BOFFIN TTS: FEW-SHOT SPEAKER ADAPTATION BY BAYESIAN OPTIMIZATION

Henry B. Moss[♣*] *Vatsal Aggarwal*[◇] *Nishant Prateek*[◇] *Javier González*[◇] *Roberto Barra-Chicote*[◇]

[♣] STOR-i Centre for Doctoral Training, Lancaster University [◇]Amazon Research Cambridge

ABSTRACT

We present BOFFIN TTS (Bayesian Optimization For FIne-tuning Neural Text To Speech), a novel approach for few-shot speaker adaptation. Here, the task is to fine-tune a pre-trained TTS model to mimic a new speaker using a small corpus of target utterances. We demonstrate that there does not exist a *one-size-fits-all* adaptation strategy, with convincing synthesis requiring a corpus-specific configuration of the hyper-parameters that control fine-tuning. By using Bayesian optimization to efficiently optimize these hyper-parameter values for a target speaker, we are able to perform adaptation with an average 30% improvement in speaker similarity over standard techniques. Results indicate, across multiple corpora, that BOFFIN TTS can learn to synthesize new speakers using less than ten minutes of audio, achieving the same naturalness as produced for the speakers used to train the base model.

Index Terms— text-to-speech, speaker adaptation, Bayesian optimization, transfer learning

1. INTRODUCTION

Given enough data, text to speech (TTS) systems can learn to convincingly mimic speakers across a wide range of acoustic and phonetic styles. However, training systems from scratch requires tens of hours of high-quality audio and reliable transcriptions, either from a single speaker to create speaker-specific models or spread across several speakers when training multi-speaker models [1, 2, 3, 4]. Training models on less data sacrifices quality and reliability [5].

To scale TTS catalogues across speakers for whom we have limited data, we adapt existing multi-speaker systems to generate new speakers - a well-studied form of transfer learning known as **speaker adaptation**[6]. Adaptation is possible in scenarios where we have just minutes of target audio and partial phoneme coverage, as the robust representation of text and subsequent mappings to coherent speech are shared between the speakers [1]. Only a small proportion of our network's capacity encodes speaker-specific information. We, therefore, need only enough utterances to learn **speaker identity** (the characteristics defining a target speaker's voice).

Existing strategies for speaker adaptation fall into two broad categories. Many approaches use pre-trained auxiliary encoding networks to extract speaker characteristics to be combined with linguistic features as inputs to a TTS model [7, 8, 9, 10]. In contrast, alternative approaches fine-tune the weights of existing multi-speaker models to synthesize new speakers [11, 12]. As fine-tuning provides the most natural adaptation across multiple TTS models [11, 12], and is applicable to any existing system (without the need for training additional encoding networks), it is the focus of this report.

Our primary contribution is to demonstrate that successful speaker adaptation requires fine-tuning of adaptation hyper-parameters (henceforth referred to as the **adaptation strategy**) for each target speaker. We carefully tune the hyper-parameters governing adaptation and introduce two additional parameters not previously used for speaker adaption, demonstrating that the optimal hyper-parameter configuration depends subtly on the acoustic and phonetic properties of the target speaker alongside attributes of the target corpus (like audio-quality and size). For example, the amount of regularization required to prevent over-fitting (of which few-shot speaker adaptation is particularly susceptible [12]), depends on the quality and quantity of adaptation utterances.

In this work, we formulate few-shot speaker-adaptation as an optimization problem - the task of finding appropriate hyper-parameter values for any given speaker. Our proposed BOFFIN[1] TTS system automatically and efficiently solves this optimization problem through the hyper-parameter tuning framework of **Bayesian optimization** (BO), providing a fully automatic speaker-adaptation system suitable for general target speakers. BO has been shown to find high-performing hyper-parameters in competitively few model fits for many machine learning tasks [13], surpassing the performance of human tuners for problems in computer vision [14], natural language processing [15], and recently for reinforcement learning in AlphaGo [16]. However, BO has yet to see wide-spread use in TTS, where grid-based and random searches are still commonplace for hyper-parameter optimization. We hope that our successes with BO for speaker adaptation will encourage its more wide-spread use across TTS.

We evaluate BOFFIN TTS across three distinct scenarios, varying both the number of speakers in the base multi-speaker model and corpora audio-quality.

*This research was completed during an internship at Amazon Research. Correspondence to h.moss@lancaster.ac.uk and agvatsal@amazon.com.

[1]Boffin: British slang for a scientific expert.

2. SYSTEM DESCRIPTION

2.1. Base Multi-Speaker Model

Our **base model** (the model we adapt to target speakers) is a Tacotron2 [17] style multi-speaker system explained in detail by [1, 18], consisting of an acoustic context-generation model and neural vocoder. Our acoustic model relies on an attention-based sequence-to-sequence network to generate context sequences (represented as mel-spectrograms) from input texts (see Figure 1). Unlike Tacotron2 which models raw graphemes, we pre-process input text with a grapheme-to-phoneme module. To condition on individual speakers, we learn a speaker-embedding from a one-hot-encoding of speaker IDs (following [2]). This dense representation of speaker characteristics is presented to the attention module alongside encoded input text, to be decoded as a speaker-specific mel-spectrogram. Model weights are tuned with an ADAM optimizer to minimize the teacher-forced L1 loss between predicted and extracted mel-spectrograms. To complete the TTS pipeline, we convert mel-spectograms to waveforms using the multi-speaker neural vocoder of [19]. This vocoder is trained across 74 speakers and suitable for generating natural speech for our wide-range of adaptation speakers.

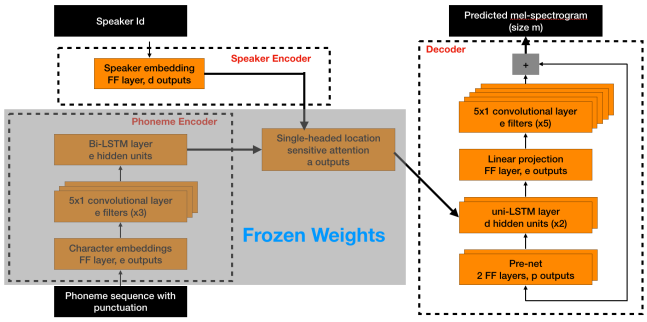


Fig. 1: Multi-speaker acoustic model architecture.

2.2. Base-line Speaker Adaptation System

Existing approaches for speaker adaptation by fine-tuning, although targeting different TTS architectures [11, 12], all share the same approach which we apply to our chosen model to form a **base-line** adaptation system. To synthesize speakers not present in the training corpus, we continue the same learning process used to train the base model, but replace the training data with utterances of only the target speaker to allow the fine-tuning of weights and the learning of a new speaker embedding with respect to this new data. To avoid over-fitting to small collections of target utterances, we hold-out 20% of adaptation data to form a validation set for early-stopping. From extensive human tuning, we know that the hyper-parameter configuration chosen for our base-model is capable of producing high-quality synthesis (achieving higher than four MOS naturalness scores for several speakers). We, therefore, expect this hyper-parameter configuration to form a competitive base-line for adaptation. Nevertheless, we later demonstrate that we can achieve a substantial improvement in adaptation quality using BOFFIN TTS.

3. BOFFIN TTS

There are two key difference between BOFFIN TTS and the base-line adaptation system. We allow the hyper-parameters controlling our adaptation to change to suit the target-speaker and, crucially, propose a framework for finding their optimal configuration in an efficient and automatic manner.

3.1. How Does BOFFIN TTS Control Adaptation?

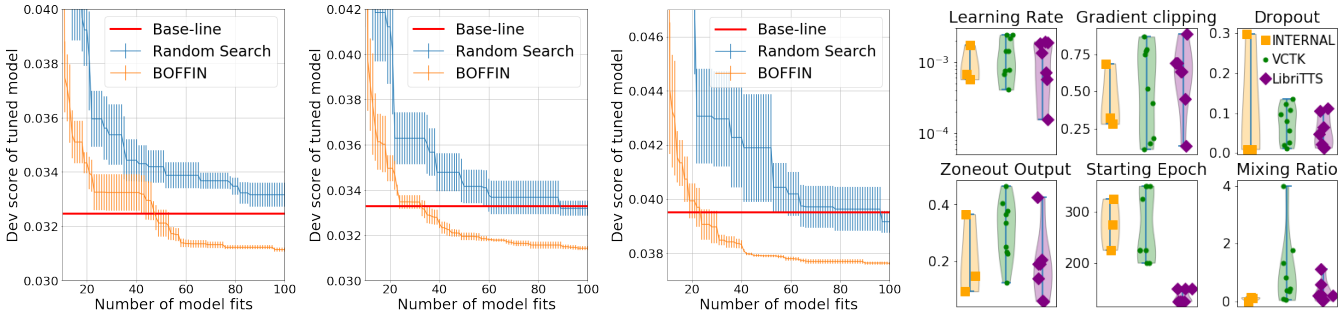
The key to effective adaptation is to learn characteristics of the target speaker without losing the generalizability of the base model (a phenomenon known as catastrophic forgetting). To this end, we believe there are nine key hyper-parameters that determine the success of adaptation. These include seven parameters already widely used in machine learning to control learning dynamics (learning rate, batch size, decay-factor and gradient-clipping threshold) and to perform regularization (dropout and two zoneout parameters[20]), alongside two parameters unique to BOFFIN TTS.

Although, tuning these seven standard hyper-parameters allowed us to learn the identity of the target speaker, the resulting models often show poor generalization capabilities. Therefore, we propose two additional hyper-parameters. Firstly, we supplement our adaptation corpus, forming a tuneable ratio of target speakers to speakers already seen by the model (a simple approach to mitigate catastrophic forgetting known as a rehearsal method). Finally, we also tune which epoch of our trained base-model from which we begin adaptation. A base model before full convergence to the base speakers can provide a model more amenable for adaptation.

In addition to hyper-parameter tuning, we also exploit the specific architecture of our chosen base-model. Rather than allowing our fine-tuning to update all model weights (as in [12]), we only allow fine-tuning of the weights in our speaker embedding and decoder modules (i.e those containing speaker-specific information, see Figure 1). We know that our encoder and attention modules are already able to facilitate synthesis across multiple speakers and we found that freezing their weights during adaptation led to more robust synthesis.

3.2. How Does BOFFIN TTS Optimize Adaptation?

Learning an optimal adaptation strategy for a target speaker is a difficult high-dimensional hyper-parameter optimization (HPO) problem. As is common in HPO, this optimization task is characterized by expensive evaluations (requiring a full adaptation to evaluate each single hyper-parameter configuration), a mixture of discrete and continuous variables, and a lack of analytical gradients for our objective function (the performance of adaptation) with respect to all our hyper-parameters. Consequently, we cannot apply gradient-based optimizers and the high-dimension of our turning task makes



(a) INTERNAL speaker A. (b) VCTK speaker p362. (c) LibriTTS speaker 114. (d) Tuned hyper-parameter values.

Fig. 2: (a, b, c): Loss of the current best hyper-parameter configuration found by each system as we adapt to three randomly selected speakers from each corpora. We plot means and standard error for BOFFIN TTS and RS based on 5 runs with different random seeds, alongside the loss achieved by the base-line adaptation system. (d): Hyper-parameter values chosen by BOFFIN TTS for multiple target speakers across three different data-sets. Each point represents a single speaker. We plot the six hyper-parameters whose optimal values show the largest variation across speakers.

a simple grid-search computationally infeasible (and likely ineffective [21]). We, therefore, use Bayesian optimization.

In a nutshell, BO is able to provide highly efficient HPO by using information from already evaluated hyper-parameter configurations to predict which untested configurations are 'likely' to perform well and therefore should be next evaluated. In particular, to choose the $t+1^{th}$ hyper-parameter for evaluation, we fit a Gaussian process model [22] to our t collected configuration-evaluation pairs $\mathcal{D}_t = \{\mathbf{x}_i, y_i\}_{i=1,..,t}$ across the hyper-parameter space $\mathcal{X}$, producing Gaussian predictions of performance at each configuration $\mathbf{x} \in \mathcal{X}$ of $y(\mathbf{x})|\mathcal{D}_t$. We then evaluate the configuration that we expect (according to our model) will provide the largest improvement over the best current best evaluation (with score $y'_t = \min_{i=1,..,t} y_i$), i.e we next evaluate configuration

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{y(\mathbf{x})|\mathcal{D}_t}\left[\max(y'_t - y(\mathbf{x}), 0)|\mathcal{D}_t\right]. \quad (1)$$

For Gaussian processes, the inner expression of (1) and its gradients have convenient analytical forms (see [23] for a comprehensive review of BO). Therefore, $\mathbf{x}_{t+1}$ can be efficiently found using a standard gradient-based optimizer.

We consider the performance of BOFFIN TTS when seeking to minimize L1 mel-spectogram loss across a held-out validation set of target speaker utterances. Although L1 loss does not necessarily correlate exactly with the perceptual quality of synthesized samples (as is the case for all objective TTS metrics), we found it informative enough to find hyper-parameters with high perceptual scores (Section 4). Adaptation to speakers from three different corpora is presented in Figure 2 (experimental details are discussed in Section 4). Our plots start after an initialization stage of 10 random hyper-parameters, as this is required to provide a meaningful initial model across $\mathcal{X}$. Note that replacing BOFFIN TTS's BO component with random search (RS) fails to substantially improve upon our baseline (not speaker-specific) adaptation system. We need a sophisticated tuner like BO to find speaker-specific adaptation strategies. In addition, Figure 2d shows that not only does the optimal hyper-parameter configuration vary between data-sets, but also across each individual speaker within each corpora. For example, our proposed *Mixing Ratio* hyper-parameter requires larger values in general across the VCTK corpus than for our other corpora, however, the optimal *Mixing Ratio* still varies substantially across just the VCTK speakers.

System	INTERNAL	VCTK	LibriTTS
base-synth	3.45 ± 0.08	3.76 ± 0.10	3.10 ± 0.10
base-truth	3.84 ± 0.08	4.05 ± 0.08	4.10 ± 0.08
adapt-synth	3.43 ± 0.10	3.6 ± 0.10	2.90 ± 0.10
adapt-truth	4.05 ± 0.08	4.09 ± 0.08	3.97 ± 0.08

Table 1: Comparing the mean naturalness scores achieved by BOFFIN TTS on target speakers (adapt-synth), by the base multi-speaker model on base speakers(base-synth), and by true audio for both target (adapt-truth) and base-model speakers (base-truth). We present each listener with samples across multiple base and adapted speakers and ask for a 5 point score from 'completely unnatural' to 'completely natural'. We print mean responses alongside 95% confidence bounds.

4. RESULTS

We have demonstrated that BOFFIN outperforms the base-line speaker adaptation system with respect to L1 loss. However, to investigate whether this lower score corresponds to an improvement in perceptual quality at inference time, we collected the perceptual evaluations of human listeners.

4.1. Experimental Protocol

To thoroughly test the performance of BOFFIN TTS, we consider three distinct corpora: (i) multi-speaker corpus with studio-quality recordings (referred to as INTERNAL[2]), (ii) the open-source VCTK corpus [24], and (iii) the LibriTTS audio-book corpus [25]. By considering a range of recording

[2]The internal corpus contains no customer voice recordings.

qualities and base-models with differing numbers of base speakers, we can understand the limitations of using BOFFIN TTS in a variety of practical settings. The architecture of our base-model remains fixed except for the more challenging LibriTTS task, where we double the size of our speaker embedding to accommodate a larger collection of base speakers. BO is performed with the Python library Emukit [3].

For each experiment, we adapt to 4 unseen speakers (from the same corpora used to train the base-model) using a random sample of 100 utterances (representing between 5 and 10 minutes of audio depending on the corpus), with 20% retained as a validation set. To evaluate each system, we compare naturalness and achieved similarity to the target speaker using a MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) test [26]. We also compare the naturalness achieved by BOFFIN TTS on target speakers with that achieved by the base multi-speaker model on its original speakers using a Mean Objective Score (MOS) test for naturalness. Each evaluation is presented to 25 native US listeners using Amazon Mechanical Turk. Statistical significance tests are performed at the p=0.01 level with Bonferroni-Holm corrections, using paired t and Mann-Whitney U tests for the MUSHRA and MOS evaluations respectively.

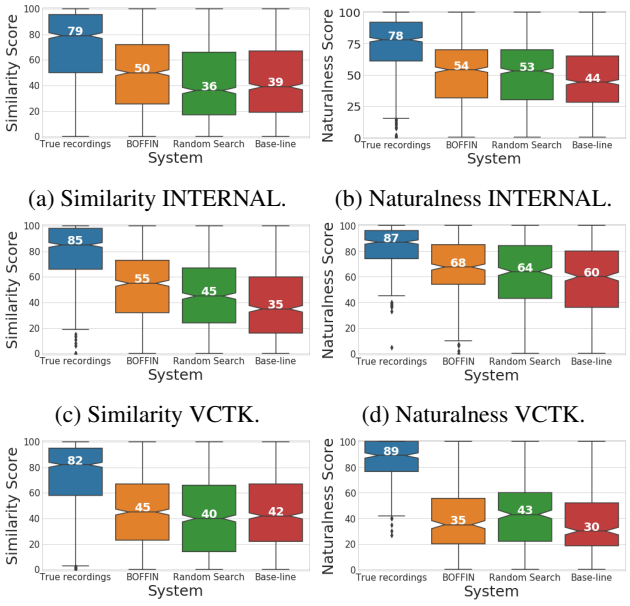
4.2. Adaptation from a base-model with few speakers

For our first experiment, we train a base-model on 4 male and 4 female proprietary speakers (each with 2.5k utterances) and adapt to 2 female and 2 child held-out speakers. Figure 3a show that BOFFIN TTS is able to achieve significant improvements in speaker similarity, with an improvement of 28% over the base-line and 39% over RS. Crucially, Figure 3b shows BOFFIN TTS' improvement in similarity does not sacrifice perceptual quality, achieving a small but statistically significant improvement in naturalness over the base-line speaker adaptation system. Moreover, Table 1 demonstrates that BOFFIN TTS is able to adapt to target speakers without a significant drop in perceptual quality from the base-model's speakers (learnt with 250 times more data).

4.3. Adaptation from a moderately rich base-model

We now consider a harder adaptation task; adapting to VCTK speakers with much higher variation in expressiveness, prosody and audio-quality than those in INTERNAL. Our base-model is trained on 22 speakers: 14 from VCTK (with 400 utterances each) supplemented with the 8 already considered in our first experiment (added to provide a more robust base-model). We adapt to 4 unseen VCTK speakers. This challenging adaptation scenario necessitates target speaker-specific adaptation strategies, with BOFFIN TTS providing significant improvements of 57% in similarity and 13% in naturalness over the base-line (Figures 3c and 3d). Moreover, Table 1 shows that BOFFIN TTS is once again able synthesize target speakers without a significant drop in naturalness than achieved for speakers already present in the base multi-speaker model.

[3] https://github.com/amzn/emukit



(a) Similarity INTERNAL. (b) Naturalness INTERNAL.

(c) Similarity VCTK. (d) Naturalness VCTK.

(e) Similarity LibriTTS. (f) Naturalness LibriTTS.

Fig. 3: MUSHRA tests for speaker similarity and naturalness. For similarity, we presented the same utterance synthesized by each system alongside a reference recording of the target speaker on another utterance and requested a rating of each system between 'definitely a different person' (0) and 'definitely the same person' (100). For naturalness, we repeat without a reference recording and instead asked for ratings between 'completely unnatural' and 'completely natural'

4.4. Adaptation from a rich base-model

To understand the limitations of BOFFIN TTS, our final experiment considers an even larger base-model containing 200 speakers (each with 200 utterances) from LibriTTS. We adapt to 4 additional unseen libriTTS speakers. LibriTTS is derived from audio-books and so contain noise, artifacts, and highly expressive voices. Consequently, although BOFFIN TTS was able to adapt to target speakers without a statistically significant drop in naturalness over the speakers used to train the base-system (Table 1) (as is consistent with our other experiments), our base-model itself was of much lower quality than our other base-models, making it difficult for our MUSHRA listeners to make a statistically significant preference in similarity across all three systems (Figures 3e and 3f).

5. CONCLUSION

We propose the few-shot speaker-adaptation framework of BOFFIN TTS. By learning adaptation strategies custom to each target speaker, BOFFIN TTS can achieve higher speaker similarity than using a *one-size-fits-all* adaptation strategy, particularly when adapting to challenging target speakers from high-performance multi-speaker models.

6. REFERENCES

[1] Javier Latorre, Jakub Lachowicz, Jaime Lorenzo-Trueba, Thomas Merritt, Thomas Drugman, Srikanth Ronanki, and Viacheslav Klimkov, "Effect of data reduction on sequence-to-sequence neural tts," in *ICASSP*, 2019.

[2] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop*, 2016.

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

[4] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *ICLR*, 2018.

[5] Yu-An Chung, Yuxuan Wang, Wei-Ning Hsu, Yu Zhang, and RJ Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *ICASSP*, 2019.

[6] Junichi Yamagishi, Takao Kobayashi, Yuji Nakano, Katsumi Ogata, and Juri Isogai, "Analysis of speaker adaptation algorithms for hmm-based speech synthesis and a constrained smaplr adaptation algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, 2009.

[7] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv:1705.02304*, 2017.

[8] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," *arXiv:1707.06588*, 2017.

[9] Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf, "Fitting new speakers based on a short untranscribed sample," *arXiv:1802.06984*, 2018.

[10] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *NeurIPS*, 2018.

[11] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou, "Neural voice cloning with a few samples," in *NeurIPS*, 2018, pp. 10019–10029.

[12] Yutian Chen, Yannis Assael, Brendan Shillingford, David Budden, Scott Reed, Heiga Zen, Quan Wang, Luis C Cobo, Andrew Trask, Ben Laurie, et al., "Sample efficient adaptive text-to-speech," *arXiv preprint arXiv:1809.10460*, 2018.

[13] Jasper Snoek, Hugo Larochelle, and Ryan P Adams, "Practical bayesian optimization of machine learning algorithms," in *NeurIPS*, 2012.

[14] James Bergstra, Daniel Yamins, and David Daniel Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," *JMLR*, 2013.

[15] Lidan Wang, Minwei Feng, Bowen Zhou, Bing Xiang, and Sridhar Mahadevan, "Efficient hyper-parameter optimization for nlp applications," in *EMNLP*, 2015.

[16] Yutian Chen, Aja Huang, Ziyu Wang, Ioannis Antonoglou, Julian Schrittwieser, David Silver, and Nando de Freitas, "Bayesian optimization in alphago," *arXiv:1812.06855*, 2018.

[17] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *Interspeech*, 2017.

[18] Nishant Prateek, Mateusz Lajszczak, Roberto Barra-Chicote, Thomas Drugman, Jaime Lorenzo-Trueba, Thomas Merritt, Srikanth Ronanki, and Trevor Wood, "In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data," *NAACL HLT 2019*, 2019.

[19] Jaime Lorenzo-Trueba, Thomas Drugman, Javier Latorre, Thomas Merritt, Bartosz Putrycz, and Roberto Barra-Chicote, "Robust universal neural vocoding," *arXiv:1811.06292*, 2018.

[20] David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Aaron Courville, and Chris Pal, "Zoneout: Regularizing rnns by randomly preserving hidden activations," *arXiv:1606.01305*, 2016.

[21] James Bergstra and Yoshua Bengio, "Random search for hyper-parameter optimization," *JMLR*, 2012.

[22] Carl Edward Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, 2003.

[23] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, Jan 2016.

[24] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[25] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv:1904.02882*, 2019.

[26] ITUR Recommendation, "Method for the subjective assessment of intermediate sound quality (mushra)," *ITU, BS*, 2001.