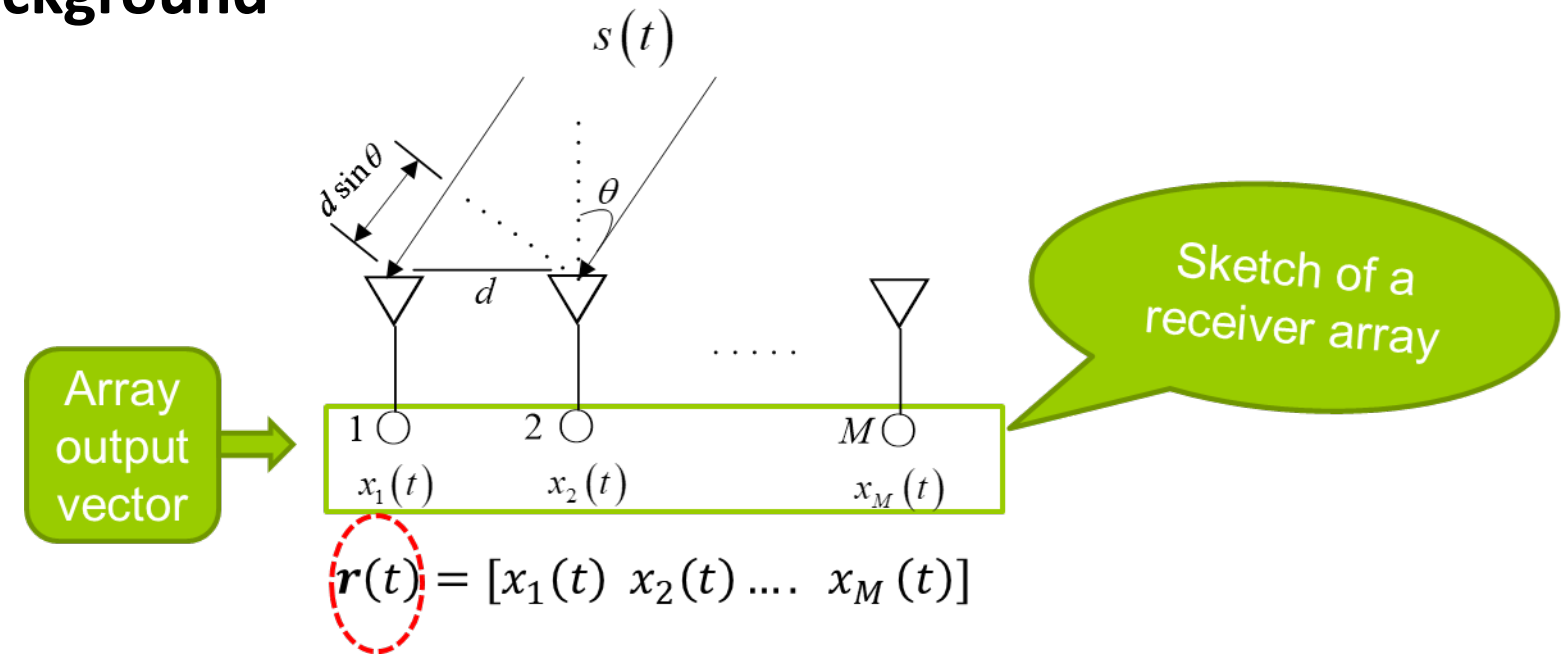
# Outline

- ❑ Background
- ❑ Existing solution and its problem
- ☐ MDNN
- ☐ Numerical Results

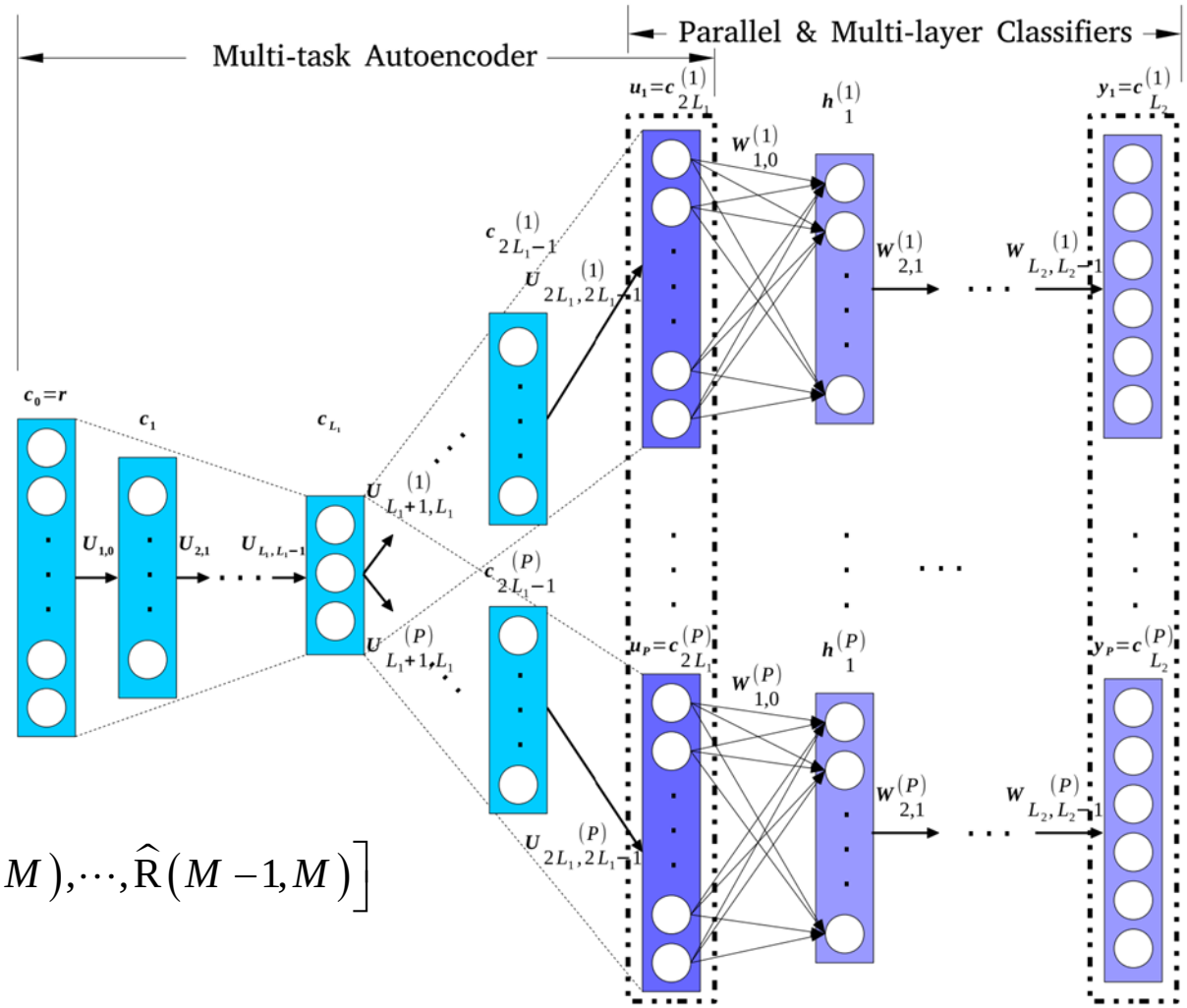
## ❑ Background




$$\mathbf{R} = \mathrm{E}\left[\mathbf{r}\mathbf{r}^{\mathrm{H}}\right] = \mathbf{A}\mathbf{R}_s\mathbf{A}^{\mathrm{H}} + \sigma_n^2\mathbf{I}_M$$

$$\mathbf{R} = \mathbf{\Phi}\mathbf{A}\mathbf{R}_s\mathbf{A}^{\mathrm{H}}\mathbf{\Phi}^{\mathrm{H}} + \sigma_n^2\mathbf{I}_M \xrightarrow{f(.)} \theta$$

# Existing solution




Input vector

$$\left[\hat{\mathrm{R}}(1,2),\cdots,\hat{\mathrm{R}}(1,M),\cdots,\hat{\mathrm{R}}(M-1,M)\right]$$

Fig. 1. Structure of proposed DNN for DOA estimation. The network consists of two parts, one is a multitask autoencoder for spatial filtering and the other is a fully connected multilayer NN for spatial spectrum estimation.

Z. -M. Liu, C. Zhang and P. S. Yu, "Direction-of-Arrival Estimation Based on Deep Neural Networks With Robustness to Array Imperfections," in *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315-7327, Dec. 2018

# Problem

Training dataset :

$$\mathbf{R} = \mathbf{\Phi A R}_s \mathbf{A}^{\mathrm{H}} \mathbf{\Phi}^{\mathrm{H}} + \sigma_n^2 \mathbf{I}_M$$

- Fixed phase error

Testing dataset :

✓ Same array phase error as the one used in training dataset

- Performance in testing stage degrades
  when phase error is different from the one in training stage.

# MDNN METHOD： Methodology

$$\mathbf{r}_{ew} = \mathbf{r} \odot \mathbf{r}^{*}$$

$$\mathbf{R}_{ew} = \mathrm{E}\left[\mathbf{r}_{ew}\mathbf{r}_{ew}^{\mathrm{T}}\right]$$

$$\mathbf{z} = \left[\hat{\mathrm{R}}_{ew}(1,2), \cdots, \hat{\mathrm{R}}_{ew}(1,M), \cdots, \hat{\mathrm{R}}_{ew}(M-1,M)\right]$$

$$\mathbf{z}_0 = \left(\mathbf{z} - mean(\mathbf{z})\right) / \left\|\mathbf{z} - mean(\mathbf{z})\right\|_2$$

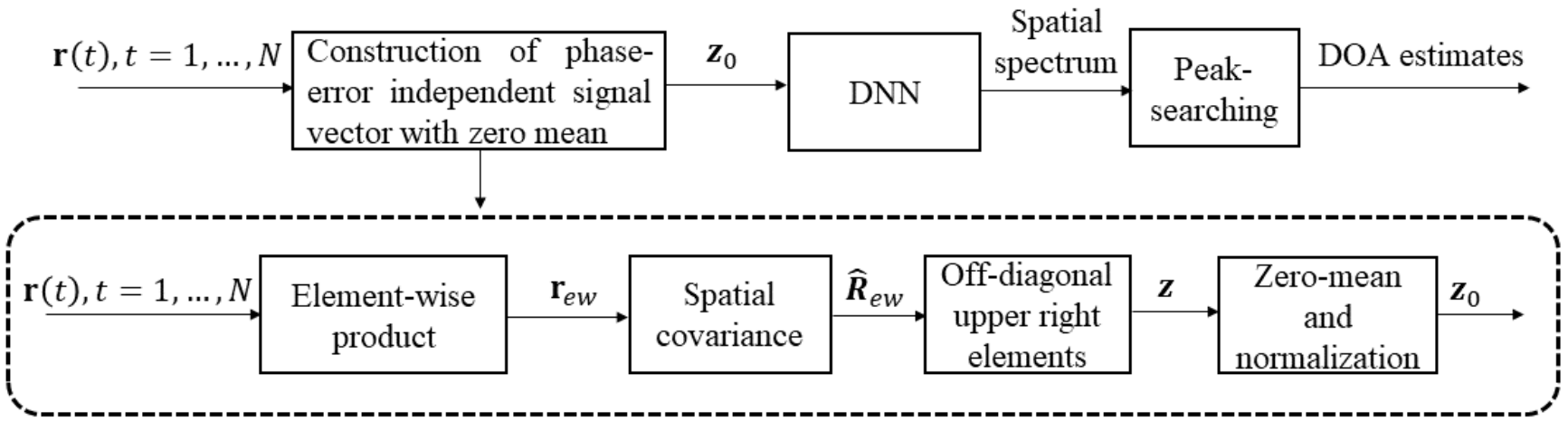



Fig. 1 Scheme of MDNN

# MDNN METHOD： DNN structure

Size of each layer of MDNN and original DNN methods

| Proposed MDNN ($J=J_0$); Original DNN ($J=J_1$) | Multitask autoencoder | | | Each Classifier | | | |
|---|---|---|---|---|---|---|---|
| | IL | HL | OL | IL | HL1 | HL2 | OL |
| | $J$ | $\left\lfloor \frac{J}{2} \right\rfloor$ | $P \times J$ | $J$ | $\left\lfloor \frac{2J}{3} \right\rfloor$ | $\left\lfloor \frac{4J}{9} \right\rfloor$ | $I_0$ |

$J_0$ and $J_1$ are the sizes of the input in the MDNN method and the original DNN method, respectively.

$$J_0 = \frac{M^2 - M}{2} \qquad J_1 = M^2 - M$$

IL, HL, and OL stand for input layer, hidden layer, and output layer, respectively

# MDNN METHOD： Training strategy

The training dataset consists of two equal-power source signals.

$$\theta_1 \in \left[-60^\circ, 60^\circ - \Delta\right)$$

$$\theta_2 = \theta_1 + \Delta$$

$$\Delta \in \left\{3^\circ, 6^\circ, \ldots, 60^\circ\right\}$$

$$\Delta = 3^\circ \quad \theta_1 \in \left\{-60^\circ, -59^\circ, \ldots, 57^\circ\right\}$$ The number of covariance vectors is equal to 117.

$$\left(117 + 114 + \cdots + 60\right) \times 10 \times 4 = 70800$$

# MDNN METHOD： Training strategy

Training dataset :

$$\mathbf{R} = \mathbf{\Phi A R}_s \mathbf{A}^{\mathrm{H}} \mathbf{\Phi}^{\mathrm{H}} + \sigma_n^2 \mathbf{I}_M$$

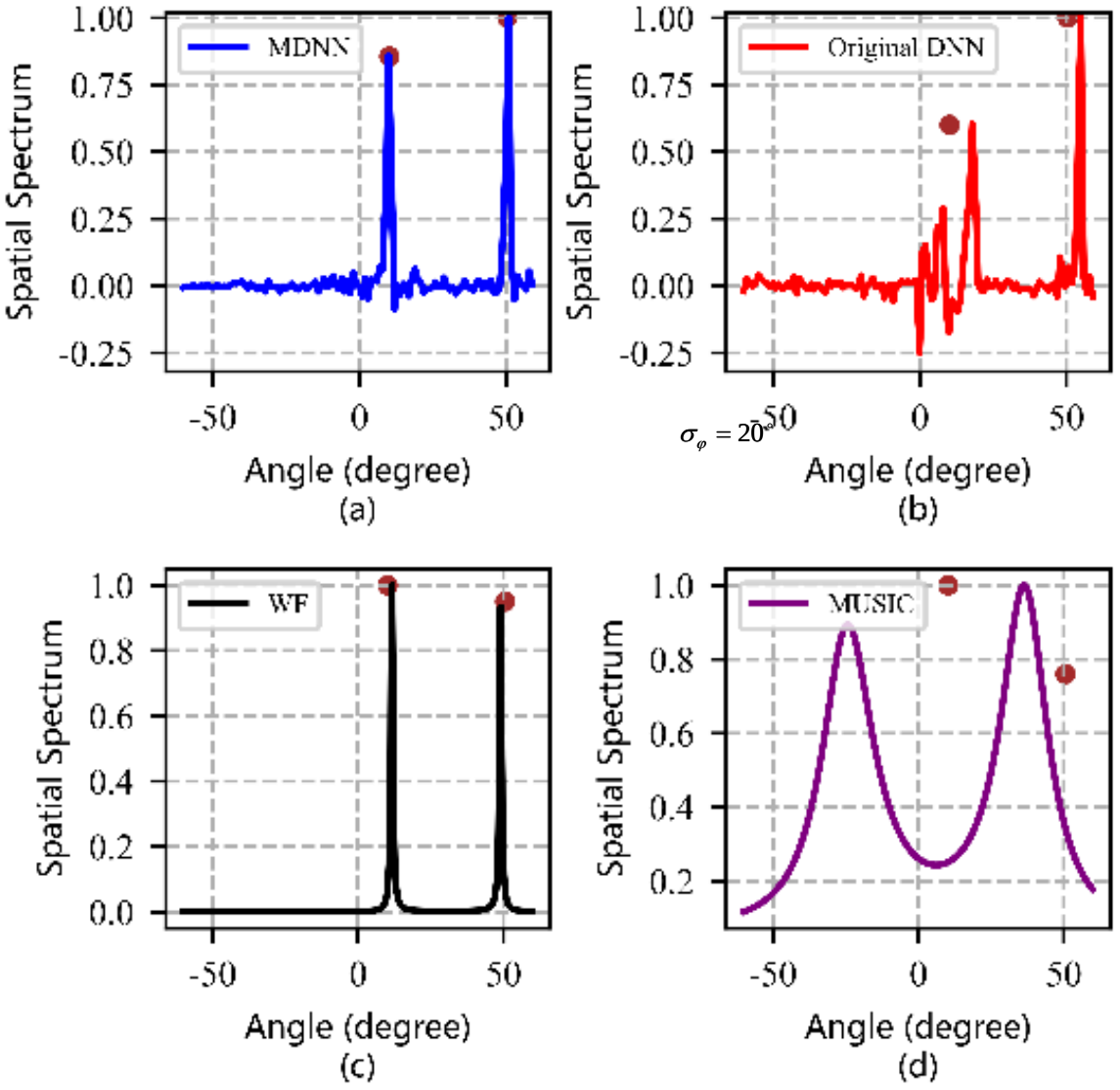
- Multiple randomly generated phase errors

Testing dataset :

- ✓ Array phase error which is different from those in training dataset

- Performance in testing stage is fairly evaluated when phase error is different from those in training stage.

# GM-MDL METHOD： Simulation Results



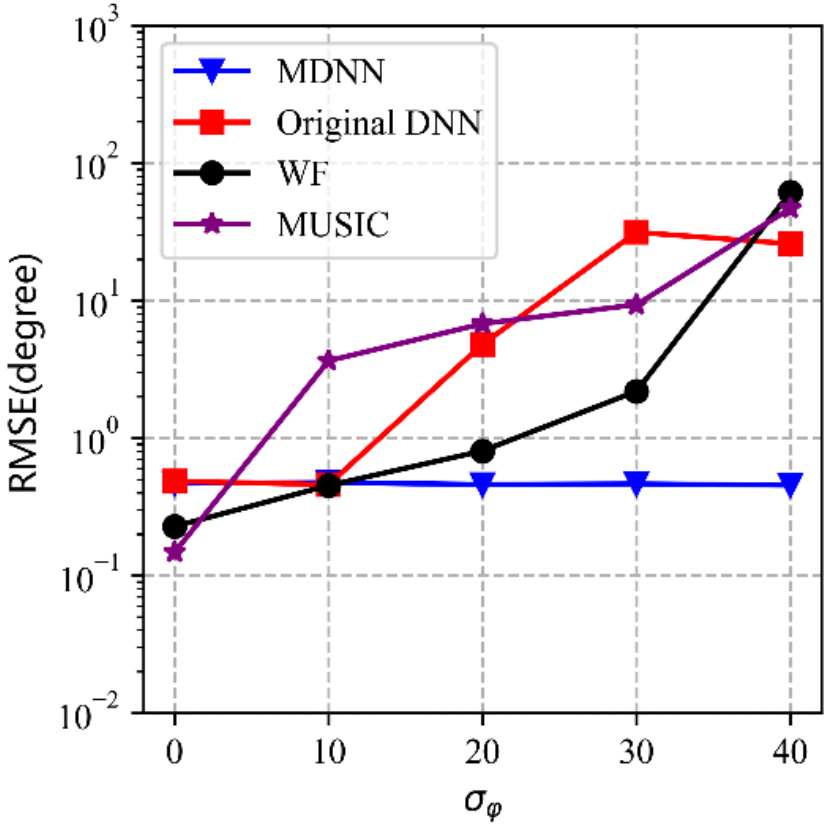

SNR = 10dB

$\sigma_\varphi = 20^\circ$

$\theta_1 = 10.2^\circ$

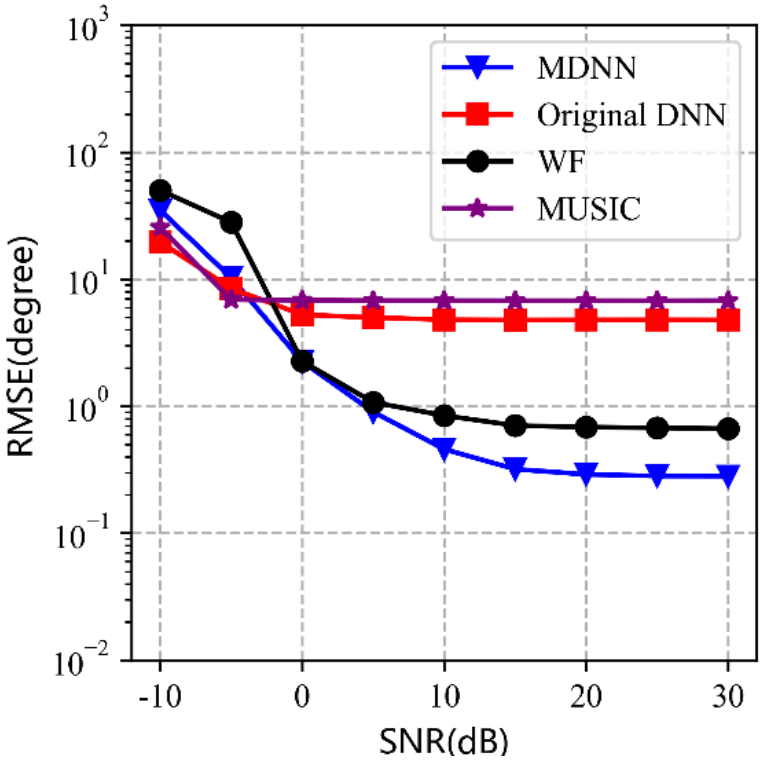
$\theta_2 = 50.5^\circ$

# MDNN METHOD： RMSE



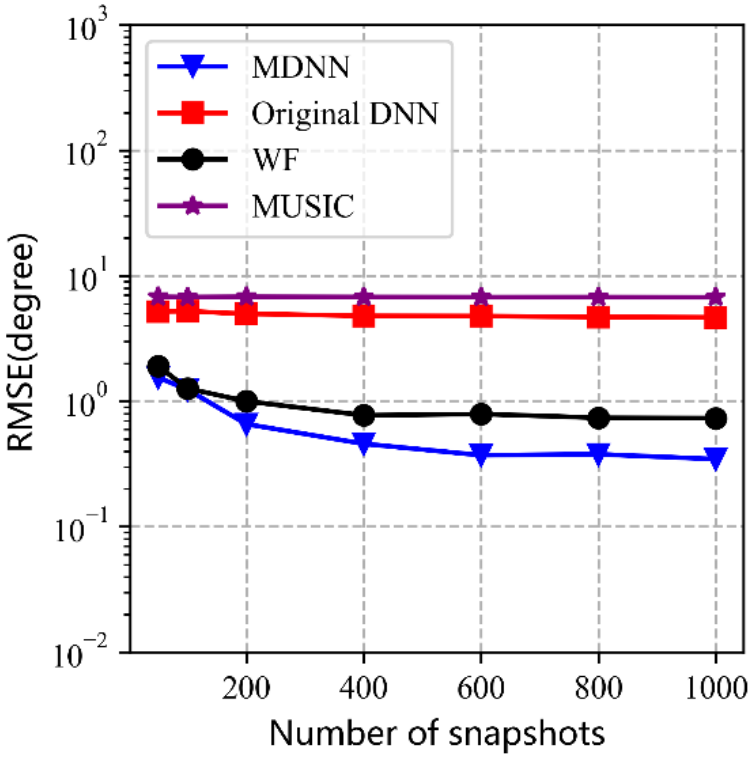

RMSE versus array phase errors

# MDNN METHOD： RMSE




RMSE versus SNR

# MDNN METHOD： RMSE




RMSE versus number of snapshots

# MDNN METHOD：Conclusions

- Use the magnitude of array data to construct an input vector, which is independent of phase errors.
- Obtain the MDNN method for DOA estimation.
- MDNN is independent of phase errors.
- MDNN is more efficient than existing DNN method.
- MDNN has limit to array geometry and angle estimation ambiguity.