ABSTRACT

Deep learning Text-to-Speech (TTS) systems have achieved impressive generated speech quality, close to human parity. However, they suffer from training stability issues and incorrect alignment between the intermediate acoustic representation and the text input. In this work, we propose Regotron, a regularized Tacotron2 version which alleviates the training issues by augmenting the objective function with an additional term, which penalizes non-monotonic alignments in the location-sensitive attention mechanism. By introducing this regularization term we demonstrate its effectiveness in stabilizing the training process, produce a monotonic attention quicker (13% of the total number of epochs compared to Tacotron2) and reduce the alignment errors during inference. Moreover, Regotron has minimal additional computational overhead, reduces common TTS mistakes and at the same time achieves improved speech naturalness according to subjective mean opinion scores (MOS) collected from 50 evaluators.

Index Terms— Speech synthesis, TTS, regularization, alignment, Tacotron2, Regotron

1. INTRODUCTION

Text-to-Speech (TTS), also referred to as speech synthesis, is the task in which the machine is challenged to generate speech from text. Successfully building such models, is of vital importance for any realistic human machine interaction system. It has therefore a plethora of applications, which vary from typical phone assistants to applications suited for people with speaking disorders.

Neural TTS systems [1, 2] have achieved impressive performance, by generating natural speech close to human parity. Most systems usually incorporate two modules. The first is utilized to map the input text (or phoneme) sequence to some acoustic features [3, 4], e.g., mel spectrograms, while the second transforms the predicted features to speech waveform [5, 6, 7]. The latter module is also called a vocoder. Some approaches in the literature also aim to tackle the problem in an end-to-end manner [8].

The module which is responsible for the text to mel mapping is usually a sequence-to-sequence model [9] based on the encoder-decoder architecture [3, 2, 4]. Different architectures have been proposed, but regardless of the underlying approach, the cornerstone is the alignment module. Tacotron2 [2] utilizes a location sensitive attention (LSA) mechanism between the (text) encoder and the decoder. ClariNet [10] incorporates a multi-hop attention mechanism, while Transformer-TTS [11] and FastSpeech [12] utilize Multihead Attention [13].

Finding proper alignments helps the model in a twofold manner. First, convergence is assisted during training, and second the model is encouraged to produce more natural spectrograms. Inaccurate alignments may therefore result in non-convergence issues and this is the reason many non-autoregressive approaches require a pretrained autoregressive teacher network [12, 11, 14, 15]. Additionally, common TTS mistakes such as repetitions and word skips are attributed to the alignment mechanism [12]. Thus, for any TTS spectrogram generating architecture, it is crucial to learn how to properly align the input text with the mel-spectrogram.

To this end, a line of research is focused on improving the training stability and alignments of such text-to-mel models. Flowtron [16] progressively stacks normalizing flow layers to stabilize training. [17] uses an acoustic model as a force aligner and [18] utilizes a reinforcement learning framework where the agent is a duration predictor. FlowTTS [19] uses a length predictor to stabilize training. In a more similar spirit to our work, [20] assumes a prior Laplacian distribution on the weights and uses multiple attention blocks. In [21] authors build upon the attention introduced in [22] and experiment with variants of this GMM-based attention within the Tacotron architecture. FeatherTTS [23] departs from the GMM-based approaches and proposes a simple Gaussian attention mechanism with (monotonically) increasing mean values. [24] introduce a guided loss term, which multiplies attention weights with an exponential term to force a “nearly diagonal” alignment
between the input text and the generated mel frame. Another line of work aims to better guide the alignment module. [25] exploits alignments from a pretrained teacher network, while in [26] authors use alignments from independent trainable auxiliary attention mechanisms. The most popular ideas, however, appear to be iterative-based, utilizing Viterbi-like algorithms for selecting the optimal (monotonic) alignment [27, 28, 29].

In this work we propose a different approach and directly tackle both training stability and alignment by adding a regularization term in the total objective of Tacotron2 architecture. Formally, we augment the loss function with a term, which penalizes non-monotonic neighboring alignment weights as described in [30]. The regularized architecture that we propose is called Regotron. Our experiments verify that Regotron has smoother training behavior compared to Tacotron2 and consistently produces monotonic alignments at an early stage (13% of total epochs) of its training process, in unseen examples, where the fully converged Tacotron2 fails to do so. Moreover, our method introduces minimal additional computational cost, compared to other methods, and has a beneficial effect on the overall loss, since Regotron achieves lower generalization error than Tacotron2. Additionally, we experimentally verify that common TTS mistakes in difficult examples are reduced when our regularization strategy is used, which highlights a side-benefit of our approach. Finally, a Mean Opinion Score (MOS) evaluation shows that Regotron achieves comparable speech naturalness to Tacotron2, even at an intermediate training stage.

Our contributions can be summarized as: 1) the introduction of an efficient regularization term that acts directly on the alignment weights, enforcing monotonicity, with minimal additional computational cost 2) enhanced training stability and generalization due to better regularization, 3) faster convergence in terms of epochs, with respect to both the total loss and MOS evaluation, 4) increased robustness to common mistakes as a side benefit of adding the regularization term. Our code is available as open-source.

2. PROPOSED METHOD

2.1. Tacotron-2

Tacotron2 [2] is an autoregressive encoder-decoder architecture equipped with a location sensitive attention (LSA) mechanism [31]. The encoder takes as input text characters and outputs contextual hidden representations which are fed to the decoder, which in turn generates the mel spectrogram.

An important part of the architecture is the LSA mechanism, which helps the decoder attend at different parts of the encoder’s hidden representations. Our implementation differs from the original paper [31]. Some implementations use only the previous alignment to compute the location features, while others use the cumulated alignments sum over all the previous timesteps. Our implementation uses a concatenation of the attention weights from the previous timestep and the cumulative alignment sum.

The resulting matrix of alignment scores (as in Fig. 2), describes how well a part of the input, i.e., character (vertical axis), is aligned with the generated mel spectrogram frames (horizontal axis). We formally describe the alignment matrix as \( \mathbf{A} \in \mathbb{R}^{N \times M} \), where \( N \) is the number of input characters and \( M \) the number of the generated mel frames. In other words \( a_{ij} \) is the probability weight which inform us how well the \( i \)-th input character is aligned with the \( j \)-th mel frame. Note here that all TTS problems require monotonic alignment, which means that if the \( i \)-th input character is mapped to the \( j \)-th mel frame then the \((i+1)\)-th input character should be mapped at frame \( k > j \).

2.2. Monotonic Alignment Loss

In order to effectively encode the monotonic alignment demand, we aim for an objective that captures monotonicity [30]. Given the alignment matrix \( \mathbf{A} \in \mathbb{R}^{N \times M} \), we follow [30] and define the “mean attended position” (or centroid position) as:

\[
\langle a_j \rangle = \frac{1}{N} \sum_{i=1}^{N} a_{ij} \cdot i
\]  

which calculates the centroid of the alignment weight for a particular mel-frame \( j \). We then use these centroids to encode the monotonicity demand by setting:

\[
\langle a_{j+1} \rangle \geq \langle a_j \rangle, \quad j \in \{1, \cdots M\}
\]  

which describes that the \( j+1 \)-th mean attended position should be larger (or equal) than the \( j \)-th. [30] relaxes Eq. (2) by introducing a hyperparameter \( \delta \) which is dynamically reweighted by the the fraction of input and output sequence lengths. Putting it all together the alignment objective can be expressed as:

\[
L_A(\mathbf{A}) = \sum_{j=1}^{M-1} \max \left\{ \frac{\langle a_j \rangle - \langle a_{j+1} \rangle + \delta \frac{N}{M}, 0} \right\}
\]  

where the hyperparameter \( \delta \) controls how much to relax the monotonicity assumption, the fraction \( N/M \) dynamically adjusts \( \delta \) according to input and output sequence lengths and finally the denominator \( N \) acts as an input-based “normalization” factor. Note that the relaxed version of Eq. (2) is satisfied, if the first term of the max operand becomes negative and the term is neglected from the total sum in Eq. (3). This means that we only penalize the terms that violate the (relaxed) monotonicity demand.

1https://github.com/efthymisgeo/regotron

2is based on the NVIDIA repo
2.3. Regotron

We now describe our proposed Regotron architecture, which is a regularized version of Tacotron2. Essentially, we enforce the location sensitive attention mechanism of Tacotron2 to produce monotonic alignments. This way we stabilize the training procedure of Tacotron2 and we also get correct alignments earlier during the training procedure. The regularization term is added to the overall Tacotron2 objective $L_T$ and the augmented Regotron loss function $L_R$ becomes:

$$L_R = L_T + \lambda L_A$$

(4)

where $\lambda$ is a positive (small) weighting value, used to adjust the effect of the second (alignment) term. In practice we only tune this hyperparameter. We also note here that the second term is applied directly to the attention matrix $A$, while the first to the generated mels.

3. EXPERIMENTAL SETUP

We follow standard practice in the literature [2, 7] and train our models using the LJSpeech (LJS) dataset [32]. We use a sampling rate of 22050 Hz and mel-spectrograms with 80 bins using the librosa mel filter defaults. We apply the STFT with a FFT size of 1024, hop size of 256 (\textasciitilde 12ms), and window size of 1024 samples. Following the repo from NVIDIA\(^3\) we choose Adam optimizer with learning rate of $10^{-3}$ and train the model for 1500 epochs. We also anneal the learning rate by a factor of 0.3 every 500 epochs. For a fair comparison we retrained Tacotron2 and the results refer to our version. All models are trained using mixed precision for faster training using a batch size of 152. For any additional hyperparameters, we refer to our code or nvidia repo. For the Regotron architecture we use the exact same hyperparameters as in Tacotron2. We additionally set $\delta = 0.01$ for all experiments and we only perform a hyperparameter search for $\lambda$ in the range $10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$. For our setup we found $10^{-5}$ to perform better.

4. EXPERIMENTS

In order to verify the benefits of our proposed Regotron we carry experiments with the vanilla Tacotron2 architecture, as well as earlier and later checkpoints of our proposed model.

4.1. Loss function analysis

In this section we analyze the loss function behavior during training. Specifically, we illustrate the respective error $L_T$ for both the vanilla as well as our architecture in Fig 1. Two things are clear from the following figures.

\(^3\)https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/SpeechSynthesis/Tacotron2

4.1.1. Training Stabilization

First of all, our regularized version does not appear to have spiky behavior, while the vanilla version appears to do so. Specifically, we can see that Vanilla Train Loss, as well as Vanilla Val Loss, are the only errors which are spiky, see the red box in Fig. 1a and its zoomed version on the same image. Note here that the purpose of the zoomed version is to make clear which losses pose spiky behavior and which do not. It is not meant to encapsulate the whole spike. It is therefore clear that our regularized version better stabilizes the training procedure when weighted appropriately, i.e for $\lambda = 10^{-3}, 10^{-4}, 10^{-5}$. For smaller weight values, e.g., $10^{-6}$ the alignment term becomes insignificant and the Regotron objective tends to the vanilla Tacotron2 one. For larger values, e.g., $10^{-2}$ the alignment term $L_A$ dominates over $L_T$, resulting in non-convergence.

4.1.2. Generalization Error

Fig. 1b illustrates the training and validation losses from epoch 1000 till the final epoch 1500. It is clear that the proposed method has lower validation error compared to the vanilla Tacotron2. This means that the generated spectrograms of the validation set are more similar to the ground truths (similarity is measured via $L_T$). An alternative metric is to measure the gap between the validation and the training loss for each model (generalization error). To do so, we averaged over the last 500 and the last 100 epochs for both models. The results are illustrated in Table 1. As it is clear, Regotron achieves both lower validation and generalization error, which implies that introducing the alignment loss has a beneficial effect on $L_T$ and results in a better-performing model (in terms of the objective).

| Model (avg. epochs) | Val Loss | Generalization |
|---------------------|----------|----------------|
| Tacotron2 (100)     | 0.4121   | 0.1319         |
| Regotron (100)      | 0.4017   | 0.1232         |
| Tacotron2 (500)     | 0.4115   | 0.1324         |
| Regotron (500)      | 0.4023   | 0.1247         |

4.2. Alignment Analysis

In this section, we analyze the alignment results from Regotron on phrases from the held-out test set. Since our model directly optimizes the monotonic alignment we expect to see different behavior from the vanilla Tacotron2. After inspecting our checkpoints we find that a very early version of our model ($\sim 13\%$ of the total training epochs) achieves better alignment in general than the vanilla Tacotron2. This result is illustrated...
Fig. 1: Both plots illustrate the train/val loss functions for the vanilla as well our modified architecture. The red box in the left subfigure is zoomed in the center right of the same image.

Fig. 2: The plots illustrate alignments from the Tacotron2 model, as well as an early (13%) and the final version of Regotron. The vertical axis illustrates the input character sequence, while the horizontal denotes the generated number of mel frames. The phrases uttered are “The jury did not believe him, and the verdict was for the defendants.” in the top row (alignments a, b, c). The horizontal row in the plot corresponds the comma in the uttered phrase. In the bottom row (alignments d, e, f) the phrase “nineteen sixty-three, merely to disarm her and to provide a justification of sorts,” is uttered.

in Fig. 2 where the first column denotes Tacotron2 alignments, the second Regotron(13%) alignments and the final Regotron alignments.

Although the Tacotoron2 alignment appears blurry is some
spots (better viewed if zoomed in), even the very early (13%) version of Regotron has figured out how to properly align and by the time of convergence it is easy to notice that Regotron’s alignments are much clearer in general. The results of Fig. 2 are not cherry picked and should be considered representative of the alignment behavior on the held-out test set (see Regotron repo for more alignments). This result can be attributed to the direct penalization of the non-monotonic alignments in the location sensitive attention mechanism.

Table 2: Robustness Analysis. Evaluation is performed on a special set of hard-to-pronounce sentences. Each kind of word error is counted at most one per sentence.

| Model       | Repeat | Skip  | Mispron. | Error Sent. |
|-------------|--------|-------|----------|-------------|
| Tacotron2   | 3      | 19    | 13       | 54%         |
| Regotron (66%) | 1      | 10    | 13       | 42%         |
| Regotron    | 0      | 9     | 10       | 36%         |

4.3. Robustness Analysis

The location sensitive attention (LSA) mechanism in both Tacotron2 and Regotron architectures, may cause wrong attention alignments between input characters and mel-spectrogram, resulting in instability with word repeating and word skipping. To evaluate the robustness of our model we choose two Regotron models (an earlier version (~ 66%) and the final one) and a Tacotron2 architecture as a baseline to compare with.

Following FastSpeech [12] we generated 50 difficult examples and manually annotated them regarding three error types. Skip errors, in which a letter/word is skipped while the nearby are uttered correctly. Mispronunciations in which a word is pronounced incorrectly, i.e., multiple letters are skipped or confused resulting in non-interpretable speech. Repeats in which the same word or letter is repeated usually in place of some other. Note that during our evaluation mispronunciations are considered a broader error class than skip errors and are therefore evaluated separately. Table 2 summarizes the results of our evaluation.

The results depict that the repeat errors can be fully alleviated by injecting the alignment term, which in turn shows that better alignment is able to tackle mistakes of this nature. The number of skip errors is also decreased but still evident, which shows that these errors are partially due to wrong alignment and partially due to model or data insufficiency. For example in phrases such as “64x64” the intermediate “times” or “x” is skipped in all models examined and is uttered as if it was “64 64”. We note here that this particular type of error could be resolved with different text normalization. In the scope of this analysis however, we aim to reveal the error types that are alleviated by the addition of the proposed regularization term solely.

The mispronunciation errors are the most difficult to tackle as seen from the results in our Table 1. Phrases s.a. “Http”, “.dll”, “exe” or even isolated letters, e.g., “c”, are not uttered correctly and the speech produced is non-interpretable by all three models. Overall, Regotron decreases all types of errors and most importantly degrades the total error percentage, i.e., the number of sentences in which at least one error occurs.

Table 3: MOS scores with 95% confidence intervals.

| Model                | MOS       |
|----------------------|-----------|
| GT                   | 4.377 ± 0.097 |
| GT (Mel + WaveGlow)  | 3.983 ± 0.118 |
| Tacotron2 (Mel + WaveGlow) | 3.898 ± 0.125 |
| RegotronEARLY (Mel + WaveGlow) | 3.989 ± 0.110 |
| RegotronLATE (Mel + WaveGlow) | 4.034 ± 0.106 |

4.4. Speech Naturalness

In order to assess the naturalness of the synthesized speech, we randomly selected 30 samples from our test set and used WaveGlow [7] as a vocoder to generate the corresponding speech clips with 4 different methods, including ground truth, Tacotron2, RegotronEARLY and RegotronLATE spectrograms, as well as the original speech clips. The purpose of including an intermediate Regotron model, is to evaluate the acceleration (in terms of training steps) in convergence w.r.t. the produced speech naturalness.

![Fig. 3: Boxplots of the ratings collected by 50 evaluators. The distributions of the answers are computed with a kernel density estimator function.](https://papers.nips.cc/paper/2019/hash/f63f65b503e22cb970527f23c9ad7db1-Abstract.html)

Overall we considered 150 speech clips and invited 50 evaluators, where each sample was rated by at least 5 participants on a Likert scale from 1 to 5 with 0.5 increments. Each evaluation test was conducted independently so to avoid possible bias when raters assigned their subjective score.
In Table 3 we present the computed subjective mean opinion scores (MOS), indicating that our fully converged version of Regotron reports marginal better MOS than Tacotron2. Furthermore, by plotting the distributions of the collected answers (see Figure 3), we observe that the early version of Regotron achieves similar MOS with Tacotron2, while the fully converged Regotron architecture presents similar performance to the speech clips that were synthesized from ground truth spectrograms. Moreover, Figure 3 shows that the fully converged Regotron (Regotron\_PLATE) manages to cut the long tail of rare bad ratings. We believe that this is related to the decreased error types, e.g., mispronunciations. Overall, the results clearly show that the proposed regularization term improves the produced speech quality.

5. CONCLUSIONS

This work presents Regotron, a regularized Tacotron2 variant, which aims to stabilize training and produce monotonic alignments. We directly penalize non-monotonic alignments by directly adding a regularization term in the total objective. We show that the loss curves becomes smoother and at the same time monotonic alignments are produced. Common TTS mistakes are reduced as a side benefit and lower generalization is gap is achieved. MOS remains unharmed from the addition of our term. Our framework is efficient since minimal additional computation is introduced.

The objective we introduce here is generic and can be applied at any TTS system, with potential modifications. We believe that this kind of solutions are in the pathway for tackling training stability and wrong-alignment issues in TTS systems efficiently. Applying our method to other TTS systems is very important in order to establish it as a plug ‘n’ play method in a wide range of speech synthesis networks.

6. REFERENCES

[1] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, “Wavenet: A generative model for raw audio,” 2016.

[2] 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 Agiomyrgiannakis, and Yonghui Wu, “Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions,” in IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, pp. 4779–4783.

[3] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Sammy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, “Tacotron: Towards end-to-end speech synthesis,” in Interspeech 2017, 2017, pp. 4006–4010.

[4] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ömer Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller, “Deep voice 3: Scaling text-to-speech with convolutional sequence learning,” in 6th International Conference on Learning Representations, ICLR 2018, 2018.

[5] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobto, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis, “Parallel WaveNet: Fast high-fidelity speech synthesis,” in Proceedings of the 35th International Conference on Machine Learning, 10–15 Jul 2018, vol. 80 of Proceedings of Machine Learning Research, pp. 3918–3926.

[6] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aäron van den Oord, Sander Dieleman, and Koray Kavukcuoglu, “Efficient neural audio synthesis,” in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018, 2018, vol. 80 of Proceedings of Machine Learning Research, pp. 2415–2424.

[7] Ryan Prenger, Rafael Valle, and Bryan Catanzaro, “Waveglow: A flow-based generative network for speech synthesis,” in IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, 2019, pp. 3617–3621.

[8] Jeff Donahue, Sander Dieleman, Mikolaj Binkowski, Erich Elsen, and Karen Simonyan, “End-to-end adversarial text-to-speech,” in 9th International Conference on Learning Representations, ICLR, 2021.

[9] Ilya Sutskever, Oriol Vinyals, and Quoc V Le, “Sequence to sequence learning with neural networks,” Advances in neural information processing systems, vol. 27, 2014.

[10] Wei Ping, Kainan Peng, and Jitong Chen, “Clarinet: Parallel wave generation in end-to-end text-to-speech,” in 7th International Conference on Learning Representations, ICLR, 2019.

[11] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, “Neural speech synthesis with transformer network,” in The Thirty-Third AAAI Conference on Artificial Intelligence, 2019.
[12] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, “Fastspeech: Fast, robust and controllable text to speech,” Advances in Neural Information Processing Systems, vol. 32, 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems 2017, 2017, pp. 5998–6008.

[14] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, “Fastspeech 2: Fast and high-quality end-to-end text to speech,” in 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021.

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

[16] Rafael Valle, Kevin J. Shih, Ryan Prenger, and Bryan Catanzaro, “Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis,” in 9th International Conference on Learning Representations, ICLR, 2021.

[17] Takuma Okamoto, Tomoki Toda, Yoshinori Shiga, and Hisashi Kawai, “Transformer-based text-to-speech with weighted forced attention,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 6729–6733.

[18] Hyunseung Chung, Sang-Hoon Lee, and Seong-Whan Lee, “Reinforce-Aligner: Reinforcement Alignment Search for Robust End-to-End Text-to-Speech,” in Proc. Interspeech 2021, 2021, pp. 3635–3639.

[19] Chenfeng Miao, Shuang Liang, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao, “Flow-TTS: A non-autoregressive network for text to speech based on flow,” in IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP), 2020, pp. 7209–7213.

[20] Qingyun Dou, Xixin Wu, Moquan Wan, Yiting Lu, and Mark J.F. Gales, “Deliberation-Based Multi-Pass Speech Synthesis,” in Proc. Interspeech 2021, 2021, pp. 136–140.

[21] Eric Battenberg, RJ Skerry-Ryan, Soroosh Mariooryad, Daisy Stanton, David Kao, Matt Shannon, and Tom Bagby, “Location-relative attention mechanisms for robust long-form speech synthesis,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 6194–6198.

[22] Alex Graves, “Generating sequences with recurrent neural networks,” in arXiv preprint arXiv:1308.0850, 2013.

[23] Qiao Tian, Chao Liu, Zewang Zhang, Heng Lu, Linghui Chen, Bin Wei, Pujiang He, and Shan Liu, “FeatherTTS: Robust and Efficient attention based Neural TTS,” in Proc. 11th ISCA Speech Synthesis Workshop (SSW 11), 2021, pp. 200–204.

[24] Hideyuki Tachibana, Katsuya Uenoyma, and Shunsuke Aihara, “Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 4784–4788.

[25] Qingyun Dou, Joshua Efiong, and Mark J.F. Gales, “Attention Forcing for Speech Synthesis,” in Proc. Interspeech 2020, 2020, pp. 4014–4018.

[26] Shilun Lin, Fenglong Xie, Li Meng, Xinlui Li, and Li Lu, “Triple M: A Practical Text-to-Speech Synthesis System with Multi-Guidance Attention and Multi-Band Multi-Time LPCNet,” in Proc. Interspeech 2021, 2021, pp. 3640–3644.

[27] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon, “Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search,” in Advances in Neural Information Processing Systems, 2020.

[28] Zhen Zeng, Jianzong Wang, Ning Cheng, Tian Xia, and Jing Xiao, “AlignTTS: Efficient feed-forward text-to-speech system without explicit alignment,” in IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2020, pp. 6714–6718.

[29] Rohan Badlani, Adrian Lancucki, Kevin J Shih, Rafael Valle, Wei Ping, and Bryan Catanzaro, “One tts alignment to rule them all,” arXiv preprint arXiv:2108.10447, 2021.

[30] Annette Rios, Chantal Amrhein, Noëmi Aepli, and Rico Sennrich, “On biasing transformer attention towards monotonicity,” in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, June 2021.

[31] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, “Attention-based models for speech recognition,” in Advances in Neural Information Processing Systems, 2015, vol. 28.

[32] Keith Ito and Linda Johnson, “The lj speech dataset,” https://keithito.com/LJ-Speech-Dataset/, 2017.