Learning Multiple Sound Source 2D Localization

Guillaume Le Moing[1,2], Phongtharin Vinayavekhin[1], Tadanobu Inoue[1],
Jayakorn Vongkulbhisal[1], Asim Munawar[1], Ryuki Tachibana[1], and Don Joven Agravante[1]

[1]IBM Research, Tokyo, Japan
[2]MINES ParisTech - PSL Research University, Paris, France; work performed during internship at IBM Research Tokyo

Introduction

Sound Source Localization

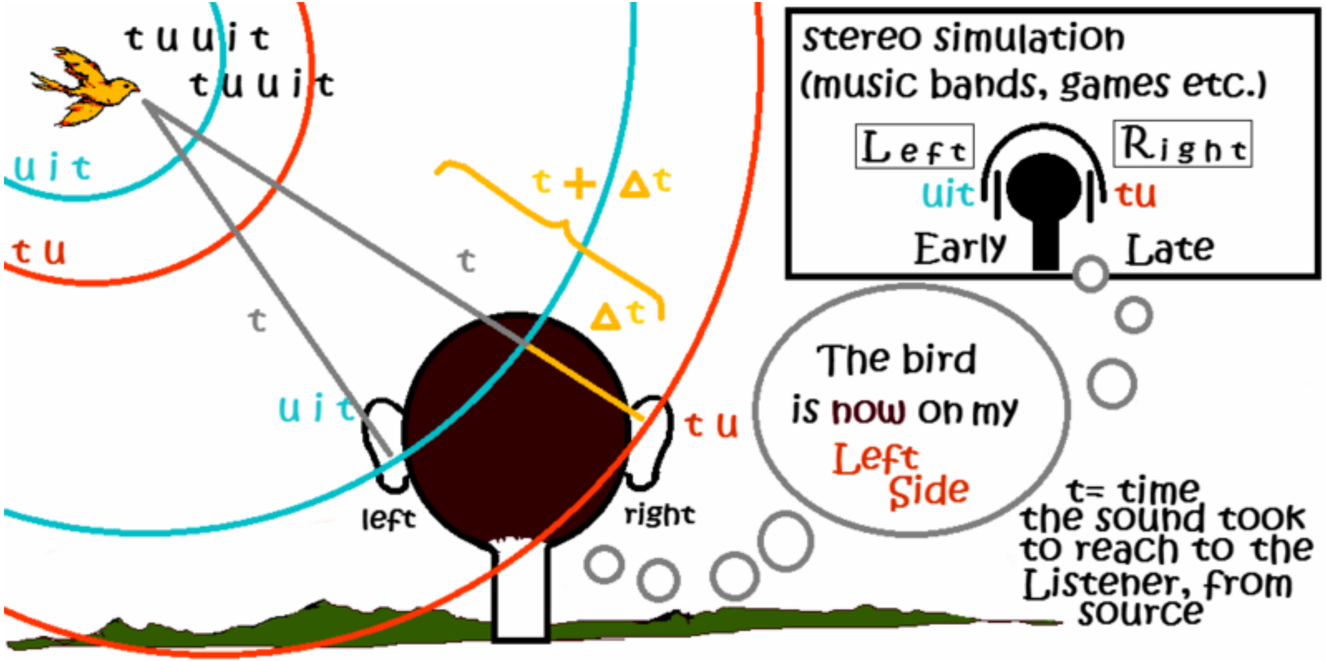
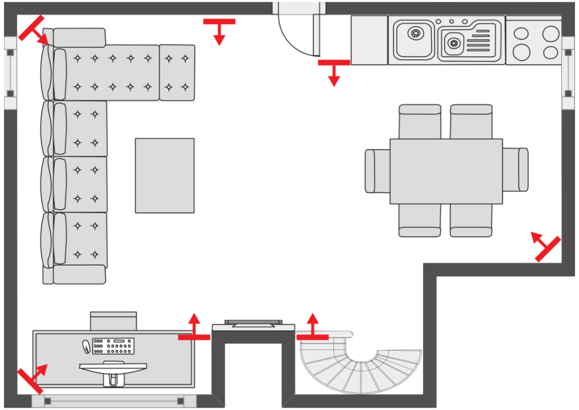


Image from Wikimedia commons:
https://commons.wikimedia.org/wiki/File:What_is_stereophonic_effect.png#filelinks

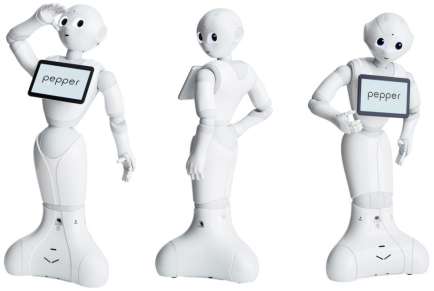
Application Areas

Healthcare, Speech Enhancement, Human-Robot Interaction etc.

DCASE 2018: task monitoring in domestic activities

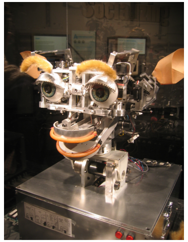


Pepper; the semi-humanoid robot



Smart Speakers and other IoT devices

Social Robot (kismet, jibo)

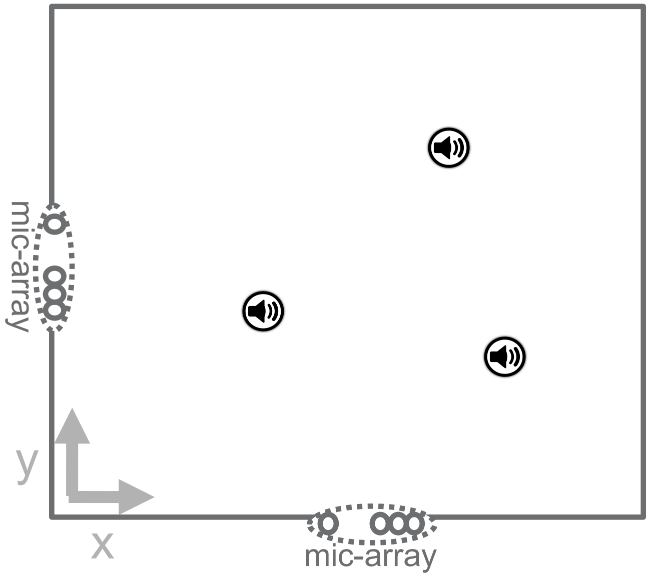


Problem Definition

Multiple Sound Source 2D Localization

Given:
- Sound from two or more microphone arrays
- Multiple sound sources

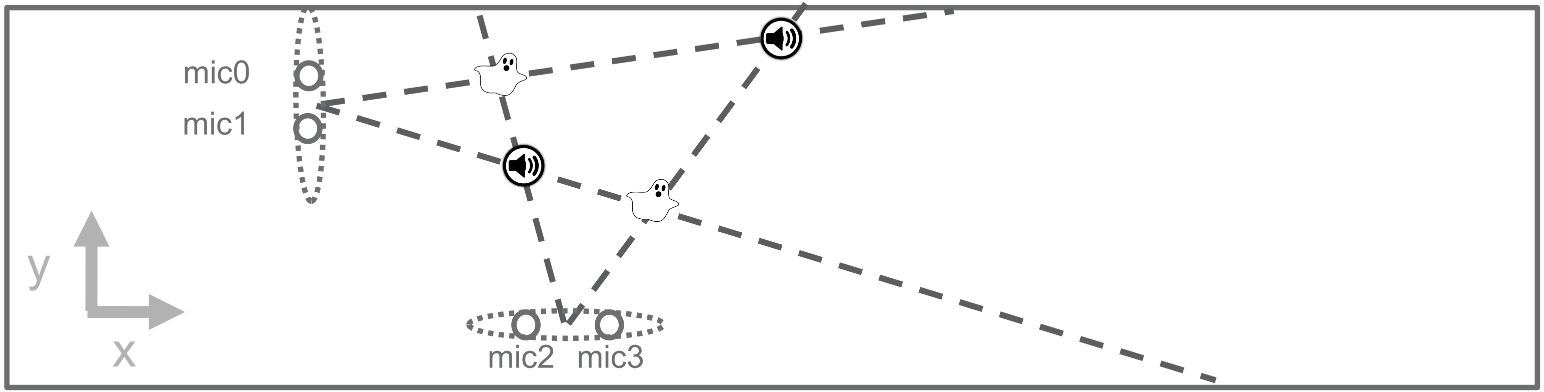
Results:
- 2D coordinates in an horizontal plane (x,y) for all sound sources.



Classical Methods on 2D Localization

Combining Direction of Arrivals (DOAs) to obtain 2D position

– *Association ambiguity problem* *[5, 6]*

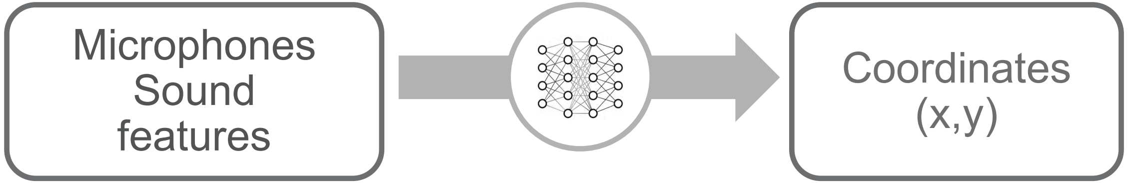


DOA COMBINATION

[5] Wing-Kin et al., “Tracking an unknown time-varying number of speakers using tdoa measurements: a random finite set approach,” IEEE Transactions on Signal Processing (2006).
[6] Alexandridis and Mouchtaris, “Multiple sound source location estimation in wireless acoustic sensor networks using doa estimates: The data-association problem,” IEEE/ACM Trans. Audio, Speech and Lang. Proc. (2018).

Our Approach

Data-driven based approach; specifically *deep-learning*.



✓ Solve association ambiguity implicitly

✓ Map sound features directly to positions

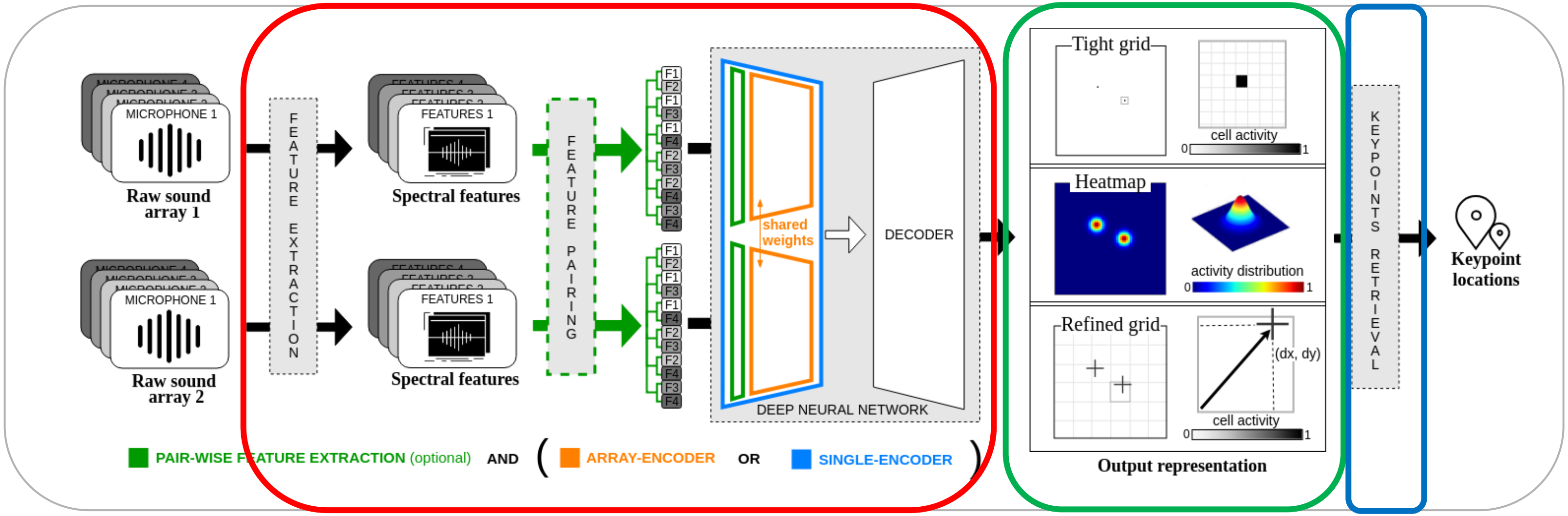
✓ Adapt to difficult acoustic conditions

✗ Need data to train

✗ Data specific to a microphone configuration

Proposed Method

Learning Multiple Sound Source 2D Localization

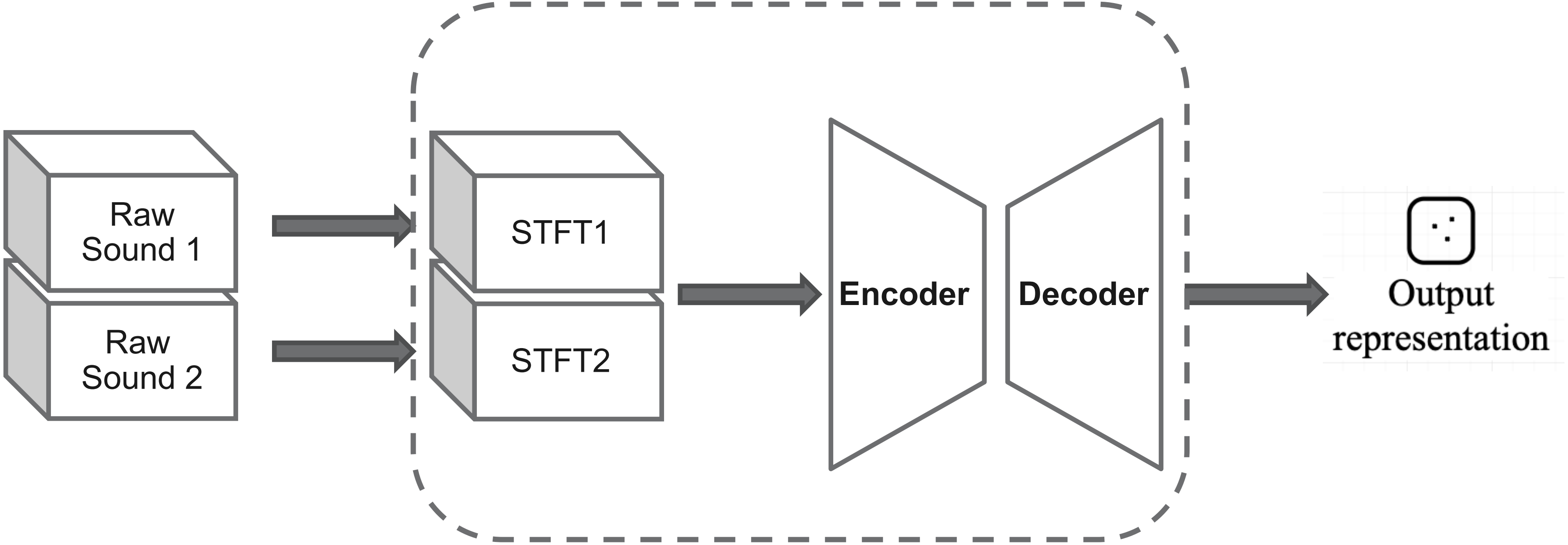


Input selection
Neural Network Architectures

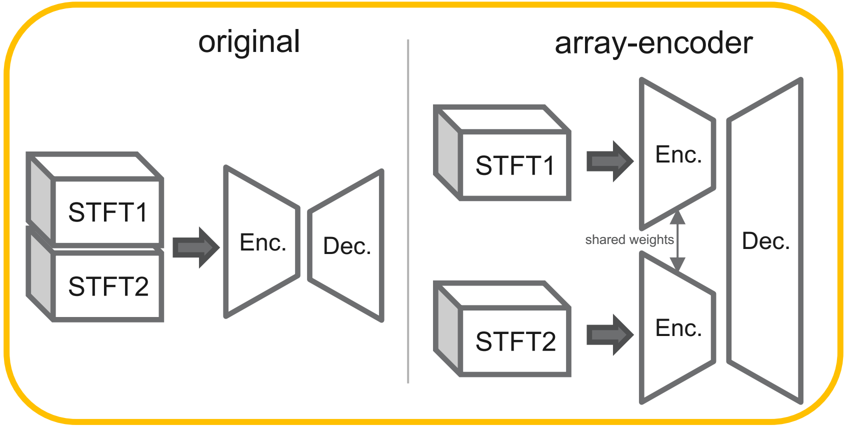
Output Representation
Loss function

Post Processing

Input Selection and Neural Network Architecture



Proposed Neural Network Architecture

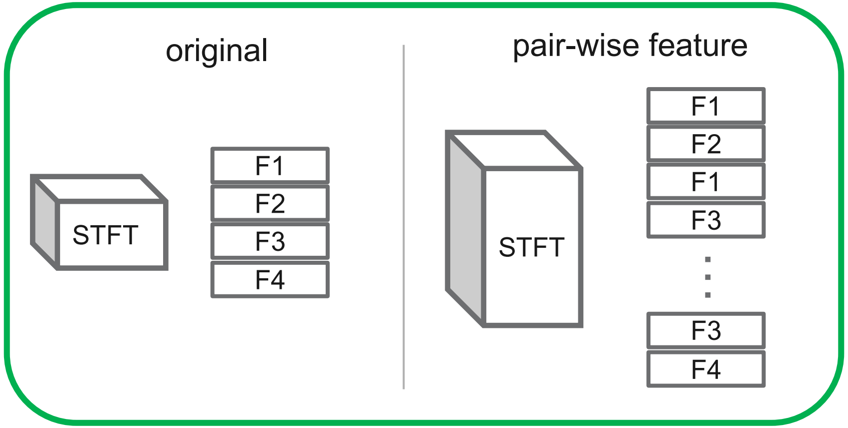


Array-Encoder

- Learn features with-in same mic-array..
- Shared data between multiple encoders.

Pair-wise Feature

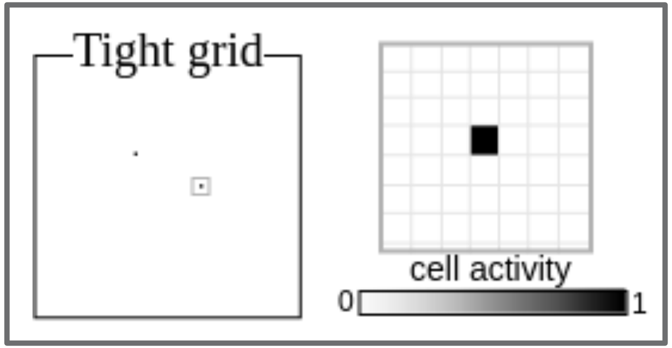
- Learn features from every mic pairs.



<u>Help network to generalize better</u>

Output Representation and Loss Function

Representing 2D coordinates (x, y) for multiple sound sources.



Representation

- M x N grid
- Active/Inactive cell

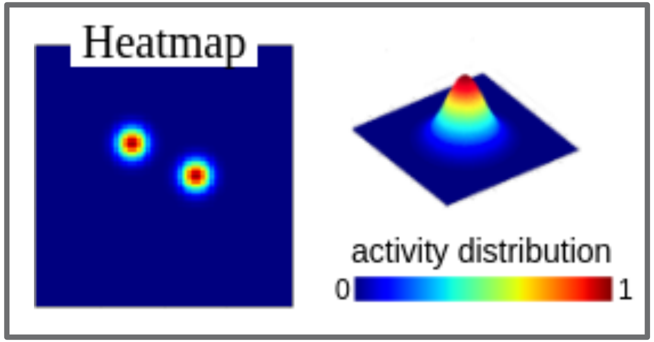
Loss Function

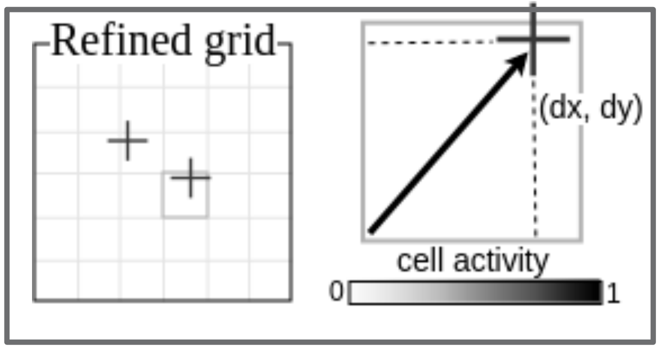
- Binary Cross Entropy (BCE)

Issue: a detailed grid (M and N) is required for accurate localization

→ difficult to train due to an imbalance of # of active/inactive cells.

Proposed Output Representation





Representation

- M x N grid
- Probability distribution

Loss Function

- Mean Squared Error (MSE)

Representation

- M x N grid
- Active/Inactive cell + Relative Location

Loss Function

- BCE + MSE

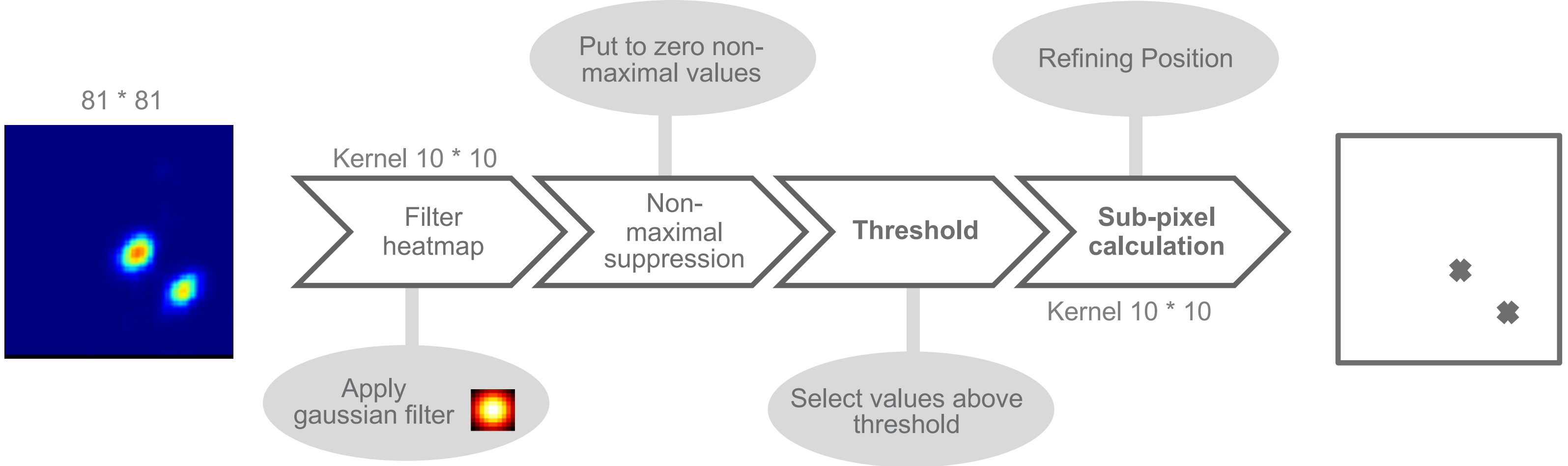
Post Processing : Keypoint Retrieval

Converting *Output Representation → Sound Source Locations (x, y); Keypoints*

Tight grid and Refined grid

– Non-maximum suppression (NMS) and Thresholding

Heat map



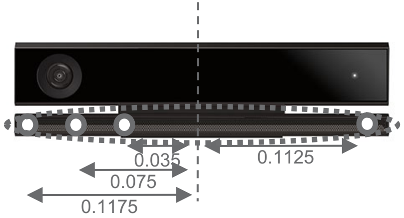
Experiment

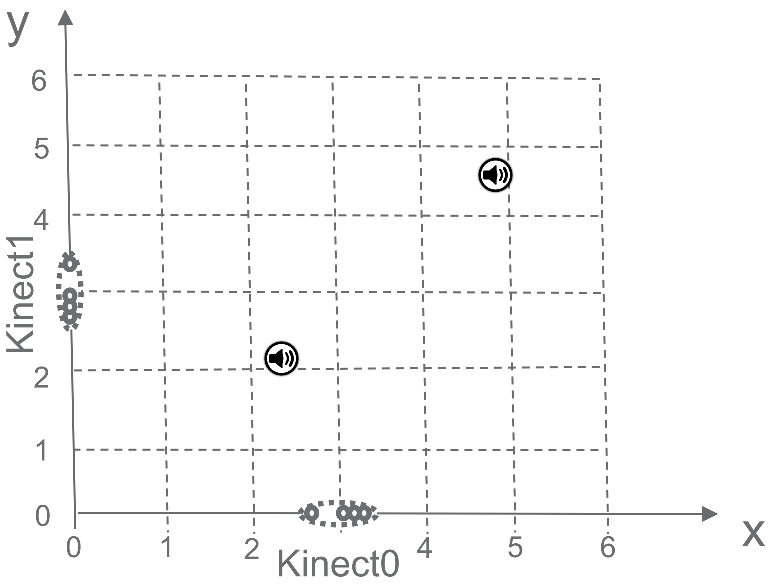
Experimental Setup

Open space 6x6 meters

One to three sound sources
- Musical excerpts (Classical & Funk, Jazz)

Recording using two linear microphone arrays



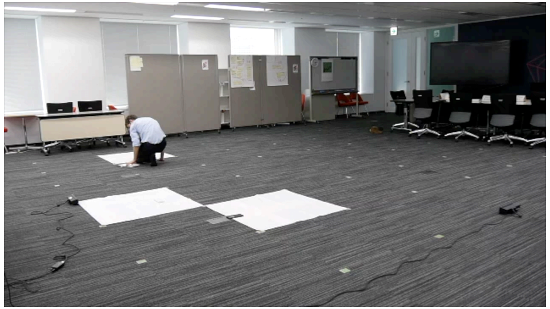


Data Collection

1. Synthetic

pyroomacoustics

2. Real-World



Dataset	Split	Excerpts	# of Srcs	Samples
Synthetic	train-S	classical-funk	1 or 2	100000
	validate-S	classical-funk	1 or 2	5000
	test-S0	classical-funk	1 or 2	5000
	test-S1	classical-funk	3	2500
	test-S2	jazz	1 or 2	5000
Real world with Augmentation	train-A	classical-funk	1 or 2	100000
	validate-A	classical-funk	1 or 2	5000
	test-A0	classical-funk	1 or 2	5000
	test-A1	classical-funk	3	2500
	test-A2	jazz	1 or 2	5000
Real world	test-R0	classical-funk	1 or 2	600
	test-R1	classical-funk	3	300
	test-R2	jazz	1 or 2	600

LESS data in Real-World

Results : Output Representation Comparison

GOAL → Which output representation perform best?

	Resolution 0.3 m			
Output rep.	**Pre (↑)**	**Rec (↑)**	**F1 (↑)**	**RMSE (↓)**
Tight Grid	0.38	0.87	0.53	0.15
Heat Map	**0.94**	**0.88**	**0.90**	**0.10**
Refined Grid	0.91	0.87	0.89	**0.10**

Train and Test (test-S0) on synthetic dataset

- Tight grid gives competitive recall, but poor precision.
- Heat map and Refined grid outperform Tight grid on large margin.

Results : Architecture Design Comparison

GOAL → Easier to generalize with proposed architecture improvement?

	TrainA0		10% of TrainA0	
DNN Arch.	**F1 (↑)**	**RMSE (↓)**	**F1 (↑)**	**RMSE (↓)**
Single Encoder	0.61	0.14	0.48	0.16
Array Encoder + Pair-Wise	**0.68**	**0.13**	**0.63**	**0.15**

Train and Test (test-A0) on real-world with augmentation dataset; Heat map; Resolution 0.3 m

- Lesser training data → Larger performance gap.
- Proposed architecture requires less data to train.

Other Results

Train with 1 or 2 sources and Test with 3 sound sources dataset.

- Generalization on the number of sound source can be observed.

Train with Classical & Funk and Test with Jazz dataset.

- Good Generalization on musical genres can be observed in synthetic data.

Comparison between synthetic and real world dataset

- Performance drop due to the lack of data diversity for training.

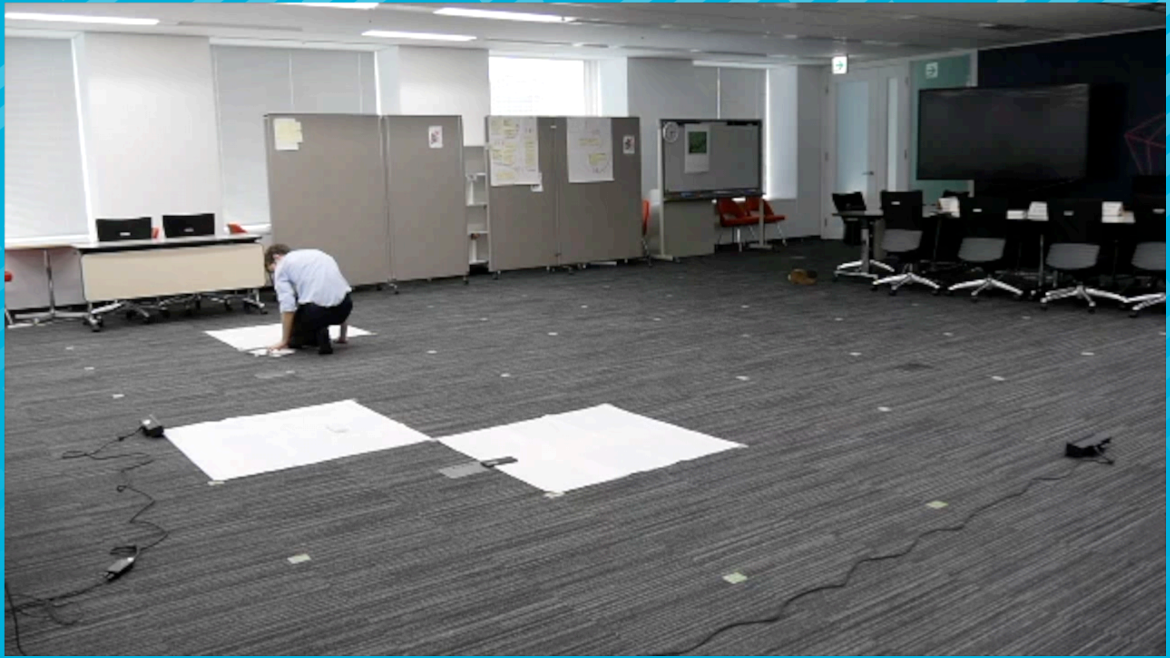
Conclusion

Proposed method to learn multiple sound source 2D localization.

- Encoding-decoding network architecture with two improvements.
- Two novel output representations.
- Extensive experiments both in synthetic and real-world data.

Future Direction : Improving result in real-world experiment.

- Use simulation to generate a large amount of labeled data.
- Train model so that the knowledge is transferrable.



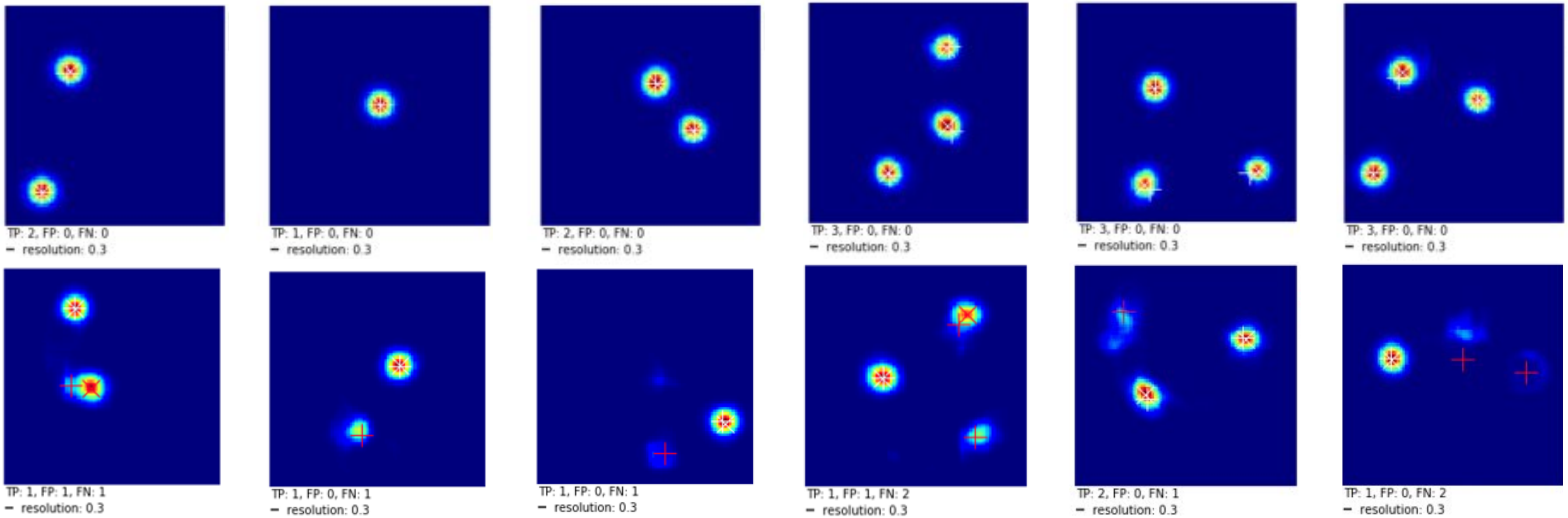
Thank you for your kind attention.

Question & Answer

Appendix

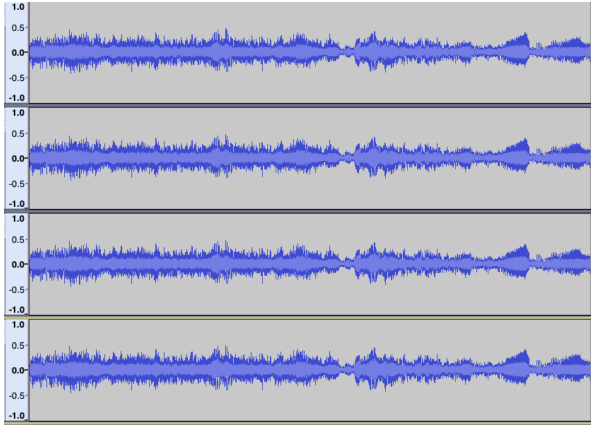
Multiple sound source 2D localization

Results on synthetic data on heatmap representation



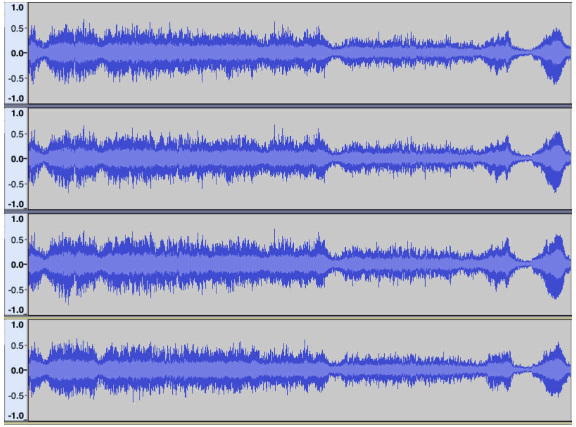
Sim-to-Real Gap in Sound

Simulation



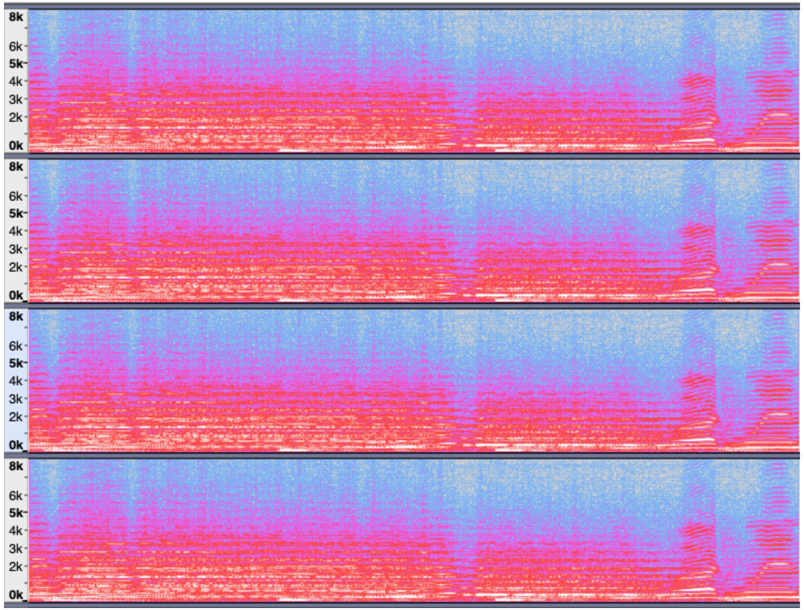
Possible causes for the gap
- Wave propagation approx.
- Reverberation
- Ambient noise

Reality

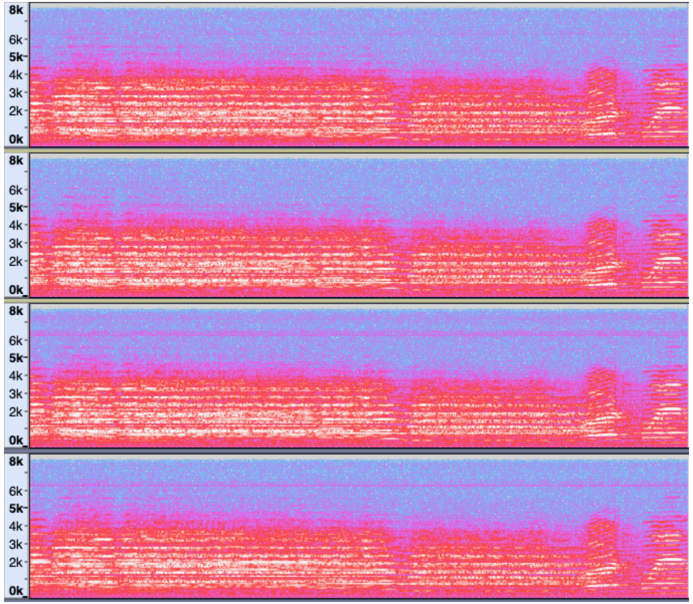


Sim-to-Real Gap in Sound - spectrum

Simulation



Reality



Architecture details

TABLE II: Deep neural network detailed architecture

Block	Filters	Kernel	Conv type	Norm	Activation
Input	Spectral features (one array): **8x9x256**				
Pair-wise feature extraction	Pairs of microphones (one array): **24x9x256**				
	Reshape: **24x9x256** $\rightarrow$ **9x1x24x256**				
	8	2x7	conv2d	bn2d	LeakyReLU
	Reshape: **9x8x12x256** $\rightarrow$ **96x9x256**				
Encoder	128	1x5	conv2d	bn2d	LeakyReLU $*5$
	64	1x3	conv2d		LeakyReLU
	32	1x3	conv2d		LeakyReLU
	16	9x4	conv2d		LeakyReLU
	Reshape: **16x1x32** $\rightarrow$ **512x1x1**				
Decoder	256	3x3	dconv2d	bn2d	ReLU
	128	3x3 / 2x2	dconv2d	bn2d	ReLU
	64	3x3	dconv2d	bn2d	ReLU
	32	3x3	dconv2d	bn2d	ReLU
	16	3x3	conv2d		ReLU
	8	3x3	conv2d		ReLU
	1 / 3	3x3	conv2d		ReLU
Output	TG-rep & HM-rep: **1x81x81** RG-rep: **3x6x6**				

Real-World Data Capturing Configuration

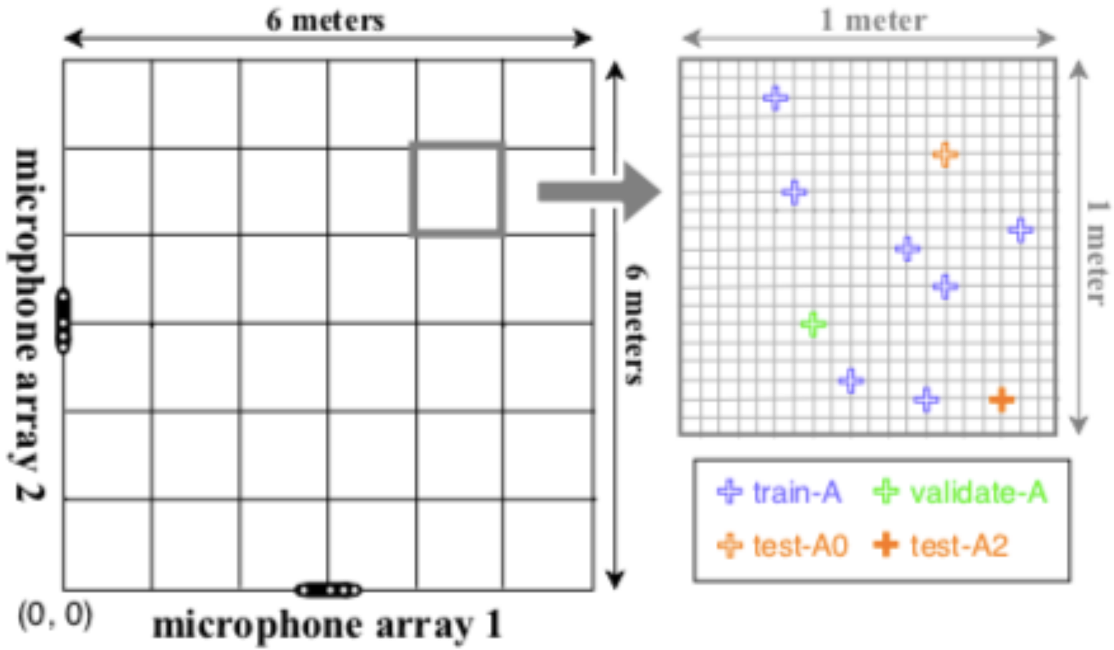


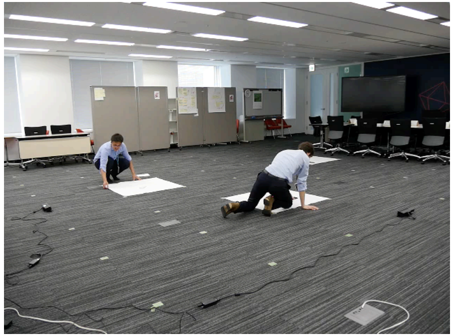
Fig. 2: Environment Layout Configuration

Data Collection

1. Synthetic

pyroomacoustics

2. Real-World

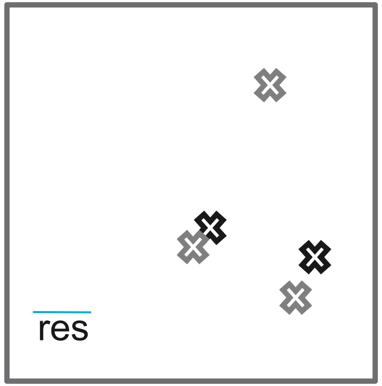


Dataset	Split	Excerpts	# of Srcs	Samples
Synthetic	train-S	classical-funk	1 or 2	100000
	validate-S	classical-funk	1 or 2	5000
	test-S0	classical-funk	1 or 2	5000
	test-S1	classical-funk	3	2500
	test-S2	jazz	1 or 2	5000
Real world with Augmentation	train-A	classical-funk	1 or 2	100000
	validate-A	classical-funk	1 or 2	5000
	test-A0	classical-funk	1 or 2	5000
	test-A1	classical-funk	3	2500
	test-A2	jazz	1 or 2	5000
Real world	test-R0	classical-funk	1 or 2	600
	test-R1	classical-funk	3	300
	test-R2	jazz	1 or 2	600

Evaluation Metrics

Output : List of sound source locations; Keypoints (x, y)

Predicted Keypoints (PK) are paired to **Groundtruth Keypoints** (GK), if they are closer than the chosen resolution threshold.



Groundtruths

Predictions

Grouping	PK	GK
True positive	O	O
False positive	O	X
False negative	X	O

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{F1-Score} = \frac{2 \times precision \times recall}{precision + recall}$$

Additional metric: ***Root Mean Square Error*** *(RMSE)* between TP

Results : Output Representation Comparison

Train and Test (test-S0) on synthetic dataset

	Resolution 0.3 m				Resolution 1.0 m			
Output rep.	**Pre (↑)**	**Rec (↑)**	**F1 (↑)**	**RMSE (↓)**	**Pre (↑)**	**Rec (↑)**	**F1 (↑)**	**RMSE (↓)**
Tight Grid	0.38	0.87	0.53	0.15	0.40	0.92	0.56	0.23
Heat Map	**0.94**	**0.88**	**0.90**	**0.10**	**0.99**	0.93	**0.96**	**0.15**
Refined Grid	0.91	0.87	0.89	**0.10**	0.98	**0.94**	**0.96**	0.17

- Tight grid gives competitive recall, but poor precision.
- Heat map and Refined grid outperform Tight grid on large margin.
- Fine (0.3 m) → Coarse (1.0 m) : increase F1-score, but higher RMSE.

Results : Synthetic, Augmented and Real World Data and Generalization on Musical Genres

Train with Classical & Funk and Test with Classical & Funk and Jazz dataset.

Heat map representation; **Array Encoder + Pair-Wise** Arch.; Metric **Resolution** 1.0 m

	Classical & Funk		Jazz	
Dataset	**F1 (↑)**	**RMSE (↓)**	**F1 (↑)**	**RMSE (↓)**
Synthetic	0.96	0.15	0.97	0.13
Real World with Augmentation	0.80	0.24	0.68	0.37
Real World	0.67	0.33	0.68	0.39

- Performance drop from synthetic to real world dataset; lack of data diversity.
- **Good generalization** on musical genres can be observed in **synthetic data.**

Results : Generalization on Sound Source Number

Train with 1 or 2 sources and Test with 1, 2 and 3 sound source dataset.

Heat map representation; **Array Encoder + Pair-Wise** Arch.; Metric **Resolution** 1.0 m

	1 sound source		2 sound sources		3 sound sources	
Dataset	**F1 (↑)**	**RMSE (↓)**	**F1 (↑)**	**RMSE (↓)**	**F1 (↑)**	**RMSE (↓)**
Synthetic	0.99	0.08	0.93	0.18	0.77	0.22
Real World with Augmentation	0.88	0.22	0.76	0.25	0.62	0.27
Real World	0.85	0.26	0.54	0.40	0.46	0.42

- **Good generalization** on the number of sound source can be observed in **all dataset.**