Efficient Strategies of Few-Shot On-Device Voice Cloning

Tasnima Sadekova, Vadim Popov, Vladimir Gogoryan, Ivan Vovk,
Andrey Drogolyub, Dmitry Polubotko, Mikhail Kudinov
Huawei Noah's Ark Lab
Moscow, Russia
`{sadekova.tasnima, vadim.popov, gogoryan.vladimir, vovk.ivan, drogolyub.andrey, dmitry.polubotko, kudinov.mikhail}@huawei.com`

Abstract

Recent advances in neural text-to-speech allowed to build multi-speaker systems capable of performing high-fidelity speech generation. However, it is often desirable to be able to add a new voice to a text-to-speech system based on only a few recordings. In this work, we study several approaches to the design of on-device voice cloning. Starting from a multi-speaker TTS system we improve its quality for a target speaker by fine-tuning the feature generation module on a small speech sample. We compare the performance of a feature generation module based on conventional Tacotron2 with step-wise monotonic attention with the ones based on Non-attentive Tacotron and Glow-TTS. We show that Non-attentive Tacotron significantly outperforms the attention-based model and demonstrate that a compact on-device TTS system of good quality can be obtained using only 1 minute of adaptation data with no more than 200 iterations of SGD corresponding to less than 1 hour of on-device training time on a consumer mobile phone.

Index Terms: Text-to-speech, voice cloning, few-shot learning, on-device learning

1 Introduction

Neural text-to-speech (TTS) systems have become very popular among the speech community during the last decade [Shen et al., 2018, van den Oord et al., 2016, Ren et al., 2019, 2020]. Modern TTS models can synthesize natural human voice in both single-speaker [Shen et al., 2018], and multi-speaker scenarios [Ren et al., 2019] as long as enough training data is available. When the volume of the training data for a speaker is limited the problem of adding his or her voice to a TTS system can be solved by means of voice cloning (VC) techniques i.e. speaker encoding and speaker adaptation. Below we refer to speaker encoding and speaker adaptation based approaches as zero-shot and few-shot VC correspondingly.

Both approaches proved to be effective in previous studies e.g. [Jia et al., 2018, Arik et al., 2018] where a neural network for speaker verification was used as encoder network for VC. In [Arik et al., 2018] it was shown that fine-tuning of the whole baseline zero-shot model on a small amount of data led to considerable improvement in both sound quality and speaker similarity. Later, in [Chen et al., 2021] FastSpeech2 [Ren et al., 2020] model was used as a backbone of a voice cloning system based on speaker adaptation with only 5k speaker-specific parameters.

On-device implementation of VC presents additional challenges to both TTS backbone models [Valin and Skoglund, 2019, Popov et al., 2020b,a] and cloning algorithms because the hardware of modern mobile phones and wearable devices is still not powerful enough to allow running a full-fledged training procedure. The goal of this work is to propose an efficient voice cloning solution capable

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

of working on-device. We start with the existing on-device TTS system based on Tacotron2 and LPCNet [Popov et al., 2020a] and build a zero-shot and a few-shot VC systems based on this pipeline in the same manner as [Jia et al., 2018] and [Tan et al., 2021] using 1 minute of adaptation data. We study approaches to development of a baseline zero-shot system and find parameters of model adaptation which are optimal for on-device training. On-device model adaptation may be useful in such scenarios as on-device speech-to-speech translation preserving user's voice. However, we should also admit that broad access to on-device voice cloning technology may potentially be used with malicious purposes such as fraudulent calls.

The paper structure is as follows: in Section 2 we describe models used in our pipelines; in Section 3 we describe our experimental setup; in Section 4 human evaluation results are discussed; we conclude in Section 5.

2 Voice cloning models

LPCNet In all experiments described below we use LPCNet [Valin and Skoglund, 2019] as a vocoder. In contrast to other popular neural vocoders [van den Oord et al., 2016, Kalchbrenner et al., 2018, Kumar et al., 2019, Kong et al., 2020], this RNN-based autoregressive model predicts excitation signal e_t which is then processed by linear-prediction filter outputting final speech samples s_t. Due to relative simplicity of predicting e_t LPCNet is capable of generating speech of good quality having a small number of parameters (1.4m) and providing highest efficiency on CPU. Despite having such a small model LPCNet can serve as a universal vocoder [Valin and Skoglund, 2019]. For the experiments described further we trained a multi-speaker LPCNet on LibriSpeech dataset [Panayotov et al., 2015].

Multi-speaker Tacotron2 We used a modified version of the small Tacotron2 model with 18m parameters described in [Popov et al., 2020a] but with additional 256-dimensional speaker embedding input. The model has decoder with 3-layer LSTM of 512 units and a decreased 4-layer postnet with kernel sizes $[5, 5, 5, 5]$. In our experiments we used model with Stepwise Monotonic Attention (SMA) [He et al., 2019]. We took the same approach to the baseline zero-shot model as [Jia et al., 2018] with the same speaker verification model used as speaker encoder. Our version of Tacotron2 predicted normalized acoustic features which were then denormalized and sent to LPCNet. Normalization was done by subtracting mean and dividing by standard deviation calculated over training dataset.

Multi-speaker Non-attentive Tacotron Non-attentive Tacotron (NAT) [Shen et al., 2020] is a feature generation model based on Tacotron2 making use of duration predictor module aimed at improving robustness of the generation process ([Ren et al., 2019, Yu et al., 2020, Kim et al., 2020]). The model is trained using a combination of the feature reconstruction loss $\mathcal{L}_{feat}$ borrowed from Tacotron2 and duration prediction loss $\mathcal{L}_{dur}$ which is a mean squared error between predicted and golden durations d and d^*: $\mathcal{L}_{dur} = \frac{1}{N}\|d - d^*\|_2^2$, where N is an input sequence length.

In order to make the model more suitable for on-device implementation we decreased the number of parameters in all modules of Non-attentive Tacotron compared to the original version. Encoder 2-layer Bi-LSTM with 512 units was replaced with 1-layer Bi-LSTM with 256 units. The size of Bi-LSTM in the duration predictor was decreased to 256 from 512. Decoder 2-layer LSTM with 1024 units was replaced with 3-layer LSTM with 512 units. For the Postnet we used 4-layer neural network with 1d-convolutions with 256 channels and kernel sizes $[5, 3, 3, 3]$. Both Tacotron2 and Non-attentive Tacotron generated 3 feature outputs per step. For multi-speaker version of NAT we used the same approach as for Tacotron2 described in Section 2

Multi-speaker Glow-TTS The third feature generation model included in our tests was Glow-TTS [Kim et al., 2020]. We considered Glow-TTS as a potential alternative to Non-attentive Tacotron because training Glow-TTS does not need golden durations. This is an important advantage for on-device training.

Glow-TTS is a flow-based [Kingma and Dhariwal, 2018] architecture making use of Monotonic Alignment Search (MAS). The main modules of Glow-TTS are text encoder, flow-based decoder and duration predictor. During training encoder f_{enc} maps input text c into a sequence of parameters of a Gaussian distribution (μ_i, σ_i) while the decoder f_{dec} maps the target spectrogram x into a sequence of latent variables z_j. The alignment $A(i, j)$ between latent variables z_j and predicted parameters

(μ_i, σ_i) is calculated via Monotonic Alignment Search. Its output is also used for training the duration predictor f_{dur}. During inference the duration predictor controls the number of samples drawn from each distribution $\mathcal{N}(z; \mu_i, \sigma_i)$ and the decoder carries out inverse transform of the sampled latents z_j. We modify Glow-TTS to output normalized acoustic features for LPCNet and to work with the input speaker embedding generated by the speaker encoding network as described above. Since Glow-TTS is designed to have a duration predictor as a separate module, we also make this duration predictor conditioned on speaker embedding. We should note though that we did not try to implement a compact version of Glow-TTS and did not change any parameters including temperature.

Feature enhancement with GAN We also considered an alternative approach based on post-filtering Tacotron2 outputs with Generative Adversarial Network [Goodfellow et al., 2014] aimed at improving sound quality and similarity in the zero-shot scenario. To ensure both good sound quality and speaker similarity and drawing inspiration from the architectures widely used in the field of voice conversion [Kaneko et al., 2019, Kameoka et al., 2018] we implemented GAN consisting of three networks: generator G, discriminator D and classifier C.

We choose to train Least Squares GAN [Mao et al., 2017] because this variant is known to be quite stable during training. Discriminator and generator adversarial losses are thus given by

$$\mathcal{L}_D = \frac{1}{2}\mathbb{E}_{t\sim T, x\sim P_t}[(1 - D(e(t), x))^2] + \frac{1}{2}\mathbb{E}_{t\sim T, x\sim Q_t}[D(e(t), G(e(t), x))^2] \tag{1}$$

$$\mathcal{L}_G^{adv} = \frac{1}{2}\mathbb{E}_{t\sim T, x\sim Q_t}[(1 - D(e(t), G(e(t), x)))^2] \tag{2}$$

where P_t denotes the distribution of all real acoustic features of the speaker t, Q_t is the distribution of all acoustic features generated by the Tacotron2 model with target speaker t, T is the set of all speakers in our training dataset and $e(\cdot)$ denotes the operation of extracting a speaker embedding for the target speaker $t \in T$ with pretrained speaker encoder (see Section 2). Both generator G and discriminator D are conditioned on the target speaker embedding $e(t)$. Speaker similarity for training set speakers is imposed by the speaker loss $\mathcal{L}_G^{spk}$ coming from the classifier C operating on speech segments of 128 frames and estimating the probabilities $\{C_t(x)\}_{t\in T}$ that this segment is pronounced by the speaker t. The classifier is trained by minimizing cross-entropy loss. We also use identity mapping loss $\mathcal{L}_G^{id}$ [Kaneko et al., 2019] in order to prevent the generator from losing linguistic information.

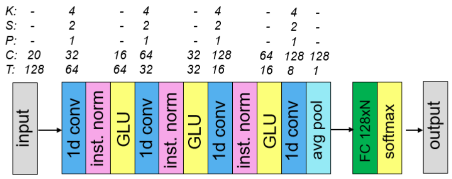

Figure 1: Classifier architecture. K is the kernel size, S is the stride, P is the zero padding, C is the number of output channels, T is the output length and N is the number of speakers in the training set.

The generator and discriminator architectures are mostly borrowed from [Kaneko et al., 2019] except that we use only 1D convolution blocks and a target speaker embedding instead of source and target speaker codes. The architecture of the classifier is given at Figure 1. All the networks utilize Instance Normalization [Ulyanov et al., 2016] and Gated Linear Unit (GLU) non-linearity [Dauphin et al., 2017].

3 Experiments

We carried out two series of subjective evaluation tests. The goal of the first testing was to select the models with the best performance on in-field data. For this experiment we used a private dataset consisting of fragments of radio shows and dialogues extracted from video clips of varying length and acoustic environment. For the second experiment aimed at estimating the optimal number of fine-tuning steps we used public data (10 speakers from LibriTTS) because we also wanted to compare our model with AdaSpeech [Chen et al., 2021].

As mentioned in Section 2, we used the same vocoder in all the experiments. Since our main goal was to enable on-device voice cloning, we choose LPCNet because of its good quality and high efficiency on CPU. We trained multi-speaker LPCNet on LibriSpeech subsets *train-clean-100* and *train-clean-360* with clean recordings. The recordings were preprocessed by a denoising algorithm.

Baseline zero-shot models We start with zero-shot models based on Tacotron2, Non-attentive Tacotron and Glow-TTS. All zero-shot models are trained on a combination of data from LibriTTS [Zen et al., 2019] and VCTK [Yamagishi et al., 2019]. To ensure quality of the training data and robustness of the attention we split long recordings into shorter ones and removed speakers with less than 5 minutes of data. We also removed leading and trailing silence for VCTK recordings. After filtering out speakers with noisy records and insufficient amount of data we had 664 speakers from LibriTTS and 105 speakers from VCTK. For all models we used speaker embedding size 256. For the Glow-TTS model we used 12 flow-blocks for decoder with 192 channels.

As our initial hypothesis was that the higher amount of speakers can be beneficial for similarity while lower sound quality could be compensated by post-filtering we also prepared an additional Tacotron2 model trained on the same LibriSpeech data as was used for training LPCNet. However, in this case we also filtered speakers with less than 5 minutes of data. After cleaning, the dataset contained data for 1100 speakers. This model was used for testing capabilities of the GAN-based filter. Below we refer to this model as *TT2-LS*.

Model with post-filtering To train the GAN-based post-filter we used the preprocesessed LibriSpeech data described above and a pretrained classifier network C with accuracy 82%. The generator network did post-processing of the recordings synthesized by *TT2-LS*. The whole GAN was trained for 1000k iterations in a standard manner. Hyperparameters $\lambda_{cls} = 0.01$ and $\lambda_{id} = 10.0$ were found to perform best. During the experiments we found that the generator is slightly unstable when processing some frames in the very beginning and in the very end of an utterance, so we do not apply post-filtering to the first and the last 10 frames. Also we replaced pitch-related features produced by the generator with those initially generated by the Tacotron2 because otherwise synthesized speech had unnatural prosody.

Model adaptation For the first experiment we collected a test set consisting of recordings made by 4 male and 6 female speakers. Test data for three of these speakers were parts of publicly available TTS datasets (Nancy [King and Karaiskos, 2011] and two speakers $p280$ and $p315$ from held-out VCTK dataset) while the remaining part contained small excerpts of radio programs or speech recordings extracted from video clips. For the second experiment we took 10 speakers from LibriTTS not used for training of the zero-shot baseline models.

The zero-shot models used speaker embeddings extracted from randomly chosen short target speaker recordings while the few-shot models were fine-tuned on few randomly chosen target speaker recordings with total duration of 1 minute. In the latter case, we averaged speaker embeddings extracted from all adaptation recordings. We also added 1 minute of recordings made by another speaker during adaptation of Tacotron2-based models as it improved attention stability. Golden durations for NAT were generated with Montreal Forced Aligner [McAuliffe et al., 2017] software. During adaptation of Tacotron2 only Decoder and Postnet modules were updated. For NAT models also Duration predictor was fine-tuned.

Listening tests In the first experiment we assessed quality of voice cloning solutions by carrying out listening tests. We asked the experts to estimate three aspects: speaker similarity, sound quality and overall naturalness of the synthesized speech. Five-point scale was used for estimating sound quality and naturalness while speaker similarity was evaluated on four-point scale. Each audio was assessed by 5 participants to ensure reliability of our results. Length of the text input varied from several words to several sentences (up to 40 words in total for each text input). Every VC model synthesized 20 speech samples with each of 10 voices.

In the second experiment we evaluated 5-point scale Mean Opinion Score (MOS) and 4-point scale similarity score with step 0.5. Each model synthesized 5 sentences from LibriTTS with each of 10 voices. Each utterance was assessed by 15 Master assessors. Because Glow-TTS and the model with GAN-based post-filter performed worse in the first experiments only Tacotron2 and Non-attentive Tacotron were included into the second testing.

Speech samples used in subjective listening tests are available at `https://fsvc-on-device.github.io`.

4 Results

Table 1 shows the results of the first subjective evaluation. *TT2-LS* and *TT2-LS-GAN* are the zero-shot Tacotron2 models trained on LibriSpeech data but in the latter case the output recordings are post-processed by the GAN-based post-filter. The number after the hyphen in the model name stands for the number of iterations of the adaptation procedure so 0 means a baseline zero-shot model. We chose the number of training iterations after which the models generated the best sound quality in our preliminary experiments.

VC model	Sim.	Nat.	Sound
TT2-LS	2.20	2.57	2.67
TT2-LS-GAN	2.32	2.97	3.14
TT2-0	2.47	2.99	3.38
TT2-1200	3.05	3.71	3.30
GlowTTS-0	2.09	2.72	2.77
GlowTTS-600	2.27	2.87	3.03
NAT-0	**2.98**	**4.01**	**3.72**
NAT-200	**3.15**	**3.84**	**3.66**

Table 1: Speaker similarity (‘Sim”), naturalness (“Nat.”) and sound quality (“Sound”) of VC models.

VC model	MOS	Similarity
TT2-0	2.21 ± 0.08	2.3 ± 0.07
TT2-200	2.88 ± 0.08	2.78 ± 0.07
TT2-1200	3.19 ± 0.08	2.94 ± 0.07
TT2-1800	3.18 ± 0.08	2.94 ± 0.06
NAT-0	3.29 ± 0.08	2.86 ± 0.07
NAT-100	3.57 ± 0.08	3.11 ± 0.07
NAT-200	$\mathbf{3.71 \pm 0.07}$	$\mathbf{3.21 \pm 0.06}$
NAT-800	3.43 ± 0.08	3.12 ± 0.07
GT	4.46 ± 0.06	3.64 ± 0.05
GT-resynth	3.77 ± 0.05	3.32 ± 0.06

Table 2: Mean Opinion Score (“MOS”) and Similarity of VC models.

Table 1 demonstrates that the adapted models consistently outperform their zero-shot counterparts in terms of similarity and that Tacotron2 and Non-attentive Tacotron give better performance than Glow-TTS. Surprisingly, in this experiment the zero-shot version of Non-attentive Tacotron performed better than the adapted one in terms of speech quality. During analysis we found that the adapted NAT models had issues with tempo for some speakers. Also we can see that GAN-based post-filtering clearly improves baseline *TT2-LS* model. In particular, it improves speaker similarity either because of introducing classifier to GAN framework or due to the correlation between speaker similarity and overall speech quality. However, despite the fact that *TT2-LS-GAN* model was trained on a dataset with more speakers, experimental results do not allow us to conclude that it is any better than *TT2-0* in terms of speaker similarity.

Few-shot adaptation of multi-speaker Glow-TTS model was problematic because of loss blowing up on early iterations so in some cases we had to do preliminary fine-tuning of two final (in reverse mode) layers of the decoder freezing the rest of the model. Eventually the overall quality of models for different speakers was uneven and the worst models had considerable pronunciation problems. The reason for such problems can be both difficulties of the alignment training without golden durations and unstable flow-based decoder.

Table 2 shows the results of the subjective evaluation of different checkpoints of Non-attentive Tacotron and Tacotron2 during speaker adaptation. NAT models clearly outperform Tacotron2. Surprisingly, even non-adapted zero-shot NAT model has higher MOS than the best adapted Tacotron2 model *TT2-1800*. This can be partly explained by the fact that low MOS correlates well with errors in speech tempo and sonic artifacts (see Figure 2). However, in contrast to the previous experiment zero-shot NAT had more issues with tempo than the adapted models which suggests that stability of the duration predictor varies across speakers. We also see that 200 iterations of NAT adaptation are enough to obtain the quality comparable to the recordings resynthesized from ground truth features. On the other hand the considerable quality gap between the ground truth recordings and the resynthesized ones clearly indicates that the universal LPCNet vocoder is a serious quality bottleneck of the whole pipeline. On the other hand overall MOS score is comparable to the one reported in [Chen et al., 2021].

On-device adaptation The results of on-device adaptation performance of *NAT* model are shown in Table 3. We tested 3 implementations of the training algorithm. For all versions we used

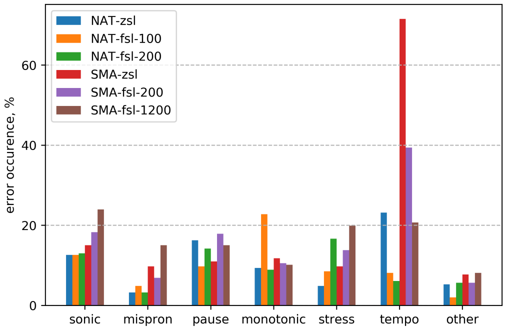

Figure 2: Typical errors per model. Subjective evaluation by experts (25 phrases, 10 assessors). All models are adapted to the same speaker from LibriTTS. *zsl* stands for non-adapted zero-shot models; *fsl-N* stands for models fine-tuned on the target speaker data for N iterations.

Table 3: On-device training performance on Kirin 980 (2 threads ARM A76 CPU). Training parameters: minibatch size: 11, max. input length: 70 characters, Adam optimizer; stopping at loss=1.25

Implementation	Max. RAM	Fine-tuning time	Power consumption
FP32	800MB	1.6h	2000 mAh
MPv1	492MB	1.3h	1700 mAh
MPv2	640MB	0.9h	1700 mAh

BLAS-Enhance module from BOLT library [BOLT Library, 2020] as a mathematical backend. The baseline non-optimized implementation with full-precision calculations (*FP32*) reached a peak performance of 18 sec/it with 800 MB maximum memory consumption. Total adaptation time with 200 iterations of Adam optimizer reached 1.6h because of CPU throttling. As we could not achieve stable convergence in half-precision mode we used mixed precision calculations with loss function calculated in fp32 while all other operations were carried out in fp16 mode. Moreover, for the versions with mixed precision we had to increase the number of iterations. For these models we chose the early stopping criterion: the adaptation stopped when loss reached the same value as *FP32* model after 200 iterations ($\mathcal{L} = 1.25$). For further optimization we fused all operations inside the LSTM cell into one integral function (*MPv1*) and then all calls of the LSTM function for each time step into a single function (*MPv2*). We also used master copy of weights technique [P. Micikevicius, 2017] in *MPv2* implementation. Our best model *MPv2* achieved a peak performance of 11 sec/it. with 55 mins total training time. We observed no quality degradation compared to the baseline *FP32* version. Inference time was comparable to the pipeline described in [Popov et al., 2020a].

5 Conclusion

In this work we have investigated and thoroughly compared various voice cloning techniques. We modified our current on-device TTS pipeline to be able to adapt to unseen voices both in zero-shot and few-shot manner. Extensive human evaluation demonstrated that the model making use of duration predictor trained on golden durations performs better and adapts faster to unseen voices. This model clearly outperforms attention-based Tacotron2 and Glow-TTS which is capable of training without explicit ground truth alignment. We also showed that despite the fact that post-processing can improve quality of the synthesized speech in some cases the same or better performance may be obtained by means of filtering out low-quality recordings from the training data. Finally, we implemented on-device training of our best performing voice cloning model and showed that it is possible to adapt TTS system to unseen voices using only 1 minute of adaptation data. The whole on-device voice cloning pipeline requires less than 1 hour of mobile CPU training and results in TTS models of reasonable quality.

References

S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou. Neural Voice Cloning with a Few Samples. In *Advances in Neural Information Processing Systems 31*, pages 10019–10029. Curran Associates, Inc., 2018.

BOLT Library. Noah's Ark Bolt Library. `https://github.com/huawei-noah/bolt`, 2020. [Online; accessed 19-October-2021].

M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, sheng zhao, and T.-Y. Liu. AdaSpeech: Adaptive Text to Speech for Custom Voice. In *International Conference on Learning Representations*, 2021.

Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 933–941, 2017.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

M. He, Y. Deng, and L. He. Robust Sequence-to-Sequence Acoustic Modeling with Stepwise Monotonic Attention for Neural TTS. In *Interspeech 2019*, 2019.

Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno, and Y. Wu. Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. In *Advances in Neural Information Processing Systems 31*, pages 4480–4490. Curran Associates, Inc., 2018.

N. Kalchbrenner, E. Elsen, K. Simonyan, et al. Efficient Neural Audio Synthesis. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2410–2419. PMLR, 10–15 Jul 2018.

H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo. StarGAN-VC: non-parallel many-to-many Voice Conversion Using Star Generative Adversarial Networks. *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 266–273, 2018.

T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo. StarGAN-VC2: Rethinking Conditional Methods for StarGAN-Based Voice Conversion. *Interspeech 2019*, pages 679–683, Sep 2019. doi: 10.21437/Interspeech.2019-2236.

J. Kim, S. Kim, J. Kong, and S. Yoon. Glow-TTS: A Generative Flow for Text-to-Speech via Monotonic Alignment Search. *ArXiv*, abs/2005.11129, 2020.

S. King and V. Karaiskos. The Blizzard Challenge 2011, 2011.

D. P. Kingma and P. Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in Neural Information Processing Systems 31*, pages 10215–10224. Curran Associates, Inc., 2018.

J. Kong, J. Kim, and J. Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *ArXiv*, abs/2010.05646, 2020.

K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, et al. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis. In *Advances in Neural Information Processing Systems 32*, pages 14910–14921. Curran Associates, Inc., 2019.

X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least Squares Generative Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, Oct 2017.

M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech 2017*, pages 498–502, 08 2017. doi: 10.21437/Interspeech.2017-1386.

P. Micikevicius. Mixed-Precision Training of Deep Neural Networks. `https://developer.nvidia.com/blog/mixed-precision-training-deep-neural-networks/`, 2017. [Online; accessed 19-October-2021].

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, 2015.

V. Popov, S. Kamenev, M. Kudinov, S. Repyevsky, T. Sadekova, V. Bushaev, V. Kryzhanovskiy, and D. Parkhomenko. Fast and lightweight on-device TTS with Tacotron2 and LPCNet. In *Interspeech 2020*, 2020a.

V. Popov, M. Kudinov, and T. Sadekova. Gaussian LPCNet for Multisample Speech Synthesis. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6204–6208, 2020b.

Y. Ren, Y. Ruan, X. Tan, T. Qin, et al. FastSpeech: Fast, Robust and Controllable Text to Speech. In *Advances in Neural Information Processing Systems 32*, pages 3171–3180. Curran Associates, Inc., 2019.

Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. *CoRR*, abs/2006.04558, 2020. URL `https://arxiv.org/abs/2006.04558`.

J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. 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, April 2018. doi: 10.1109/ICASSP.2018.8461368.

J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu. Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling. *ArXiv*, abs/2010.04301, 2020.

D. Tan, H. Huang, G. Zhang, and T. Lee. CUHK-EE voice cloning system for ICASSP 2021 m2voc challenge. *CoRR*, abs/2103.04699, 2021. URL `https://arxiv.org/abs/2103.04699`.

D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR*, abs/1607.08022, 2016. URL `http://arxiv.org/abs/1607.08022`.

J. Valin and J. Skoglund. LPCNet: Improving Neural Speech Synthesis through Linear Prediction. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5891–5895, May 2019. doi: 10.1109/ICASSP.2019.8682804.

J. Valin and J. Skoglund. A Real-Time Wideband Neural Vocoder at 1.6kb/s Using LPCNet. In G. Kubin and Z. Kacic, editors, *Interspeech 2019*, 2019.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. In *9th ISCA Speech Synthesis Workshop*, pages 125–125, 2016.

J. Yamagishi, C. Veaux, and K. MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92), 2019.

C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, et al. Durian: Duration informed attention network for speech synthesis. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 2027–2031. ISCA, 2020.

H. Zen, R. Clark, R. J. Weiss, V. Dang, Y. Jia, Y. Wu, Y. Zhang, and Z. Chen. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Interspeech*, 2019.