Master Thesis

Gaspard Ulysse Fragnière

August 2022

1 State-of-the-art

The problem of Acoustical Source Localization (ASL) is an important problem. It was many applications suach as smart assistant (e.g. Google Home, Alexa, ...), industrial applications, **TODO: add more?**. Traditionnaly this problem is tackled with methods based on the physics of sound propagation (e.g. TDoA, beamforming) or with statistcal inference (e.g.Sparse Bayesian Learning).

The recent success of Deep Learning (DL) based method in other field of research (e.g. Computer Vision) led to believe that Deep Neural Networks (DNN) based approaches could provide state-of-the-art result in solving the ASL problem. Castellini et al. (2021), Kujawski et al. (2019), Lee et al. (2021), Ma and Liu (2019), Pinto et al. (2021) and Xu et al. (2021) propose state-of-the-arts DL-based methods for Source Characterization.

A common issue faced while implementing DL-based methods is that significant quantities of well structures data are required. In the litterature, the data has been obtained using the following approaches:

- **Real Mesurement**: To create the different samples of such a dataset, sounds emitted with a loudspeaker or human voices are recorded in a real acoustic enviromnent. Eventhough such a method allows for the creation of perfectly realistic samples, it does not come without any issue. Indeed, it is very tedious and time consuming to record in different environment. Additionally, all the environment for measurement need to physically exists, which limits the quantity of possible samples. Moreover, to build a high quality data set, expensive equipment is required to have an accurate groundtruth (i.e. precisely identify the location of the sources). In the literature, He et al. (2018) and Ferguson et al. (2018) have used such an approach.
- **Synthetic Data**: The sounds used are artificial (i.e. white noise, sine wave). The room acoustic is also simulated. Indeed the dry sound is convolved with a simulated Room Impulse Response (RIR) to mimic the effect of room acoustics (e.g. reverberation). Compared to real measurement, this approach allows sample in more diverse environment. Indeed RIR for rooms of arbitrary size, different source position as well as different dry signals can be used for the training. The issue with such a method is the important amount of time and storage required for the creation of the datasets. E.g. Chakrabarty and Habets (2017), Perotin et al. (2018) and Adavanne et al. (2018) created their datasets in this way.
- **Semi-synthetic data**: The creation of such a dataset is similar the creation of synthetic dataset. The difference lies in the fact that the dry sound source used and the RIR are measured and not simulated. Then, the samples of such a dataset are generated by convolving

dry sounds with RIR. This method is not the best suited, since it is very time-consuming to generate a data set with enough samples for training a DL-based algorithm. Indeed, measuring all the RIR lead to the issues faced with real measurement. Takeda and Komatani (2016) use such an approach for to obtain their data.

Moreover it is to be noted that none of these methods are suitable for online data generation. Indeed, any of the above mentionned method do not allow for creating random sample while training DL-based algorithm. To use such datasets for training, they need to be fully created (and stored) before any training can occur.

1.1 DL-based data generation

In the past years, DL-based approaches have shown to be able to learn and realistically reproduce very complicated data structures (e.g. generation of pictures of faces in the field of Computer Vision). Those breakthroughs lead to believe that similar data generation methods could be used to fix the above-mentionned issues (e.g. offline training, lack of variance in the different samples, ...).

Moreover, it is relevant to note that the data used for source characterization in Castellini et al. (2021), Lee et al. (2021), Ma and Liu (2019), Xu et al. (2021) is the Cross Power Spectra (CPS), i.e. a direct representation of the signals received in the array of microphone. Indeed those approaches do not use direct recording of microphone input but instead features extracted from the raw data. This is crucial because it means that recording, simulating or generating raw microphone data is no longer necessary, if features (e.g. CPS) could be generated directly. We therefore need to identify what acoustic quantities:

- have already been generated using a DL approach
- are potential feature for a Source Characterization Algorithm.

1.1.1 Generation of Signal

Neekhara et al. (2019), Kumar et al. (2019), Engel et al. (2019) use Generative Adversial Network (GAN) to generate realistic audio waveform. Neekhara et al. (2019) and Kumar et al. (2019) specifically focus on the generation of audio waveform conditioned on a spectogram (cGAN). On the other hand, Engel et al. (2019) design a GAN to generate realistic audio waveform of single music notes played by an instrument. The data generated in those approaches is single-channel data, but maybe it could be extended to multi-channel to simulate the different signals recorded in an array of microphone. It is relevant to note that the GAN designed by Neekhara et al. (2019) is the one implemented in Vargas et al. (2021) in order to compare the accuracy of a network for single source DoA estimation when trained with different sound classes.

1.1.2 Generation of Impulse Response

Papayiannis et al. (2019) introduce a GAN approach to generate artificial Acoustical Impulse Response (AIR) of different environment in order to generate data for a NN used for classification of acoustic environment.

Ratnarajah et al. (2021) proposes a fast method (NN-based) for generating Room Impulse Response (RIR). The input paramaters of the networks used for creating the IR are the desired dimensions of the rectangular room, listener position, speaker position and reverberation time (T_{60}).

TODO: is it worth mentionning both papers ?

This is relevant for a problem at hand because if we are to be able to generate impulse responses with known source and listener position, we could simply convolve them with the dry source sounds. This way, we could generate raw microphone signal and use them to train a DL-based algorithm for source characterization.

1.1.3 Generation of potential NN feature

Bianco et al. (2020) proposes an approach to generate another acoustic feature: the phase of the relative transfer function (RTF) between two microphones. In this paper a Variational Auto Encoder (VAE) is designed to simultaneously generate phases of RTF and classifying them by their Direction of Arrival (DoA).

Gerstoft et al. (2020) use a GAN to generate Sample Cross Spectra Matrices (CSM). for a given DoA. In their approach, the GAN is trained with data only coming from one DoA, making it unable to generate sample for different DoA. This approach could be extended by creating a conditional Generative Adversial Network (cGAN) taking as input the DoA. Such a GAN would receive a DoA as input and use it to produce a CSM corresponding to the received DoA.

1.1.4 Other possible approaches to generate the data

In Hübner et al. (2021) introduce a low complexity model-based method for generating samples of microphones phases. This method proposed is not based on DL. Indeed, it is based on a statistical noise model, a deterministic direct-path model for the point source, and a statistical model. The claim of this paper is that the low complexity of the proposed model makes it suited for online training data generation.

Vera-Diaz et al. (2021) introduce a CNN for denoising (i.e. removing the effects of reverberation and multipath effects) on the Generalization Cross Correlation (GCC) matrix of an array of microphone. More specifically than a CNN, the network used has a encoder-decoder structure. This means that a possible approach to create the data we want, would be to attempt to invert network proposed. With this we could realistically add noise to GCC matrices and hence making it suitable for training.

2 Fundamentals

2.1 GAN

Goodfellow et al. (2020) introduce a new approach to solve the problem of generative model. The goal is to learn the probability disitrbution that was used to generate samples, by observing them.

The method introduced in this paper is called Generarive Adversial Network (GAN). The idea is to create a game (in the game theory sense) where two networks compete against each other. The first network is called a discriminator and its goal is to determine real from fake samples. The other network is a generator with the aim to produce data realisitic enough that the discriminator cannot determine that it is fake.

The generator takes as input a random vector and use it generate a sample. This random vector is referred to as latent variable. The vector space of latent variable is called latent space. After training, the generator should be a mapping from the latent space to the data space. The latent space is a representation of smaller dimension of the data space.

The discriminator takes as input a real or generated sample and predicts its authenticity. The discriminator is a simple binary classifier. The real sample come from the datasets and the fake are output of the generator.

Both models are trained simultaneously. First the generator create a batch of fake sample. This batch is then fed to the discriminator, alongside a batch of real samples. For each of them, the discriminator makes a prediction about their authenticity. The discriminator then gets updated based on how accurate it was at classifying samples and the generator based on how many times fake sample were able to fool the discriminator. We can see here that the training of both networks is supervised, on the contrary of typical generative models.

In a gaming theory framework, both networks are competing in a zero-sum game. This means that if one network performs well at its task and gets rewarded by little weights update, the other must have performed poorly and hence gets penalized by heavy weights updated. E.g. if the discriminator was successful at classifying all samples, it means that the generator had not been able to fool the discriminator by producing any realistic fake samples.

TODO: include more maths and/or more graphs?

2.2 DCGAN

Radford et al. (2015) introduce GAN, but unlike in Goodfellow et al. (2020), the architecture of the discriminator and generator is not achieved with regular perceptron, but with Convolutional layer. We first remind how a regular perceptron layer works. For an input vector $\mathbf{x} \in \mathbb{R}^k$ and output vector $\mathbf{y} \in \mathbb{R}^l$, a perceptron layer consists of two parts:

- an activation $\mathbf{a} = \mathbf{W}\mathbf{x} + \mathbf{b}$
- a non-linearity $y = f(\mathbf{x}; \theta) = \sigma(\mathbf{a})$

where $\mathbf{W} \in \mathbb{R}^{l \times k}$ are the weights of the perceptron and $\mathbf{b} \in \mathbb{R}^l$ its bias. $\sigma(\cdot)$ is called the activation function and is the source of non linearity in neural networks. An example of such activation function is the Rectified Linear Unit (ReLU) where:

$$\sigma_{\text{ReLU}}(\mathbf{a}) = [\max(a_0, 0), \ldots, \max(a_{l-1}, 0)]^T \tag{1}$$

We can now introduce the architecture of a convolutional layer. Without loss of generality, we assume that our network reecive as input an square image with a single channel, i.e. $\mathbf{H_0} \in \mathbb{R}^{d_0 \times (m \times m)}$ with $d_0 = 1$, transforms it with one square kernel, i.e. $\mathbf{K} \in \mathbb{R}^{s \times (r \times r)}$ with $s = 1$ to output another square image with a single channel, i.e. $\mathbf{H_1} \in \mathbb{R}^{d_1 \times (n \times n)}$, with $d_1 = 1$

The relationship between input image $\mathbf{H_0} \in \mathbb{R}^{m \times m}$, output image $\mathbf{H_1} \in \mathbb{R}^{n \times n}$ and kernel $\mathbf{K} \in \mathbb{R}^{r \times r}$ is defined as:

- an activation $\mathbf{A} = \mathbf{H_0} * \mathbf{K}$
- a non-linearity $\mathbf{H_1}_{(i,j)} = \sigma(\mathbf{A}_{(i,j)})$

Radford et al. (2015) proposes a set of constraint that makes the use of convulotional layers possible in GAN setting.

2.3 WGAN

Arjovsky et al. (2017) introduce a new type of Generative Adversial Network, namely the Wasserstein GAN (WGAN). The claim is that WGAN improves the stability in learning and get rid of typical problem of the traditional GAN approach such as Mode Collapse or Convergence failure. **TODO: here need to explain mode collapse ?**.

2.3.1 The Earth-Mover or Wasserstein distance

The goal of WGAN, remains the same as GAN, namely solving the problem of generative model. More precisely, this mean approximating the probability distribution P_r of some data by a distribution P_θ. For this reason, it is necessary to have metrics to quantify distance between two probability distributions P_r and P_m. Example of such distances are the Kullback-Leibler divergence or the Jensen-Shannon divergence. In WGAN, the distance used is the Earth-Mover (EM) distance or Wasserstein-1, defined as

$$W(\mathbb{P}_r, \mathbb{P}_m) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_m)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||] \tag{2}$$

Where $\Pi(\mathbb{P}_r, \mathbb{P}_m)$ is the set of all joint distribution $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_m$. Informally, $\gamma(x, y)$ shows how much "mass" must be carried to transform $\mathbb{P}_r$ into $\mathbb{P}_m$. The EM distance is then "the cost" of the optimal "transport".

TODO: here explain why the earth-mover is better than the the other probaility distances -> use https://www.alexirpan.com/2017/02/22/wasserstein-gan.html

Unfortunately the EM distance is intractable, due to the infinum part in its equation. But using the Kantorovich-Rubinstein duality, it can be reformulated as:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{||f||_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)] \tag{3}$$

Where the supremum is over 1-Lipschitz functions. It is important to note that if we replace $||f||_L \leq 1$ by $||f||_L \leq K$, i.e. consider also the K-Lipschitz functions, then we obtain $K \cdot W(\mathbb{P}_r, \mathbb{P}_\theta)$. Hence, for a family of functions $\{f_w\}_{w\in\mathcal{W}}$ (all functions being K-Lipschitz), we can consider solving the optimization problem:

$$\max_{w\in\mathcal{W}} \mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z\sim p(z)}[f_w(g_\theta(z))] \tag{4}$$

Note: we can approximate the solution of the above mentionned problem with a Neural Network with weights $\mathcal{W}$. $\mathcal{W}$ need to be compact to assure that the function f_w are K-Lipschitz. Therefore in order to have $\mathcal{W}$ being compact, Arjovsky et al. (2017) proposes to simply clip the weights, such that they lay in a small box $[-c, c]^l$, e.g. with $c = 0.01$

2.3.2 Necessary changes to turn a GAN into a WGAN

Implementation of a WGAN requires a few changes from implementation of a regular GAN, i.e.

- Use a linear activation function in the output layer of the "discriminator" model (instead of sigmoid). The "discriminator" then becomes a critic that quantify the realness of a sample, instead of discriminating between real or fake.

- Use Wasserstein loss to train the critic and generator models that promote larger difference between scores for real and generated images.

- Constrain critic model weights to a limited range after each mini batch update (e.g. [-0.01,0.01]). As seen above, this is to ensure, that the function estimated in the for approximting the Wasserstein distance are K-lipschitz, a necessary condition.

- Update the critic model more times than the generator each iteration (e.g. 5). Contrary as in a GAN, this is not a issue. Indeed, switching to the EM distance, allow for more stability when training the two networks in the WGAN. Moreover, the fact that the EM distance is continuous and differentiable means that we should train the critic until optimality.

- Use the RMSProp version of gradient descent with small learning rate and no momentum (e.g. 0.00005).

2.4 WGAN-GP

As we have seen above, WGAN improve the stability in training the critic ($\approx$ discriminator). But it is still subject to poor sample generation, convergence failure or mode collapse. This is due to the weight clipping happening while training the critic. In order to remedy to this, Gulrajani et al. (2017) propose to replace weight clipping by the introduction of a penalization of the norm of gradient of the critic with respect to its input. The issue is that trying to orient the critic to 1-Lipschitz function by weight clipping, biases the critic for too simple function. Gulrajani et al. (2017) observes that implementing the Lipschitz constraint using weights clipping leads to either exploding or vanishing gradient, unless the threshold c used for the clipping is carefully fine-tuned.

2.4.1 The gradient penalty

In order to enforce the Lipschitz constraint, Gulrajani et al. (2017) proposes to add a penalty term to the loss function. The loss function then becomes:

$$L = L' + P \tag{5}$$

where:

- Original loss function:
$$L' = \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[D(\mathbf{x})] \tag{6}$$

- Penalty:
$$P = \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}}[(||\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}) - 1||)^2] \tag{7}$$

The goal of the penalty is to enforce the 1-Lipschitz constraint. Indeed, by definition, a function is 1-Lipschitz if and only if its gradient norm is smaller or equal to 1 everywhere. It can be easily seen that the penalty is here to enforce this constraint. In order to make such a penalty tractable, a soft version of the penalty is considered, where the constraint is only enforced on a the gradient norm of a few random samples $\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}$.

The sampling distribution $\mathbb{P}_{\hat{\mathbf{x}}}$ is defined by sampling uniformly on on a line between a pair of points respectively sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$. This was proven experimentally to give sufficiently good results.

In Gulrajani et al. (2017), the penalty coefficient λ was set always to 10 in all experiences done. No batch normalization was used in Gulrajani et al. (2017). They claim that batch normalization shifts the discriminator problem from trying to match a single input to a single output from trying to macth a batch input to a batch output. This makes the penalty invalid, since the penalization is perfomed with each input individually and batch normalization introduce correlation between samples.Instead of batch normalization, Gulrajani et al. (2017) recomnends layer normalizations.

3 Our approach (TODO: title to change)

We decided that it made sense to try to generate the Cross Spectral Matrix (CSM), as done in Gerstoft et al. (2020) and extend his work to create a network to generate CSM, conditionally on Direction of Arrival (DoA). Indeed, such a network would allow us to have the online generation of labeled data required to train the network **TODO: find name of network for Source Characterization**. By providing a DoA (i.e. a label) to the network, we would generate the corresponding CSM data.

More specifically than the CSM, we thought it would make more sense to generate separately the eigenvalues and eigenvectors its eigendecomposition. A CSM $\hat{\mathbf{C}}$ can be decomposed as :

$$\hat{\mathbf{C}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H \tag{8}$$

where $\mathbf{V} = [\mathbf{v}_1^T, \ldots, \mathbf{v}_M^T] \in \mathbb{C}^{M \times M}$, $\mathbf{v}_i$ being the ith eigenvector and where $\mathbf{\Lambda} \in \mathbb{R}^{M \times M}$ is a diagonal matrix, where λ_{ii} is the ith eigenvalue, corresponding to the ith eigenvector.

Indeed, since we choose a Generative Adversial Approach, the data will be generated using two networks: a generator and a discriminator/critic. Those two network are competing against each other: the goal of the generator is to produce data realistic enough so that discriminator can not tell it is fake. The goal of the discriminator is to tell whether a given input is real or has been generated. Both the generator and the discriminator have to be trained simultaneously until convergence. A typical issue occuring during the training, is that the discriminator becomes too good at discerning real from fake sample and hence the generator does not improve anymore.

Generating the eigenvalues and eigenvectors instead of the CSM is done in order to help the generator. This allow to normalize all the eigenvectors before feeding them to the discriminator, whether they are real or generated. The eigenvalues can also be scaled the biggest of them is equal to one. **TODO: develop on that**

3.1 Generation of Eigenvalues

3.1.1 WGAN-GP

-> Implementation from this repo https://github.com/henry32144/wgan-gp-tensorflow/blob/master/WGAN-GP-celeb64.ipynb

3.1.2 Networks Architecture

3.1.3 Performance

- Comparison between Generated and Real - Loss Function

3.2 Generation of Eigenvectors

-> Implementation from this repo https://github.com/henry32144/wgan-gp-tensorflow/blob/master/WGAN-GP-celeb64.ipynb

3.2.1 WGAN-GP

3.3 Data Augmentation

3.4 Generation of Cross-Correlation Matrix

References

Sharath Adavanne, Archontis Politis, and Tuomas Virtanen. Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1462–1466. IEEE, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

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.

Paolo Castellini, Nicola Giulietti, Nicola Falcionelli, Aldo Franco Dragoni, and Paolo Chiariotti. A neural network based microphone array approach to grid-less noise source localization. *Applied Acoustics*, 177:107947, 2021.

Soumitro Chakrabarty and Emanuël AP Habets. Broadband doa estimation using convolutional neural networks trained with noise signals. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 136–140. IEEE, 2017.

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.

Eric L Ferguson, Stefan B Williams, and Craig T Jin. Sound source localization in a multipath environment using convolutional neural networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2386–2390. IEEE, 2018.

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.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017. URL `http://arxiv.org/abs/1704.00028`.

Weipeng He, Petr Motlicek, and Jean-Marc Odobez. Deep neural networks for multiple speaker detection and localization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 74–79. IEEE, 2018.

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.

Adam Kujawski, Gert Herold, and Ennes Sarradj. A deep learning method for grid-free localization and quantification of sound sources. *The Journal of the Acoustical Society of America*, 146(3): EL225–EL231, 2019.

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. URL `https://proceedings.neurips.cc/paper/2019/file/6804c9bca0a615bdb9374d00a9fcba59-Paper.pdf`.

Soo Young Lee, Jiho Chang, and Seungchul Lee. Deep learning-based method for multiple sound source localization with high resolution and accuracy. *Mechanical Systems and Signal Processing*, 161:107959, 2021.

Wei Ma and Xun Liu. Phased microphone array for sound source localization with deep learning. *Aerospace Systems*, 2(2):71–81, 2019.

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

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

Lauréline Perotin, Romain Serizel, Emmanuel Vincent, and Alexandre Guérin. Crnn-based joint azimuth and elevation localization with the ambisonics intensity vector. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 241–245. IEEE, 2018.

Wagner Gonçalves Pinto, Michaël Bauerheim, and Hélène Parisot-Dupuis. Deconvoluting acoustic beamforming maps with a deep neural network. 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

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.

Ryu Takeda and Kazunori Komatani. Sound source localization based on deep neural networks with directional activate function exploiting phase information. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 405–409. IEEE, 2016.

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.

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

Pengwei Xu, Elias JG Arcondoulis, and Yu Liu. Deep neural network models for acoustic source localization. In *Berlin Beamforming Conference*, 2021.