layers and how it increases the receptive field of the layers. Also, similar to language modelling tasks, the output of the model is a categorical distribution for the next timestep. The model achieves this through softmax layer followed by cross-entropy loss to maximize the log-likelihood of the data with respect to the parameters of the model. In the softmax layer, though there are 65,536 possible output values (16-bit integers) they transform the data according to ITU-T (1988) and have a 256 dimensional softmax output. They also used Gated Activation Units (Van Oord et al., 2016) as the activation functions, along with residual and skip connections.

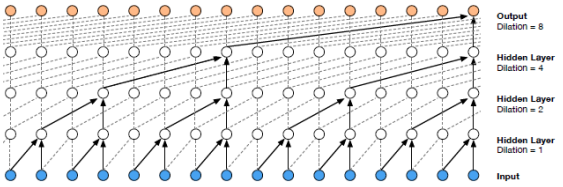

Figure 3: Stack of dilated causal convolutional layers [3]

The authors use an internal North American English dataset containing 24.6 hours of speech data. They also trained WaveNets conditioned on linguistic features of text and logarithmic fundamental frequency (log F_0) values, which are obtained using external models.

3.2 Tacotron2

Tacotron2 (Shen et al., 2018), an end-to-end speech synthesis architecture by Google, is among the initial state-of-the-art approaches using deep learning architectures for text-to-speech systems and generated outputs which were almost indistinguishable human speech. Like WaveNet, Tacotron2 is also an autoregressive model. However, unlike the

[2] Figure taken from the original paper (Hunt and Black, 1996)

[3] Figure taken from the original paper (Oord et al., 2016)

WaveNet, Tacotron2 first generates the mel spectrograms from the input text. These mel spectrograms are then converted to speech waveforms by the use of *vocoders*. The architecture consists of two components. First, a recurrent sequence-to-sequence network with attention mechanism which predicts the mel spectrograms of the desired output. Second, a vocoder (generally a neural network) that transforms the mel spectrograms to speech waveforms in time-domain.

In first stage of generating the mel spectrograms, the encoder consists of convolutional layers to model longer term context, followed by a bidirectional LSTM layer. This forms the encoder output which is passed through a location sensitive attention that summarizes the entire input for the decoder, at each timestep. The decoder comprises of two uni-directional LSTM layers which receives, along with the attention vector, the prediction from previous timesteps via a pre-net (2-layer feed-forward neural network). The pre-net serves the purpose to bottleneck the information flow, which the authors claim was essential for the model learning. After this, the output is passed through two feed-forward neural networks– one for predicting the mel spectrogram frame and other to predict the "stop token" (indicating end of generation). The predicted mel spectrogram frame is also passed through a post-net consisting of 5 convolutional layers, in order to rectify some discrepancies, which is again added to the original prediction. The architecture is shown in Figure 4.

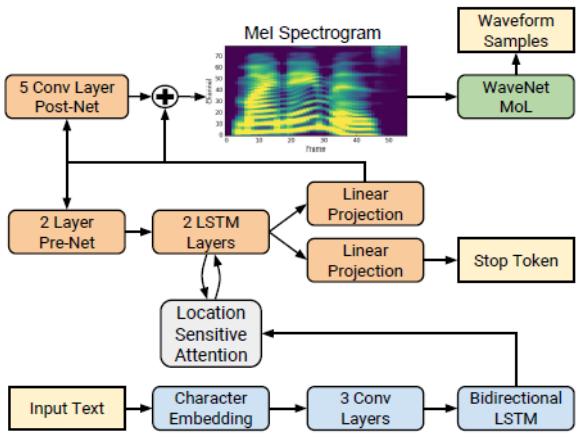

Figure 4: Architecture of Tacotron2 [4]

The model is trained on an internal North American English dataset which contains 24.6 hours of speech data by a single professional female speaker.

[4]Figure taken from the original paper (Shen et al., 2018)

The MSE loss is minimized over the ground-truth compared with the output of the model before post-net and after post-net. Dropouts with probability 0.5 are used for regularization. During inference, a dropout of 0.5 is kept in the pre-net layers to allow variation in the outputs of the model.

The authors slightly modify the architecture of WaveNet to train a vocoder to invert these generated mel spectrograms to the waveforms. Also, instead of predicting discrete signal sample outputs using a softmax layer, they use a 10-component mixture of logistic distributions (MoL) as per Salimans et al. (2017), to generate 16-bit samples at 24000 Hz. To compute this distribution, the WaveNet output is passed through a linear layer to predict the mean, log scale, mixture weight for each of the components. For this task, they use the negative log-likelihood loss.

3.3 FastSpeech

The tremendous success of Tacotron2 caught the attention of many researchers, proving the potential of text-to-speech systems. One of the concerns around Tacotron2 was that it takes too long to generate outputs owing to its autoregressive nature. So, Microsoft came up with FastSpeech (Ren et al., 2019), with the main objective of developing a non-autoregressive model for text-to-speech synthesis. Like Tacotron2, this model also has two components - mel spectrogram generation followed by neural vocoders. The authors boast a daunting 270x speed up in mel spectrogram generation and 38x speed up in overall end-to-end speech synthesis.

The core problem and tricky part of developing a non-autoregressive model is determining the phone durations in the output speech, in parallel. This was solved by exploiting the learnings of autoregressive models via the teacher-student learning paradigm. In solving this, they also achieved more control over the prosody and speech of generated speech, which are inherently linked to phone durations and the pauses in-between. Even after achieving all this, the output of the model maintains a high quality, making it a powerful model. The authors also mention that autoregressive models suffer the problem of word skipping and repeating due to wrong attention alignments between text and speech. They further claim that this problem is alleviated in case of FastSpeech, thus producing better quality speech.

The model is trained on the LJSpeech dataset which contains a total of 24 hours of speech. Input to the model are the phonemes corresponding to

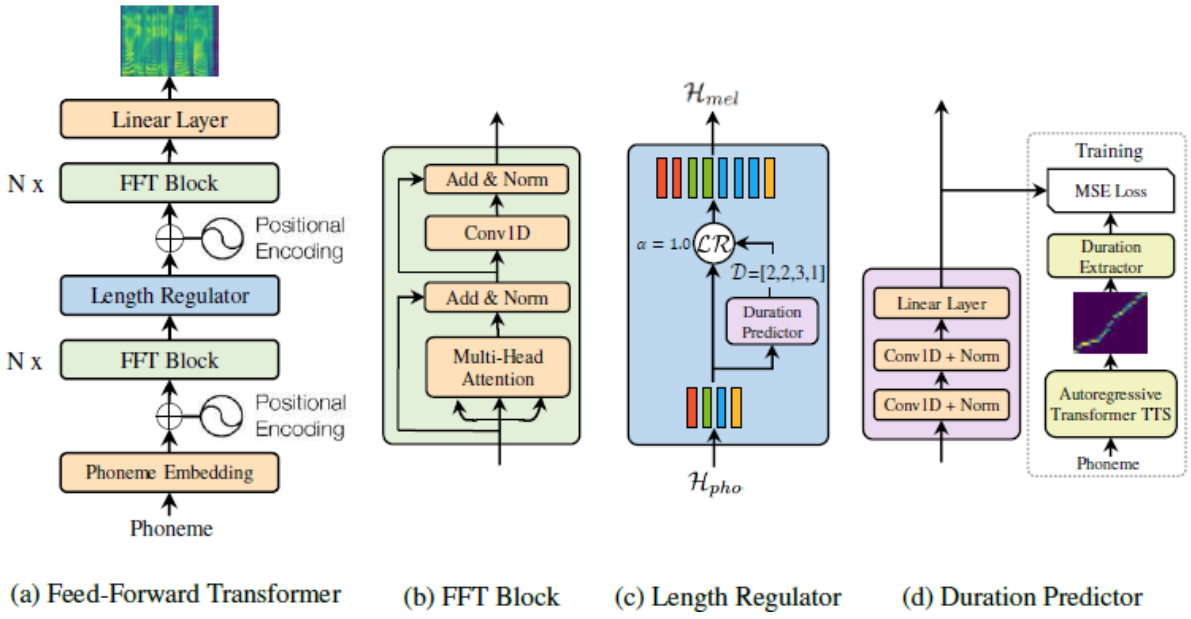

Figure 5: Architecture of FastSpeech [5]

the text obtained via grapheme-to-phoneme conversion. The ground-truth mel spectrograms are generated using the same signal sampling parameters as used for Tacotron2. It should be noted that the mel spectrograms used in the training of TTS models must be the same that are used in the training of vocoders. If not, the output speech, though intelligible, often consists or various artifacts like increase or drop in pitch, white noise, etc. Thus, most models have adopted the signal sampling parameters of Tacotron2 to be the default.

The model consists of a feed-forward Transformer network which drops the traditional encoder-attention-decoder architecture. This feed-forward Transformer is similar to the Transformer model from Vaswani et al. (2017), comprising of multi-head attention with residual and skip connections, layer normalization, and positional embeddings. The overall FastSpeech architecture is shown in Figure 5. They also train a length regulator which specifies the duration for which a specific phoneme is to be uttered. It consists of training a duration predictor which is a relatively small network to estimate the learned alignment predictions of another autoregressive model. The outputs of duration predictor specify the number of times the hidden states corresponding to a phoneme have to be repeated. It also has a tuning parameter α, which can speed up or slow down the output speech.

[5]Figure taken from the original paper (Ren et al., 2019)

3.4 ForwardTacotron

ForwardTacotron[6]is a non-autoregressive text-to-mel model without any attention mechanisms. It brings together the best of Tacotron and FastSpeech models. The techniques for the non-autoregressive training are adopted from the FastSpeech (Ren et al., 2019) model, while the architecture is motivated by the Tacotron model to eliminate the memory-intensive transformer blocks. The architecture can be seen in Fig 6. The *Prenet* and *Postnet* are CBHG (Convolutional Bank Highway GRU) blocks introduced in Tacotron. There are three major differences between ForwardTacotron and FastSpeech:

1. While the duration predictor (discussed below) for FastSpeech is trained with the main model, for the ForwardTacotron model, it is trained separately from rest of the model.

2. Unlike FastSpeech, ForwardTacotron consists of pitch and energy predictors (same architecture as the duration predictor). Outputs of these predictors are passed through a single convolutional layer and added to the input embeddings, before the length regulator.

3. FastSpeech makes use of knowledge distillation by training on the mel spectrograms generated by the external alignment model.

[6]The developers of this model have not released any paper. Refer https://github.com/as-ideas/ForwardTacotron

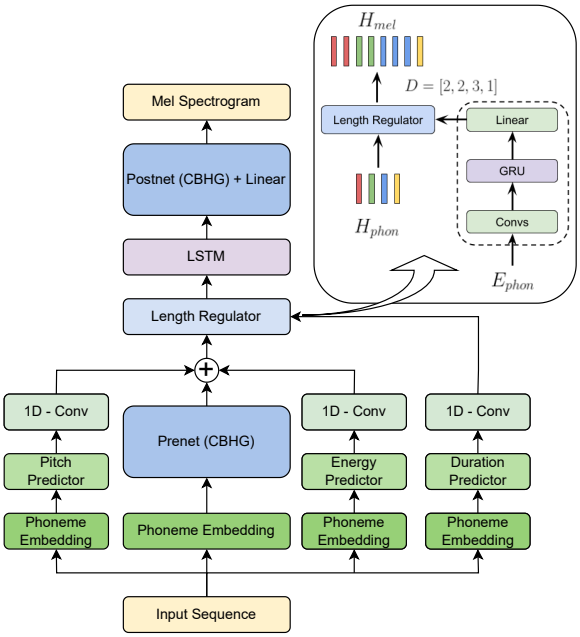

Figure 6: Architecture of ForwardTacotron with the Duration Predictor and Length Regulator (the cutout is reproduced) [7]

However, ForwardTacotron is trained using the original mel spectrograms.

Length Regulator

The mismatch between the number of mel spectrogram frames and number of input tokens is handled by the length regulator using the duration predictor outputs. The encoder outputs are repeated as per the duration predicted for each input token. For fractional duration, the closest integer is chosen.

Series Predictors

Along with the main model, three sequence-to-sequence models are trained for predicting duration, pitch, and energy corresponding to each input token. The ground truth for these are obtained from the alignments learnt by the external alignment model. The duration is the number of mel spectrogram frames that attended the input token. The pitch is calculated by averaging over the corresponding aligned frames of raw pitch obtained from original speech. The energy is calculated by averaging the norm of aligned mel spectrogram frames for the input token. During training, teacher forcing is used by passing the ground truth values instead of series predictor outputs. The series predictor outputs are used only during the inference.

[7] https://github.com/as-ideas/ForwardTacotron

3.5 Recent Advancements

Apart from the models discussed above, there have been other text-to-mel models which improved upon their predecessors in some way. However, the above mentioned models introduced new ideas that significantly improved over the previous works. Following the progress of these models, there has been consistent work in improving the text-to-mel models for solving specific problems. While the audio quality is close to the natural speech for all such models, the new architectures either reduce the training and inference time or allow variations in the generated speech.

Parallel Tacotron 2 (Elias et al., 2021) eliminates the need for an external aligner for durations by differential duration modelling, while being non-autoregressive in nature. LightSpeech (Luo et al., 2021) leverages the neural architecture search (NAS) to find a lightweight and efficient model that can faithfully replicate the performance of FastSpeech with minor degradation. FastSpeech 2 (Ren et al., 2021a) improves upon FastSpeech by introducing variance adapter containing pitch and energy predictors and extends it FastSpeech 2s which directly generates the speech waveform (fully end-to-end speech synthesis). PortaSpeech (Ren et al., 2021b) discusses that there are two types of generative non-autoregressive TTS models- variational autoencoders and normalizing flows. Both these approaches have their advantages and disadvantages. PortaSpeech finds a middle ground and brings together the best of both worlds with only slight performance degradation. The most recent VQTTS (Du et al., 2022) breaks the traditional pipeline of text-to-mel followed by a vocoder. It argues that mel spectrogram is highly correlated in both time and frequency axes, making it difficult for a model to predict the mel spectrograms from a text. So it introduces new paradigm where the text is first converted to a self-supervised vector quantized acoustic feature followed by a vector to waveform model (modified HiFiGAN).

4 Vocoders

Most of the latest neural architectures for text-to-speech first generate the mel spectrograms which are then converted to raw speech waveforms via vocoders. There have been many different approaches to develop these vocoders, out of which the deep learning based neural networks work exceptionally well. However, with the increasing

speed of speech generation by TTS models, the speed of vocoders started becoming the bottleneck in the end-to-end TTS systems. So people started attempting to improve the vocoders by making use of developing deep learning architectures. The naturalness of generated speech being one of the most important aspect in text-to-speech, GANs became a preferred choice of research direction in developing vocoders.

4.1 Waveglow

WaveGlow (Prenger et al., 2019), by NVIDIA, is one of the initial neural network based vocoders which produced high quality outputs. It is a flow-based, non-autoregressive, generative model which combines WaveNet (Oord et al., 2016) and Glow (Kingma and Dhariwal, 2018). The core idea of a flow based network is the make the function represented by the network to be invertible. This is achieved by forcing each layer in the network to be a bijective mapping. The motivation behind doing this is to be able to compute the log-likelihood and directly minimize it. So, if $\boldsymbol{z}$ is a random noise sample or latent space representation, then the model transforming $\boldsymbol{z}$ to output $\boldsymbol{x}$ can be written as

$$\boldsymbol{x} = \boldsymbol{f}_0 \circ \boldsymbol{f}_1 \circ ... \circ \boldsymbol{f}_k(\boldsymbol{z}) \tag{2}$$

$$\boldsymbol{z} = \boldsymbol{f}_k^{-1} \circ \boldsymbol{f}_{k-1}^{-1} \circ ... \circ \boldsymbol{f}_0^{-1}(\boldsymbol{x}) \tag{3}$$

The Equation 3 is possible due to the invertible nature of the flow network. These sequence of transformations are referred to as *normalizing flows*. To work with this framework, the paper introduces an *affine coupling layer*. The equations governing the function of this layer are given in Equations 4.

$$\begin{gathered} \boldsymbol{x}_a, \boldsymbol{x}_b = \text{split}(\boldsymbol{x}) \\ (\log \boldsymbol{s}, \boldsymbol{t}) = WN(\boldsymbol{x}_a, \text{mel spectrogram}) \\ \boldsymbol{x}_b\prime = \boldsymbol{s} \odot \boldsymbol{x}_b + \boldsymbol{t} \\ \boldsymbol{f}_{\text{coupling}}^{-1}(\boldsymbol{x}) = \text{concat}(\boldsymbol{x}_a, \boldsymbol{x}_b\prime) \end{gathered} \tag{4}$$

where, $\boldsymbol{x}$ is the input to the network, and $\odot$ is the element-wise dot-product.

The split function implies a spilt in the channel dimension. So half of the channels are processed and concatenated with the other unprocessed half. One interesting aspect to note here is that the transformation $WN()$ need not be invertible. The reason for this is that the channels which are input to $WN()$ are also the ones that are passed unprocessed to the output of the layer, and hence do not affect the overall invertibility. Thus, this $WN()$ transformation can be replaced by a neural network, which in this paper is similar to the WaveNet. In this affine coupling layer, the upsampled mel spectrograms are introduced to condition the generated output. In the whole model, 12 such coupling layers are used. Overall architecture is shown in Figure 7.

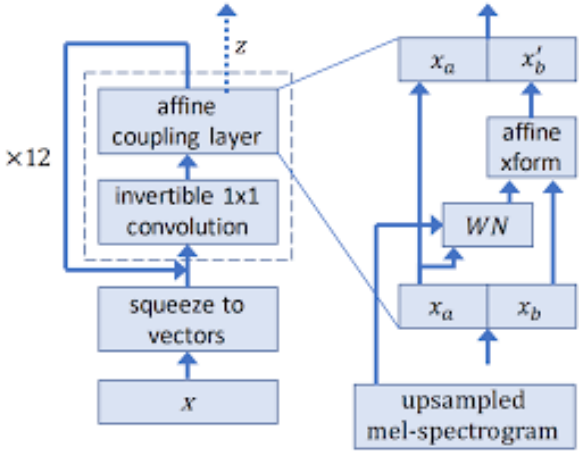

Figure 7: Architecture of WaveGlow [8]

The WaveGlow model works very well and produces high quality speech outputs. Owing to the flow based nature of the model, training the model is also very simple. Also due to its autoregressive nature, it is very fast in producing the outputs.

4.2 MelGAN

The outputs of GANs have been highly realistic due to the adversarial loss introduced in the architecture. However, for a long time people believed that training GANs to generate high quality speech waveforms is very tough and challenging. MelGAN (Kumar et al., 2019), is the first paper to show that by introducing some architectural changes and inculcating domain knowledge, it is possible to train GAN models that produce coherent speech waveforms. The model is non-autoregressive and fully convolutional making it extremely fast (2x faster than real-time on CPU). Although the WaveGlow model outperforms MelGAN in terms of MOS, the number of parameters of WaveGlow are almost 20 times that of MelGAN.

The generator of MelGAN is a fully convolutional model with transposed convolutional layers followed by residual blocks with dilated convolutions. The residual blocks ensure better long range correlation by effectively increasing the receptive

[8] Figure taken from the original paper (Prenger et al., 2019)

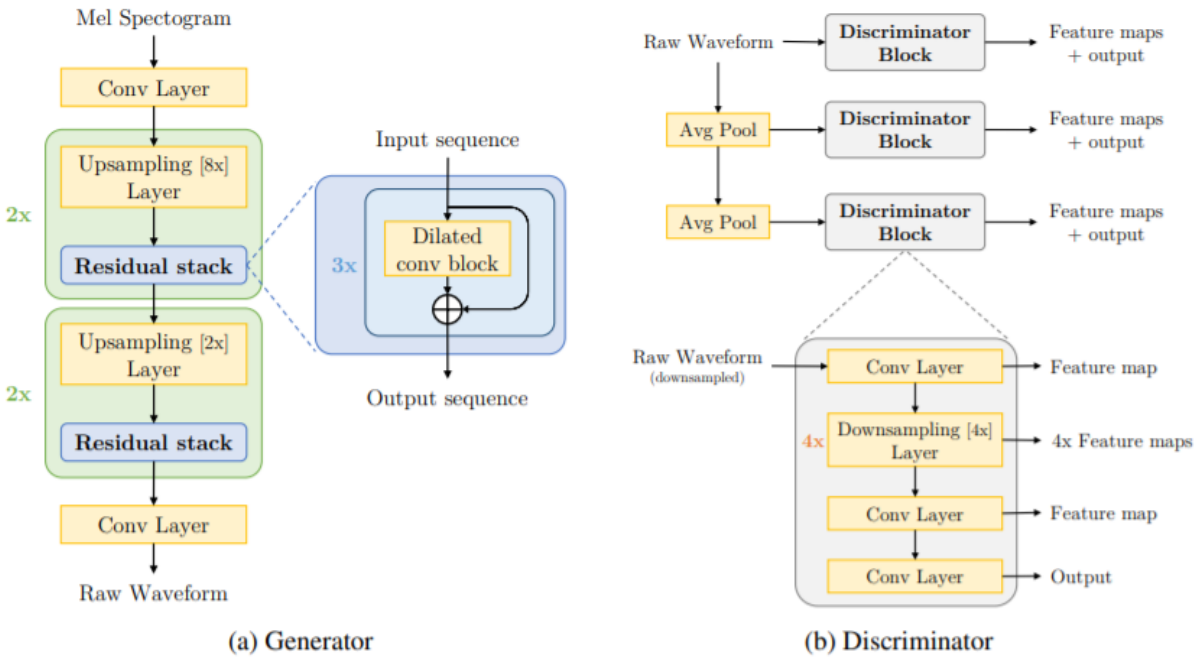

Figure 8: Architecture of MelGAN [9]

fields of output timesteps. The authors mention that according to (Odena et al., 2016), the kernel size and stride of the transposed convolution layers must be chosen carefully, else "checkerboard" artifacts are introduced in the output. The training objective is a combination of adversarial loss (slight variant) and feature matching task. Feature matching consists of L1 loss between the feature maps of real and fake audio that are learnt by the discriminator.

The discriminator follows a multi-scale architecture with 3 discriminators - one operating on raw audio and others on downsampled versions by factors of 2 and 4. Weight normalization is used in discriminator as well as generator. The architecture is shown in Figure 8.

4.3 Parallel WaveGAN

Parallel WaveGAN (Yamamoto et al., 2020), is another approach to develop GAN models for speech synthesis. The model is fully convolutional and non-autoregressive (fast parallel computations), is based on the WaveNet architecture, and leverages the advantages of GAN. Hence, the name Parallel WaveGAN. The paper proposes a new loss function to train models for generating speech, called as the *multi-resolution STFT loss*. The model only has 1.44 million parameters which almost 80 times less than WaveGlow and 4 times less than MelGAN. This shows the potential of GAN architectures in providing high quality results with smaller models.

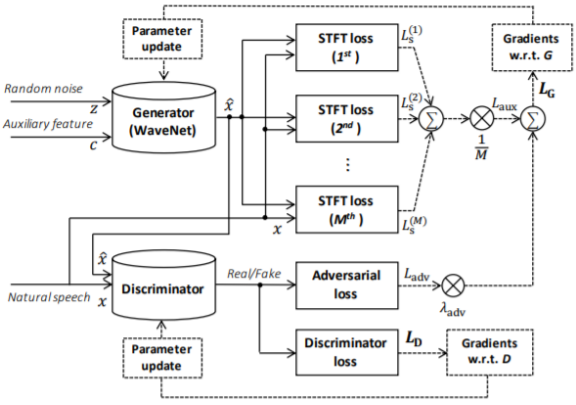

Figure 9: Parallel WaveGAN training framework [10]

For the generator model of GAN, a WaveNet based architecture is trained with noise as input and conditioned on the mel spectrograms. The distinctive feature of Parallel WaveGAN is that the use of non-causal convolutions for the models, instead of causal convolutions. The architecture also makes use of residual and skip connections. The discriminator is also a stack of non-causal dilated 1-D convolutions. The training process has been

[9] Figure taken from the original paper (Kumar et al., 2019)

[10] Figure taken from the original paper (Yamamoto et al., 2020)

depicted in Figure 9. All convolutional layers have weight normalization, both in the discriminator and the generator.

The speech corpus used for training the model was recorded by a female professional Japanese speaker. The dataset is said to be phonetically and prosaically balanced. It contains approximately 24 hours of speech, with a sampling rate of 24 kHz.

4.4 HiFi-GAN

HiFi-GAN (Kong et al., 2020) is another GAN based vocoder which uses the multi-resolution discriminative framework. This model has excellent performance, with the audio quality close to human speech. Owing to the success of this architecture, the recent vocoders compare their results with this model to emphasize the quality of their outputs. It consists of one generator and two discriminators. HiFi-GAN is trained using the adversarial loss (used in original GAN architecture) and additional two losses for improving the training stability. The generator and discriminators are fully convolutional.

Multi-Period Discriminator (MPD):
Since the human speech is generated from the periodic vibrations of glottis, the resulting speech waveform also contains the corresponding sinusoidal signals. MPD aims at capturing these diverse periodic patterns. It consists of a mixture of sub-discriminators each of which looks at equally spaced samples in the raw input audio. If p is the defined spacing, then the 1D input of length T is reshaped into the 2D shape of (T/p $\times$ p). This is passed through a stack of strided 2D convolution layers with kernel width set to 1, followed by leaky ReLU.

Multi-Scale Discriminator (MSD):
The MSD is similar to the multi-resolution discriminator of MelGAN, discussed in Section 4.2. It also consists of mixture of sub-discriminators which look at the input at different resolutions-original, $\times 2$ average pooled, and $\times 4$ average pooled. This allows the discriminator to capture long-term dependencies and consecutive patterns in the input.

Additional Loss Terms:
Apart from the adversarial GAN loss, HiFi-GAN includes two more loss terms. First is the mel spectrogram reconstruction loss which calculates the L1 distance between the mel spectrogram of generated speech waveform and that of the original audio. ParallelWaveGAN had proved that efficacy of using such a loss in improving the perceptual quality of the generated audio. Another loss is the feature matching loss which calculates the L1 distance of every intermediate feature of the discriminator between the original audio and the generated speech. The final loss for generator is the weighted sum of adversarial loss along with the above two losses, while that for the discriminator is only the adversarial loss.

4.5 GAN Vocoder

As we have seen, all the recent vocoders are GAN models with amazing results. They outperform many of the existing autoregressive and flow-based vocoders in both objective and subjective metrics. Furthermore, the size of these GAN models are much smaller, allowing them to synthesize speech orders of magnitude faster than other vocoders, like Waveglow. You et al. (2021) hypothesizes that the success of these GAN-based vocoders is due to the multi-resolution discriminating framework, and not the specific architectures chosen for the generator.

The paper compares the performance of six different generator architectures while maintaining the multi-resolution discriminating framework. The six generators are that of MelGAN, Parallel WaveGAN, HiFi-GAN, Universal MelGAN (Jang et al., 2020), VocGAN (Yang et al., 2020), and their own generator architecture. The LJSpeech dataset is used as is, without any modification. The results of the training can be seen in Figure 10. Further, the objective score Mean Cepstral Distortion (MCD) and the subjective score Mean Opinion Score (MOS) support the hypothesis that the outputs of all these vocoders are not perceptually distinguishable.

4.6 CARGAN

Chunked Autoregressive GAN (Morrison et al., 2022) (aka CARGAN), is among the most recent GAN-based vocoders and performs exceptionally well. As the name suggests, it is an autoregressive architecture but instead of generating one sample at a time, it produces a chunk of samples in one go. Thus, CARGAN is a hybrid model which strikes a trade-off between better quality of autoregressive models and fast training and inference of non-autoregressive models. The model reduces the pitch errors by 40-60% and reduces the training

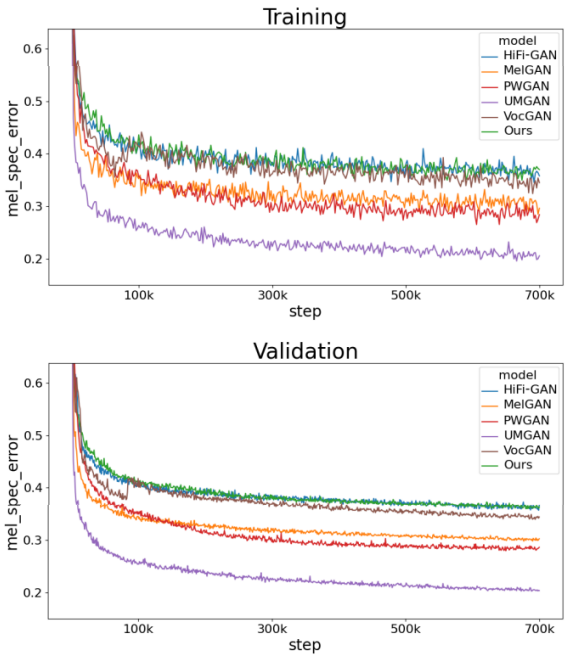

Figure 10: Results of Multi-Resolution Discriminative training for Vocoder [11]

time by 58% when compared to previous state-of-the-art vocoders.

The paper claims that certain artifacts are introduced by the existing vocoders, which correspond to the inability of the generator to correctly learn and predict the pitch and periodicity of the speech waveform. Autoregressive architectures contain the inductive bias for learning the relationship between the pitch and phase. Consider a perfectly periodic signal (sampling rate r) which instantaneous frequency $\boldsymbol{f} = \{f_1, ..., f_T\}$ and the instantaneous phase $\phi = \{\phi_1, ..., \phi_T\}$, then we have

$$\phi_t = \phi_{t-1} + \frac{2\pi}{r} f_t \tag{5}$$

So, we can see that there is a cumulative summation operation for finding the phase. Since autoregression considers the information from the previously generated output, this operation is inherent to it. Specifically, the model can find the ϕ_{t-1} and f_t based on the previously generated samples. Though the above equation is true only for a single sample generation, we can inductively show that it holds while generating chunk of samples as well. The architecture can be seen in Figure 11.

The model is fully convolutional and consists of three components- autoregressive conditioning stack to summarize previous k-samples, a generator

[11] Figure taken from the original paper (You et al., 2021)

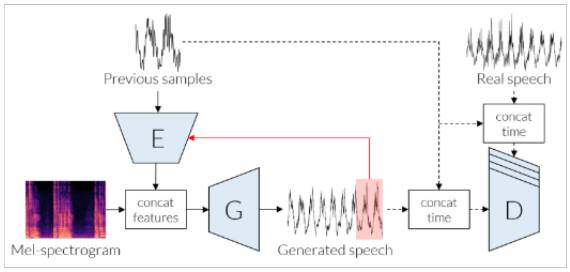

Figure 11: Training of CARGAN [12]

which combines the previous output with the input conditioning to generate the raw waveform, and a bunch of discriminators for the adversarial training. The first two components are trained to minimize the adversarial, mel spectrogram reconstruction, and feature matching losses, as in HiFiGAN. The discriminator is trained with only the adversarial loss. During training, each data sample is generated by randomly choosing a starting frame index for the mel spectrogram of an audio and considering fixed number of frames after it.

5 Voice Conversion

Voice conversion is the task of changing the voice of input speech waveform to a desired voice of a different speaker, keeping the linguistic content the same. As we have seen for vocoders, GANs have become capable of generating excellent speech waveforms from mel-spectrograms. This suggests that the GANs are able to learn the general characteristics of speech waveform. So, now we will look at some GAN-based models for the task of voice conversion. All these models are a specific type of GAN architecture, that is, CycleGAN. This architecture intuitively seems like the perfect choice for voice conversion, with two distinct domains– voices of source speaker and target speaker. Also, we wish to preserve some specific information from the source domain (linguistic information) when transforming it to the target domain.

5.1 CycleGAN-VC Models

Many different CycleGAN-based models were developed to tackle the voice conversion problem. We will look at a specific set of these models from the CycleGAN-VC series. In these models, Mel-cepstral coefficients (MCEPs) of the corresponding speech signals formed the input and output

[12] Figure taken from the original paper (Morrison et al., 2022)

of these models. The first among these was proposed in (Kaneko and Kameoka, 2018), called as CycleGAN-VC. All the generators and discriminators are gated convolutional neural networks (CNNs with Gated Linear Unit as activation) with residual connections. While the original CycleGAN paper showed the effectiveness of identity-mapping loss (Eq. 6) for colour preservation, in CycleGAN-VC it is used to impose the condition of preserving the linguistic information.

$$\begin{aligned}\mathscr{L}_{id}\left(G_{X\to Y}, G_{Y\to X}\right) = \mathbb{E}_{y\sim P_{\text{Data}}(y)}\left[\|G_{X\to Y}(y) - y\|_1\right] \\ + \mathbb{E}_{x\sim P_{\text{Data}}(x)}\left[\|G_{Y\to X}(x) - x\|_1\right]\end{aligned} \tag{6}$$

CycleGAN-VC2 (Kaneko et al., 2019) improves over CycleGAN-VC by introducing three new techniques. Firstly, it introduces a new loss term, which the authors call as the two-step adversarial loss (Eq 7) which added to the original adversarial loss to obtain the final loss. For this loss, additional discriminator D'_X for X domain (similarly D'_Y for Y domain) is introduced. For the reverse direction, similar loss term is calculated. Both these loss together form the two-step adversarial loss. Next, the architecture of generator is changed to 2-1-2D CNN. So, the initial and final layers are 2D convolutional layers, while the middle part, where the main conversion occurs, is 1D convolutional layers with residual connections. Finally, the discriminator is changed from FullGAN to a PatchGAN.

$$\begin{aligned}\mathscr{L}_{adv2}\left(G_{X\to Y}, G_{Y\to X}, D'_X\right) = \mathbb{E}_{x\sim P_X(x)}\left[\log D'_X(x)\right] \\ + \mathbb{E}_{x\sim P_X(x)}\left[\log\left(1 - D'_X\left(G_{Y\to X}\left(G_{X\to Y}(x)\right)\right)\right)\right]\end{aligned} \tag{7}$$

CycleGAN-VC/VC2 take MCEPs as inputs and generate the MCEPs of the target speech. However, mel-spectrograms are known to capture the characteristics of speech that correspond to human hearing. The authors observed that these models compromised the time-frequency structure of the outputs, when trained with mel-spectrograms instead of MCEPs. So, CycleGAN-VC3 (Kaneko et al., 2020) was developed with time-frequency adaptive normalization (TFAN) module to handle mel-spectrograms as inputs and outputs. Mathematically, the output of TFAN is evaluated according to Eq 8.

$$f' = \gamma(x)\frac{f - \mu(f)}{\sigma(f)} + \beta(x) \tag{8}$$

The $\mu(f)$ and $\sigma(f)$ denote the channel-wise mean and standard deviation of a given feature f. Further, for a given timestep x, the scale $\beta(x)$ and bias $\gamma(x)$ are applied in an element-wise manner. This allows the TFAN module to adjust the scale and bias of the features, while taking the timestep x under consideration.

5.2 MaskCycleGAN-VC

The latest in the series of CycleGAN-VC models is the MaskCycleGAN-VC (Kaneko et al., 2021). While the CycleGAN-VC3 allows training with mel-spectrograms instead of MCEPs, it introduces a new module with learnable parameters called TFAN. Also, this TFAN module needs to be interchanged with every instance normalization layer in the CycleGAN-VC2 model, due to which the increase in parameters is significant. To avoid this, MaskCycleGAN-VC introduces a novel auxiliary masked training called *filling in frames (FIF)*. This can be seen in Figure 12. Apart from FIF, the authors experimented with three other masking techniques, but found that FIF provided the best results.

In FIF, a random temporal mask (all values along frequency axis are equal) is applied on the input and passed to the generator. The generator is expected to learn the characteristics of mel-spectrograms of the specific domain and fill in the missing frames. So, this auxiliary task helps the model learn the time-frequency structures in a mel-spectrogram in a self-supervised manner. The MelGAN vocoder was used for converting mel-spectrograms to raw speech waveforms.

6 Summary

In this paper, we looked at the various approaches that have been developed by the people to solve the challenging task of speech synthesis. There was a significant performance boost in going from the classical approaches involving signal processing techniques to deep learning based architectures. The approach of first converting the text to mel spectrograms followed by the use of vocoders to convert it into raw waveforms seems to work extremely well. Many other attempts at TTS also use this approach in their models. In case of Indian languages, a thorough study of such approaches is yet to be done. We also discussed some approaches for tackling the task of voice conversion.

The performance metrics as reported by the corresponding papers has be presented in Table 1. Since ForwardTacotron did not have a paper and did not conduct the MOS experiments, the table

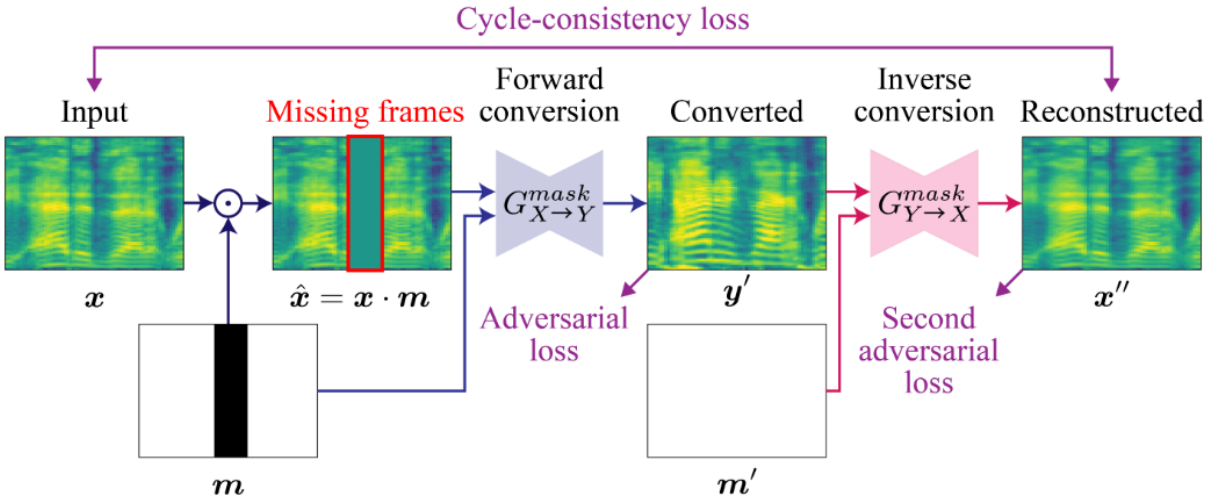

Figure 12: Filling in Frames training for MaskCycleGAN-VC [13]

Model	Dataset	Task	Metric	Value
WaveNet	North American English	Text-to-Speech	MOS	4.21
Tacotron2 + WaveNet	North American English	Text-to-Speech	MOS	4.52
FastSpeech + WaveGlow	LJSpeech	Text-to-Speech	MOS	3.84
WaveGlow	LJSpeech	Vocoder	MOS	3.81
MelGAN	LJSpeech	Vocoder	MOS	3.79
Parallel WaveGAN	Japanese Dataset	Vocoder	MOS	4.06
HiFiGAN	LJSpeech	Vocoder	MOS	4.36

Table 1: Performance metrics of various speech synthesis models

does not include the model. Similarly, the CAR-GAN did not present any MOS evaluation results and hence not present in the table.

References

Francis Charpentier and Eric Moulines. 1989. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. In *Proc. First European Conference on Speech Communication and Technology (Eurospeech 1989)*, pages 2013–2019.

Chenpeng Du, Yiwei Guo, Xie Chen, and Kai Yu. 2022. VQTTS: High-fidelity text-to-speech synthesis with self-supervised VQ acoustic feature.

Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Ye Jia, R.J. Skerry-Ryan, and Yonghui Wu. 2021. Parallel Tacotron 2: A Non-Autoregressive Neural TTS Model with Differentiable Duration Modeling. In *Proc. Interspeech 2021*, pages 141–145.

A.J. Hunt and A.W. Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376 vol. 1.

ITU-T. 1988. Recommendation g. 711. pulse code modulation (pcm) of voice frequencies.

Won Jang, Dan Lim, and Jaesam Yoon. 2020. Universal melgan: A robust neural vocoder for high-fidelity waveform generation in multiple domains. *arXiv preprint arXiv:2011.09631*.

Takuhiro Kaneko and Hirokazu Kameoka. 2018. Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2100–2104.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. 2019. Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. 2020. Cyclegan-vc3: Examining and improving cyclegan-vcs for mel-spectrogram conversion. In *Proceedings of the Annual Conference of the International Speech Communication Association*.

[13] Figure taken from the original paper (Kaneko et al., 2021)

Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. 2021. Maskcyclegan-vc: Learning non-parallel voice conversion with filling in frames. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Rubeena A Khan and JS Chitode. 2016. Concatenative speech synthesis: A review. *International Journal of Computer Applications*, 136(3):1–6.

Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033. Curran Associates, Inc.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Renqian Luo, Xu Tan, Rui Wang, Tao Qin, Jinzhu Li, Sheng Zhao, Enhong Chen, and Tie-Yan Liu. 2021. Lightspeech: Lightweight and fast text to speech with neural architecture search. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5699–5703.

Max Morrison, Rithesh Kumar, Kundan Kumar, Prem Seetharaman, Aaron Courville, and Yoshua Bengio. 2022. Chunked autoregressive GAN for conditional waveform synthesis. In *International Conference on Learning Representations*.

Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and checkerboard artifacts. *Distill*.

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

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021a. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*.

Yi Ren, Jinglin Liu, and Zhou Zhao. 2021b. PortaSpeech: Portable and high-quality generative text-to-speech. In *Advances in Neural Information Processing Systems*, volume 34, pages 13963–13974. Curran Associates, Inc.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. 2017. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif A. Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu. 2018. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. 2016. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE.

Jinhyeok Yang, Junmo Lee, Youngik Kim, Hoon-Young Cho, and Injung Kim. 2020. VocGAN: A High-Fidelity Real-Time Vocoder with a Hierarchically-Nested Adversarial Network. In *Proc. Interspeech 2020*, pages 200–204.

Jaeseong You, Dalhyun Kim, Gyuhyeon Nam, Geumbyeol Hwang, and Gyeongsu Chae. 2021. GAN Vocoder: Multi-Resolution Discriminator Is All You Need. In *Proc. Interspeech 2021*, pages 2177–2181.