Flowtron: an Autoregressive Flow-based Generative Network for Text-to-Speech Synthesis

Rafael Valle [1]   Kevin Shih [1]   Ryan Prenger [1]   Bryan Catanzaro [1]

Abstract

In this paper we propose Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis with control over speech variation and style transfer. Flowtron borrows insights from IAF and revamps Tacotron in order to provide high-quality and expressive mel-spectrogram synthesis. Flowtron is optimized by maximizing the likelihood of the training data, which makes training simple and stable. Flowtron learns an invertible mapping of data to a latent space that can be manipulated to control many aspects of speech synthesis (pitch, tone, speech rate, cadence, accent). Our mean opinion scores (MOS) show that Flowtron matches state-of-the-art TTS models in terms of speech quality. In addition, we provide results on control of speech variation, interpolation between samples and style transfer between speakers seen and unseen during training. Code and pre-trained models will be made publicly available at https://github.com/NVIDIA/flowtron.

1. Introduction

Current speech synthesis methods do not give the user enough control over how speech actually sounds. Automatically converting text to audio that successfully communicates the text was achieved a long time ago (Umeda et al., 1968; Badham et al., 1983). However, communicating only the text information leaves out all of the acoustic properties of the voice that convey much of the meaning and human expressiveness. Nearly all the research into speech synthesis since the 1960s has focused on adding that non-textual information to synthesized speech. But in spite of this, the typical speech synthesis problem is formulated as a text to speech problem in which the user inputs only text.

Taming the non-textual information in speech is difficult

because the non-textual is unlabeled. A voice actor may speak the same text with different emphasis or emotion based on context, but it is unclear how to label a particular reading. Without labels for the non-textual information, models have fallen back to unsupervised learning. Recent models have achieved nearly human-level quality, despite treating the non-textual information as a black box. The model's only goal is to match the patterns in the training data (Shen et al., 2017; Arik et al., 2017b;a; Ping et al., 2017). Despite these models' excellent ability to recreate the non-textual information in the training set, the user has no insight into or control over the non-textual information.

It is possible to formulate an unsupervised learning problem in such a way that the user can gain insights into the structure of a data set. One way is to formulate the problem such that the data is assumed to have a representation in some latent space, and have the model learn that representation. This latent space can then be investigated and manipulated to give the user more control over the generative model's output. Such approaches have been popular in image generation for some time now, allowing users to interpolate smoothly between images and to identify portions of the latent space that correlate with various features (Radford et al., 2015; Kingma & Dhariwal, 2018).

In audio, however, approaches have focused on embeddings that remove a large amount of information and are obtained from assumptions about what is interesting. Recent approaches that utilize deep learning for expressive speech synthesis combine text and a learned latent embedding for prosody or global style (Wang et al., 2018; Skerry-Ryan et al., 2018). A variation of this approach is proposed by (Hsu et al., 2018), wherein a Gaussian mixture model (GMM) encoding the audio is added to Tacotron to learn a latent embedding. These approaches control the non-textual information by learning a bank of embeddings or by providing the target output as an input to the model and compressing it. However, these approaches require making assumptions about the dimensionality of the embeddings before hand and are not guaranteed to contain all the non-textual information it takes to reconstruct speech, including the risk of having dummy dimensions or not enough capacity, as the appendix sections in (Wang et al., 2018;

---

[1]NVIDIA Applied Deep Learning Research (ADLR). Correspondence to: Rafael Valle <rafaelvalle@nvidia.com>.

Skerry-Ryan et al., 2018; Hsu et al., 2018) confirm. They also require finding an encoder and embedding that prevents the model from simply learning a complex identity function that ignores other inputs. Furthermore, these approaches focus on fixed-length embeddings under the assumption that variable-length embeddings are not robust to text and speaker perturbations. Finally, most of these approaches do not give the user control over the degree of variability in the synthesized speech.

In this paper we propose Flowtron: an autoregressive flow-based generative network for mel-spectrogram synthesis with control over acoustics and speech. Flowtron learns an invertible function that maps a distribution over mel-spectrograms to a latent z space parameterized by a spherical Gaussian. With this formalization, we can generate samples containing specific speech charateristics manifested in mel-space by finding and sampling the corresponding region in z-space. In the basic approach, we generate samples by sampling a zero mean spherical Gaussian prior and control the amount of variation by adjusting its variance. Despite its simplicity, this approach offers more speech variation and control than Tacotron.

In Flowtron, we can access specific regions of mel-spectrogram space by sampling a posterior distribution conditioned on prior evidence from existing samples (Kingma & Dhariwal, 2018; Gambardella et al., 2019). This approach allows us to make a monotonous speaker more expressive by computing the region in z-space associated with expressive speech as it is manifested in the prior evidence. Finally, our formulation also allows us to impose a structure to the z-space and parametrize it with a Gaussian mixture, for example. In this approach related to (Hsu et al., 2018), speech charateristics in mel-spectrogram space can be associated with individual components. Hence, it is possible to generate samples with specific speech characteristics by selecting a component or a mixture thereof [1].

Although VAEs and GANs (Hsu et al., 2018; Bińkowski et al., 2019; Akuzawa et al., 2018) based models also provide a latent prior that can be easily manipulated, in Flowtron this comes at no cost in speech quality nor optimization challenges.

We find that Flowtron is able to generalize and produce sharp mel-spectrograms by simply maximizing the likelihood of the data while not requiring any additional Prenet or Postnet layer (Wang et al., 2017), nor compound loss functions required by most state of the art models like (Shen et al., 2017; Arik et al., 2017b;a; Ping et al., 2017; Skerry-Ryan et al., 2018; Wang et al., 2018; Bińkowski et al., 2019).

Flowtron is optimized by maximizing the likelihood of the training data, which makes training simple and stable. It

learns an invertible mapping of the a latent space that can be manipulated to control many aspects of speech synthesis. Our mean opinion scores (MOS) show that Flowtron matches state-of-the-art TTS models in terms of speech quality. In addition, we provide results on control of speech variation, interpolation between samples, and style transfer between seen and unseen speakers with similar and different sentences. To our knowledge, this work is the first to show evidence that normalizing flow models can also be used for text-to-speech synthesis. We hope this will further stimulate developments in normalizing flows.

2. Related Work

Earlier approaches to text-to-speech synthesis that achieve human like results focus on synthesizing acoustic features from text, treating the non-textual information as a black box. (Shen et al., 2017; Arik et al., 2017b;a; Ping et al., 2017). Approaches like (Wang et al., 2017; Shen et al., 2017) require adding a critical Prenet layer to help with convergence and improve generalization (Wang et al., 2017). Furthermore, such models require an additional Postnet residual layer and modified loss to produce "*better resolved harmonics and high frequency formant structures, which reduces synthesis artifacts.*"

One approach to dealing with this lack of labels for underlying non-textual information is to look for hand engineered statistics based on the audio that we believe are correlated with this underlying information.

This is the approach taken by models like (Nishimura et al., 2016; Lee et al., 2019), wherein utterances are conditioned on audio statistics that can be calculated directly from the training data such as F_0 (fundamental frequency). However, in order to use such models, the statistics we hope to approximate must be decided upon a-priori, and the target value of these statistics must be determined before synthesis.

Another approach to dealing with the issue of unlabeled non-textual information is to learn a latent embedding for prosody or global style. This is the approach taken by models like (Skerry-Ryan et al., 2018; Wang et al., 2018), wherein in a bank of embeddings or a latent embedding space of prosody is learned from unlabelled data. While these approaches have shown promise, manipulating such latent variables only offers a coarse control over expressive characteristics of speech.

A mixed approach consists of combining engineered statistics with latent embeddings learned in an unsupervised fashion. This is the approach taken by models like Mellotron (Valle et al., 2019b). In Mellotron, utterances are conditioned on both audio statistics and a latent embedding of acoustic features derived from a reference acoustic representation. Despite its advantages, this approach still requires

---

[1]What is relevant statistically might not be perceptually.

determining these statistics before synthesis.

3. Flowtron

Flowtron is an autoregressive generative model that generates a sequence of mel spectrogram frames $p(x)$ by producing each mel-spectrogram frame based on previous mel-spectrogram frames $p(x) = \prod p(x_t|x_{1:t-1})$. Our setup uses a neural network as a generative model by sampling from a simple distribution $p(z)$. We consider two simple distributions with the same number of dimensions as our desired mel-spectrogram: a zero-mean spherical Gaussian and a mixture of spherical Gaussians with fixed or learnable parameters.

$$z \sim \mathcal{N}(z; 0, I) \qquad (1)$$

$$z \sim \sum_k \hat{\phi}_k \mathcal{N}(z; \hat{\mu}_k, \hat{\Sigma}_k) \qquad (2)$$

These samples are put through a series of invertible, parametrized transformations f, in our case affine transformations that transform $p(z)$ into $p(x)$.

$$x = f_0 \circ f_1 \circ \ldots f_k(z) \qquad (3)$$

As it is illustrated in (Kingma et al., 2016), in autoregressive normalizing flows the t-th variable z'_t only depends on previous timesteps $z_{1:t-1}$:

$$z'_t = f_k(z_{1:t-1}) \qquad (4)$$

By using parametrized affine transformations for f and due to the autoregressive structure, the Jacobian determinant of each of the transformations f is lower triangular, hence easy to compute. With this setup we can train Flowtron by maximizing the log-likelihood of the data, which can be done using the change of variables:

$$\log p_\theta(x) = \log p_\theta(z) + \sum_{i=1}^{k} \log |\det(J(f_i^{-1}(x)))| \quad (5)$$

$$z = f_k^{-1} \circ f_{k-1}^{-1} \circ \ldots f_0^{-1}(x) \qquad (6)$$

For the forward pass through the network, we take the mel-spectrograms as vectors and process them through several "steps of flow conditioned on the text and speaker ids. A step of flow here consists of an affine coupling layer, described below.

3.1. Affine Coupling Layer

Invertible neural networks are typically constructed using coupling layers (Dinh et al., 2014; 2016; Kingma &

Dhariwal, 2018). In our case, we use an affine coupling layer (Dinh et al., 2016). Every input x_{t-1} produces scale and bias terms, s and b respectively, that affine-transform the succeeding input x_t:

$$(\log s_t, b_t) = NN(x_{1:t-1}, text, speaker) \qquad (7)$$

$$x'_t = s_t \odot x_t + b_t \qquad (8)$$

Here $NN()$ can be any autoregressive causal transformation. This can be achieved by time-wise concatenation of a 0-valued vector to the input provided to $NN()$. The affine coupling layer preserves invertibility for the overall network, even though $NN()$ does not need to be invertible. This follows because the first input of $NN()$ is a constant and due to the autoregressive nature of the model the scaling and translation terms s_t and b_t only depend on $x_{1:t-1}$ and the fixed $text$ and $speaker$ vectors. Accordingly, when inverting the network, we can compute s_t and b_t from the preceding input $x_{1:t-1}$, and then invert x'_t to compute x_t, by simply recomputing $NN(x_{1:t-1}, text, speaker)$.

With an affine coupling layer, only the s_t term changes the volume of the mapping and adds a change of variables term to the loss. This term also serves to penalize the model for non-invertible affine mappings.

$$\log |\det(J(f_{coupling}^{-1}(x)))| = \log |s| \qquad (9)$$

With this setup, it is also possible to revert the ordering of the input x without loss of generality. Hence, we choose to revert the order of the input at every even step of flow and to maintain the original order on odd steps of flow. This allows the model to learn dependencies both forward and backwards in time while remaining causal and invertible.

3.2. Model architecture

Our text encoder modifies Tacotron's by replacing batch-norm with instance-norm. Our decoder and NN architecture, depicted in Figure 1, removes the *essential* Prenet and Postnet layers from Tacotron. We use the content-based tanh attention described in (Vinyals et al., 2015). We use the Mel Encoder described in (Hsu et al., 2018) for Flowtron models that predict the parameters of the Gaussian mixture.

Unlike (Ping et al., 2017; Gibiansky et al., 2017), where site specific speaker embeddings are used, we use a single speaker embedding that is channel-wise concatenated with the encoder outputs at every token. We use a fixed dummy speaker embedding for models not conditioned on speaker id. Finally, we add a dense layer with a sigmoid output the flow step closest to z. This provides the model with a gating mechanism as early as possible during inference to avoid extra computation.
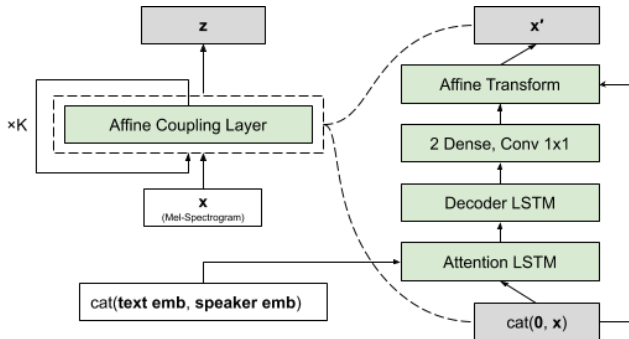
Figure 1: Flowtron network. Text and speaker embeddings are channel-wise concatenated. A 0-valued vector is concatenated with x in the time dimension.

3.3. Inference

Once the network is trained, doing inference is simply a matter of randomly sampling z values from a spherical Gaussian, or Gaussian Mixture, and running them through the network, reverting the order of the input when necessary. During training we used $\sigma^2 = 1$. The parameters of the Gaussian mixture are either fixed or predicted by Flowtron. In section 4.3 we explore the effects of different values for σ^2. In general, we found that sampling z values from a Gaussian with a lower standard deviation from that assumed during training resulted in mel-spectrograms that sounded better, as found in (Kingma & Dhariwal, 2018), and earlier work on likelihood-based generative models (Parmar et al., 2018). During inference we sampled z values from a Gaussian with $\sigma^2 = 0.5$, unless otherwise specified. The text and speaker embeddings are included at each of the coupling layers as before, but now the affine transforms are inverted in time, and these inverses are also guaranteed by the loss.

4. Experiments

This section describes our training setup and provides quantitative and qualitative results. Our quantitative results show that Flowtron has mean opinion scores (MOS) that are comparable to that of state of the art models for text to mel-spectrogram synthesis such as Tacotron 2. Our qualitative results display many features that are not possible or not efficient with Tacotron and Tacotron 2 GST. These features include control of the amount of variation in speech, interpolation between samples and style transfer between seen and unseen speakers during training.

We decode all mel-spectrograms into waveforms by using a single pre-trained WaveGlow (Prenger et al., 2019) model trained on a single speaker and available on github (Valle et al., 2019a). During inference we used $\sigma^2 = 0.7$. In consonance with (Valle et al., 2019b), our results suggests that WaveGlow can be used as an universal decoder.

Although we provide images to illustrate our results, they can best be appreciated by listening. Hence, we ask the readers to visit our website [2] to listen to Flowtron samples.

4.1. Training setup

We train our Flowtron, Tacotron 2 and Tacotron 2 GST models using a dataset that combines the LJSpeech (LJS) dataset (Ito et al., 2017) with two proprietary single speaker datasets with 20 and 10 hours each (Sally and Helen). We will refer to this combined dataset as LSH. We also train a Flowtron model on the *train-clean-100* subset of LibriTTS (Zen et al., 2019) with 123 speakers and 25 minutes on average per speaker. Speakers with less than 5 minutes of data and files that are larger than 10 seconds are filtered out. For each dataset we use at least 180 randomly chosen samples for the validation set and the remainder for the training set.

The models are trained on uniformly sampled normalized text and ARPAbet encodings obtained from the CMU Pronouncing Dictionary (Weide, 1998). We do not perform any data augmentation. We adapt the public Tacotron 2 and Tacotron 2 GST repos to include speaker embeddings as described in Section 3.

We use a sampling rate of 22050 Hz and mel-spectrograms with 80 bins using librosa mel filter defaults. We apply the STFT with a FFT size of 1024, window size of 1024 samples and hop size of 256 samples ($\sim 12ms$).

We use the ADAM (Kingma & Ba, 2014) optimizer with default parameters, 1e-4 learning rate and 1e-6 weight decay for Flowtron and 1e-3 learning rate and 1e-5 weight decay for the other models, following guidelines in (Wang et al., 2017). We anneal the learning rate once the generalization error starts to plateau and stop training once the the generalization error stops significantly decreasing or starts increasing. The Flowtron models with 2 steps of flow were trained on the LSH dataset for approximately 1000 epochs and then fine-tuned on LibriTTS for 500 epochs. Tacotron 2 and Tacotron 2 GST are trained for approximately 500 epochs. Each model is trained on a single NVIDIA DGX-1 with 8 GPUs.

We find it faster to first learn to attend on a Flowtron model with a single step of flow and large amounts of data than multiple steps of flow and less data. After the model has learned to attend, we transfer its parameters to models with more steps of flow and speakers with less data. Thus, we first train Flowtron model with a single step of flow on the LSH dataset with many hours per speaker. Then we fine tune this model to Flowtron models with more steps of flow. Finally, these models are fine tuned on LibriTTS with an optional new speaker embedding.

---

[2]https://nv-adlr.github.io/Flowtron

4.2. Mean Opinion Score comparison

We provide results that compare mean opinion scores (MOS) from real data from the LJS dataset, samples from a Flowtron with 2 steps of flow and samples from our implementation of Tacotron 2, both trained on LSH. Although the models evaluated are multi-speaker, we only compute mean opinion scores on LJS. In addition, we use the mean opinion scores provided in (Prenger et al., 2019) for ground truth data from the LJS dataset.

We crowd-sourced mean opinion score (MOS) tests on Amazon Mechanical Turk. Raters first had to pass a hearing test to be eligible. Then they listened to an utterance, after which they rated pleasantness on a five-point scale. We used 30 volume normalized utterances from all speakers disjoint from the training set for evaluation, and randomly chose the utterances for each subject.

The mean opinion scores are shown in Table 1 with 95% confidence intervals computed over approximately 250 scores per source. The results roughly match our subjective qualitative assessment. The larger advantage of Flowtron is in the control over the amount of speech variation and the manipulation of the latent space.

Source	Flows	Mean Opinion Score (MOS)
Real	-	4.274 ± 0.1340
Flowtron	3	3.665 ± 0.1634
Tacotron 2	-	3.521 ± 0.1721

Table 1: Mean Opinion Score (MOS) evaluations with 95% confidence intervals for various sources.

4.3. Sampling the prior

The simplest approach to generate samples with Flowtron is to sample from a prior distribution $z \sim \mathcal{N}(0, \sigma^2)$ and adjust σ^2 to control amount of variation. Whereas $\sigma^2 = 0$ completely removes variation and produces outputs based on the model bias, increasing the value of σ^2 will increase the amount of variation in speech.

4.3.1. SPEECH VARIATION

To showcase the amount of variation and control thereof in Flowtron, we synthesize 10 mel-spectrograms and sample the Gaussian prior with $\sigma^2 \in \{0.0, 0.5, 1.0\}$. All samples are generated conditioned on a fixed speaker *Sally* and text "*How much variation is there?*" to illustrate the relationship between σ^2 and variability.

Our results show that despite all the variability added by increasing σ^2, all the samples synthesized with Flowtron still produce high quality speech.

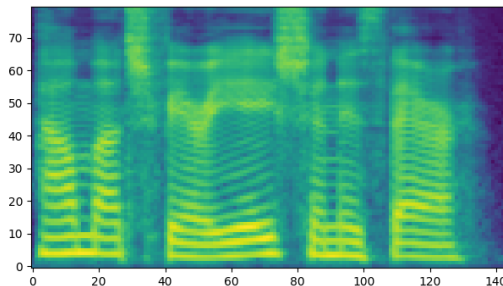Figure 2 also shows that unlike most SOTA models (Shen

et al., 2017; Arik et al., 2017b;a; Ping et al., 2017; Skerry-Ryan et al., 2018; Wang et al., 2018; Bińkowski et al., 2019), Flowtron generates sharp harmonics and well resolved formants without a compound loss nor Prenet or Postnet layers.
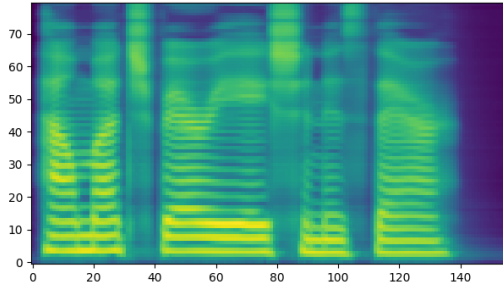
(a) $\sigma^2 = 0$
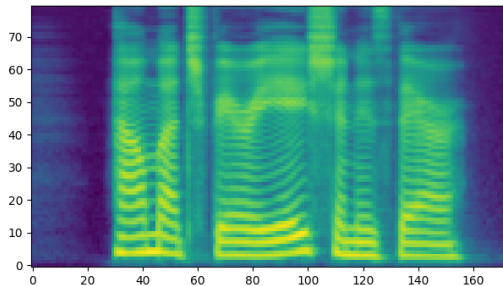
(b) $\sigma^2 = 0.5$

(c) $\sigma^2 = 1$

Figure 2: Mel-spectrograms generated with Flowtron using different σ^2. This parameter can be adjusted to control mel-spectrogram variability during inference.

Now we show that adjusting σ^2 is a simple and valuable approach that provides more variation and control than Tacotron, without sacrificing speech quality. For this, we synthesize 10 samples with Tacotron 2 using different values for the Prenet dropout probability $p \in \{0.45, 0.5, 0.55\}$. We scale the outputs of the dropout output such that the mean of the output remains equal to the mean with $p = 0.5$,

the value used during training. Although we also provide samples computed on values of $p \in [0, 1]$ in our supplemental material, we do not include them in our results because they are unintelligible.

In Figure 3 below we provide scatter plots from sample duration in seconds. Our results show that whereas $\sigma^2 = 0$ produces samples with no variation in duration, larger values of σ^2 produces samples with more variation in duration. Humans manipulate word and sentence length to express themselves, hence this is valuable.
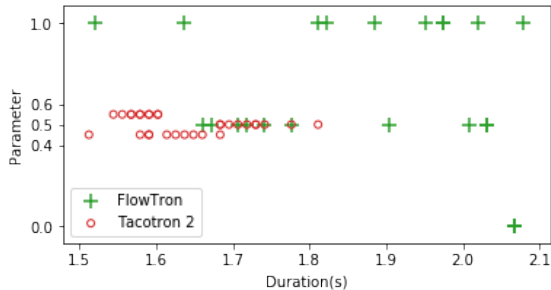

Figure 3: Sample duration in seconds given parameters σ^2 and p. These results show that Flowtron provides more variation in sample duration than Tacotron 2.

In Figure 4 we provide scatter plots of F_0 contours extracted with the YIN algorithm (De Cheveigné & Kawahara, 2002), with minimum F_0, maximum F_0 and harmonicity threshold equal to 80 Hz, 400 Hz and 0.3 respectively. Our results show a behavior similar to the previous sample duration analysis. As expected, $\sigma^2 = 0$ provides no variation in F_0 contour[3], while increasing the value of σ^2 will increase the amount of variation in F_0 contours.

Our results in Figure 4 also show that the samples produced with Flowtron are considerably less monotonous than the samples produced with Tacotron 2. Whereas increasing σ^2 considerably increases variation in F_0, modifying p barely produces any variation. This is valuable because expressive speech is associated with non-monotonic F_0 contours.

4.3.2. INTERPOLATION BETWEEN SAMPLES

With Flowtron we can perform interpolation in z-space to achieve interpolation in mel-spectrogram space. For this experiment we evaluate Flowtron models with and without speaker embeddings. For the experiment with speaker embeddings we choose the Sally speaker and the phrase "*It is well known that deep generative models have a rich latent space.*". We generate mel-spectrograms by sampling $z \sim \mathcal{N}(0, 0.8)$ twice and interpolating between them over 100 steps.

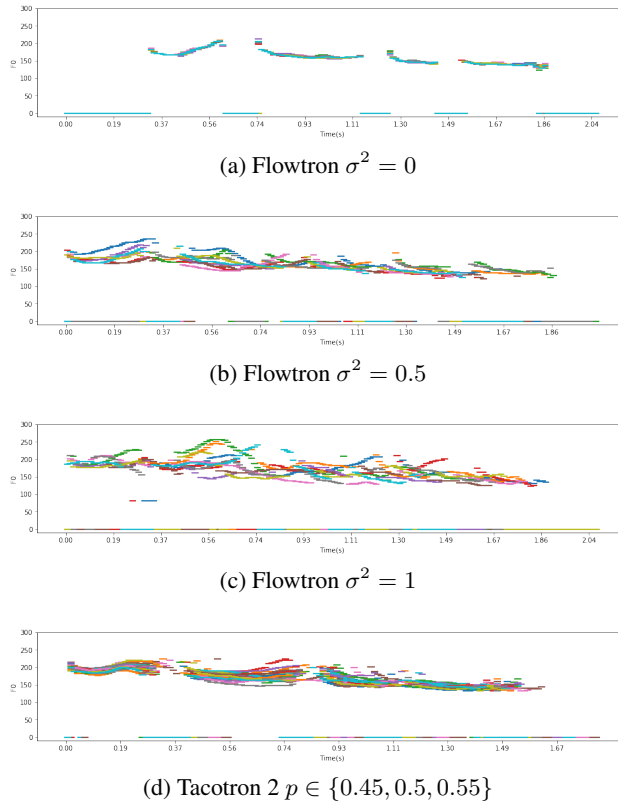[3]Variations in $\sigma^2 = 0$ are due to different z for WaveGlow.

(a) Flowtron $\sigma^2 = 0$

(b) Flowtron $\sigma^2 = 0.5$

(c) Flowtron $\sigma^2 = 1$

(d) Tacotron 2 $p \in \{0.45, 0.5, 0.55\}$

Figure 4: F_0 contours obtained from samples generated by Flowtron and Tacotron 2 with different values for σ^2 and p. Flowtron provides more expressivity than Tacotron 2.

For the experiment without speaker embeddings we interpolate between Sally and Helen using the phrase "*We are testing this model.*". First, we perform inference by sampling $z \sim \mathcal{N}(0, 0.5)$ until we find two z values, z_h and z_s, that produce mel-spectrograms with Helen's and Sally's voice respectively. We then generate samples by performing inference while linearly interpolating between z_h and z_s.

Our same speaker interpolation samples show that Flowtron is able to interpolate between multiple samples while producing correct alignment maps. In addition, our different speaker interpolation samples show that Flowtron is able to blurry the boundaries between two speakers, creating a speaker that combines the characteristics of both.

4.4. Sampling the posterior

In this approach we generate samples with Flowtron by sampling a posterior distribution conditioned on prior evidence containing speech characteristics of interest, as described in (Gambardella et al., 2019; Kingma & Dhariwal, 2018). In this experiment, we collect prior evidence z_e by performing a forward pass with the speaker id to be used during

inference[4], observed mel-spectrogram and text from a set of samples with characteristics of interest. If necessary, we time-concatenate each z_e with itself to fulfill minimum length requirements defined according to the text length to be said during inference.

Tacotron 2 GST (Wang et al., 2018) has an equivalent posterior sampling approach, in which during inference the model is conditioned on a weighted sum of global style tokens (posterior) queried through an embedding of existing audio samples (prior). For Tacotron 2 GST, we evaluate two approaches: in one we use a single sample to query a style token in the other we use an average style token computed over multiple samples.

4.4.1. SEEN SPEAKER WITHOUT ALIGNMENTS

In this experiment we compare Sally samples from Flowtron and Tacotron 2 GST generated by conditioning on the posterior computed over 30 Helen samples with the highest variance in fundamental frequency. The goal is to make a monotonic speaker sound expressive. Our experiments show that by sampling from the posterior or interpolating between the posterior and a standard Gaussian prior, Flowtron is able to make a monotonic speaker gradually sound more expressive. On the other hand, Tacotron 2 GST is barely able to alter characteristics of the monotonic speaker.

4.4.2. SEEN SPEAKER WITH ALIGNMENTS

We use a Flowtron model with speaker embeddings to illustrate Flowtron's ability to learn and transfer acoustic characteristics that are hard to express algorithmically but easy to perceive acoustically, we select a female speaker from LibriTTS with a distinguished nasal voice and oscillation in F_0 as our source speaker and transfer her style to a male speaker, also from LibriTTS, with acoustic characteristics that sound different from the female speaker. Unlike the previous experiment, this time the text and the alignment maps are transferred from the female to the male speaker.

Figure 5 is an attempt to visualize the transfer of these acoustic qualities we described. It shows that after the transfer, the lower partials of the male speaker oscillate more and become more similar to the female speaker.

4.4.3. UNSEEN SPEAKER STYLE

We compare samples generated with Flowtron and Tacotron 2 GST with speaker embeddings in which we modify a speaker's style by using data from the same speaker but from a style not seen during training. Whereas Sally's data used during training consists of news article readings, the evaluation samples contain Sally's interpretation of the somber and vampiresque novel Born of Darkness.

[4]To remove this speaker's information from z_e
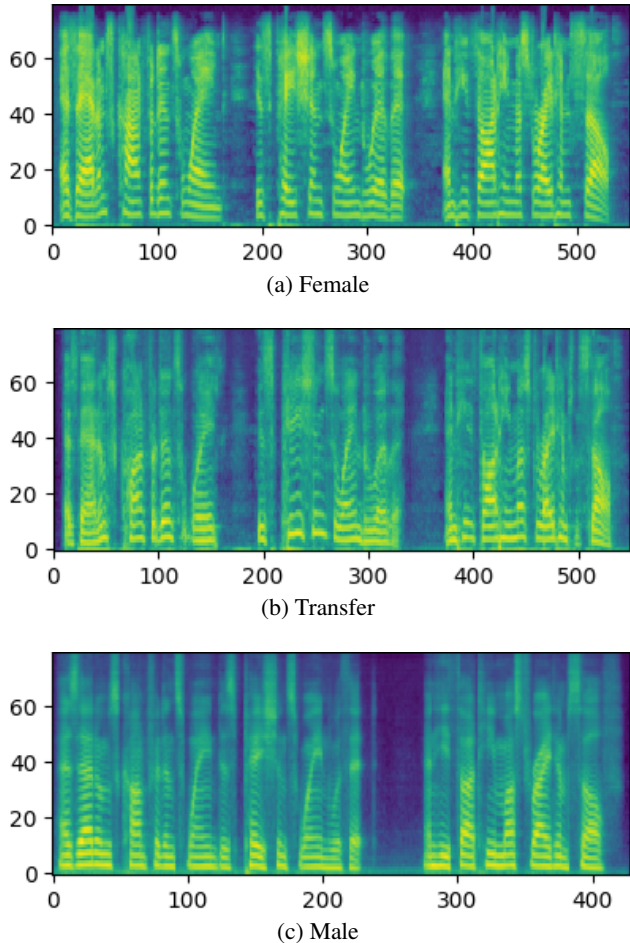

(a) Female

(b) Transfer

(c) Male

Figure 5: Mel-spectrograms from a female speaker, male speaker and a sample where we transfer the acoustic characteristics from the female speaker to the male speaker. It shows that the transferred sample is more similar to the female speaker than the male speaker.

Our samples show that Tacotron 2 GST fails to emulate the somber style from Born of Darkness's data. We show that Flowtron succeeds in transferring not only to the somber style in the evaluation data, but also the long pauses associated with the narrative style.

4.4.4. UNSEEN SPEAKER

In this experiment we compare Flowtron and Tacotron 2 GST samples in which we transfer the speaking style of a speaker not seen during training. Both models use speaker embeddings.

For these experiments, we consider two speakers. The first comes from speaker ID 03 from RAVDESS, a dataset with emotion labels. We focus on the label "surprised". The second speaker is Richard Feynman, using a set of 10 audio samples collected from the web.

For each experiment, we use the Sally speaker and the sentences "*Humans are walking on the street?*" and "*Surely you are joking mister Feynman.*", which do not exist in RAVDESS nor in the audio samples from Richard Feynman.

The samples generated with Tacotron 2 GST are not able to emulate the surprised style from RAVDESS nor Feynman's prosody and acoustic characteristics. Flowtron, on the other hand, is able to make Sally sound surprised, which is drastically different from the monotonous baseline. Likewise, Flowtron is able to pick up on the prosody and articulation details particular to Feynman's speaking style, and transfer them to Sally.

4.5. Sampling the Gaussian Mixture

In this last section we showcase visualizations and samples from Flowtron Gaussian Mixture (GM). First we investigate how different mixture components and speakers are correlated. Then we provide sound examples in which we modulate speech characteristics by translating one of the the dimensions of an individual component.

4.5.1. VISUALIZING ASSIGNMENTS

For the first experiment, we train a Flowtrom Gaussian Mixture on LSH with 2 steps of flow, speaker embeddings and fixed mean and covariance (Flowtron GM-A). We obtain mixture component assignments per mel-spectrogram by performing a forward pass and averaging the component assignment over time and samples. Figure 6 shows that whereas most speakers are equally assigned to all components, component 7 is almost exclusively assigned to Helen's data.
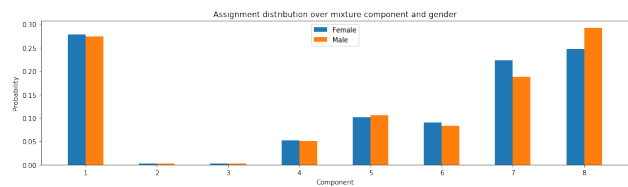

Figure 6: Component assignments for Flowtron GM-A. Unlike LJS and Sally, Helen is almost exclusively assigned to component 7.

In the second experiment, we train a Flowtron Gaussian Mixture on LibriTTS with 1 step of flow, without speaker embeddings and predicted mean and covariance (Flowtron GM-B). Figure 7 shows that Flowtron GM assigns more probability to component 7 when the speaker is male than when it's female. Conversely, the model assigns more probability to component 6 when the speaker is female than when it's male.
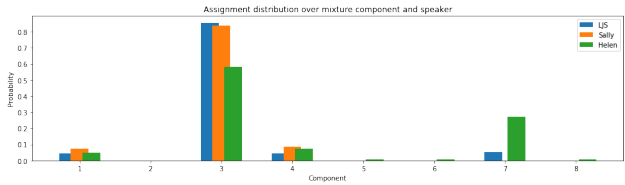

Figure 7: Component assignments for Flowtron GM-B. Components 7 and 8 are assigned different probabilities according to gender, suggesting that the information stored in the components is gender dependent.

4.5.2. TRANSLATING DIMENSIONS

In this subsection, we use the model Flowtron GM-A described previously. We focus on selecting a single mixture component and translating one of its dimensions by adding an offset.

The samples in our supplementary material show that we are able to modulate specific speech characteristics like pitch and word duration. Although the samples generated by translating one the dimensions associated with pitch height have different pitch contours, they have the same duration. Similarly, our samples show that translating the dimension associated with length of the first word does not modulate the pitch of the first word. This provides evidence that we can modulate these attributes by manipulating these dimensions and that the model is able to learn a disentangled representation of these speech attributes.

5. Discussion

In this paper we propose a new text to mel-spectrogram synthesis model based on autoregressive flows that is optimized by maximizing the likelihood and allows for control of speech variation and style transfer. Our results show that samples generated with FlowTron achieve mean opinion scores that are similar to samples generated with state-of-the-art text-to-speech synthesis models. In addition, we demonstrate that at no extra cost and without a compound loss term, our model learns a latent space that stores non-textual information. Our experiments show that FlowTron gives the user the possibility to transfer charactersitics from a source sample or speaker to a target speaker, for example making a monotonic speaker sound more expressive.

Our results show that despite all the variability added by increasing σ^2, the samples synthesized with FlowTron still produce high quality speech. Our results show that FlowTron learns a latent space over non-textual features that can be investigated and manipulated to give the user more control over the generative models output. We provide many examples that showcase this including increasing variation in mel-spectrograms in a controllable manner, transferring

the style from speakers seen and unseen during training to another speaker using sentences with similar or different text, and making a monotonic speaker sound more expressive.

Flowtron produces expressive speech without labeled data or ever seeing expressive data. It pushes text-to-speech synthesis beyond the expressive limits of personal assistants. It opens new avenues for speech synthesis in human-computer interaction and the arts, where realism and expressivity are of utmost importance. To our knowledge, this work is the first to demonstrate the advantages of using normalizing flow models in text to mel-spectrogram synthesis.

References

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. Expressive speech synthesis via modeling expressions with variational autoencoder. *arXiv preprint arXiv:1804.02135*, 2018.

Arik, S., Diamos, G., Gibiansky, A., Miller, J., Peng, K., Ping, W., Raiman, J., and Zhou, Y. Deep voice 2: Multi-speaker neural text-to-speech. *arXiv preprint arXiv:1705.08947*, 2017a.

Arik, S. O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., et al. Deep voice: Real-time neural text-to-speech. *arXiv preprint arXiv:1702.07825*, 2017b.

Badham, J., Lasker, L., Parkes, W. F., Rubinstein, A. B., Broderick, M., Coleman, D., and Wood, J. Wargames, 1983.

Bińkowski, M., Donahue, J., Dieleman, S., Clark, A., Elsen, E., Casagrande, N., Cobo, L. C., and Simonyan, K. High fidelity speech synthesis with adversarial networks. *arXiv preprint arXiv:1909.11646*, 2019.

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

Dinh, L., Krueger, D., and Bengio, Y. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Gambardella, A., Baydin, A. G., and Torr, P. H. Transflow learning: Repurposing flow models without retraining. *arXiv preprint arXiv:1911.13270*, 2019.

Gibiansky, A., Arik, S., Diamos, G., Miller, J., Peng, K., Ping, W., Raiman, J., and Zhou, Y. Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in neural information processing systems*, pp. 2962–2970, 2017.

Hsu, W.-N., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Cao, Y., Jia, Y., Chen, Z., Shen, J., et al. Hierarchical generative modeling for controllable speech synthesis. *arXiv preprint arXiv:1810.07217*, 2018.

Ito, K. et al. The LJ speech dataset, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.

Lee, J., Choi, H.-S., Jeon, C.-B., Koo, J., and Lee, K. Adversarially trained end-to-end korean singing voice synthesis system. *arXiv preprint arXiv:1908.01919*, 2019.

Nishimura, M., Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. Singing voice synthesis based on deep neural networks. In *Interspeech 2016*, pp. 2478–2482, 2016. doi: 10.21437/ Interspeech.2016-1027. URL http://dx.doi.org/ 10.21437/Interspeech.2016-1027.

Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A., and Tran, D. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018.

Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., Raiman, J., and Miller, J. Deep voice 3: 2000-speaker neural text-to-speech. *arXiv preprint arXiv:1710.07654*, 2017.

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

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *arXiv preprint arXiv:1712.05884*, 2017.

Skerry-Ryan, R., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., Weiss, R. J., Clark, R., and Saurous, R. A. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. *arXiv preprint arXiv:1803.09047*, 2018.

Umeda, N., Matsui, E., Suzuki, T., and Omura, H. Synthesis of fairy tales using an analog vocal tract. In *Proceedings of 6th International Congress on Acoustics*, pp. B159–162, 1968.

Valle, R., Li, J., Prenger, R., and Catanzaro, B. Mellotron github repo, 2019a. URL https://github.com/ NVIDIA/mellotron.

Valle, R., Li, J., Prenger, R., and Catanzaro, B. Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. *arXiv preprint arXiv:1910.11997*, 2019b.

Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. Grammar as a foreign language. In *Advances in neural information processing systems*, pp. 2773–2781, 2015.

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al. Tacotron: A fully end-to-end text-to-speech synthesis model. *arXiv preprint arXiv:1703.10135*, 2017.

Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y., and Saurous, R. A. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*, 2018.

Weide, R. L. The cmu pronouncing dictionary. *URL: http://www.speech.cs.cmu.edu/cgi-bin/cmudict*, 1998.

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