Relative Transfer Function Inverse Regression from Low Dimensional Manifold

Ziteng Wang, Emmanuel Vincent, Yonghong Yan

***Abstract*—In room acoustic environments, the Relative Transfer Functions (RTFs) are controlled by few underlying modes of variability. Accordingly, they are confined to a low-dimensional manifold. In this letter, we investigate a RTF inverse regression problem, the task of which is to generate the high-dimensional responses from their low-dimensional representations. The problem is addressed from a pure data-driven perspective and a supervised Deep Neural Network (DNN) model is applied to learn a mapping from the source-receiver *poses* (positions and orientations) to the frequency domain RTF vectors. The experiments demonstrate the difficulty of this task: the model provides better prior knowledge of the RTF than the free field assumption, however, it fails to compete with the linear interpolation technique in small distances.**

***Index Terms*—relative transfer function, inverse regression, deep neural network, manifold learning**

I. Introduction

THE acoustic properties of a room environment can be fully characterized by the collection of the Acoustic Impulse Responses (AIRs). An AIR relates two arbitrary poses inside the room: one for the source and the other for the receiver. In the multichannel setup, where the receiver includes multiple microphones, the Relative Transfer Function (RTF) [1] is more often used. The estimation of the RTF is essential in many applications, such as sound field reproduction [2], dereverberation [3], source localization [4] and source separation [5]. Over the years, the RTF estimation is based merely on the measured signals [6]–[8]. While the prior knowledge of the RTF given the source-receiver pose could provide additional performance benefits [9], it remains less studied.

The RTF involves two AIRs, the knowledge of which usually relies on explicit physical models. For instance, the classical image-source method [10] is widely used for simulating small room acoustics. The method assumes the response to be contributed by multiple virtual image sources, and the simulation requires the room to be rectangular and the wall reflection coefficients known. If the room geometry can be estimated in practice, an approximation of the AIR can be obtained accordingly [11], [12]. Considering a predefined pose space only, the AIRs can be alternatively parameterized based on the harmonic solution to the wave equation [13]. Moreover, in stationary rooms, AIR measurements can be collected in advance and help the modeling. With a limited number of measurements, compressed sensing with sparsity is introduced to interpolate AIR for the early parts [14] or in the low frequency bins [15]. When dense sampling is achievable, the simple data-based linear interpolation turns out to be an effective approach [16], [17].

In some common scenarios, e.g., in conference rooms or cars, the stationary assumption made above is reasonable since the layout does not often change over time. Indeed, the AIRs in this case are confined to a low-dimensional manifold because the source-receiver pose now works as the only varying degree of freedom. This geometrical property was revealed by the manifold learning paradigm [18] that was first introduced to parameterize linear systems. It was then adopted to RTF modeling [19], [20] and applied in supervised [21] and semi-supervised source localization tasks [22], [23], which was to associate the manifold with the source poses. The RTF encodes the Interchannel Level Difference (ILD) in decibels and the Interchannel Phase Difference (IPD) in radians. The low-dimensional structure of the ILD and IPD was shown in a binaural setup [24]. Based on a local linearity assumption on the manifold, the method of Probabilistic Piecewise Affine Mapping (PPAM) was proposed for a 2D sound localization task. Specially, PPAM learned a bijective mapping between the poses and the ILD and IPD, whereas only the interaural-to-pose regression was discussed for localization. The bijective mapping has also been generalized to the cases of multiple sources [25], co-localization of audio sources in images [26] and partially latent response variables [27].

In this letter, we raise the problem of RTF inverse regression, which is defined as approximating the high-dimensional responses from their low-dimensional representations. The question is: how to acquire prior knowledge of the RTF given the source-receiver pose? As deriving an explicit physical model could not always be possible but representative examples of the acoustic environment can be collected in advance, especially with modern smart devices, the problem is addressed from a pure data-driven perspective. A Deep Neural Network (DNN) model is proposed to directly generate the frequency domain RTF vector given the source-receiver pose. The DNN model is advantageous in that it learns a globally nonlinear mapping from the data examples while preserving a local linearity, which matches the manifold structure implicitly. In the experiments, the acoustic space is sampled uniformly and the DNN model is tested in unseen poses. The evaluation is based on an absolute prediction error measure, while applying the generated RTFs to specific applications is not in the scope here.

The rest of this letter is organized as follows. The relevant definitions are given in Section II. The RTF inverse regression task and the possible solutions including the supervised DNN model are explained in Section III. The experimental setups and results are presented in Section IV and conclusions are drawn in Section V.

II. Definitions

In a reverberant room environment, the RTFs represent the coupling between a pair of microphones in response to the source signal. Denote the source as s and the two AIRs from the source to the microphones as h_1, h_2, the observations at time t are written as

$$a_m(t) = h_m(t) * s(t) + v_m(t) \qquad m = 1, 2 \tag{1}$$

where $*$ denotes convolution and v_m is the noise component in the mth microphone. Under the narrowband approximation, the signals in the frequency domain are given by

$$A_m(l, f) = H_m(f)S(l, f) + V_m(l, f) \tag{2}$$

where l is the frame index, f is the frequency index, A_m, S and V_m are the Short Time Fourier Transform (STFT) coefficients of a_m, s and v_m, respectively, and H_m is the Fourier transform of h_m. The RTF is defined as

$$H(f) = \frac{H_2(f)}{H_1(f)}. \tag{3}$$

The RTFs are known to be governed by these parameters: the size and geometry of the room, the reflection coefficients of the walls, and the source-receiver poses. Assuming the room characteristics to be stationary over time, the RTFs are thus confined to a low-dimensional manifold that can be associated with the source-receiver poses as

$$H(f) = g(\Theta_s, \Theta_r) \tag{4}$$

where $g : \mathbb{R}^L \rightarrow \mathbb{R}^D$ is defined as the mapping function and Θ_s, Θ_r are the pose parameters in the intrinsic Cartesian coordinate system on the embedded manifold. The pose parameters include the position coordinates (x, y, z) and the direction variables (azimuth angle, elevation angle and rotation angle).

In anechoic conditions, the translation from the source-receiver pose to the RTF is straightforward under the free field assumption, which is given by the direct sounds with (3) and

$$H_m(f) = \frac{\exp(j \cdot 2\pi f ||\Theta_{r_m} - \Theta_s||_2 / c)}{4\pi ||\Theta_{r_m} - \Theta_s||_2} \tag{5}$$

where j is the complex unit, $|| \cdot ||_2$ denotes the Euclidian norm and c denotes the speed of sound [13]. However, the association becomes complex in reverberant conditions.

III. RTF inverse regression

The task of RTF inverse regression is to learn the mapping function g from a given set of pairwise examples $\{\mathcal{X} : \Theta_s, \Theta_r;\ \mathcal{Y} : H(f)\}$, while no physical constraint is involved. Three possible solutions are discussed in the following.

Linear interpolation is the intuitive way. Provided that $\mathcal{X}, \mathcal{Y}$ have the same geometric structure in their separate space, the response of a new pose $\widehat{\mathcal{X}}$ can be estimated as

$$\widehat{\mathcal{Y}} = \frac{1}{\sum \alpha_i} \sum_{i=1}^{I} \alpha_i \mathcal{Y}_i \tag{6}$$

where $\mathcal{Y}_i$ is the response of the neighboring pose $\mathcal{X}_i$ and α_i is the weighting parameter correlated to the spatial distance from $\mathcal{X}_i$ to $\widehat{\mathcal{X}}$.

Although PPAM was not proposed for this task, an instantiation of g was realized based on a piecewise linear approximation [25]. The acoustic space is divided into K local regions and each is characterized by the affine transform:

$$\mathcal{Y} = \sum_{k=1}^{K} \mathbb{I}(k)(\mathbf{A}_k \mathcal{X} + \mathbf{b}_k) + \epsilon \tag{7}$$

where $\mathbb{I}(k) = 1$ if $\mathcal{X}$ lies in the kth local region, and 0 otherwise. $\mathbf{A}_k \subset \mathbb{R}^{L \times D}$ is the weight matrix, $\mathbf{b_k}$ is the bias vector and ϵ is the error term described by the Gaussian distribution.

DNNs feature multiple hidden layers trained through error backpropagation. Theoretically, the expression in (7) can be approximated by a neural network with one hidden layer [28], which can be described as

$$\mathcal{Y} = \mathbf{W}^{(1)} \zeta(\mathbf{W}^{(0)} \mathcal{X} + \mathbf{b}^{(0)}) + \mathbf{b}^{(1)} \tag{8}$$

where $\mathbf{W}^{(i)}, \mathbf{b}^{(i)}$ are the ith layer parameters and $\zeta(\cdot)$ is a nonlinear activation function. More details about the proposed DNN model are given as below.

A. Inputs and targets

The inputs of the DNN model are the pose parameters $\{\Theta_s, \Theta_r\} \subset \mathbb{R}^L$, the effective dimension of which is dependent on the degree of freedom in the system, since fixed input values should make no difference to the performance.

The target RTFs are complex vectors, which are represented by the real-valued ILD and IPD as

$$\text{ILD} = 20\log_{10}|H(f)| \tag{9}$$

$$\text{IPD} = \arg(H(f)). \tag{10}$$

The sine and cosine of the IPD are computed and concatenated with the ILD to form the final target vector for training, which is of $D = 513 \times 3$ dimensions as STFT is performed in 1024 points. The setup follows that in [25], nevertheless, alternative targets could be the real and imaginary parts of the RTF.

Since data normalization is known to help the training, a normalized target RTF is considered as

$$\overline{H(f)} = \frac{H(f)}{H_d(f)} \tag{11}$$

where $H_d(f)$ is calculated using (3) and (5). This could provide marginal benefit in the latter experiments.

B. DNN Architecture and training

The DNN model is a basic feed-forward neural network with all the layers linearly connected. The ReLU activation function is used for the hidden layers and linear activation is used for the output layer. Given the previous target selection, a local normalization is enforced to the IPD output nodes:

$$o_{f,-} = \frac{o_{f,-}}{\sqrt{o_{f,s}^2 + o_{f,c}^2}} \tag{12}$$

which means that the sum of squares of the sine part $o_{f,s}$ and the cosine part $o_{f,c}$ in the fth frequency bin should equal to one.

In the training stage, the layer weights are initialized with Gaussian samples (zero mean and deviation $\sqrt{1/\text{in\_size}}$) and the bias vectors are initialized with zeros. The model is optimized under the Mean Squared Error (MSE) loss criterion. The Adam method [29] is used to update the model parameters and the learning rate is adjusted adaptively. Layer normalization [30] is applied to the hidden layers to speed up the model convergence. Other regularization techniques such as dropout and batch normalization, are not found helpful here.

C. Evaluation metric

As far as we know, there is no established measure to evaluate the approximation error of a high-dimensional vector. A mean absolute error metric is chosen here to straightly show the performance in each frequency:

$$\mu_f = \frac{1}{N}\sum_{n=1}^{N} ||\widehat{\mathcal{Y}}_n(f) - \mathcal{Y}_n(f)||_1 \tag{13}$$

where $\widehat{\mathcal{Y}}_n$ is the predicted response of the nth test sample. The 95% Confidence Interval (CI) of μ_f is also calculated to give an idea of the error distribution. The ILD and IPD prediction errors are treated separately.

IV. Experiments and results

A. Experimental setup validation

In the first place, we validate the experimental setup before reporting further results on simulated data. The CAMIL dataset[1] is considered for this task as it includes real-world recordings labeled with true poses. A simulated dataset is first created following its original setup and the models are then tested on both the real and simulated data. We consider the experimental setup is validated if similar trends are observed in the performance. Note that the CAMIL dataset consists of binaural recordings that involves Head-Related Transfer Functions (HRTFs) and changes the definition of RTF inverse regression slightly. Nevertheless, this dataset is to our knowledge the only qualified public dataset, and the previous analysis can be generalized to this case.

The simulation setup is illustrated in Fig. 1(a). The room size is 4×6×3 m and the reverberation time is set to be 300 ms. A pair of microphones with cardioid directivity are used to imitate a dummy-head, while the AIRs (simulated using the image-source method [10]) are convolved with real measured HRTFs[2]. The receiver is positioned at [2, 1, 1.4], with microphone distance 0.18 m, and the source is fixed at 3 meters away in the front. The receiver pose is defined by an azimuth angle in the range of $[-160°, 160°]$ and an elevation angle in the range of $[-60°, 60°]$. Data samples are generated every $2°$ and there are 9600 poses in total. For each pose, a one-second white noise signal is emitted from the source and then the captured microphone signals are used to calculate the corresponding RTF [25]. This setup is one way to measure the RTF in practise, though it can be directly computed from the two AIRs with (3) in the simulation.

[1]https://team.inria.fr/perception/the-camil-dataset/

[2]https://dev.qu.tu-berlin.de/projects/measurements

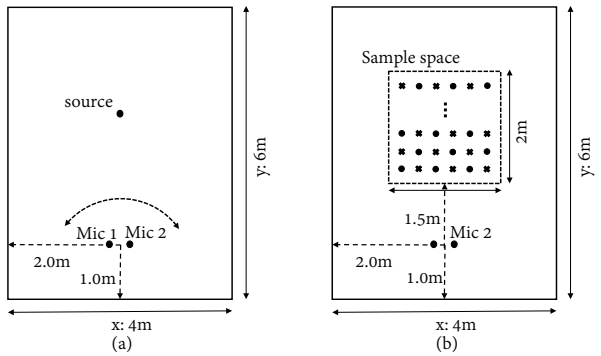

Fig. 1. Top view of the simulation setup: (a) simulated CAMIL dataset, (b) uniform acoustic space sampling. One every two samples (marked as dots in the sample space 2×2×1 m) is used in the training.

For the DNN model, 3 hidden layers each with 1024 nodes are used throughout based on heuristic search. One every two samples in the dataset is used for training. Half of the left is used for development and the other half for testing. Early stopping is applied to avoid overfitting when the development set error no longer decreases after a patience of 5 epoches. The method of PPAM [25] (source code provided therein) is investigated and the parameter K is chosen from {64, 128, 256} to achieve the best performance. It is worth to note again that PPAM was not proposed for the task here. Linear interpolation is also considered, but in a simplified way:

$$\widehat{\mathcal{Y}}_n(f) = [\mathcal{Y}_{n,1}(f) + \mathcal{Y}_{n,2}(f)]/2 \tag{14}$$

where $\mathcal{Y}_{n,1}$ and $\mathcal{Y}_{n,2}$ are the response vectors of the spatially adjacent poses. The interpolated IPD is normalized as in (12).

The results on the real and simulated CAMIL datasets are given in Table I. The mean values μ_f are further averaged over the frequencies. Similar trends are observed: DNN achieves lower prediction errors than PPAM in ILD but slightly higher errors in IPD, and both methods fail to compete with linear interpolation. This should validate our simulation setup. The reason that DNN and PPAM perform relatively better on the simulated data could be due to the simplification of the simulation setup.

TABLE I
ILD/IPD prediction errors on the real (left part) and simulated (right part) CAMIL datasets.(mean±CI bound)

	ILD	IPD	ILD	IPD
PPAM	1.73±.057	0.33±.015	1.42±.066	0.27±.013
DNN	1.42±.047	0.34±.014	1.23±.042	0.29±.013
Linear	1.03±.035	0.18±.011	1.07±.034	0.20±.011

The results also accord with the finding that the acoustic responses are locally linear in small distances [20]. Meanwhile, linear interpolation has the largest parameter size here, which is $N(D + L)$ with N being the size of the training examples. To show that the DNN model implicitly learns the RTF manifold structure, the target ILDs and the DNN generated ones are visualized in the low-dimensional space in Fig. 2. The manifolds resemble each other in the sense that the samples are clearly organized according to the poses.

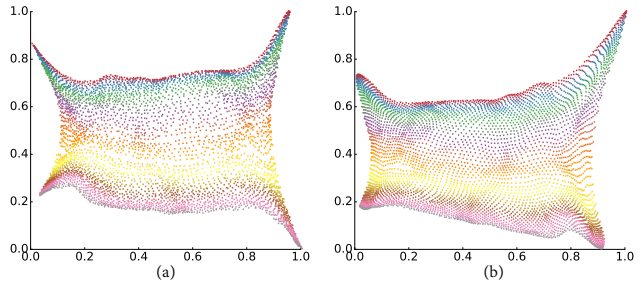

Fig. 2. 2D visualization of the manifold using local tangent space alignment in the scikit-learn toolbox: (a) target ILDs, (b) DNN generated ILDs. Samples with the same elevation angle have the same color.

B. Performance on simulated data

In the rest of the experiments, a more general setup is considered, e.g., in conference rooms, where the microphone positions are fixed and the source moves around in limited space. Following the previous setup, the receiver pose is fixed and the source pose has three degrees of freedom while HRTF is no longer used. The acoustic space is sampled uniformly as shown in Fig. 1(b). Dense data samples are generated every 1 cm and there are in total $200\times100\times100$ poses in the training set. 10,000 extra poses are randomly chosen for the evaluation.

The performance results are given in Table II. The model is evaluated w.r.t. sample distances $\{1,\ 2,\ 4,\ 8\}$ cm in the training set and larger sample distance also means less training data. The prediction errors clearly go up with larger sample distances. This is more obvious for linear interpolation (using (14) with interpolation samples along the z coordinate which reports lower errors than along the x, y coordinates), the mean errors in ILD and IPD are more than double in the 2 cm case than that in the 1 cm case. DNN loses to linear interpolation again in small distances but slightly surpasses it starting from 4 cm. Note that the local linearity of the RTF manifold under the Euclidean distance measure [20] holds within around 3.5 cm in this case. Here the performance of using $H_d(f)$, that is the prior knowledge of the RTF responses we can have from the free field assumption, to approximate the targets is given by: ILD, 3.53±0.053 and IPD, 0.62 ± 0.010.

TABLE II
ILD/IPD PREDICTION ERRORS OF DNN AND LINEAR INTERPOLATION W.R.T. SAMPLE DISTANCE. (MEAN±CI BOUND)

	1cm	2cm	4cm	8cm
DNN-ILD	3.01±.044	3.03±.046	3.10±.047	3.23±.049
Linear-ILD	0.92±.017	2.21±.036	3.18±.049	3.43±.052
DNN-IPD	0.46±.008	0.48±.009	0.50±.009	0.54±.009
Linear-IPD	0.17±.005	0.38±.008	0.53±.009	0.57±.010

The mean errors in different frequencies are plotted in Fig. 3. The errors in the low frequencies are relatively smaller, especially below the spatial aliasing frequency f_a as shown by $H_d(f)$ (direct) and the DNN model. In the high frequencies, the target values vary more rapidly on the manifold and the variations become harder to be captured by linearity. It is shown that DNN outperforms linear interpolation (LI) only in the high frequencies.

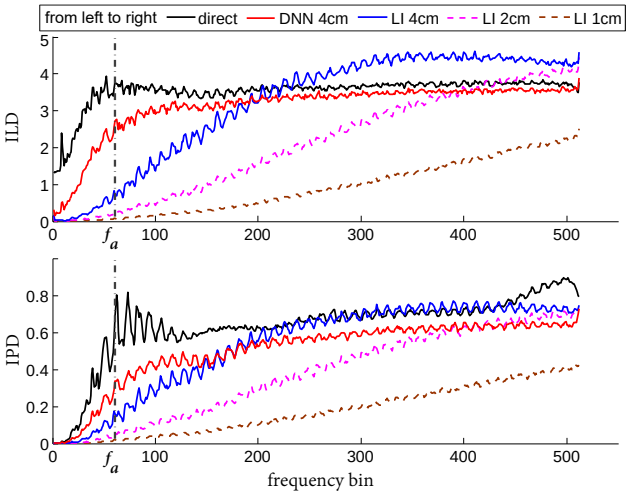

Fig. 3. The mean errors in ILD and IPD along the frequency axis. f_a marks the spatial aliasing frequency.

TABLE III
ILD/IPD PREDICTION ERRORS W.R.T. SNR. (MEAN±CI BOUND)

	30dB	20dB	10dB
DNN-ILD	3.03±.046	3.03±.046	3.05±.046
Linear-ILD	2.31±.036	2.32±.036	2.36±.037
DNN-IPD	0.48±.009	0.48±.009	0.48±.009
Linear-IPD	0.40±.008	0.40±.008	0.41±.008

Considering the usage of white noise source signals in the RTF measurement process, there exist measurement errors: ILD 0.93±0.027, and IPD 0.14±0.004, which are given by running 2000 separate simulations for the same pose. The mean errors are close to that of the 1 cm case, which is one reason that no denser sampling is considered. The ambient noise is also considered as another factor and diffuse noise is manually added to the source signals in the training data at Signal-to-Noise Ratios (SNRs) $\{30,\ 20,\ 10\}$ dB. The results for the 2 cm sample distance are given in Table III. The predictions errors differ slightly in the three cases, which means that both methods are quite robust to noise.

V. Conclusion

The prior knowledge of the RTFs is favorable in many applications but it is less studied before. In this letter, we raised the RTF inverse regression problem and addressed it in the simplified stationary environments. A DNN model was trained to directly generate the high-dimensional acoustic responses given their low-dimensional pose representations. The DNN model implicitly captured the RTF manifold and performed better than the free field model and better than linear interpolation in large distances. The superior performance of linear interpolation in small distances also supported the local linearity property of the RTFs on the manifold.

Acknowledgment

The authors would like to thank Antoine Deleforge and Sharon Gannot for their helpful discussions and comments.

REFERENCES

[1] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1614–1626, 2001.

[2] T. Betlehem and T. D. Abhayapala, "Theory and design of sound field reproduction in reverberant rooms," *The Journal of the Acoustical Society of America*, vol. 117, no. 4, pp. 2100–2111, 2005.

[3] Y. Lin, J. Chen, Y. Kim, and D. D. Lee, "Blind channel identification for speech dereverberation using l1-norm sparse learning." in *NIPS*, 2007, pp. 921–928.

[4] X. Li, L. Girin, R. Horaud, and S. Gannot, "Estimation of relative transfer function in the presence of stationary noise based on segmental power spectral density matrix subtraction," in *IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 320–324.

[5] Z. Koldovský, J. Málek, and S. Gannot, "Spatial source subtraction based on incomplete measurements of relative transfer function," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 8, pp. 1335–1347, 2015.

[6] I. Cohen, "Relative transfer function identification using speech signals," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 451–459, 2004.

[7] R. Talmon, I. Cohen, and S. Gannot, "Relative transfer function identification using convolutive transfer function approximation," *IEEE Transactions on audio, speech, and language processing*, vol. 17, no. 4, pp. 546–555, 2009.

[8] M. Taseska and E. A. Habets, "Relative transfer function estimation exploiting instantaneous signals and the signal subspace," in *23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 404–408.

[9] R. Talmon and S. Gannot, "Relative transfer function identification on manifolds for supervised GSC beamformers," in *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*. IEEE, 2013, pp. 1–5.

[10] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[11] A. Asaei, M. E. Davies, H. Bourlard, and V. Cevher, "Computational methods for structured sparse component analysis of convolutive speech mixtures," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 2425–2428.

[12] A. Asaei, M. Golbabaee, H. Bourlard, and V. Cevher, "Structured sparsity models for reverberant speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 3, pp. 620–633, 2014.

[13] P. Samarasinghe, T. Abhayapala, M. Poletti, and T. Betlehem, "An efficient parameterization of the room transfer function," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 12, pp. 2217–2227, 2015.

[14] R. Mignot, L. Daudet, and F. Ollivier, "Room reverberation reconstruction: Interpolation of the early part using compressed sensing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 11, pp. 2301–2312, 2013.

[15] R. Mignot, G. Chardon, and L. Daudet, "Low frequency interpolation of room impulse responses using compressed sensing," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 1, pp. 205–216, 2014.

[16] T. Nishino, S. Kajita, K. Takeda, and F. Itakura, "Interpolating head related transfer functions in the median plane," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 1999, pp. 167–170.

[17] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 126–130.

[18] R. Talmon, D. Kushnir, R. R. Coifman, I. Cohen, and S. Gannot, "Parametrization of linear systems using diffusion kernels," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1159–1173, 2012.

[19] B. Laufer, R. Talmon, and S. Gannot, "Relative transfer function modeling for supervised source localization," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2013, pp. 1–4.

[20] B. Laufer-Goldshtein, R. Talmon, and S. Gannot, "A study on manifolds of acoustic responses," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 203–210.

[21] R. Talmon, I. Cohen, and S. Gannot, "Supervised source localization using diffusion kernels," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2011, pp. 245–248.

[22] B. Laufer, R. Talmon, and S. Gannot, "Semi-supervised sound source localization based on manifold regularization," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 8, pp. 1393–1407, 2016.

[23] B. Laufer-Goldshtein, R. Talmon, and S. Gannot, "Semi-supervised source localization on multiple-manifolds with distributed microphones," *arXiv preprint arXiv:1610.04770*, 2016.

[24] A. Deleforge and R. Horaud, "2D sound-source localization on the binaural manifold," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2012, pp. 1–6.

[25] A. Deleforge, F. Forbes, and R. Horaud, "Acoustic space learning for sound-source separation and localization on binaural manifolds," *International Journal of Neural Systems*, vol. 25, no. 01, 2015.

[26] A. Deleforge, R. Horaud, Y. Y. Schechner, and L. Girin, "Co-localization of audio sources in images using binaural features and locally-linear regression," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 4, pp. 718–731, 2015.

[27] A. Deleforge, F. Forbes, and R. Horaud, "High-dimensional regression with Gaussian mixtures and partially-latent response variables," *Statistics and Computing*, vol. 25, no. 5, pp. 893–911, 2015.

[28] A. Pinkus, "Approximation theory of the mlp model in neural networks," *Acta Numerica*, vol. 8, pp. 143–195, 1999.

[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.