Master Thesis

Gaspard Ulysse Fragnière

August 2022

1 How to LaTeX

How to make a reference to a paper: [7]

2 Draft

2.1 Grumiaux

[7] is a survey of several methods for sound source localization (SSL). Tradiontionally, this problem has been tackled using Signal Processing based methods. But in the recent years, methods based on deep learning have been developped and showed better results than traditional approaches. Those methods have been compiled in this paper. The survey is organized in the different following sections:

- **Section I**: Introduction
- **Section II**: Acoustic Environment and Sound Source Configuration
- **Section III**: Conventional SSL methods
- **Section IV**: Neural Network Architectures for SSL
- **Section V**: Input Features
- **Section VI**: Outputs strategies
- **Section VII**: Data
    1. Synthetic Data
    2. Real data
    3. Data augmentation techniques
- **Section VIII**: Learning Strategies
- **Section IX**: Conclusions and Perspectives

We are interested in the section about Synthetic Data and Data augmentation. Indeed those sections can be used as a statring point for building the state of the art.Its goal is to answer the following questions:

- Are there **existing methods** to generate virtually:
  - measured time data (single channel/multi-channel)?
  - measured source spectra (single channel/multi-channel)?
  - measured cross-spectral matrices in stationary environments (multi-channel only)?
- What **measurement scenarios** are used in the literature (time-stationary/non-stationary sources, number of microphones, temporal dimensions...)?
- What are the **existing setups** in multi-channel data generation with neural networks (conditioning variables, network architectures (convolutional, recurrent, Transformer,...), generative algorithms (GAN/VAE), ...)

In [7], a classical method about data generation is introduced. The idea is the following: Simulate the Room Impulse Response (RIR) in order to simulate realistically room acoustics (e.g. reverberation). Then convolve dry audio signals with the RIR simulated. This can provide suited training data, since RIR for rooms of different size, different source prosition as well as different dry signals can be used for the training.

In [3], the datasets is created in the following way: a speaker with a visual marker is placed in front of camera and binaural microphone setup (dummy head). "The loud-speaker that emits fixed-length full-spectrum sounds is moved in front of the dummy-head/camera device and for each loud-speaker location, both the emitted sound and the image location of the visual marker are recorded. → not so useful

In [15], a GAN is used to simulate data. The GAN used in [15] is an implementation of [11]

[8] proposed a low-complexity model-based training data generation method that includes a deterministic model for the direct path and a statistical model for late reverberation. It has been demonstrated that the SSL neural network, trained using the data generated by this method, achieves comparable localization performance as the same architecture trained on a dataset generated by the usual ISM.

An investigation of several simulation methods was done by [5], with extensions of ISM, namely, ISM with directional sources, and ISM with a diffuse field due to scattering. [5] compared the simulation algorithms via the training of an MLP (in both regression and classification modes) and showed that ISM with scattering effects and directional sources leads to the best SSL performance.

[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.

Paper read and not useful:

- Deleforge 2013 [2]
-

2.2 Bianco

- **in:** Relative Transfer Function (RTF)
- **out:** Relative Transfer Function (RTF)

- **measurement scenario:** Binaural microphone.
- **setup:** VAE (Semi supervised learning)

In [1] a VAE is used to perform SSL. The idea is the following: based on VAEs to encode the phase of the relative transfer function (RTF) between two microphones to a latent parametric distribution. The resulting model estimates DOA and generates RTF phase.

VAEs learn from unlabeled data explicit latent codes for generating samples, and are inspiring examples of representation learning.

There is a link between DOA and RTF. Indeed, the RTF phase is encoded as a function of source azimuth (direction of arrival, DOA). Similarly as in [6], the goal of the NN (GAN or VAE) is to learn the distribution of a a quantity that is a function of the DOA.

The experiments show, only **two labeled samples per DOA** permit the VAE-SSL to obtain better performance than SRP-PHAT (State of the art).

2.2.1 Bianco: the maths:

The goal in [1] is to create a VAE to generate an acoustics feature. The acoustics feature of interests here is the Relative Transfer Function (RTF). Model: we consider the following model:

$$d_i = s * a_i + u_i \tag{1}$$

with

- $i \in \{1, 2\}$: the microphone index
- d_i: time domain acoustic recording at microphone i
- s the acoustic source
- a_i: the impulse response (IR) at microphone i
- u_i: the noise at microphone i

Then we can define the RTF $H(k)$ as

$$H(k) = \frac{A_1(k)}{A_2(k)} \tag{2}$$

with $A_i(k)$ the Fourier transform of a_i and k the frequency.
Then $H(k)$ can be estimated by (with d_1 as reference):

$$\hat{H}(k) = \frac{S_{d1d2}}{S_{d1d1}} \tag{3}$$

with

- $S_{d1d1} = D_1(k)^* D_1(k)$: the Power Spectral Density (PSD)
- $S_{d2d1} = D_2(k)^* D_1(k)$: the cross-PSD for a single frame.

The estimator is biased since we ignore the PSD of the noise. For each FFT frame a vector is obtained of RTF is obtained $\hat{\mathbf{h}} = [\hat{H}(1), \ldots, \hat{H}(K)]^T \in \mathbb{C}^K$ for K frequencies bin. We use RTFs estimated using a single frame as input to the VAE-SSL.

the nth input sample and the supervised CNN is a sequence of RTFs frames :

$$\mathbf{x}_n = \text{vec}(\text{phase}(\hat{\mathbf{H}}_n)) \in \mathbb{R}^{KP} \tag{4}$$

with

- $\hat{\mathbf{H}}_n = [\hat{\mathbf{h}}_n \ldots \hat{\mathbf{h}}_{n+P-1}] \in \mathbb{C}^{K \times P}$
- $K = N_{\text{FFT}}/2$
- P: the number of RTF frame in the sequence.

2.3 Neekhara

- **in:** (perceptually informed) spectogram (or text on a more basic level)
- **out:** natural sounding audio waveform
- **measurement scenario:** -
- **setup:** GAN (for amplitude estimation)

[11] is concerned with finding a solution for Text to speech (TTS) problem. The claim is that using a GAN approach, they have been able to outperform by far naive approaches (user review) and being 100x faster than other DL approaches. More specifically this paper was concerned with creating a mapping from language to **perceptually informed spectogram**. Indeed the difficluty of the problem lays in the fact that perceptually informed spectograms are not invertible. Indeed a spectogram is a compact representation of a signal where much of the information contained in a audio waveform has been lost. More specifically the problem at hand is phase estimation and magnitude estimation. Therefore a predictive model is required to fill the missing information and create natural sounding sound.

2.4 NEURIPS

- **in:** mel-spectogram
- **out:** audio waveform
- **measurement scenario:** -
- **setup:** non-autoregressive feed-forward convolutional architecture to perform audio waveform generation in a GAN setup

[10] also introduces a method for the TTS problem. Moreover in the introduction of the paper, there is a comparison of different method for text to speech (i.e. audio wave generation):

- **Pure signal processing approaches:** "The main issue with these pure signal processing methods is that the mapping from intermediate features to audio usually **introduces noticeable artifacts**"

- **Autoregressive NN based models:** An autoregressive model is a model that relies on past values to predict current ones. In this sense, an autoregressive model must be sequential. "These methods have produced state-of-the-art results in text-to-speech synthesis and other audio generation tasks. Unfortunately, inference with these models is inherently slow and inefficient because audio samples must be generated sequentially. Thus auto-regressive models are **usually not suited for real-time applications**."
- **Non autoregressive models:** "While inference is fast on the GPU, the large size of the model makes it **impractical for applications with a constrained memory budget.**"
- **GAN for audio:** However their results show that adversarial loss alone is not sufficient for high quality waveform generation; it requires a KL-divergence based distillation objective as a critical component. To this date, making them work well in this domain has been challenging

In [10], a GAN (MelGAN) is introduced. The goal of this GAN is to perform audio waveform generation.

2.5 EngelGanSynth:

- **in:** instruments note from datasets NSynth (Single channel)
- **out:** instruments audio waveform.
- **measurement scenario:** -
- **setup:** GAN

[4] is concerned with demonstrating that GANs can in fact generate high-fidelity and locally-coherent naudio by modeling log magnitudes and instantaneous frequencies with sufficient frequency resolution in the spectral domain.

Using a GAN approach and the NSythn datasets (Dataset of standardized notes from instrument) to generate notes.

TODO: According to Adam, this paper is to generate spectogram and not audio wave form. → check again

2.6 gerstoft

- **in:** True Sample Covariance Matrix with DOA
- **out:** Generator, i.e. probability distribution function
- **measurement scenario:** -
- **setup:** (Wasserstein) GAN

[6] is concerned with training a GAN for generating audio based features.

In many array processing techniques (i.e. beamfroming) first compute the Sample Covariance Matrix (SCM). The idea of the (Wasserstein) GAN here is to generate many SCM. More specifically, the goal of the GAN is to learn the **joint** probability distribution function of the **observable data**

(array data) and the **target variable** (Wasserstein distance). The wasserstein distance is a metric to measure distance between probability distribution on a given metric space.

The idea is first to consider a model that describe relationship between **array data** and **DOA**. Using this model, another relationship betwen **CSM** and **array data** can be defined. Hence we have a mathematical relationship between **DOA** and **CSM**.

Idea: maybe it would be possible to generate for the pdf os SCM for several DOA. Moreover, a link between SCM and cross spectral matrix should be investigated.

2.7 Vera-Diaz

- **in:** GCC
- **out:** (x,y,z)
- **measurement scenario:** pair of microphone
- **setup:** (Deep) CNN

In ASL prblem, the source's position can be estimated with at least three Time Difference of Arrival (TDoA) measurements with hyperbolic trilateration methods. Signals captured in everyday scenarios are contaminated with noise and multipath effects. Directly measuring TDoA in those cases is a difficult task that produces inaccurate localization results. [16] is concerned with finding a way to denoise those signals. Other methods uses GCC instead of TDoA. SUch methods are more robust to noise and multipath effects, but not fully immune to it.

The contribution of this paper is a DNN named DeepGCC. This DNN takes as input a GCC(Generalized Cross Correlation) Matrix and estimates a Gaussian function. Then the SSL problem is solved by replacing the GCC typically used by DeepGCC(GCC) and then using a classical beamforming approach.

$\rightarrow$ this is actually the inverse of what we want. Hence if we could reverse this nectwork, maybe we could do something of the following:

$$(x, y, z) \rightarrow GCC \rightarrow CSM \tag{5}$$

Where the last step is done thanks to the relation between GCC and CSM via Fourier transform.

2.8 Hübner

- **in:** -
- **out:** data for phase based DOA approximation. (RTF)
- **measurement scenario:** -
- **setup:** CNN

This paper is concerned with dealing with a common issue when creating DL algorithm to solve the SSL problem, namely the complexity to gather data for training/validating. Indeed, the two current way to obtain data is by either generating them using simulation or by recording them in real life. Both methods require significant amount of resources (resp. time or storage)

TODO: Use those quotes to structure a bit the the state-of-the-art

quotes from Hübner: 2 "DOA estimation methods can be categorized into classical model-based methods and data-driven methods, which are prevalently implemented using deep neural networks (DNN)."

"Most of the DL methods include a feature extraction step rather than using the raw microphone signals. Popular features include:

- (i) the eigendecomposition of the spatial covariance matrix [14] (similar to MUSIC)
- (ii) generalized cross-correlation (GCC) based features [15–18]
- (iii) modal coherence [19]
- (iv) the Ambisonics intensity vectors [20]
- (v) phase and magnitude spectra [21] and
- (vi) phase spectra [11, 12]. Many of the features are phase-based as motivated by physical models and classical DOA estimators [9].

"

"One way to generate training data for DL-based DOA estimation is by recording sound emitted from a source (e.g., loudspeaker, human) in real acoustic environments [16, 17]. This approach is time-consuming and for high-quality datasets a precise ground truth position is essential, which requires expensive measurement equipment."

"Another popular method is the convolution of signals (e.g., speech) with room impulse responses (RIRs) that have either been recorded [14, 18] or simulated based on the source-image method [11, 12, 20, 21, 23]. The main drawbacks of these data generation methods are excessive time and storage consumption."

Conclusion: "We proposed a low complexity model-based training data generation method for phase-based DOA estimation. The proposed method models the microphone phases directly in the frequency domain to avoid computationally costly operations as present in state-of-the-art methods. The low computational complexity of the proposed method allows for online training data generation, which allows faster proto- typing, and paves the way for **applications with a high data demand** such as moving sound sources simulation or large microphone arrays. An evaluation using measured RTFs yielded **comparable results** for phase-based DOA estimation when using **the proposed method and the computationally expensive source-image method for training data generation.**"

2.9 46107

[9] is concerned with room simulation -> precisely not what we want to do.

2.10 Papayiannis

[12] is concerned with the generation of Acoustic Impulse Response.

Reverberation are a good representation of the acoustic environment. In numerous taskm it is useful to be able to know in what environment a speech recording has been made (for instance), based on the reverbaration present in the recording. For this purpose, ML classifier have been built,

in order to classify different enviroment. A feature that is typically used for such a classification is the Acoustic Impulse Response represented as Finite Impulse Response (FIR).

The issue is that a lot of IR are required for building a classifier. An IR is typically measured, hence the number that can be created is limited (time wise). For this reason, [12] introduce a way to generate (with GAN) artificial AIR from measured AIR. → data augmentation.

"This is an alternative to the process of measuring many more AIRs, by moving the source and receiver at various positions in the same real room. Repeating the process for a number of rooms expands the available dataset, without the need for any additional data collection"

"A challenge to overcome during training is related to the motivation for this work, which is the high-dimensionality of AIRs. This is overcome by using a proposed low-dimensional representation for acoustic environments. The representation describes sparse early reflection using the parameters estimated in [6] and uses established acoustic parameters to represent the late reverberation."

-> **Question for Adam: Are IR dependent on the source-receiver positions ? -> it is never mentionned in [12]**

Data representation: During training, AIRs are presented to the networks using **taps of FIR filters → to investigate**. The taps represent the sound pressure at the position of a receiver placed in the room, with the room excited by a source placed within its boundaries.

-> Hence the data generated is dependent on the dimension of the room. GAN trained for a specific room. But then the position of the receiver and source a randomly chosen (not conditional)

"The task of room classification is to identify a known room at unknown source and receiver positions. With the transformation being class invariant, the available training AIRs from a room will be used to artificially generate AIRs at new source and receiver positions from the same room."

→ the main contribution of this paper is to propose a low dimension representation of AIR that is still informative (i.e. contains info about early reflection, late reverberation) → look more into that (chapter: Proposed low-dimensional representation)

In this work, the generation of AIRs is based on training one GAN for each of the 7 rooms, part of the training database. Therefore, 7 GANs are trained and each one of them is used to generate a number of AIRs as if they were measured in the corresponding rooms.

Conclusion: as it is, this paper is not useful. For it to be useful for the task at hand, the GAN would need be conditionned on:

- Source and Receiver positions
- Room dimension (maybe less important)

2.11 Ratnarajah

[13] proposes a fast method (NN-based) for generating Room Impulse Response (RIR). The input paramaters used are the following:

- rectangular room dimensions
- listener position
- speaker position
- reverberation time (T_{60})

An issue with IR generator is that they are computationally very expensive. Moreover for NN training significant amount of data are required, hence offline data generation leads to high storage needs.

"To generate RIRs for a given acoustic environment, we propose a one-dimensional conditional generator network. Our generator network takes room geometry, listener and speaker positions, and T_{60} as inputs, which are the common input used by all traditional RIR generators, and generates RIRs as raw-waveform audio. Our FAST-RIR generates RIRs of length 4096 at 16 kHz frequency."

$\rightarrow$ all together this paper seems to provide good results and seems applicable for the task at hand. $\rightarrow$ **TODO: write about this in State-of-the-Art alongside Papayiannis and ask Adam if it makes sense to keep both (probably not)**

2.12 Thoughts about the different data generation methods:

In the thesis task, the following three data generation approach are listed (LHS) Whereas in the meeting, we mentionned the following three approaches (RHS):

- Measurement $\leftrightarrow$ experimental measurements
- Syntetic $\leftrightarrow$ calculation of theoretical model [10]
- Semi-synthetic $\leftrightarrow$ virtual measurement [11]

In the state of the art, semi-synthetic approach must be mentionned explained and its limitation must be stated. Moreover a time were such an approach was used need to be mentionned.

Semi synthetic data generation approach are not the best suited for several reason. First, the data produce is not usually not as accurate as actual measurement. Moreover, generating sufficient data is extremely tedious and is often more time consuming than model training, since extremley significant quantity of data are required to train a NN. Indeed to have a good measurement a lot of IR must be created.

Question: is semi synthetic when the RIR is measured and full synthetic when the RIR is generated with "maths" ? $\rightarrow$ wait for Adam email and use response to rewrite.

2.13 Difference between VAE and GAN

In the context of image generation:

"GANs generally produce better photo-realistic images but can be difficult to work with. Conversely, VAEs are easier to train but don't usually give the best results.

I recommend picking VAEs if you don't have a lot of time to experiment with GANs and photorealism isn't paramount.

There are exceptions such as Google's VQ-VAE 2 which can compete with GANs for image quality and realism. There is also VAE-GAN and VQ-VAE-GAN.

As a note, GANs and VAEs are not specifically for images and can be used for other data types/structures." (this is from a comment on stackstats -> not really usable as a source).

TODO: $\rightarrow$ read [14]

from [12]: "A generative model represents the joint probability $P(x, y)$, which is in contrast to classification DNNs that estimate the posterior $P(x|y)$. Recent advancements in deep learning led to the proposal of alternatives to the traditional method for the estimation of parametric model

distributions. The two dominant methods in the modern literature are GANs and Variational Autoencoders (VAEs). Both follow a similar formulation that uses back-propagation to train network layers, which are able to estimate the generative model by filtering noise drawn from a known prior. In the literature review conducted for this work, GANs have shown to be widely adopted in the field of audio processing across different tasks such as SED [14], speech recognition [15], speech enhancement [16] and dereverberation [17, 18]. Furthermore, variants of the original GAN in [19] exist, which can be adapted in the future to lead to more exciting applications of the method proposed in this work, such as Conditional GANs [20], DualGANs [21] and many others. GANs are therefore chosen as the estimation mechanism for the generative models in this paper."

3 Goal

The goal of the project is to create a GAN to generate either:

- cross-spectral matrix
- the complex-valued sound pressure vector at the different microphone

with the corresponding labels (DoA).

In some way the goal of this project is to create a GAN to realistically add noise to either a cross spectral matrix or complex-valued signal sound pressure vector of an array of microphone. This noise pdf should be estimated instead of being modelled. Indeed a GAN approach is necessary to reduce the computation time and the storage required. Indeed we want to be able to generate randomly and in real time labeled data.

4 How to train a GAN:

1. feed sample to the discriminator to make it learn what a real sample is.
2. When the discriminator can distinguish recognize real sample, we need to feed it fake sample and make sure it can recognize them as fake
3. When the discriminator is good enough at his job, we can start training the generator. The generator takes as input a random input vector and create a sample
4. The knowledge whether the sample was fake or not is revealed to both networks. Based upon this, the generator and discriminator need to adapt their behaviour:
    - there is always a winner and a loser
    - if the discriminator successfully detect the image as fake, it remains unchanged an5d the generator need to change its behaviour
    - On the other hand, if the discriminator fail to detect the image as fake, it has to adapt its behaviour and the generator stays unchanged.
5. We repeat this process until the generator is so good that the discriminator can no longer detect the fakes.

5 Important words and relationships

- **SSL**: Sound source localization
- alternatively **SSL**: Semi-Supervised Learning
- **ASL**: Acoustics Source Localization
- **Source Spectra**
- **GCC**: Generalized Cross Correlation
- **CSM**: Cross Spectral Matrix (note that there is a connection between the CSM and the GCC features via the Fourier transform)
- **SCM**: Sample Covariance Matrix
- **CB map**: Conventional Beamforming map (also known as Acoustic Power Map???)
- **conditioning variable**: splitting the data up into bins based on the values of these features, and then training a model for each bin. Then examining the differences between the models. Usually this is done to learn something about the benefits of using the different features, and about the relationships between features and outputs.
- **(Time) Stationary process**: process with mean, variance and autocorrelation structure not changing over time (constant overtime)
- **Autoregressive**: An autoregressive model specifies that the output variable depends linearly on its own previous values and on a stochastic term
- Data Covariance Matrix $\leftrightarrow$ Cross Spectral Matrix $\leftrightarrow$ Cross Power Spectrum.
- relationship between pressure vector in the microphone array $\mathbf{p}$ and the Cross-Spectral Matrix $\mathbf{C}$
- SED: Sound Event Detection

$$\mathbf{C} = \mathbf{p}\mathbf{p}^H \tag{6}$$

6 Papers found summary:

6.1 Bianco

In [1]:

- **in:** Relative Transfer Function (RTF)
- **out:** Relative Transfer Function (RTF)
- **measurement scenario:** Binaural microphone.
- **setup:** VAE (Semi supervised learning)

6.2 Neekhara

In [11]:

- **in:** (perceptually informed) spectogram (or text on a more basic level)
- **out:** natural sounding audio waveform
- **measurement scenario:** -
- **setup:** GAN (for amplitude estimation)

6.3 NEURIPS

In [10]:

- **in:** mel-spectogram
- **out:** audio waveform
- **measurement scenario:** -
- **setup:** non-autoregressive feed-forward convolutional architecture to perform audio waveform generation in a GAN setup

6.4 Engel

in [4]:

- **in:** instruments note from datasets NSynth (Single channel)
- **out:** instruments audio waveform.
- **measurement scenario:** -
- **setup:** GAN

6.5 Gerstoft

In [6]:

- **in:** True Sample Covariance Matrix with DOA
- **out:** Generator, i.e. probability distribution function for Sample Covariance Matrix.
- **measurement scenario:** -
- **setup:** (Wasserstein) GAN

6.6 Vera-Diaz

In [16]:

- **in:** GCC
- **out:** (x,y,z)
- **measurement scenario:** pair of microphone
- **setup:** (Deep) CNN

References

[1] Michael J Bianco, Sharon Gannot, and Peter Gerstoft. Semi-supervised source localization with deep generative modeling. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020.

[2] Antoine Deleforge, Florence Forbes, and Radu Horaud. Variational em for binaural sound-source separation and localization. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 76–80. IEEE, 2013.

[3] Antoine Deleforge, Radu Horaud, Yoav Y Schechner, and Laurent Girin. Co-localization of audio sources in images using binaural features and locally-linear regression. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(4):718–731, 2015.

[4] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[5] Femke B Gelderblom, Yi Liu, Johannes Kvam, and Tor Andre Myrvoll. Synthetic data for dnn-based doa estimation of indoor speech. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4390–4394. IEEE, 2021.

[6] Peter Gerstoft, Herbert Groll, and Christoph F Mecklenbräuker. Parametric bootstrapping of array data with a generative adversarial network. In *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 1–5. IEEE, 2020.

[7] Pierre-Amaury Grumiaux, Srđan Kitić, Laurent Girin, and Alexandre Guérin. A survey of sound source localization with deep learning methods. *The Journal of the Acoustical Society of America*, 152(1):107–151, 2022.

[8] Fabian Hübner, Wolfgang Mack, and Emanuël AP Habets. Efficient training data generation for phase-based doa estimation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 456–460. IEEE, 2021.

[9] Chanwoo Kim, Ananya Misra, Kean Chin, Thad Hughes, Arun Narayanan, Tara Sainath, and Michiel Bacchiani. Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home. pages 379–383, 2017.

[10] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[11] Paarth Neekhara, Chris Donahue, Miller Puckette, Shlomo Dubnov, and Julian McAuley. Expediting tts synthesis with adversarial vocoding. *arXiv preprint arXiv:1904.07944*, 2019.

[12] Constantinos Papayiannis, Christine Evers, and Patrick A Naylor. Data augmentation of room classifiers using generative adversarial networks. *arXiv preprint arXiv:1901.03257*, 2019.

[13] Anton Ratnarajah, Shi-Xiong Zhang, Meng Yu, Zhenyu Tang, Dinesh Manocha, and Dong Yu. Fast-rir: Fast neural diffuse room impulse response generator. 10.48550. *arXiv preprint ARXIV.2110.04057*, 2021.

[14] Yuanyuan Sun, Lele Xu, Lili Guo, Ye Li, and Yongming Wang. A comparison study of vae and gan for software fault prediction. In Sheng Wen, Albert Zomaya, and Laurence T. Yang, editors, *Algorithms and Architectures for Parallel Processing*, pages 82–96, Cham, 2020. Springer International Publishing.

[15] Elizabeth Vargas, James R Hopgood, Keith Brown, and Kartic Subr. On improved training of cnn for acoustic source localisation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:720–732, 2021.

[16] Juan Manuel Vera-Diaz, Daniel Pizarro, and Javier Macias-Guarasa. Acoustic source localization with deep generalized cross correlations. *Signal Processing*, 187:108169, 2021.