NP PD trained MMSE MIMO chest

Zhao-Jie, Luo

janny00kevin@gmail.com

Advisor: Professor Carrson C. Fung

National Yang Ming Chiao Tung University

Feb. 25 2025

Problem Formulation

Our objective is to minimize the expected mean square error:

$$\min_{\hat{\mathbf{h}}} \mathbb{E}_{\mathbf{y},\mathbf{h}} \left[\left\| \mathbf{h} - \hat{\mathbf{h}} \right\|_2^2 \right]$$

and it can be written in epigraph form as:

$$\begin{aligned} &\min_{t,\mathbf{h}} t \\ &\quad s.t.\ \mathbb{E}_{\mathbf{y},\mathbf{h}} \left[\left\| \mathbf{h} - \hat{\mathbf{h}} \right\|_2^2 \right] \leq t \end{aligned}$$

Primal-Dual Optimization Mehthod (1)

We use parameterize channel estimator so that $\hat{\mathbf{h}} = \phi(\mathbf{y};\boldsymbol{\theta})$, with $\boldsymbol{\theta}$ denoting the parameters of the neural network.

Then the Lagrangian function of (15) can be written as

$$\begin{aligned}\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right) &= t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right] - t\right) \\ &= t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right] - t\right)\end{aligned}$$

Primal-Dual Optimization Mehthod (2)

It is uncertain whether or not the duality gap equals zero.
However, the stationary point of $\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right)$ can be found via the KKT conditions by solving for the primal and dual variables alternately using gradient descent and ascent, respectively:

$$\begin{aligned}
\boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \alpha_{\boldsymbol{\theta},k}\lambda_k \nabla_{\boldsymbol{\theta}_k} \mathbb{E}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta}_k)\|_2^2\right] \\
t_{k+1} &= t_k - \alpha_{t,k}(1-\lambda_k) \\
\lambda_{k+1} &= \left[\lambda_k + \alpha_{\lambda,k}\left(\mathbb{E}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta}_{k+1})\|_2^2\right] - t_{k+1}\right)\right]_+
\end{aligned}$$

Sample MMSE Problem Formulation

Change the MMSE problem to sample average

$$\min_{\widehat{\mathbf{h}}} \quad \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{h}_i - \widehat{\mathbf{h}}_i\|_2^2,$$

it's epigraph form is

$$\begin{aligned} \min_{\widehat{\mathbf{h}},t} \quad & t \\ \text{s.t.} \quad & \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{h}_i - \widehat{\mathbf{h}}_i\|_2^2 \leq t. \end{aligned}$$

Neural Network Based Method for Sample MMSE

Let $\widehat{\widetilde{\mathbf{h}}} \triangleq [\widehat{\mathbf{h}}^T, t]^T \in \mathbb{C}^{(n_T n_R+1)\times 1}, \mathbf{c}^T \triangleq [\mathbf{0}^T, 1] \in \mathbb{R}^{1\times(n_T n_R+1)}$, $\mathbf{D} \triangleq [\mathbf{I}_{n_T n_R}, \mathbf{0}] \in \mathbb{R}^{n_T n_R \times (n_T n_R+1)}$,
then the problem becomes

$$\begin{aligned} \min_{\widehat{\widetilde{\mathbf{h}}}} \quad & \mathbf{c}^T \widehat{\widetilde{\mathbf{h}}} \\ \text{s.t.} \quad & \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{h}_i - \mathbf{D}\widehat{\widetilde{\mathbf{h}}}_i\|_2^2 \leq \mathbf{c}^T \widehat{\widetilde{\mathbf{h}}}. \end{aligned}$$

Change the algorithm to a neural network based (MLP) method

$$\widehat{\widetilde{\mathbf{h}}} = \boldsymbol{\phi}(\mathbf{y}_c, \boldsymbol{\theta}),$$

where $\mathbf{y}_c \triangleq \begin{bmatrix} \mathbf{y}_R \\ \mathbf{y}_I \end{bmatrix} \in \mathbb{R}^{2n_R T\times 1}$ is a real value vector concatenated real and imaginary part of $\mathbf{y}$.

Varibles Update w/ Primal-Dual Method (2/11 edited)

Then the Lagrangian function is

$$\mathcal{L}(\widehat{\widetilde{\mathbf{h}}}, \lambda) = \mathbf{c}^T \boldsymbol{\phi}(\mathbf{y}_{c_i}, \boldsymbol{\theta}) + \lambda \left(\frac{1}{N} \sum_{i=1}^{N} \left(\|\mathbf{h}_i - \mathbf{D}\boldsymbol{\phi}(\mathbf{y}_{c_i}, \boldsymbol{\theta})\|_2^2 - \mathbf{c}^T \boldsymbol{\phi}(\mathbf{y}_{c_i}, \boldsymbol{\theta}) \right) \right)$$

then we can use primal-dual method to alternately update the variables

$$\lambda^{(k+1)} = \lambda^{(k)} + \frac{\alpha_\lambda^{(0)}}{k+1} \frac{1}{n} \sum_{i=1}^{n} \left(\|\mathbf{h}_i - \mathbf{D}\boldsymbol{\phi}(\mathbf{y}_{c_i}, \boldsymbol{\theta}^{(k)})\|_2^2 - \mathbf{c}^T \boldsymbol{\phi}(\mathbf{y}_{c_i}, \boldsymbol{\theta}^{(k)}) \right), \quad (1)$$

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha_{\boldsymbol{\theta}}^{(k)} \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_i}(\boldsymbol{\theta}^{(k)}, \lambda^{(k+1)}), \quad (2)$$

where n represents the mini-batch size, $\frac{\alpha_\lambda^{(0)}}{k+1}$ is a monotonically decreasing step size, and $\alpha_{\boldsymbol{\theta}}^{(k)}$ is updated using the Adam optimizer. To save computations, we choose to do the dual update first.

Data Normalization

We standard normalize the dataset before training to let the model learn the features better.

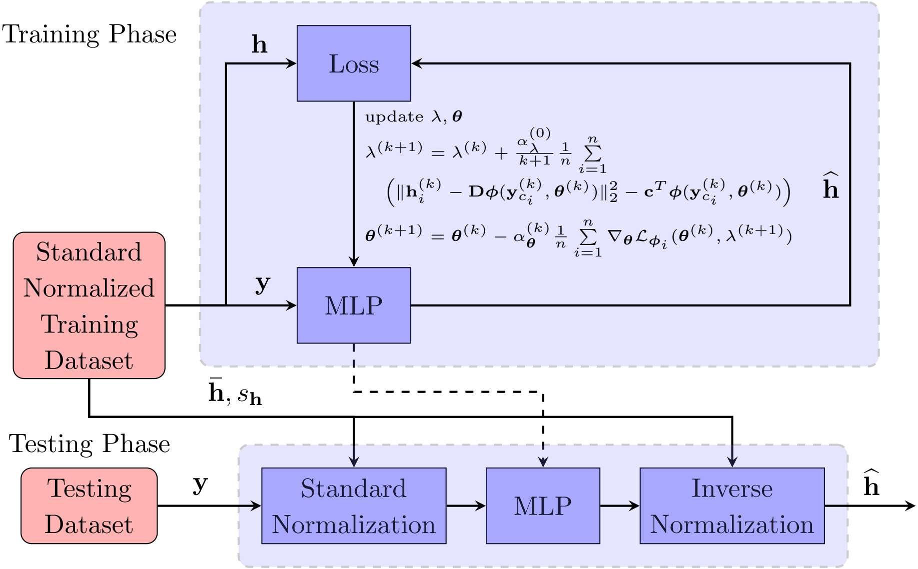
The sample mean of the dataset is $\bar{\mathbf{h}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_i$, where N is the size of training or testing dataset. The sample variance is $s_{\mathbf{h}}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{h}_i - \bar{\mathbf{h}})^H(\mathbf{h}_i - \bar{\mathbf{h}})$, then we can use the standard normalized dataset $\frac{\mathbf{h}_i - \bar{\mathbf{h}}}{s_{\mathbf{h}}}$ to train the model.
In the testing process, silmilar to upscaling case, standard normalize $\mathbf{y}$ as $\frac{\mathbf{y} - \bar{\mathbf{h}}}{s_{\mathbf{h}}}$ for input, where $\bar{\mathbf{h}}$ and $s_{\mathbf{h}}^2$ is the sample mean and sample variance from the training dataset. And restore $\widehat{\mathbf{h}}$ by multiplying $s_{\mathbf{h}}$ and adding $\bar{\mathbf{h}}$ back (inverse normalization).
If we use the sample mean and variance from the testing dataset, we must first collect the entire dataset, though they perform almost the same. However, in practice, the receiver obtains the received signal one by one.

Algorithm of Sample Average

Algorithm 1: neural network based (MLP or CNN) primal-dual sample MSE learning

```
Result: $\boldsymbol{\theta}$, $\lambda$
Input: $\mathbf{y}$
Initialization: $\boldsymbol{\theta}^{(0)},\ \lambda^{(0)} = 1,\ \alpha_{\boldsymbol{\theta}}^{(0)} = 10^{-4},\ \alpha_{\lambda}^{(0)} = 10^{-4}$
do
    do
        Obtain $\boldsymbol{\phi}(\mathbf{y}_c^{(k)}, \boldsymbol{\theta}^{(k)})$
        Update primal and dual variables:
        $\lambda^{(k+1)} = \lambda^{(k)} + \frac{\alpha_{\lambda}^{(0)}}{k+1} \frac{1}{n} \sum_{i=1}^{n} \left( \|\mathbf{h}_i^{(k)} - \mathbf{D}\boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)})\|_2^2 - \mathbf{c}^T \boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)}) \right)$
        $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha_{\boldsymbol{\theta}}^{(k)} \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_i}(\boldsymbol{\theta}^{(k)}, \lambda^{(k+1)})$
    while minibatches with each size $n$;
    Shuffle the dataset;
while $\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}}(\boldsymbol{\theta}, \lambda)\|_2 > \epsilon$ or $\frac{1}{n} \sum_{i=1}^{n} \left( \|\mathbf{h}_i^{(k)} - \mathbf{D}\boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)})\|_2^2 - \mathbf{c}^T \boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)}) \right) > 0$
  or $\lambda < 0$ or $\lambda \cdot \frac{1}{n} \sum_{i=1}^{n} \left( \|\mathbf{h}_i^{(k)} - \mathbf{D}\boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)})\|_2^2 - \mathbf{c}^T \boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)}) \right) > \epsilon$ or
  $k <$ NumEpoch × NumBatch;
Return result;
```

Block Diagram for Sample MMSE NN PD



update $\lambda, \boldsymbol{\theta}$

$$\lambda^{(k+1)} = \lambda^{(k)} + \frac{\alpha_\lambda^{(0)}}{k+1} \frac{1}{n} \sum_{i=1}^{n} \left(\|\mathbf{h}_i^{(k)} - \mathbf{D}\boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)})\|_2^2 - \mathbf{c}^T \boldsymbol{\phi}(\mathbf{y}_{c_i}^{(k)}, \boldsymbol{\theta}^{(k)}) \right)$$

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha_{\boldsymbol{\theta}}^{(k)} \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\phi}_i}(\boldsymbol{\theta}^{(k)}, \lambda^{(k+1)})$$

Simulation Result of Rayleigh fading channel (MLP)

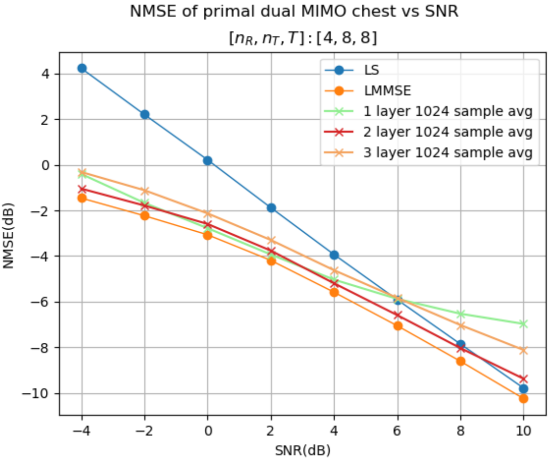


Figure: 2 hidden layers performs the best, 3 hidden layers overfitting, and the curve of 1 hidden layer looks like PD DRL method.

CNN with Sample Average (1)

Let the input of the CNN be $\mathbf{Y}_c \triangleq \{\mathrm{Re}\{\mathbf{Y}\}, \mathrm{Im}\{\mathbf{Y}\}, |\mathbf{Y}|\} \in \mathbb{R}^{3\times n_R \times T}$, where the three input channels correspond to the real part, imaginary part, and magnitude of the received signal $\mathbf{Y}$.
Then the CNN model can be represented as

$$\mathbf{W}_{out} f(\mathbf{W}_1 \mathrm{vec}(\mathbf{K} * \mathbf{Y}_c + \mathbf{b}_{conv}) + \mathbf{b}_1) + \mathbf{b}_{out} \ \in \mathbb{R}^{(2n_R n_T+2)\times 1},$$

where $\mathbf{K} \in \mathbb{R}^{8\times 3\times 3\times 3}$ is the filter of 8 output channels, 3 input channels, and filter size 3×3 with stride $= 1$; $\mathbf{b}_{conv} \in \mathbb{R}^{8\times(n_R-2)\times(T-2)}$ is the bias of the convolutional layer;
$\mathbf{W}_1 \in \mathbb{R}^{8(n_R-2)(T-2)\times 1024}, \mathbf{b}_1 \in \mathbb{R}^{1024\times 1}, \mathbf{W}_{out} \in \mathbb{R}^{(2n_R n_T+2)\times 1024}$, $\mathbf{b}_{out} \in \mathbb{R}^{(2n_R n_T+2)\times 1}$ are the weights and biases of the fully connected layers, and $f(\cdot)$ is the activation function, here I use tanh as before.
For the case $\{n_R = 4, n_T = 4, T = 4\}$, there are about 1M parameters in the 2 hidden layers MLP; and about 167K in the CNN. There are 1M trainning datapoints evenly distributed from SNR = -4 to 10 dB.

CNN with Sample Average (2)

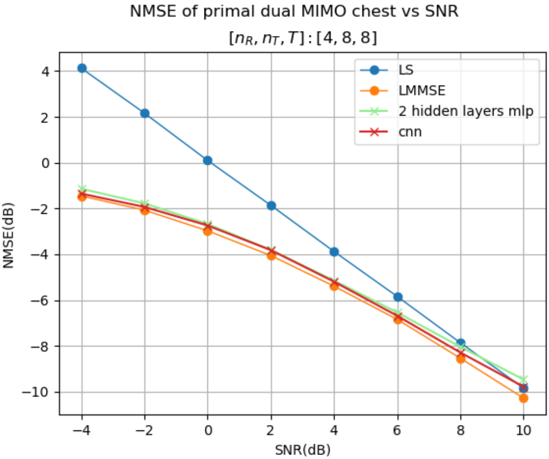


Figure: Although the elements of **H** are independent, multiplying by the pilot signal introduces patterns that the convolutional layer can capture.

Policy Gradient (1)

We have the policy gradient theorem:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\tau}[G(\tau)] = \mathbb{E}_{\tau}\left[\sum_{t=0}^{T-1} G(\tau)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(A_t|S_t)\right]$$

At each time step, $t = 1, ..., T-1$:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\tau}[G(\tau)] = \mathbb{E}_{\tau}\left[G(\tau)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(A_t|S_t)\right]$$

And we can estimate the policy gradient with sample mean:

$$\widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_{\tau}[G(\tau)] = \frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}} G(\tau)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(A_t|S_t)$$

Policy Gradient (2)

Our goal is to minimize the mean square error, by substituting $\mathbb{E}_\tau[G(\tau)]$, $\pi_{\boldsymbol{\theta}}(A_t|S_t)$ with $\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h}-\phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right]$, and $\pi_{\boldsymbol{\theta}}(\hat{\mathbf{h}}|\mathbf{y})$.
Thus, we obtain the estimated policy gradient for our problem:

$$\widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h}-\phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right] = \frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}\left\|\mathbf{h}-\widehat{\phi}(\mathbf{y};\boldsymbol{\theta})\right\|_2^2 \nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}\left(\hat{\mathbf{h}}|\mathbf{y}\right) \tag{3}$$

where $\hat{\phi}(\mathbf{y};\boldsymbol{\theta}) = \hat{\mathbf{h}}$ is the sampled output of the policy.

Ensemble Average with Policy Gradient (1)

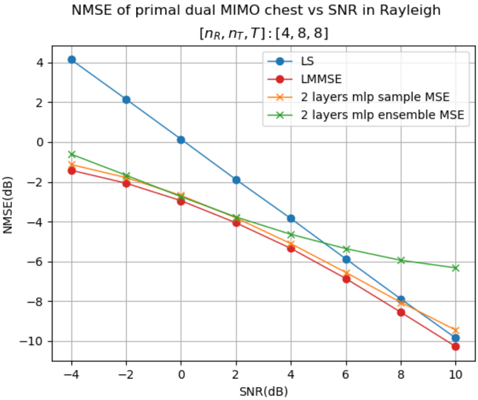
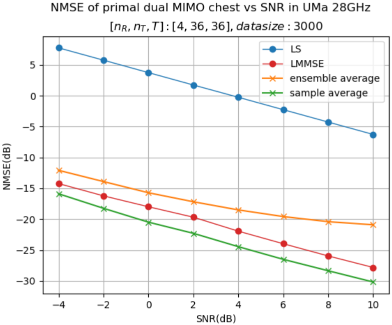


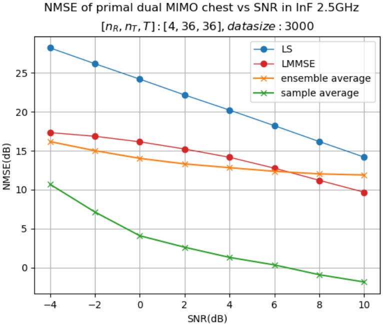
Figure: change sample to ensemble average, but it performs the same as before

Ensemble Average with Policy Gradient (2)

The performances of 2 layers MLP trained with standard normalized dataset using policy gradient and PD method and testing with standard normalized $\mathbf{y}$ in



(a) UMa 28GHz channel



(b) InF 2.5GHz channel

Single SNR in Rayleigh Fading Channel

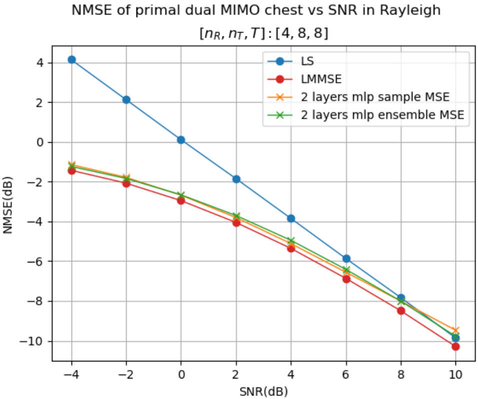
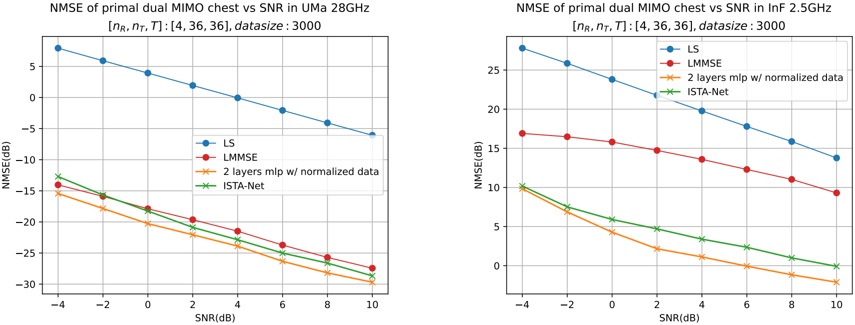


Figure: In Rayleigh fading channel, here I only use 1 single SNR training dataset to train both, and test them uder corresponding SNR testing dataset.

Updates on 10/8



NMSE of primal dual MIMO chest vs SNR in UMa 28GHz
$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$

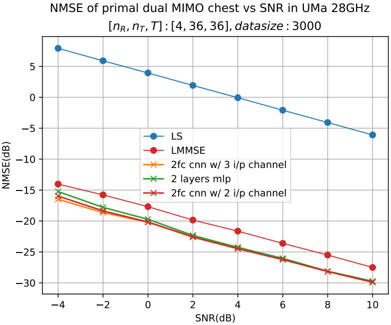
NMSE of primal dual MIMO chest vs SNR in InF 2.5GHz
$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$

Figure: Comparison of PD sample mean chest performance and the ISTA-Net

Updates on 10/17

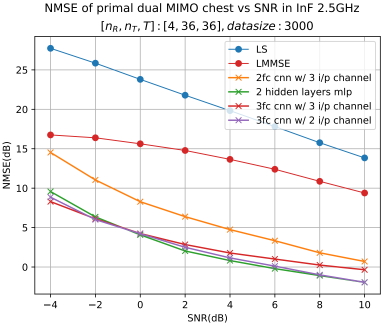
- With ensemble MSE, I tried 10 times larger dataset(from 1M to 10M) with different minibatch size (10k, 100k, 1M, 10M), but the results remain the same.

- With sample MSE, replace MLP with CNN (1 convolutional layer):
    - In UMa, they perform almost the same, even when only input the real and imaginary part.
    - In InF, the original CNN model underperforms MLP. After adding 1 fully connected layer then it performs close to MLP, even when only input 2 channels (real and imaginary part).



NMSE of primal dual MIMO chest vs SNR in UMa 28GHz

$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$



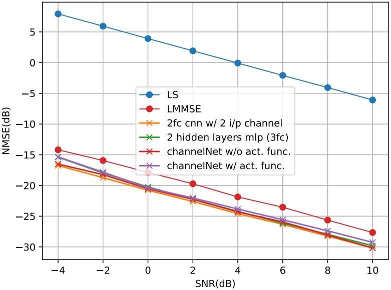
NMSE of primal dual MIMO chest vs SNR in InF 2.5GHz

$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$

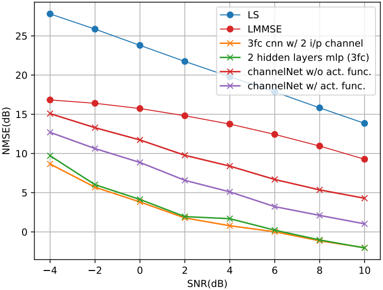
Updates on 11/7

ChannelNet [1] contains 3 convolutional layers with 256 filters of size 3×3 and 3 fully connected layers with 1024 and 2048 units respectively and 2 dropout layers. About 40M parameters comparing to about 1M in MLP and 3fc CNN and 167K in 2fc CNN. But there is no performance gain in UMa, and it performs worse in InF.

NMSE of primal dual MIMO chest vs SNR in UMa 28GHz
$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$

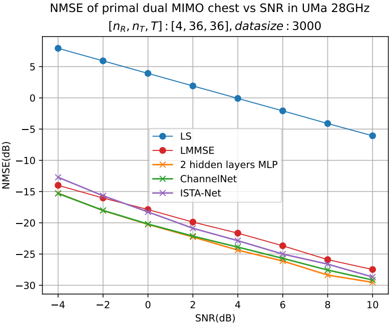


NMSE of primal dual MIMO chest vs SNR in InF 2.5GHz
$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$

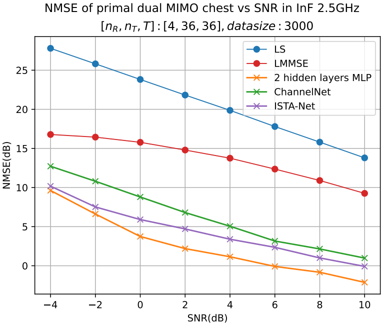


[1] A. M. Elbir, A. Papazafeiropoulos, P. Kourtessis and S. Chatzinotas, ”Deep Channel Learning for Large Intelligent Surfaces Aided mm-Wave Massive MIMO Systems,” in IEEE Wireless Communications Letters, vol. 9, no. 9, pp. 1447-1451, Sept. 2020

Compare with channelNet & ISTA-Net (11/27)


NMSE of primal dual MIMO chest vs SNR in UMa 28GHz

$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$


NMSE of primal dual MIMO chest vs SNR in InF 2.5GHz

$[n_R, n_T, T] : [4, 36, 36], datasize : 3000$

MISO System model with passive diagonal IRS (12/12)

Consider single N_T antennas BS, N_I antennas IRS, single antenna UE MISO downlink case, the receive signal

$$\begin{aligned}\mathbf{y}^T &= (\mathbf{h}_D^T + \mathbf{h}_{I,U}^T \boldsymbol{\Psi} \mathbf{H}_{B,I})\mathbf{X} + \mathbf{w}^T \\ &= (\mathbf{h}_D^T + \boldsymbol{\psi}^T \underbrace{\mathrm{Diag}(\mathbf{h}_{I,U})\mathbf{H}_{B,I}}_{\mathbf{H}_C^T})\mathbf{X} + \mathbf{w}^T,\end{aligned}$$

where $\mathbf{y} \in \mathbb{C}^{T\times 1}$ is the receive signal, $\mathbf{h}_D \in \mathbb{C}^{N_T\times 1}$ is the direct channel from BS to UE, $\mathbf{H}_{B,I} \in \mathbb{C}^{N_I\times 1}$ is the channel from BS to IRS, $\boldsymbol{\Psi} = \mathrm{Diag}(\boldsymbol{\psi}) \in \mathbb{C}^{N_I\times N_I}$ is the reflective coefficient matrix of the IRS, $\mathbf{h}_{I,U} \in \mathbb{C}^{N_I\times N_T}$ is the channel from IRS to UE, then $\mathbf{H}_C \in \mathbb{C}^{N_T\times N_I}$ is the cascade channel, and $\mathbf{X} \in \mathbb{C}^{N_T\times T}$ is the pilot signal matrix.

If the direct channel is blocked, then

$$\mathbf{y} = \mathbf{X}^T \mathbf{H}_C \boldsymbol{\psi} + \mathbf{w}. \tag{4}$$

MIMO System model with passive IRS (12/12)

Consider N_R antennas UE MIMO downlink case, the vectorized receive signal [2]

$$\begin{aligned}
\mathbf{y} = \mathrm{vec}(\mathbf{Y}) &= \mathrm{vec}(\mathbf{H}_{I,U}\boldsymbol{\Psi}\mathbf{H}_{B,I}\mathbf{X} + \mathbf{W}) \\
&= (\mathbf{X}^T\mathbf{H}_{B,I}^T \circledast \mathbf{H}_{I,U})\boldsymbol{\psi} + \mathbf{w} \\
&= (\mathbf{X}^T\mathbf{H}_{B,I}^T \circledast \mathbf{I}_{N_R}\mathbf{H}_{I,U})\boldsymbol{\psi} + \mathbf{w} \\
&= \underbrace{(\mathbf{X}^T \otimes \mathbf{I}_{N_R})}_{\widetilde{\mathbf{X}} \in \mathbb{C}^{TN_R \times N_T N_R}} \underbrace{(\mathbf{H}_{B,R}^T \circledast \mathbf{H}_{R,U})}_{\mathbf{H}_C \in \mathbb{C}^{N_T N_R \times N_I}} \boldsymbol{\psi} + \mathbf{w}, \\
&= \widetilde{\mathbf{X}}\mathbf{H}_C\boldsymbol{\psi} + \mathbf{w}
\end{aligned}$$

where $\circledast$ is the Khatri-Rao product (column-wise Kronecker product). The system above (MISO, MIMO) is the case when the IRS coefficients remain the same through the whole estimating time interval.

[2] Liu, Shuangzhe & TRENKLER, OTZ. (2008). Hadamard, Khatri-Rao, Kronecker and other matrix products. International Journal of Information & Systems Sciences. 4.

MIMO System model with passive IRS (12/26)

[3] Breaks the training interval T into T/N_T subblocks of length N_T:

$$\mathbf{x}_t\text{pilots} = \left[\quad \underbrace{\mathbf{X} \qquad \cdots \qquad \mathbf{X}}_{\text{repeated } T/N_T \text{ times}} \quad \right], \ \mathbf{X} \in \mathbb{C}^{N_T \times N_T},$$

$$\boldsymbol{\psi}_t\text{pilots} = \left[\underbrace{\boldsymbol{\psi}_1 \cdots \boldsymbol{\psi}_1}_{\text{repeated}N_T\text{times}} \quad \cdots \underbrace{\boldsymbol{\psi}_{\frac{T}{N_T}} \cdots \boldsymbol{\psi}_{\frac{T}{N_T}}}_{\text{repeated}N_T\text{times}} \right], \ \boldsymbol{\psi}_b \in \mathbb{C}^{N_I \times 1}.$$

Thus in a subblock b, the IRS coefficients $\boldsymbol{\Psi}_b$ remain the same

$$\mathbf{Y}_b = \text{vec}(\mathbf{H}_{I,U}\boldsymbol{\Psi}_b\mathbf{H}_{B,I}\mathbf{X} + \mathbf{W}),$$
$$\text{where } \boldsymbol{\psi}_b = \text{diag}(\boldsymbol{\Psi}_b).$$

[3] A.L. Swindlehurst, G. Zhou, R. Liu, C. Pan and M. Li, “Channel estimation with reconfigurable intelligent surface - A general framework,” *Proceedings of the IEEE*, vol. 110(9), pp. 1312-1338, Sep. 2022.

MIMO System model with passive IRS (12/26)

$$
\begin{aligned}
\mathbf{y}_b &= \mathrm{vec}(\mathbf{Y}_b) = \mathrm{vec}(\mathbf{H}_{I,U}\boldsymbol{\Psi}_b\mathbf{H}_{B,I}\mathbf{X} + \mathbf{W}) \\
&= (\mathbf{X}^T\mathbf{H}_{B,I}^T \circledast \mathbf{H}_{I,U})\boldsymbol{\psi}_b + \mathbf{w} \qquad (\because \mathrm{vec}(\mathbf{A}\mathbf{W}_d\mathbf{B}) = (\mathbf{B}^T \circledast \mathbf{A})\mathbf{w}) \\
&= (\mathbf{X}^T\mathbf{H}_{B,I}^T \circledast \mathbf{I}_{N_R}\mathbf{H}_{I,U})\boldsymbol{\psi}_b + \mathbf{w} \\
&= \underbrace{(\mathbf{X}^T \otimes \mathbf{I}_{N_R})}_{\widetilde{\mathbf{X}} \in \mathbb{C}^{N_T N_R \times N_T N_R}} \underbrace{(\mathbf{H}_{B,I}^T \circledast \mathbf{H}_{I,U})}_{\mathbf{H}_C \in \mathbb{C}^{N_T N_R \times N_I}} \boldsymbol{\psi}_b + \mathbf{w} \\
&= \widetilde{\mathbf{X}}\mathbf{H}_C\boldsymbol{\psi}_b + \mathbf{w} \qquad ((\mathbf{A}\mathbf{C}) \circledast (\mathbf{B}\mathbf{D}) = (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \circledast \mathbf{B}))
\end{aligned}
$$

The result $\mathbf{y}_b$ from each of the T/N_T subblocks then forms a column of the following combined equation:

$$
\mathbf{Y} = \left[\mathbf{y}_1, \cdots \mathbf{y}_{\frac{T}{N_T}}\right] = \widetilde{\mathbf{X}}\mathbf{H}_C\boldsymbol{\Psi} + \mathbf{W} \ \in \mathbb{C}^{N_T N_R \times \frac{T}{N_T}},
$$

$$
\text{where } \boldsymbol{\Psi} = \left[\boldsymbol{\psi}_1, \cdots \boldsymbol{\psi}_{\frac{T}{N_T}}\right] \in \mathbb{C}^{N_I \times \frac{T}{N_T}}.
$$

MIMO System model with passive IRS (12/26)

Let $\mathbf{X}\mathbf{X}^H = \mathbf{I}_{N_T}$, $\widetilde{\mathbf{X}} \triangleq (\mathbf{X}^T \otimes \mathbf{I}_{N_R})$, then

$$\begin{aligned}
\widetilde{\mathbf{X}}^H \widetilde{\mathbf{X}} &= ((\mathbf{X}^T \otimes \mathbf{I}_{N_R})^T)^* (\mathbf{X}^T \otimes \mathbf{I}_{N_R}) = (\mathbf{X}^* \otimes \mathbf{I}_{N_R})(\mathbf{X}^T \otimes \mathbf{I}_{N_R}) \\
&\qquad (\because\ (\mathbf{A} \otimes \mathbf{B})^T = (\mathbf{A}^T \otimes \mathbf{B}^T)\ \&\ (\mathbf{A} \otimes \mathbf{B})^* = (\mathbf{A}^* \otimes \mathbf{B}^*)) \\
&= (\mathbf{X}^* \mathbf{X}^T) \otimes (\mathbf{I}_{N_R} \mathbf{I}_{N_R}) = \mathbf{I}_{N_T} \otimes \mathbf{I}_{N_R} = \mathbf{I}_{N_T N_R} \\
&\qquad (\because\ (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC} \otimes \mathbf{BD})\ \&\ \mathbf{X}^* \mathbf{X}^T = \mathbf{X}\mathbf{X}^H = \mathbf{I}_{N_T}),
\end{aligned}$$

multiplying on the right of $\mathbf{Y}$ by $\widetilde{\mathbf{X}}^H$, we get

$$\widetilde{\mathbf{X}}^H \mathbf{Y} = \widetilde{\mathbf{X}}^H \widetilde{\mathbf{X}} \mathbf{H}_C \boldsymbol{\Psi} + \widetilde{\mathbf{X}}^H \mathbf{W} = \mathbf{H}_C \boldsymbol{\Psi} + \widetilde{\mathbf{W}},$$

then vectorize it:

$$\begin{aligned}
\widetilde{\mathbf{y}} &\triangleq \mathrm{vec}(\widetilde{\mathbf{X}}^H \mathbf{Y}) = (\boldsymbol{\Psi}^T \otimes \mathbf{I}_{N_T N_R}) \mathbf{h}_c + \widetilde{\mathbf{w}} \\
&= \widetilde{\boldsymbol{\Psi}} \mathbf{h}_c + \widetilde{\mathbf{w}} \ \in \mathbb{C}^{T N_R \times 1} \qquad (5)
\end{aligned}$$

Problem Formulation (12/26)

The MMSE problem can be formulated as

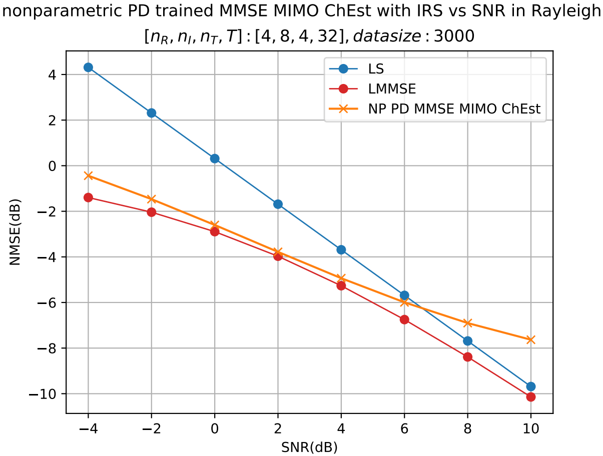
$$\begin{aligned} \min_{\boldsymbol{\theta}, \widetilde{\boldsymbol{\Psi}}} \quad & \mathbb{E}\left[\| \mathbf{h}_c - \phi\left(\widetilde{\mathbf{y}}(\widetilde{\boldsymbol{\Psi}}), \boldsymbol{\theta} \right) \|_2^2 \right] \\ \text{s.t.} \quad & \widetilde{\boldsymbol{\Psi}} = \boldsymbol{\Psi} \otimes \mathbf{I}_{N_R N_T} \\ & |[\boldsymbol{\Psi}]_{i,j}| = 1, \quad \forall i, j \end{aligned}$$

Simulation Configuration (1/9)

Parameter	Value	Parameter	Value
N_R, N_I, N_T, T	4, 8, 4, 32	Number of batches	10
$H_{B,I}, H_{I,U}$	$\sim \mathcal{CN}(0,1)$	Trainning dataset	100K
IRS coefficient $\boldsymbol{\Psi}$	$\mathbf{I}_{N_I}$	Testing dataset	3K
Transmitted pilot $\mathbf{X}$	$\mathbf{I}_{N_T}$	Epoch	300
Batch size	10K	Learning rate	10^{-5}

Using 2 hidden layers MLP with 1024 units each, and the activation function $f(\cdot)$ is tanh:
$\boldsymbol{\phi}(\widetilde{\mathbf{y}}_c, \boldsymbol{\theta}) = \mathbf{W}_{out} f(\mathbf{W}_2 f(\mathbf{W}_1 \widetilde{\mathbf{y}}_c + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_{out}$, where $\widetilde{\mathbf{y}}$ is split into real and imaginary parts and concatenate to $\widetilde{\mathbf{y}}_c$, $\mathbf{W}_1 \in \mathbb{R}^{1024\times256}$, $\mathbf{W}_2 \in \mathbb{R}^{1024\times1024}$, $\mathbf{W}_{out} \in \mathbb{R}^{1024\times258}$, $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{1024\times1}$, $\mathbf{b}_{out} \in \mathbb{R}^{258\times1}$.
And using Algorithm (1) with the ground truth $\mathbf{h}_c$ to train the model.

Simulation Result (only identity) (1/9)



Simulation Result with Hadamard matrix (1/16)

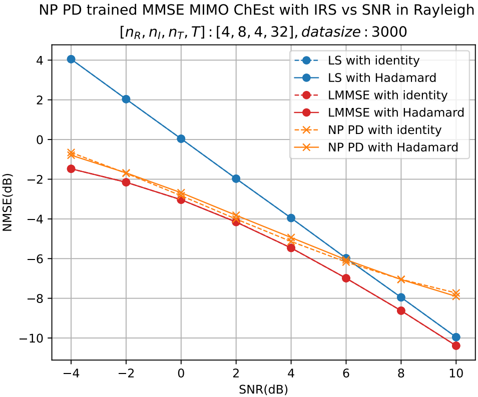
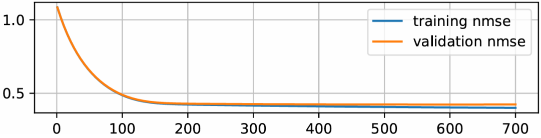


Figure: The dashed lines are the performances of using identity matrix as the IRS coefficients, and the solid lines are the performances of using Hadamard matrix. It shows that there is no significant difference in Rayleigh fading channel (The dashed lines are overlapped by the solid lines).

Overfitting (1/16)

While training, I noticed that the model was overfitting. The training loss kept decreasing while the validation loss remained the same.



Therefore, I tried to add dropout layers into the MLP. I tried different dropout rates (0.001, 0.01, 0.1, 0.3, 0.5), it helps to alliviate the overfitting, but the performance is still the same.
Then I increased the training dataset size from 1M back to 10M (100 batched with 100K datapoints each),

Simulation Result with 10M dataset (Rayleigh) (1/16)

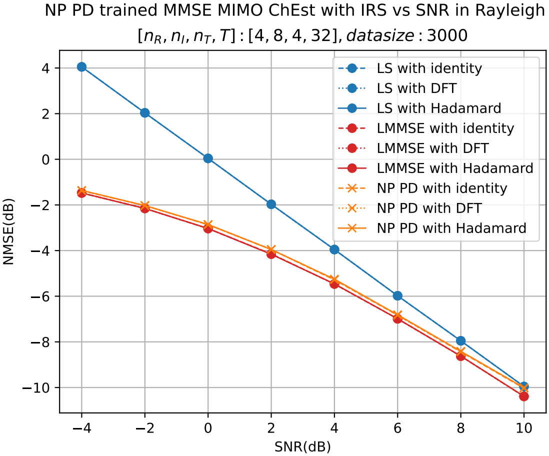


Figure: Increasing the training dataset size from 1M to 10M, comparing identity, DFT and Hadamard matrix for IRS coefficient.

Rician fading channel Configuration (1) (1/16)

[4] The distance-dependent path loss model is given by

$$L(d) = C_0 \left(\frac{d}{D_0} \right)^{-\alpha},$$

where C_0 is the path loss at the reference distance $D_0 = 1$ meter (m), d denotes the individual link distance, and α denotes the path loss exponent.

To account for small-scale fading, we assume the Rician fading channel model for all channels involved. Thus, the channel $\mathbf{G}$ is given by

$$\mathbf{G}_{B,I} = \sqrt{\frac{\beta_{B,I}}{1+\beta_{B,I}}}\mathbf{G}_{B,I}^{\text{LoS}} + \sqrt{\frac{1}{1+\beta_{B,I}}}\mathbf{G}_{B,I}^{\text{NLoS}},$$

[4] Q. Wu and R. Zhang, ”Intelligent Reflecting Surface Enhanced Wireless Network via Joint Active and Passive Beamforming,” in IEEE Transactions on Wireless Communications, vol. 18, no. 11, pp. 5394-5409, Nov. 2019.

Rician fading channel Configuration (1) (1/16)

where $\mathbf{G}_{B,I}^{\text{LoS}}$ and $\mathbf{G}_{B,I}^{\text{NLoS}}$ are the deterministic LoS and Rayleigh fading components, respectively, and $\beta_{B,I}$ is the Rician factor.

$$\mathbf{G}_{B,I}^{\text{LoS}} = \mathbf{a}_I(\theta^I, \phi^I)\mathbf{a}_B^H(\theta^B),$$

where $\mathbf{a}_I(\theta^I, \phi^I)$ and $\mathbf{a}_B(\theta^B)$ are the array response vectors at the IRS and BS, respectively.

$$\begin{aligned}\mathbf{a}_B(\theta^B) &= [1, e^{j2\pi d^B \sin(\theta^B)/\lambda}, \cdots, e^{j2\pi d^B (N_T-1)\sin(\theta^B)/\lambda}]^T,\\ \mathbf{a}_I(\theta^I, \phi^I) &= \mathbf{a}_{Ix}(\theta^I, \phi^I) \otimes \mathbf{a}_{Iz}(\theta^I, \phi^I),\end{aligned}$$

where

$$\begin{aligned}\mathbf{a}_{Ix}(\theta^I, \phi^I) &= [1, e^{j2\pi d^I \sin(\theta^I)\cos(\phi^I)/\lambda}, \cdots, e^{j2\pi d^I (N_{Ix}-1)\sin(\theta^I)\cos(\phi^I)/\lambda}]^T,\\ \mathbf{a}_{Iz}(\theta^I, \phi^I) &= [1, e^{j2\pi d^I \cos(\theta^I)\sin(\phi^I)/\lambda}, \cdots, e^{j2\pi d^I (N_{Iz}-1)\cos(\theta^I)\sin(\phi^I)/\lambda}]^T.\end{aligned}$$

where d_I, d_B are the represents the antenna spacing, λ is the wavelength, $\theta^I, \phi^I, \theta^B$ are the azimuth and elevation angles of the IRS and BS, respectively, and $N_I = N_{Ix}N_{Iz}$

Rician fading channel Configuration (2) (1/16)

Then the channel can be written as

$$\mathbf{H}_{B,I} = \sqrt{L(d_{B,I})}\left(\sqrt{\frac{\beta_{B,I}}{1+\beta_{B,I}}}\mathbf{G}_{B,I}^{\mathrm{LoS}} + \sqrt{\frac{1}{1+\beta_{B,I}}}\mathbf{G}_{B,I}^{\mathrm{NLoS}}\right) \tag{6}$$

Parameter	Value	Parameter	Value
N_R, N_I, N_T, T	4, 8, 4, 32	C_0	-30 (dB)
N_{Ix}, N_{Iz}	2, 4	α_{BI}, α_{IU}	2.0, 2.8
$\theta_B, \theta_I, \phi_I$	0 (radius)	Training dataset	10M
β_{BI}, β_{IU}	$\infty, 0$	Epoch	500
d_{BI}, d_{IU}	51, 3 (m)	Learning rate	10^{-4}

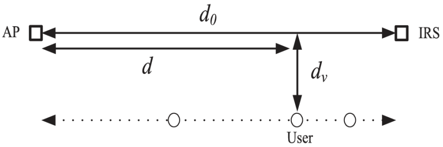


Figure: Simulation setup from [4].

Simulation Result under Rician fading result (1/16)

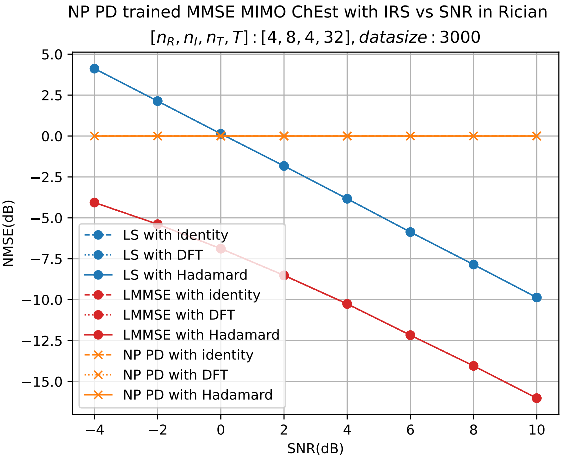


Figure: It didn't learn anything so far, I will try standard normalization method to see if it can help to improve the performance.

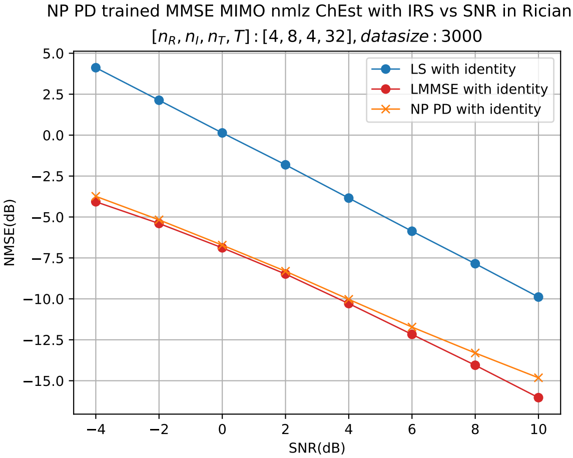
Discussion on 1/16

- By the larger dataset (10M) for solving the overfitting problem, the NN model can approach LMMSE in Rayleigh fading channel.
- In Rician fading channel (6), the NN model didn't learn anything so far. try to use standard normalization method to see if it can help to improve the performance.
- The channel model in Rician fading channel (6) didn't consider the frequency-dependent path loss model, shadowing effect, and micro/macro fading components like in [5] eq (26).
- Professor explained the path loss component and the shadow fading $\mathbf{s}$ and how it turn into the macro fading diagonal matrix $\mathbf{A}$ in (5) eq(26).
- Instead of the system configuration in (4), try to find out the circular configuration to see how the NN model performs.

[5] T. Chao, C. C. Fung, Z. -E. Ni and M. Servetnyk, "Joint Beamforming and Aerial IRS Positioning Design for IRS-Assisted MISO System With Multiple Access Points," in IEEE Open Journal of the Communications Society, vol. 5, pp. 612-632, 2024.

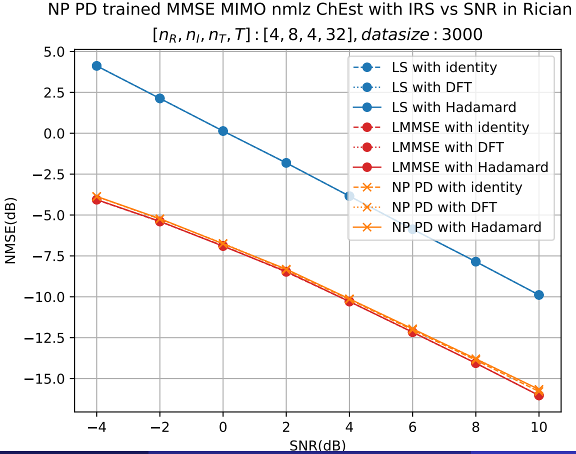
Simulation Result under Rician w/ 1M dataset (2/6)

Using the same standard normalization method as in the block diagram: fig(10),



Simulation Result under Rician w/ 10M dataset (2/6)

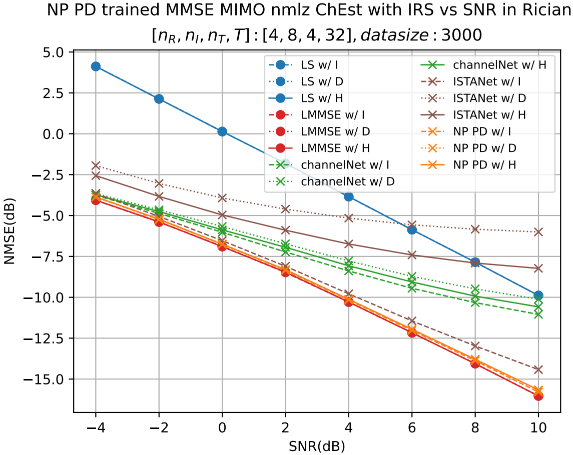
In this configuration, β_{BI}, β_{IU} are set to $\infty, 0$ respectively. Therefore $\mathbf{H}_{B,I}, \mathbf{H}_{I,U}$ are pure deterministic LoS and Gaussian distribution channels. Then the channel $\mathbf{H}_C = \mathbf{H}_{B,I}^T \circledast \mathbf{H}_{I,U}$ is also Gaussian distributed. Therefore, LMMSE solution here is equal to MMSE.



Simulation Result vs channelNet & ISTA-Net (2/13)

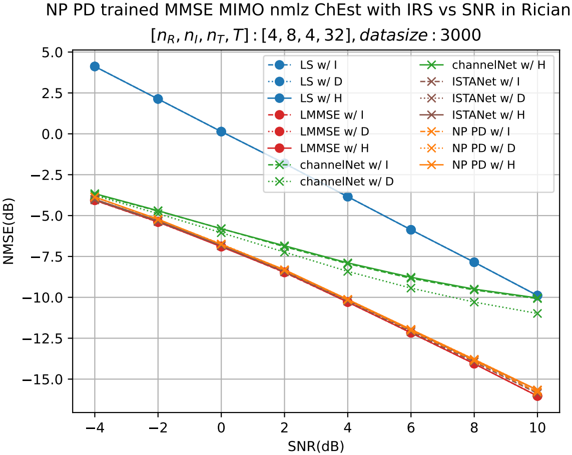
Using the same training dataset & standard normalization method to train both channelNet & ISTA-Net.

This result is incorrect because of the bug in the code.



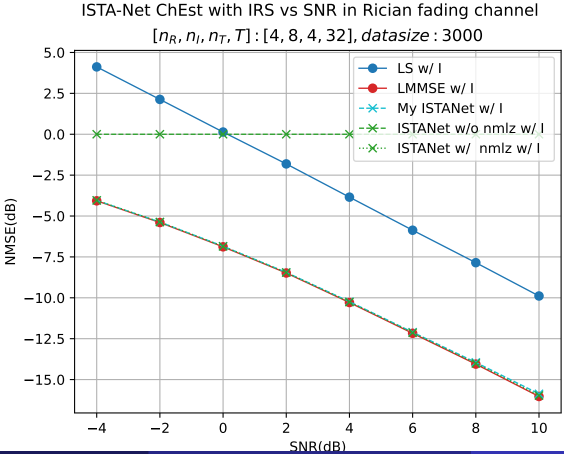
Simulation Result vs channelNet & ISTA-Net (2/25)

After fixing the bug in the code, the result is as follows.



Simulation Result vs channelNet & ISTA-Net (2/25)

Comparing my implementation (cyan dashed) with Natthakrit's (green dotted) to verify its correctness, as both almost overlap with LMMSE. It doesn't learn anything if the standard normalization method is not applied (green dashed).


ISTA-Net ChEst with IRS vs SNR in Rician fading channel
$[n_R, n_I, n_T, T] : [4, 8, 4, 32], datasize : 3000$
NMSE(dB)
SNR(dB)
LS w/ I
LMMSE w/ I
My ISTANet w/ I
ISTANet w/o nmlz w/ I
ISTANet w/ nmlz w/ I