VOCBENCH: A NEURAL VOCODER BENCHMARK FOR SPEECH SYNTHESIS

Ehab A. AlBadawy[*§], *Andrew Gibiansky*[†], *Qing He*[†], *Jilong Wu*[†], *Ming-Ching Chang*[§], *Siwei Lyu*[‡]

[†]Facebook AI, USA
[§] University at Albany, State University of New York, USA
[‡]University at Buffalo, State University of New York, USA

ABSTRACT

Neural vocoders, used for converting the spectral representations of an audio signal to the waveforms, are a commonly used component in speech synthesis pipelines. It focuses on synthesizing waveforms from low-dimensional representation, such as Mel-Spectrograms. In recent years, different approaches have been introduced to develop such vocoders. However, it becomes more challenging to assess these new vocoders and compare their performance to previous ones. To address this problem, we present VocBench, a framework that benchmark the performance of state-of-the-art neural vocoders. VocBench uses a systematic study to evaluate different neural vocoders in a shared environment that enables a fair comparison between them. In our experiments, we use the same setup for datasets, training pipeline, and evaluation metrics for all neural vocoders. We perform a subjective and objective evaluation to compare the performance of each vocoder along a different axis. Our results demonstrate that the framework can show competitive efficacy and quality of the synthesized samples for each vocoder. VocBench framework is available at https://github.com/facebookresearch/vocoder-benchmark.

Index Terms: speech synthesis, vocoders, Mel-Spectrograms, GAN, VocBench, benchmark, evaluation.

1. INTRODUCTION

Throughout the years, speech synthesis techniques have gone through different phases of improvements, from knowledge-based approaches [1, 2, 3] to data-based ones [4, 5, 6]. To date, there are two types of speech synthesis algorithms, *text to speech*, which converts input text to audio signals, and *voice conversion*, which transforms an input audio to different identities or styles. Regardless of this difference, most of the recent speech synthesis approaches [7, 8, 9] rely on *neural vocoders* to generate the final waveform for more natural-sounding speech synthesis.

In this context, a vocoder is designed to synthesize waveform from the lower feature dimension, such as Mel-spectrograms. For many years, the state-of-the-art (SOTA) methods used DSP-based approaches [10] for vocoder development. While the advantage of fast speech generation time, the quality of the synthesized waveform is largely limited due to the assumptions under the heuristics. In recent years, more sophisticated vocoders have been developed based on the use of deep neural networks for more enhanced quality for the generated speech. These methods include (1) *autoregressive* approaches [11, 12], (2) *Generative Adversarial Networks (GANs)* approaches [13, 14, 15], and (3) *diffusion based* approaches [16, 17]. Due to the different variables in the evaluation process, datasets selection, hardware configuration, and evaluation metrics used, how best to compare and evaluate these different approaches remains an open challenge.

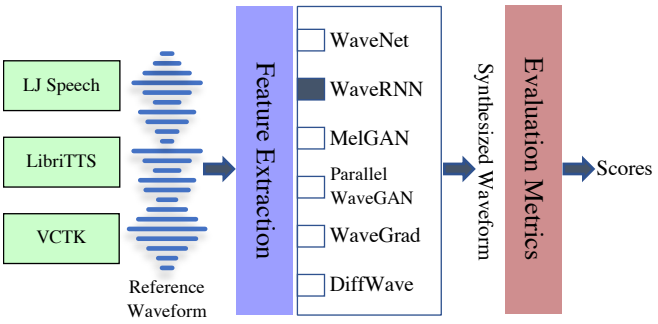

Fig. 1. An overview of the proposed VocBench framework.

In this work, we present the VocBench framework, a comprehensive benchmark for vocoder quality and speed evaluations. More specifically, we build VocBench to train and test neural vocoders in a shared environment with public datasets. We construct three datasets, including one single-speaker and two multi-speaker scenarios, and then train six vocoders covering three different categories: *autoregressive*, *GAN*, and *diffusion* based approaches. All vocoders are trained and evaluated following the same pipeline. We design two main experiments. First, we test the efficacy of each vocoder in synthesizing the waveform from lower-dimensional features such as Mel-Spectrogram. Second, we test the generalizability of each vocoder in synthesizing speech for speakers who are not included in the training set. Figure 1 provides an overview of the proposed framework.

Recently, various studies have been conducted for neu-

[*]Work done during an internship at Facebook AI.

ral vocoder evaluation. Govalkar *et al.* [18] conducted a study with six autoregressive-based vocoders and two additional phase reconstructions vocoders. Airaksinen *et al.* [19] adopted classical methods for vocoder design in their study. Both of these works used MUSHRA [20] as their main evaluation metric to compare the performance of each vocoder. In this study, we extend the vocoder implementations to include both GAN and diffusion-based models. Additionally, we carry on the evaluation using both subjective and objective metrics. We use the Mean Opinion Score (MOS) test as a subjective evaluation. To evaluate each of the different vocoders objectively, we used the following four different evaluation metrics: Structural Similarity Index Measure (SSIM) [21], Fréchet Audio Distance (FAD) [22], Log-mel Spectrogram Mean Squared Error (LS-MSE), and Peak Signal-to-Noise Ratio (PSNR). More details about the experiment setup and evaluation metrics are presented in § 3.

2. NEURAL VOCODERS

We next describe the three main categories of the neural vocoders used in our study: the autoregressive models (§ 2.1), GAN based models (§ 2.2), and diffusion models (§ 2.3).

2.1. Autoregressive Models

The key feature of the autoregressive models is that they are designed as probabilistic models to predict the probability of each waveform sample based on the previous samples. This allows generating a natural, high-quality speech signal. However, due to the sample-by-sample generation process, the overall synthesis speed is slow compared to other methods. In the following, we will consider two main autoregressive models: WaveNet and WaveRNN.

The **WaveNet** [11] model works on the waveform level to achieve long-range temporal dependency through the depth of the model. It combines a stack of causal filters and dilated convolutions to help their receptive fields grow exponentially with the depth. We use the open-source implementation from [23] with different configurations of input types and loss functions. More details are provided in § 3. The autoregressive **WaveRNN** [12] architecture utilizes a recurrent neural network (RNN) for sequential modeling of the target waveform. A single layer RNN with a dual softmax layer is used.

2.2. GAN Based Models

GAN-based vocoders have shown remarkable performance often exceeding autoregressive models in the speed and quality of the synthesized speech. The main idea of GANs [24] use a *generator* to model the waveform signal in the time domain and a *discriminator* to assess the quality of the generated speech. We consider two representative models, MelGAN and Parallel WaveGAN among the different variants of GAN-based vocoders.

MelGAN [13] takes the standard GAN architecture for fast waveform generation. A fully convolutional model is used for high-quality Mel-Spectrogram inversion. With fewer parameters compared to the autoregressive model, MelGAN achieves higher real-time factor on both GPU and CPU without the need of hardware-specific optimization.

The **Parallel WaveGAN** [15] architecture is distillation-free, fast, and requires only small memory footprint for waveform synthesis. Parallel WaveGAN jointly optimizes the waveform-domain adversarial loss and multi-resolution short-time Fourier transform (STFT) loss.

2.3. Diffusion Based Models

Diffusion probabilistic models are generative models entailing two main processes: *diffusion* and *reverse* [25]. The diffusion process is defined as a Markov chain that gradually adds Gaussian noise to the original signal until it gets destroyed. The reverse process, on the other hand, is a denoising process that progressively removes the added Gaussian noise and restores the original signal. We included two diffusion-based vocoders in our study: WaveGrad and DiffWave.

The **WaveGrad** [16] model architecture is built on prior works from score matching [26] and diffusion probabilistic models [25]. The WaveGrad model takes a white Gaussian noise as input, and condition on the Mel-Spectrogram to iteratively refine the signal via a gradient-based sampler.

DiffWave [17] is a versatile diffusion probabilistic model for waveform synthesis that works well under both conditional and unconditional scenarios. Using a white Gaussian noise as input, DiffWave performs a Markov chain process with a constant number of steps to gradually generate a structured waveform [27, 28, 25]. The model is trained to optimize a choice of variational bound on the data likelihood.

3. DATASET AND EXPERIMENTS

3.1. Dataset and Feature Extraction

We use three datasets in this study: LJ Speech for the *single-speaker* scenario as well as LibriTTS and VCTK for the *multi-speaker* scenarios. For all of the three different datasets, the train, validation, and test splits are fixed across the different vocoders that are used in our study.

The **LJ Speech** dataset [29] consists of $13,100$ short audio clips of a single speaker reading passages from 7 non-fiction books. A transcription is provided for each clip. The length of each clip varies from 1 to 10 seconds, and the total length is approximately 24 hours. We reserve the first 20 clips for testing, and the following 10 clips for validation. The rest of the clips are used for training.

The **LibriTTS** dataset [30] is a multi-speaker English corpus of approximately 585 hours of reading English speech at a 24kHz sampling rate. It is derived from the original materials (MP3 audio files from LibriVox and text files from Project Gutenberg) of the LibriSpeech corpus. We use *train-clean*-100 and *train-clean*-360 subsets for training with about 1150 speakers and 25 minutes of recordings on average per speaker. For validation and test splits, we use the *dev-clean* and *test-clean* subsets respectively.

Table 1. Evaluation results for the four objective metrics (SSIM, LS-MSE, PSNR, and FAD) and the 5-scale MOS with 95% confidence intervals evaluated on the three datasets: LJ Speech, LibriTTS, and VCTK. We welcome researchers to submit or update their results at our GitHub repository https://github.com/facebookresearch/vocoder-benchmark for comparisons.

Metric	Corpus	WaveNet	WaveRNN	MelGAN	Parallel WaveGAN	WaveGrad	DiffWave	Griffin-Lim	Ground Truth
SSIM	LJ Speech	0.66	0.62	0.89	0.84	0.76	0.82	0.90	-
	LibriTTS	0.056	0.53	0.91	0.86	0.71	0.74	0.89	-
	VCTK	0.46	0.43	0.88	0.79	0.59	0.64	0.86	-
LS-MSE	LJ Speech	0.006	0.010	0.001	0.002	0.006	0.006	0.001	-
	LibriTTS	0.008	0.008	0.001	0.001	0.005	0.006	0.001	-
	VCTK	0.009	0.010	0.001	0.002	0.007	0.007	0.001	-
PSNR	LJ Speech	23.20	20.36	28.53	26.70	22.57	22.51	28.77	-
	LibriTTS	21.54	21.17	29.98	28.62	22.94	22.18	29.03	-
	VCTK	21.36	20.40	30.40	28.17	21.54	21.22	28.77	-
FAD	LJ Speech	1.05	3.43	1.51	**0.92**	3.12	3.62	2.69	0.31
	LibriTTS	1.55	2.60	2.95	**1.41**	3.10	3.74	4.27	1.23
	VCTK	**0.99**	3.59	1.76	1.22	4.10	5.59	3.92	0.61
MOS	LJ Speech	3.68±0.037	3.96±0.089	3.73±0.075	3.99±0.059	3.85±0.068	**4.07±0.060**	3.68±0.082	4.10±0.059
	LibriTTS	3.75±0.107	3.74±0.099	3.50±0.086	**3.82±0.069**	3.48±0.083	3.80±0.073	3.36±0.092	4.03±0.065
	VCTK	**3.95±0.032**	3.94±0.089	3.75±0.074	3.87±0.068	3.77±0.074	3.86±0.069	3.66±0.079	3.98±0.064

The **VCTK** corpus [31] includes speech data uttered by 110 English speakers with various accents. Each speaker reads out about 400 sentences selected from a newspaper. We randomly select 85% of the samples for training data, 10% for validation, and 5% for testing.

Log-spectrogram computations. The speech signals in the three datasets are re-sampled to 24 kHz. We extract the 80-dimensional Mel-Spectrogram features using 40 ms Hanning window, 12.5 ms frameshift, 1024-point FFT, and 0 Hz & 12 kHz lower & upper-frequency cutoffs. We then perform log dynamic range compression on the resulting Mel-Spectrogram features followed by a min-max normalization.

3.2. Training Setup

For training each of the vocoders in our study, we conduct a hyperparameter search and report the best model configuration on the three different datasets described in § 3.1. Our framework is implemented on the PyTorch library, and training is performed on a Tesla V100 GPU. For reproducibility, we use the Amazon Web Services (AWS) to compute the evaluation metrics. Specifically, for CPU computations, we use `c5.4xlarge` AWS instance with 16 vCPU of 3.6GHz Intel Xeon Processors. For GPU computations, we use `p3.2xlarge` AWS instance with 8 vCPU of 2.3GHz Intel Xeon Processors and one NVIDIA Tesla V100 GPU.

For each of the vocoders, we start from the original configuration provided in the respective open-source implementation. However, for WaveNet, there are different configurations that vary in terms of input types and loss functions. For the input, we can use either raw waveform or pre-processed waveform using μ-law compression. For the loss function, there are two different options: Mixture of Logistics (MoL-loss) or a single Gaussian distribution (*normal-loss*). We run different versions of the WaveNet model using each configuration and report the one with the best performance. We found that on LJ Speech and VCTK, it is better to use μ-law compression on the input waveform; and on LibriTTS, raw waveform input yields the best results. For the loss function, using *normal-loss* helps to increase the overall performance.

3.3. Evaluation

Our aim is to evaluate multiple vocoders along different axes numerically and qualitatively. The choice of metrics is crucial for evaluation, and we consider the following metrics:

- **Mean Opinion Score (MOS)** is a subjective numerical measure of the human-judged overall quality after listening to a sample. We conducted the MOS study on each of the vocoder models with three different datasets. Each MOS test consists of 400 participants asked to rate the quality of each sample between 1-5 (1:bad - 5:excellent). We report the MOS for each vocoder as well as the ground truth over 20 samples from the test set.
- **Structural Similarity Index Measure (SSIM)** [21] is a quantitative metric that measures the similarity between two given images in the original study. We perform SSIM in the frequency domain to compare the synthetic spectrogram with the real-world sample.
- **Fréchet Audio Distance (FAD)** [22] measures the quality and diversity of the generated samples. FAD score is the distance between two multivariate Gaussian distributions estimated on the sets of embeddings, *i.e.* the background

and evaluation embeddings. To generate these feature embeddings, FAD use a VGG model [32] trained on a large YouTube dataset as the audio classifier.

- **Log-mel Spectrogram Mean Squared Error (LS-MSE)** is computed between the ground truth spectrogram sample and a generated one. We use the computation in § 3.1 to obtain the Log-mel Spectrogram for the synthesized speech samples. The LS-MSE can be interpreted as a measure of how close the low-dimensional representation of the spectrogram is when compared to the ground truth spectrogram.
- **Peak Signal-to-Noise Ratio (PSNR)** is the ratio of the power of a peak signal, which is the magnitude of the best-case output of a signal to the power of the noise at the peak measured in dB. We apply PSNR computation in the frequency domain, where the peak signal of the output is 1 and the distorting noise is represented by LS-MSE.

3.4. Results and Discussion

Table 1 shows the results of the five objective and subjective evaluation metrics described in § 3.3. Each of the metrics are computed using 20 audio samples from each dataset. For MOS, we report the mean value as well as the 95% confidence intervals. We use the Griffin-Lim vocoder [10] as a baseline to compare with each of the other vocoders in our study.

FAD and MOS metrics show close correlation especially for GAN-based vocoders. Both metrics have the same best-performing models in each dataset except LJ Speech. MOS reports that Diffwave is the best performing vocoder for LJ Speech with a MOS score of 4.07 ± 0.06, while Parallel WaveGAN achieves the best FAD score of 0.92 (which is the second-best in terms of MOS 3.99 ± 0.059). For LibriTTS dataset, Parallel WaveGAN has the best performance for both FAD (1.41) and MOS (3.82 ± 0.69). As for VCTK dataset, WaveNet achieves lower FAD (0.99) and higher MOS (3.95 ± 0.032) scores compared to other vocoders.

Observe that when using FAD and MOS metrics, each of the different models achieves their best performance on LJ Speech dataset, while having lower performance on VCTK and LibriTTS, respectively. This is due to the fact that LJ Speech is a single-speaker dataset which makes it easier to train and evaluate on. On the other hand, VCTK and LibriTTS are multi-speaker datasets. When LibriTTS is used for speaker generalizability, the scenario is more challenging as suggested from our experimental results.

Table 2 shows the model complexity of each vocoder and how that affects the voice synthesis computation time. We compare the following aspects of the neural vocoder models: the model parameter size, the number of Floating Point Operations per Second (FLOPS) of a speech sample, total training iterations, and Real-Time Factor (RTF).

The autoregressive models, namely WaveNet and WaveRNN, have a consistent number of parameters (3.79 and 4.35 Million parameters respectively) and FLOPS (89.65 and 94.98 GFLOPS respectively) when compared to other models. We exclude the RTF computation for the autoregressive model as they are significantly slower compared to other vocoders in our study. For real-time applications, custom kernels are used for autoregressive models such as LPCNET [33].

Table 2. Space and time complexity for vocoders under evaluation in terms of: (1) the number of parameters, (2) computation FLOPS, and (3) their corresponding RTF using on GPU and CPU setup. #Param for GANs (MelGAN and Parallel WaveGAN) is only for the generator and for a single step of inference for the diffusion models (WaveGrad and DiffWave).

Model	#Param (M)	GFLOPS	RTF	
			GPU	CPU
WaveNet	3.79	89.65	-	-
WaveRNN	4.35	94.98	-	-
MelGAN	3.05*	3.01	0.001	0.029
Parallel WaveGAN	1.34*	31.26	0.002	0.576
WaveGrad	15.81*	33.75	0.381	9.858
DiffWave	2.62*	31.70	0.070	4.452

For GAN-based vocoders, we report the number of parameters and FLOPS for the generator. MelGAN has fewer number of FLOPS (3.01 GLOPS) compared with the Parallel WaveGAN (31.26 GLOPS). This difference is also reflected in the RTF values, where MelGAN has RTF 0.001 RTF for GPU and 0.029 for CPU. On the other hand, Parallel WaveGAN achieves 0.002 RTF on GPU and 0.576 on CPU.

In Diffusion-based vocoders, WaveGrad has a relatively higher number of parameters (15.81 Million parameters) compared to DiffWave, while both models maintain the same order of magnitude for the number of FLOPS (33.75 and 31.70 GFLOPS respectively). We report the computation of a single step of the inference for both the number of model parameters and FLOPS. In our experiments, during inference we use 50 steps noise scheduler for WaveGrad and 6 steps for DiffWave, following the original implementation. This explains the higher RTF obtained for both vocoders in comparison to GAN-based, where WaveGrad has 0.381 RTF on GPU and 9.858 on CPU, respectively. DiffWave reports 0.070 and 4.452 RTF on GPU and CPU respectively.

4. CONCLUSION

We present VocBench, a framework for a general-purpose benchmark of neural vocoders on the speech synthesis task. VocBench provides the speech community a standard and comprehensive approach for neural vocoders evaluation. Our study includes results of both the objective and subjective differences for the vocoders. We have open-sourced our toolkit for training and evaluating neural vocoders on GitHub. We welcome the community to contribute and share their implementations and evaluations against SOTA vocoders.

References

[1] M. Tatham, "An integrated knowledge base for speech synthesis and automatic speech recognition," *Journal of Phonetics*, vol. 13, no. 2, pp. 175–188, 1985.

[2] F. Grigoras, H.-N. Teodorescu, L. C. Jain, and V. Apopei, "Fuzzy and knowledge-based control for speech synthesis," in *ECC*, 1999.

[3] G. K. Anumanchipalli, Y.-C. Cheng, J. Fernandez, et al., "KLATTSTAT: Knowledge-based parametric speech synthesis," in *Seventh ISCA Workshop*, 2010.

[4] K. Tokuda, T. Yoshimura, T. Masuko, et al., "Speech parameter generation algorithms for HMM-based speech synthesis," in *ICASSP*. IEEE, 2000, pp. 1315–1318.

[5] Y. Wang, R. Skerry-Ryan, D. Stanton, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv*, 2017.

[6] N. Li, S. Liu, Y. Liu, et al., "Neural speech synthesis with transformer network," in *AAAI*, 2019.

[7] J. Shen, R. Pang, R. J. Weiss, et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018.

[8] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *SLT Workshop*. IEEE, 2018.

[9] E. A. AlBadawy and S. Lyu, "Voice conversion using speech-to-speech neuro-style transfer," in *Proc. Interspeech*, 2020, pp. 4726–4730.

[10] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on acoustics, speech, and signal processing*, 1984.

[11] A. van den Oord, S. Dieleman, H. Zen, et al., "WaveNet: A generative model for raw audio," in *arXiv*, 2016.

[12] N. Kalchbrenner, E. Elsen, K. Simonyan, et al., "Efficient neural audio synthesis," in *ICML*, 2018.

[13] K. Kumar, R. Kumar, T. de Boissiere, et al., "MelGAN: Generative adversarial networks for conditional waveform synthesis," *arXiv*, 2019.

[14] G. Yang, S. Yang, K. Liu, et al., "Multi-band melgan: Faster waveform generation for high-quality text-to-speech," in *SLT workshop*, 2021, pp. 492–498.

[15] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*, 2020.

[16] N. Chen, Y. Zhang, H. Zen, et al., "WaveGrad: Estimating gradients for waveform generation," in *ICLR*, 2020.

[17] Z. Kong, W. Ping, J. Huang, et al., "DiffWave: A versatile diffusion model for audio synthesis," in *ICLR*, 2020.

[18] P. Govalkar, J. Fischer, F. Zalkow, et al., "A comparison of recent neural vocoders for speech signal reconstruction," in *ISCA*, 2019.

[19] M. Airaksinen, L. Juvela, B. Bollepalli, et al., "A comparison between straight, glottal, and sinusoidal vocoding in statistical parametric speech synthesis," *IEEE Trans Audio Speech Lang Process*, 2018.

[20] M. Schoeffler, F.-R. Stöter, B. Edler, et al., "Towards the next generation of web-based experiments: A case study assessing basic audio quality following the ITU-R recommendation BS. 1534 (MUSHRA)," in *1st Web Audio Conference*, 2015, pp. 1–6.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process*, 2004.

[22] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A metric for evaluating music enhancement algorithms," *arXiv*, 2018.

[23] R. Yamamoto, M. Andrews, M. Petrochuk, et al., "r9y9/wavenet_vocoder," https://github.com/r9y9/wavenet_vocoder, 2018.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative adversarial nets," *NeurIPS*, vol. 27, 2014.

[25] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *NeurIPS*, 2020.

[26] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput*, 2011.

[27] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*, 2015.

[28] A. Goyal, N. R. Ke, S. Ganguli, and Y. Bengio, "Variational walkback: Learning a transition operator as a stochastic recurrent net," *NeurIPS*, 2017.

[29] K. Ito and L. Johnson, "The LJ speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[30] H. Zen, V. Dang, R. Clark, et al., "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[31] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," https://datashare.ed.ac.uk/handle/10283/2651, 2017.

[32] S. Hershey, S. Chaudhuri, D. P. Ellis, et al., "CNN architectures for large-scale audio classification," in *ICASSP*. IEEE, 2017, pp. 131–135.

[33] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *ICASSP*. IEEE, 2019, pp. 5891–5895.