Title

Expediting TTS Synthesis with Adversarial Vocoding

Authors

Neekhara, Paarth
Donahue, Chris
Puckette, Miller
et al.

Publication Date

2019

DOI

10.21437/interspeech.2019-3099

Peer reviewed

Expediting TTS Synthesis with Adversarial Vocoding

*Paarth Neekhara[1], *Chris Donahue[2], Miller Puckette[2], Shlomo Dubnov[2], Julian McAuley[1]

[1]UC San Diego Department of Computer Science
[2]UC San Diego Department of Music
* Equal contribution

pneekhar@eng.ucsd.edu, cdonahue@ucsd.edu

Abstract

Recent approaches in text-to-speech (TTS) synthesis employ neural network strategies to vocode perceptually-informed spectrogram representations directly into listenable waveforms. Such vocoding procedures create a computational bottleneck in modern TTS pipelines. We propose an alternative approach which utilizes generative adversarial networks (GANs) to learn mappings from perceptually-informed spectrograms to simple magnitude spectrograms which can be heuristically vocoded. Through a user study, we show that our approach significantly outperforms naïve vocoding strategies while being hundreds of times faster than neural network vocoders used in state-of-the-art TTS systems. We also show that our method can be used to achieve state-of-the-art results in unsupervised synthesis of individual words of speech.

1. Introduction

Generating natural-sounding speech from text is a well-studied problem with numerous potential applications. While past approaches were built on extensive engineering knowledge in the areas of linguistics and speech processing (see [1] for a review), recent approaches adopt neural network strategies which learn from data to map linguistic representations into audio waveforms [2, 3, 4, 5, 6]. Of these recent systems, the best performing [4, 6] are both comprised of two functional mechanisms which (1) map language into *perceptually-informed spectrogram* representations (i.e., time-frequency decompositions of audio with logarithmic scaling of both frequency and amplitude), and (2) *vocode* the resultant spectrograms into listenable waveforms. In such two-step TTS systems, using perceptually-informed spectrograms as intermediaries is observed to have empirical benefits over using representations which are simpler to convert to audio [4]. Hence, vocoding is central to the success of state-of-the-art TTS systems, and is the focus of this work.

The need for vocoding arises from the non-invertibility of perceptually-informed spectrograms. These compact representations exclude much of the information in an audio waveform, and thus require a predictive model to fill in the missing information needed to synthesize natural-sounding audio. Notably, standard spectrogram representations discard phase information resulting from the short-time Fourier transform (STFT), and additionally compress the linearly-scaled frequency axis of the STFT magnitude spectrogram into a logarithmically-scaled one. This gives rise to two corresponding vocoding subproblems: the well-known problem of *phase estimation*, and the less-investigated problem of *magnitude estimation*.

Vocoding methodology in state-of-the-art TTS systems [4, 6] endeavors to address the joint of these two subproblems, i.e., to transform perceptually-informed spectrograms directly into waveforms. Specifically, both systems use WaveNet [7] conditioned on spectrograms. This approach is problematic as it necessitates running WaveNet once per individual audio sample (e.g. 22050 times per second), bottlenecking the overall TTS system as the language-to-spectrogram mechanisms are comparatively fast.[1] Given that joint solutions currently necessitate such computational overhead, it may be methodologically advantageous to combine solutions to the individual subproblems.

Before endeavoring to develop individual solutions to magnitude and phase estimation, we first wished to discover which (if any) of the two represented a greater obstacle to vocoding. To answer this, we conducted a user study examining the effect that common heuristics for each subproblem have on the perceived naturalness of vocoded speech (Table 1).[2] Our study demonstrated that combining an ideal solution to *either* magnitude or phase estimation with a heuristic for the other results in high-quality speech. Hence, we can focus our research efforts on *either* subproblem, in the hopes of developing methods which are more computationally efficient than existing end-to-end strategies.

In this paper, we seek to address the magnitude estimation subproblem, which has received less attention in comparison to phase estimation [8, 9, 10, 11]. We propose a learning-based method which uses Generative Adversarial Networks [12] to learn a stochastic mapping from perceptually-informed spectrograms into simple magnitude spectrograms. We combine this magnitude estimation method with a modern phase estimation heuristic, referring to this method as *adversarial vocoding*. We show that adversarial vocoding can be used to expedite TTS synthesis and additionally improves upon the state of the art in unsupervised generation of individual words of speech.

1.1. Summary of contributions

- For both real spectrograms and synthetic ones from TTS systems, we demonstrate that our proposed vocoding method yields significantly higher mean opinion scores than a heuristic baseline and faster speeds than state-of-the-art vocoding methods.
- We show that our method can effectively vocode highly-compressed (13:1) audio feature representations.
- We show that our method improves the state of the art in unsupervised synthesis of individual words of speech.
- We measure the perceived effect of inverting the primary sources of compression in audio features. We observe that coupling solutions to either compression source with a heuristic for the other result in high-quality speech.

[1]In our empirical experimentation with open-source codebases, the autoregressive vocoding phase was over 1500 times slower on average than the language to spectrogram phase.

[2]Sound examples: chrisdonahue.com/advoc_examples

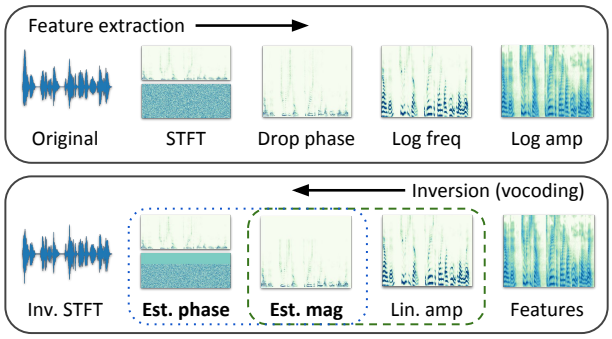

Figure 1: *Depiction of stages in common audio feature extraction pipelines and corresponding inversion. The two obstacles to vocoding are (1) estimating linear-frequency magnitude spectra from log-frequency mel spectra (outlined in green dashed line), and (2) estimating phase information from magnitude spectra (outlined in blue dotted line). We focus on magnitude estimation in this paper, observing that coupling an ideal solution to this subproblem with a phase estimation heuristic can produce high-quality speech (Table 1).*

2. Audio feature preliminaries

The typical process of transforming waveforms into perceptually-informed spectrograms involves several cascading stages. Here, we describe spectrogram methodology common to two state-of-the-art TTS systems [4, 6]. A visual representation is shown in Figure 1.

Extraction The initial stage consists of decomposing waveforms into time and frequency using the STFT. Then, the phase information is discarded from the complex STFT coefficients leaving only the linear-amplitude magnitude spectrogram. The linearly-spaced frequency bins of the resultant spectrogram are then compressed to fewer bins which are equally-spaced on a logarithmic scale (usually the mel scale [13]). Finally, amplitudes of the resultant spectrogram are made logarithmic to conform to human loudness perception, then optionally clipped and normalized.

Inversion To heuristically invert this procedure (vocode), the inverse of each cascading step is applied in reverse. First, logarithmic amplitudes are converted to linear ones. Then, an appropriate magnitude spectrogram is estimated from the mel spectrogram. Finally, appropriate phase information is estimated from the magnitude spectrogram, and the inverse STFT is used to render audio.

Unless otherwise specified, throughout this paper we operate on waveforms sampled at 22050Hz using an STFT with a window size of 1024 and a hop size of 256. We compress magnitude spectrograms to 80 bins ($melBins = 80$) equally spaced along the mel scale from 125Hz to 7600Hz. We apply log amplitude scaling and normalize resultant mel spectrograms to have 120dB dynamic range. Precisely recreating this representation [14] is simple in our codebase.[3]

3. Measuring the effect of magnitude and phase estimation on speech naturalness

The audio feature extraction pipelines outlined in Section 2 have two sources of compression: the discarding of phase information and compression of magnitude information. Conventional wisdom suggests that the primary obstacle to inverting such features is phase estimation. However, to the best of our knowledge, a systematic evaluation of the individual contributions of magnitude and phase estimation on perceived naturalness of vocoded speech has never been reported.

[3]Code: `github.com/paarthneekhara/advoc`

Table 1: *Ablating the effect of heuristics for magnitude and phase estimation on mean opinion score (MOS) of speech naturalness with 95% confidence intervals.* ***Bolded*** *entries show that coupling an ideal solution to either subproblem (real data used as a proxy) with a good heuristic for the other yields speech with only 2–9% lower MOS than real speech ($p < 0.05$).*

Magnitude est. method	Phase est. method	MOS
Ideal (real magnitudes)	*Ideal* (real phases)	4.30 ± 0.06
Ideal (real magnitudes)	Griffin-Lim w/ 60 iters	3.70 ± 0.07
Ideal (real magnitudes)	Local Weighted Sums	$\mathbf{4.09 \pm 0.06}$
Mel pseudoinverse	*Ideal* (real phases)	$\mathbf{4.04 \pm 0.06}$
Mel pseudoinverse	Griffin-Lim w/ 60 iters	2.48 ± 0.09
Mel pseudoinverse	Local Weighted Sums	2.51 ± 0.09

To perform such an evaluation, we mix and match methods for estimating both STFT magnitudes and phases from log-amplitude mel spectrograms. A common heuristic for magnitude estimation is to project the mel-scale spectrogram onto the pseudoinverse of the mel basis which was originally used to generate it. As a phase estimation baseline, state-of-the-art TTS research [4, 6] compares to the iterative Griffin-Lim [8] strategy with 60 iterations. We additionally consider the more-recent Local Weighted Sums (LWS) [9] strategy which, on our CPU, is about six times faster than 60 iterations of Griffin-Lim. As a proxy for an ideal solution to either subproblem, we also use magnitude and phase information extracted from real data.

We show human judges the same waveform vocoded by six different magnitude and phase estimation combinations (inducing a comparison) and ask them to rate the naturalness of each on a subjective 1 to 5 scale (full user study methodology outlined in Section 5.1). Mean opinion scores are shown in Table 1, and we encourage readers to listen to our sound examples linked from the footnote on the first page to help contextualize.

From these results, we conclude that an ideal solution to *either* magnitude or phase estimation can be coupled with a good heuristic for the other to produce high-quality speech. While the ground truth speech is still significantly more natural than that of ideal+heuristic strategies, the MOS for these methods are only 2-9% worse than the ground truth ($p < 0.05$). Of these two problems, we focus on building magnitude estimation strategies as the conventional heuristic (pseudoinverse) is comparatively primitive to heuristics used for phase estimation.

As a secondary conclusion, we observe that—for our speech data—using LWS for phase estimation from real spectrograms yields significantly higher MOS than using Griffin-Lim. Given that it is faster *and* yields significantly more natural speech, we recommend that all TTS research use LWS as a phase estimation baseline instead of Griffin-Lim. Henceforth, all of our experiments that require phase estimation use LWS.

4. Adversarial vocoding

Our goal is to invert a mel spectrogram feature representation into a time domain waveform representation. In the previous section, we demonstrated the potential of the magnitude estimation subproblem for achieving this goal in combination with the LWS phase estimation heuristic. A common heuristic for

magnitude estimation is performed by multiplying the mel spectrogram with the approximate inverse of the mel transformation matrix. Since the mel spectrogram is a lossy compression of the magnitude spectrogram, a simple linear transformation is an oversimplification of the magnitude estimation problem.

In order to improve on heuristic magnitude estimation, we formulate it as a generative modeling problem and propose a Generative Adversarial Network (GAN) [12] based solution.[4] GANs are generative models which seek to learn latent structure in the distribution of data. They do this by mapping samples $\boldsymbol{z}$ from a prior distribution p_Z to samples $\boldsymbol{y}$, $G : \boldsymbol{z} \rightarrow \boldsymbol{y}$. For our purpose, we use a variation of GAN called *conditional* GAN [17] to model the conditional probability distribution of magnitude spectrograms given a mel spectrogram. The pix2pix method [18] demonstrates that this conditioning information can be a structurally-rich image, extending GANs to learn stochastic mappings from one image domain (spectrogram domain in our case) to another. We adapt it for our task.

The conditional GAN objective to generate appropriate magnitude spectrograms $\boldsymbol{y}$ given mel spectrograms $\boldsymbol{x}$ is:

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) =& \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x}, \boldsymbol{y})] + \\ & \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x}, \boldsymbol{z}))],\end{aligned} \quad (1)$$

where the generator G tries to minimize this objective against an adversary D that tries to maximize it. i.e $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D)$. In such a conditional GAN setting, the generator tries to "fool" the discriminator by generating *realistic* magnitude spectrograms that correspond to the conditioning mel spectrogram. Previous works [18, 19] have shown that it is beneficial to add a secondary component to the generator loss in order to minimize the L_1 distance between the generated output $G(\boldsymbol{x}, \boldsymbol{z})$ and the target $\boldsymbol{y}$. This way, the adversarial component encourages the generator to generate more realistic results, while the L_1 objective ensures the generated output is close to the target.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[||\boldsymbol{y} - G(\boldsymbol{x}, \boldsymbol{z})||_1]. \quad (2)$$

Our final objective therefore becomes:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G). \quad (3)$$

Here, λ is a hyperparameter which determines the trade-off between the L_1 loss and adversarial loss.

4.1. Network architecture

Figure 2 shows our setup for adversarial inversion of the mel spectrogram into a magnitude spectrogram.

Generator The generator network G takes as input the linear-amplitude mel spectrogram representation x of shape $(n, melBins)$ and generates a magnitude spectrogram of shape $(n, 513)$; $n = 256$ (nearly 3 seconds) in all of our experiments. The generator first estimates the magnitude spectrorgram through a fixed (non trainable) linear projection of the mel spectrogram using the approximate inverse of the mel transformation matrix. The estimated magnitude spectrogram goes through a convolution based encoder-decoder architecture with skip connections as in pix2pix [18]. Past works [20, 18] have noted that generators similar to our own empirically learn to ignore latent codes leading to deterministic models. We adopt the same policy of using dropout at both training and test time to force the model to be stochastic (as our task is not a one-to-one mapping). Additionally, we also train a smaller generator *(Advoc - small)* with fewer convolutional layers and fewer convolutional channels. We omit the specifics of our architecture for brevity, however we point to our codebase (link in footnote of previous page) for precise model implementations.

[4] GANs have been previously used for phase estimation [11] and to enhance speech both before [15] and after [16] vocoding.

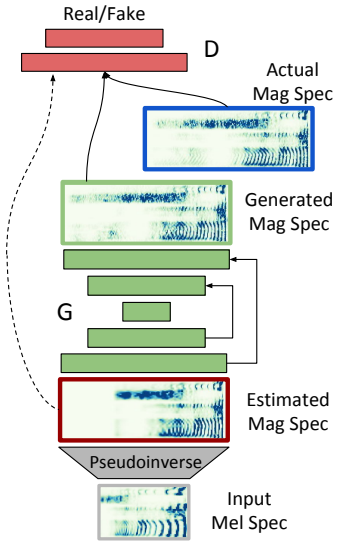

Figure 2: *Adversarial Vocoder Model: The generator performs an image-to-image translation from the estimated magnitude spectrogram to the actual magnitude spectrogram guided by an adversarial loss from the discriminator and the L_1 distance between the generated and actual magnitude spectrogram*

Discriminator Previous works have found that training generators similar to our own using just an L_1 or L_2 loss produces images with reasonable global structure (spatial relationships preserved) but poor local structure (blurry) [21, 22]. As in [18], we combine an L_1 loss with a discriminator which operates on *patches* (subregions) of a spectrogram to help improve the "sharpness" of the output. Our discriminator takes as input the estimated spectrogram and *either* the generated or real magnitude spectrogram. Thus, in order to satisfy the discriminator, the generator needs to produce magnitude spectrograms that both correspond to the mel spectrogram *and* look realistic.

To complete our adversarial vocoding pipeline, we combine generated magnitude spectrograms with LWS-estimated phase spectrograms and use the inverse STFT to synthesize audio.

5. Experiments

We focus our primary empirical study on the publicly available LJ Speech dataset [23], which is popularly used in TTS research [24, 25]. The dataset contains 13k short audio clips (24 hours) of a single speaker reading from non-fiction books.

Audio is processed using the feature extraction process described in Section 2. We train three models for $melBins \in \{20, 40, 80\}$ to study the feasibility of our technique for varying levels of mel compression. Each of the models is trained for 100,000 mini-batch iterations using a batch size of 8 which corresponds to 12 hours of wall clock training time using a NVIDIA 1080Ti GPU. We set the regularization parameter $\lambda = 10$ and use the Adam optimizer [26] ($\alpha = 0.0002$).

Table 2: *Comparison of vocoding methods on mel spectrograms with* 80 *bins. We display comparative mean opinion scores from two separate user studies for vocoding spectrograms extracted from real speech (MOS-Real) and spectrograms generated by a state-of-the-art TTS method (MOS-TTS) with* 95% *confidence intervals.* × *RT denotes the speed up over real time; higher is faster. MB denotes the size of each model in megabytes.*

Source	MOS-Real	MOS-TTS	× RT	MB
Real data	4.16 ± 0.06	4.28 ± 0.07	1.000	
Pseudoinverse	2.91 ± 0.10	2.12 ± 0.09	8.836	0.2
WaveNet [7]	3.98 ± 0.07	3.87 ± 0.07	0.003	95.0
WaveGlow [24]	4.09 ± 0.06	3.89 ± 0.07	1.229	334.7
AdVoc	3.78 ± 0.07	2.91 ± 0.08	3.111	207.7
AdVoc-small	3.68 ± 0.07	3.09 ± 0.07	3.437	16.0

5.1. Vocoding LJ Speech mel spectrograms

In this study we are concerned with vocoding both real mel spectrograms extracted from the LJ Speech dataset *and* mel spectrograms generated by a language-to-spectrogram model [6] trained on LJ Speech. We compare both our large (*AdVoc*) and small (*AdVoc-small*) adversarial vocoder models to the mel pseudoinverse magnitude estimation heuristic combined with LWS (*Pseudoinverse*), a *WaveNet* vocoder [6], and the recent *WaveGlow* [24] method. We cannot directly compare to the *Parallel WaveNet* approach because it is an end-to-end TTS method rather than a vocoder [27].

We randomly select 100 examples from the holdout dataset of LJ Speech and convert them to mel spectrograms. We also synthesize mel spectrograms for each transcript of these same examples using the language-to-spectrogram module from Tacotron 2 [6]. We vocode both the real and synthetic spectrograms to audio using the five methods outlined in the previous paragraph. Audio from each method can be found in our sound examples (footnote of first page).

To gauge the relative quality of our methods against others, we conduct two mean opinion score (comparative) studies with human judges on Amazon Mechanical Turk. In the first user study, judges evaluate a batch of six versions of the same utterance: the original utterance and the spectrogram of that utterance vocoded by the five aforementioned methods. In the second user study, we show each judge a batch consisting of the real utterance and five vocodings of a synthetic spectrogram with the same transcript. In all user studies, the ordering of the waveforms is randomized in each batch but the waveforms in a batch always pertain to the same utterance. Judges are asked to rate the naturalness of each on a subjective 1–5 scale with 1 point increments. Each batch is reviewed by 8 different reviewers resulting in 800 evaluations of each strategy. We display mean opinion scores in Table 1. We also include the speed of each method (relative to real time) as measured on GPU, and the sizes of each model's parameters in megabytes.

Our results demonstrate that—for both real *and* synthetic spectrograms—our adversarial magnitude estimation technique (AdVoc) significantly outperforms magnitude estimation using the pseudoinverse of the mel basis. Our method is more than $1000\times$ faster than the autoregressive WaveNet vocoder and $2.5\times$ faster than WaveGlow vocoder.

Additionally, we train our models to perform magnitude estimation on representations with higher compression. Specifically, we train our model to vocode mel spectrograms with 20, 40 and 80 bins. We compare our adversarial magnitude estimation method against magnitude estimation using the pseudoinverse of the mel basis. We conduct a comparative user study using the same methodology as previously outlined. Our results in Table 3 demonstrate that our model can vocode highly compressed mel spectrogram representations with relatively little drop in the perceived audio quality as compared to the pseudoinversion baseline (audio examples in footnote of first page).

Table 3: *Comparison of heuristic and adversarial vocoding of spectrograms with different levels of mel compression. Adversarial vocoding can vocode highly compressed mel spectrograms with relatively less drop in speech naturalness as compared to a heuristic.*

Source	$melBins$	MOS
Real data		4.05 ± 0.07
Pseudoinverse	20	2.68 ± 0.10
Pseudoinverse	40	2.84 ± 0.10
Pseudoinverse	80	3.25 ± 0.09
AdVoc	20	3.75 ± 0.07
AdVoc	40	3.79 ± 0.07
AdVoc	80	3.86 ± 0.07

Table 4: *Combining our adversarial vocoding approach with GAN-generated mel spectrograms outperforms our prior work in unsupervised generation of individual words by all metrics.*

	Quantitative	*Qualitative*	
Source	Inception score	Acc.	MOS
Real data	8.01 ± 0.24	0.95	3.9 ± 0.15
WaveGAN [28]	4.67 ± 0.01	0.58	2.3 ± 0.18
SpecGAN [28] + Griffin-Lim	6.03 ± 0.04	0.66	1.9 ± 0.17
MelSpecGAN + AdVoc	6.63 ± 0.03	0.71	3.4 ± 0.20

5.2. Unsupervised audio synthesis

In this section we are concerned with the *unsupervised* generation of speech (as opposed to supervised generation in the case of TTS). We focus on the SC09 digit generation task proposed in our previous work [28], where the goal is to learn to generate examples of spoken digits "zero" through "nine" *without* labels. We first train a GAN to generate mel spectrograms of spoken digits (*MelSpecGAN*), then train an adversarial vocoder to generate audio conditioned on those spectrograms. Using a pretrained digit classifier, we calculate an Inception score [29] for our approach, finding it to outperform our previous state-of-the-art results by 9%. We also calculate an "accuracy" by comparing human labelings to classifier labels for our generated digits, finding that our adversarial vocoding-based method outperforms our previous results (Table 4).

6. Conclusion

In this work we have shown that solutions to *either* the magnitude or phase estimation subproblems within common vocoding pipelines can result in high-quality speech. We have demonstrated a learning-based method for magnitude estimation which significantly improves upon popular heuristics for this task. We demonstrate that our method can integrate with an existing TTS pipeline to provide comparatively fast waveform synthesis. Additionally, our method has advanced the state of the art in unsupervised small-vocabulary speech generation.

7. Acknowledgements

The authors would like to thank Bo Li for helpful discussions about this work. This research was supported by the UC San Diego Chancellors Research Excellence Scholarship program. Thanks to NVIDIA for GPU donations which were used in the preparation of this work.

8. References

[1] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, "Deep Voice: Real-time neural text-to-speech," in Proc. *ICML*. JMLR. org, 2017, pp. 195–204.

[3] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-speaker neural text-to-speech," in Proc. *NIPS*, 2017, pp. 2962–2970.

[4] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in Proc. *ICLR*, 2018.

[5] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," in Proc. *INTERSPEECH*, 2017.

[6] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in Proc. *ICASSP*, 2018.

[7] 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," *arXiv:1609.03499*, 2016.

[8] D. W. Griffin, Jae, S. Lim, and S. Member, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoustics, Speech and Sig. Proc*, pp. 236–243, 1984.

[9] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in Proc. *International Conference on Digital Audio Effects*, 2010, pp. 397–403.

[10] K. Li, B. Wu, and C.-H. Lee, "An iterative phase recovery framework with phase mask for spectral mapping with an application to speech enhancement." in Proc. *INTERSPEECH*, 2016.

[11] K. Oyamada, H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, and H. Ando, "Generative adversarial network-based approach to signal reconstruction from magnitude spectrogram," in Proc. *EUSIPCO*, 2018.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in Proc. *NIPS*, 2014, pp. 2672–2680.

[13] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[14] B. McFee, J. W. Kim, M. Cartwright, J. Salamon, R. M. Bittner, and J. P. Bello, "Open-source practices for music signal processing research: Recommendations for transparent, sustainable, and reproducible audio research," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 128–137, 2019.

[15] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for STFT spectrograms," in Proc. *INTERSPEECH*, 2017.

[16] K. Tanaka, T. Kaneko, N. Hojo, and H. Kameoka, "Synthetic-to-natural speech waveform conversion using cycle-consistent adversarial networks," in *IEEE Spoken Language Technology Workshop*, 2018.

[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2014.

[18] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proc. *CVPR*, 2017, pp. 5967–5976.

[19] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," in Proc. *INTERSPEECH*, 2017.

[20] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv:1511.05440*, 2015.

[21] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," in Proc. *CVPR*, 2016.

[22] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in Proc. *ECCV*, 2016.

[23] K. Ito, "The LJ Speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[24] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in Proc. *ICASSP*, 2018.

[25] R. Yamamoto., "WaveNet vocoder," https://github.com/r9y9/wavenet_vocoder, 2018.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. *ICLR*, 2015.

[27] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in Proc. *NIPS*, 2017.

[28] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," in Proc. *ICLR*, 2019.

[29] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in Proc. *NIPS*, 2016.