Adaptive Neural Signal Detection for Massive MIMO

Presenter: Haoran Wan

Motivation

- Learning based MIMO detection algorithms have been proposed recently, such as **DetNet** and **OAMPNet**, and they are showing potential.

- However, **neither** approach is effective in practice.
    - DetNet’s training is unstable for realistic channels.
    - OAMPNet suffers a large performance gap (4-7dB at symbol error rate of 10^-3).

- Can a receiver optimize its detector for **every realization** of the channel matrix?

Key Ideas

- It uses a **neural network** architecture that strikes a balance between flexibility and complexity.
  - DetNet is too large for online training
  - OAMPNet performs poorly with online learning infrastructure.

- An online training algorithm that exploits the locality of channel matrices at receiver in both **frequency** and **time** domain.

Background

Received vector:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

The channel matrix **H** is assumed known, and the goal is:

$$\hat{\mathbf{x}} = arg\ min_{x \in \mathcal{X}^{N_t}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2$$

And it's an NP-hard problem.

Iterative Framework for MIMO Detection

This paper proposes that many MIMO detection algorithms can be transformed into an iterative form:

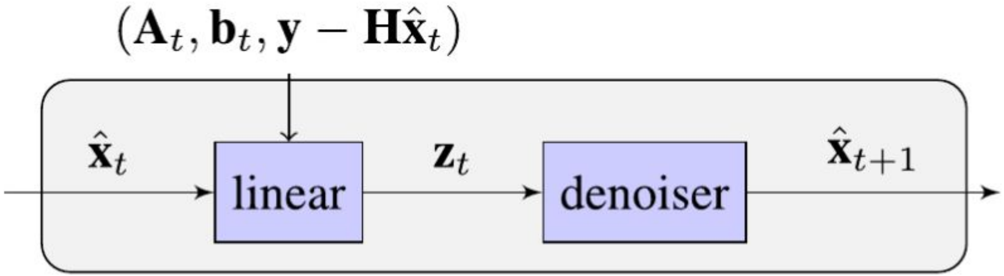
Residual error from the last round.

$$\mathbf{z}_t = \hat{\mathbf{x}}_t + \boxed{\mathbf{A}_t}\left(\boxed{\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_t}\right) + \boxed{\mathbf{b}_t}$$

$$\hat{\mathbf{x}}_{t+1} = \boxed{\eta_t}(\mathbf{z}_t)$$

Transformation in this round for estimation refinement.

Denoiser



Optimal Denoiser for Gaussian Noise

Input is $\mathbf{z}_t$, output is $\hat{\mathbf{x}}_{t+1}$, we want the final $\hat{\mathbf{x}}$ to be as close to $\mathbf{x}$as possible.

$$arg\ min_{\eta_0,\ldots,\eta_M}\mathbb{E}[||\hat{\mathbf{x}}-\mathbf{x}||_2|\mathbf{z}_0,\ldots,\mathbf{z}_{M+1}]$$

For each specific denoiser, we want:

$$arg\ min_{\eta_t}\mathbb{E}[||\hat{\mathbf{x}}-\mathbf{x}||_2|\mathbf{z}_t]$$

So the output of the denoiser should be:

$$\mathbb{E}[||\mathbb{E}[\mathbf{x}]-\mathbf{x}||_2]=0$$

$$\eta_t(\mathbf{z}_t)=\mathbb{E}[\mathbf{x}|\mathbf{z}_t]$$

Optimal Denoiser for Gaussian Noise

If we assume that $\mathbf{z}_t - \boldsymbol{x}$ has an i.i.d. Gaussian distribution with diagonal covariance matrix $\sigma_t^2 \mathbf{I}_{N_t}$, each element of the output should be $\mathbb{E}[x_i|z]$.

$$z - x_i \sim N(0, \sigma_t)$$

$$x_i \sim N(z, \sigma_t)$$

So the output of the denoiser should be

$$\beta_t(z; \sigma_t^2) = \frac{1}{Z} \sum_{x_i \in \mathcal{X}} x_i \exp\left(-\frac{\|z - x_i\|^2}{\sigma_t^2}\right)$$

$$Z = \sum_{x_j \in \mathcal{X}} \exp\left(-\frac{\|z - x_j\|^2}{\sigma_t^2}\right)$$

DetNet and OAMPNet

DetNet:

$$\mathbf{q}_t = \hat{\mathbf{x}}_{t-1} - \theta_t^{(1)}\mathbf{H}^H\mathbf{y} + \theta_t^{(2)}\mathbf{H}^H\mathbf{H}\hat{\mathbf{x}}_{t-1}$$

$$\mathbf{u}_t = [\Theta_t^{(3)}\mathbf{q}_t + \Theta_t^{(4)}\mathbf{v}_{t-1} + \theta_t^{(5)}]_+$$

$$\mathbf{v}_t = \Theta_t^{(6)}\mathbf{u}_t + \theta_t^{(7)}$$

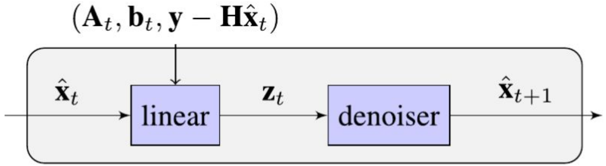
$$\hat{\mathbf{x}}_t = \Theta_t^{(8)}\mathbf{u}_t + \theta_t^{(9)}$$

OAMPNet:

$$\mathbf{z}_t = \hat{\mathbf{x}}_t + \theta_t^{(1)}\mathbf{H}^H(v_t^2\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I})^{(-1)}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_t)$$

$$\hat{\mathbf{x}}_{t+1} = \eta_t(\mathbf{z}_t; \sigma_t^2)$$

MMNet Design – i.i.d. Gaussian Channel



The model is

$$\mathbf{z}_t = \hat{\mathbf{x}}_t + \theta_t^{(1)} \mathbf{H}^H (\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_t)$$

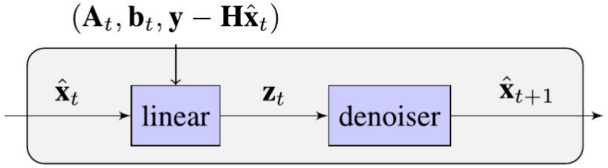
The denoiser is

$$\hat{\mathbf{x}}_{t+1} = \eta_t(\mathbf{z}_t; \sigma_t^2)$$

$$\mathbf{A}_t = \theta_t^{(1)} \mathbf{H}^H$$

$$\sigma_t^2 = \frac{\theta_t^{(2)}}{N_t} \left(\frac{\|\mathbf{I} - \mathbf{A}_t \mathbf{H}\|_F^2}{\|\mathbf{H}\|_F^2} \left[\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_t\|_2^2 - N_r \sigma^2 \right]_+ + \frac{\|\mathbf{A}_t\|_F^2}{\|\mathbf{H}\|_F^2} \sigma^2 \right)$$

MMNet Design – i.i.d. Arbitrary Channel



The model is

$$\mathbf{z}_t = \hat{\mathbf{x}}_t + \Theta_t^{(1)}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_t)$$

The denoiser is

$$\hat{\mathbf{x}}_{t+1} = \eta_t(\mathbf{z}_t; \sigma_t^2)$$

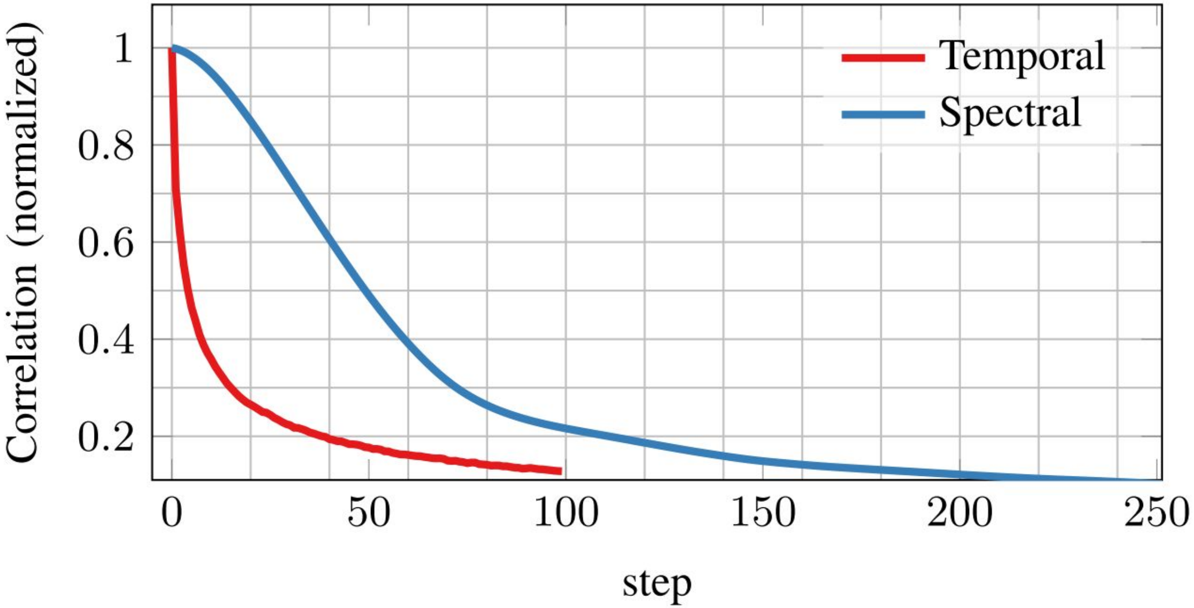
$$\mathbf{A}_t = \boldsymbol{\Theta}_t^{(1)}$$

$$\boldsymbol{\sigma}_t^2 = \frac{\boldsymbol{\theta}_t^{(2)}}{N_t}\left(\frac{\|\mathbf{I} - \mathbf{A}_t\mathbf{H}\|_F^2}{\|\mathbf{H}\|_F^2}\left[\|\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}_t\|_2^2 - N_r\sigma^2\right]_+ + \frac{\|\mathbf{A}_t\|_F^2}{\|\mathbf{H}\|_F^2}\sigma^2 \right)$$

Online Training – Channel Locality to Reduce Complexity

Temporal: c … nearby points are co…

Spectral: dif… the frequencies …



Both forms of channel locality reduce the cost of training.

Online Trai

Algorithm 1 MMNet online training

```
M ← Construct MMNet with parameters ϑ = {Θ_t^(1), θ_t^(2)}_{t=1}^T
ϑ ← Initialize model parameters randomly
for n ∈ {1, 2, ...} do                                  ▷ n keeps the time step
    for f ∈ {1, 2, ..., F} do
        H[f] ← Measured channel at time step n and frequency f
        #TrainIterations ← Set to Φ if f ≠ 1 and Ψ otherwise
        for it∈ {1, 2, ..., #TrainIterations} do
            D ← Generate random (x, y) batch on H[f] using (1)
            L ← Find the loss in (15) for M over the samples in D
            ϑ ← Compute the model updates using ∇_ϑ L
        end for
        M_n[f] ← M.copy( )                               ▷ Store the parameters ϑ
    end for
end for
```

n is the interval that the channel stays the same, a.k.a coherence time.

$\Phi = 3 - 10, \Psi = 1000$

Computational Complexity

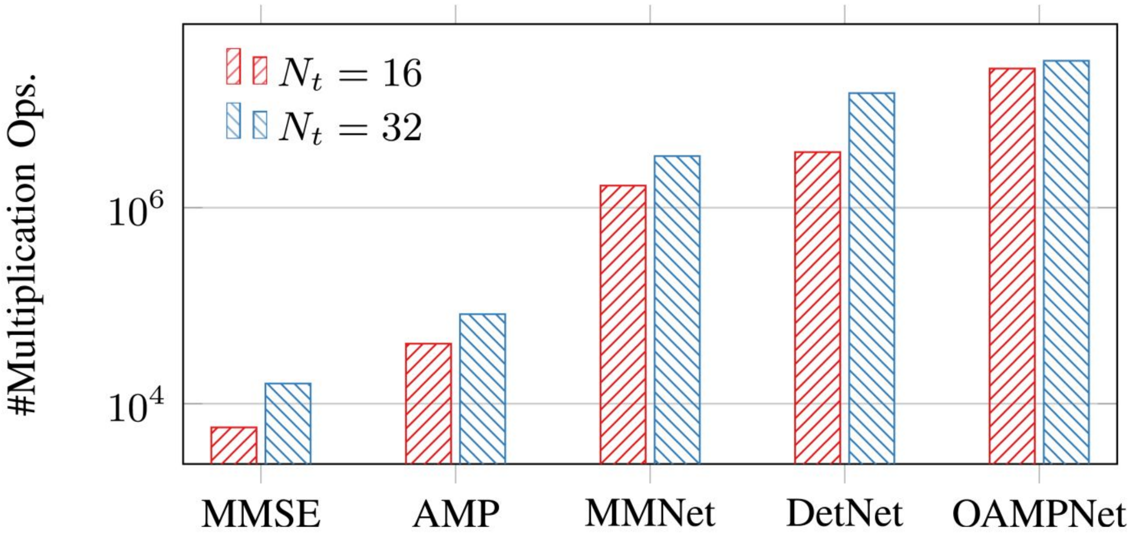
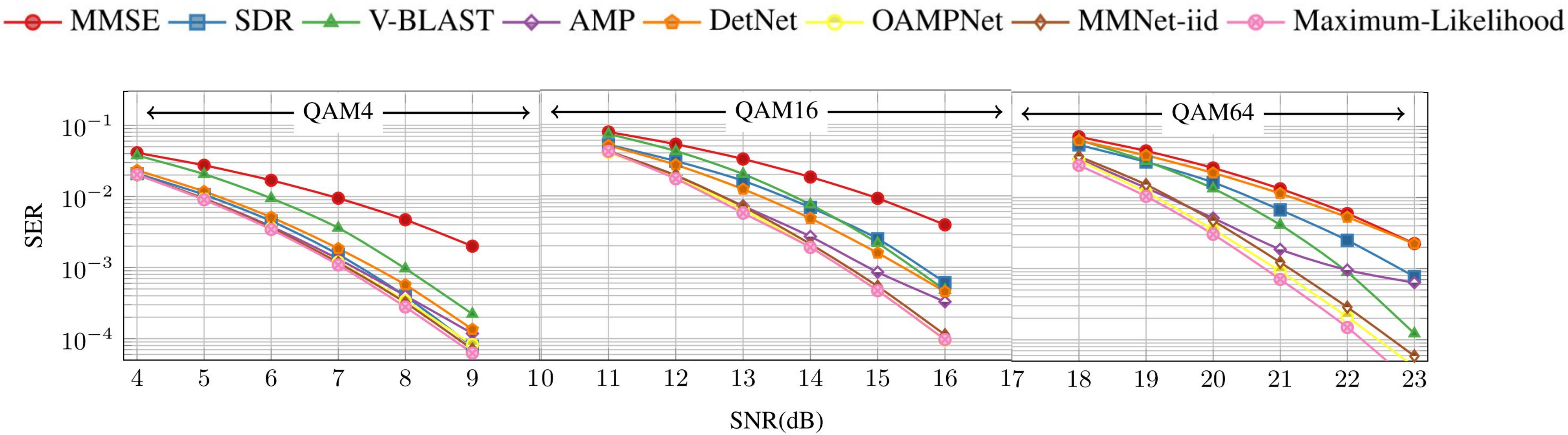


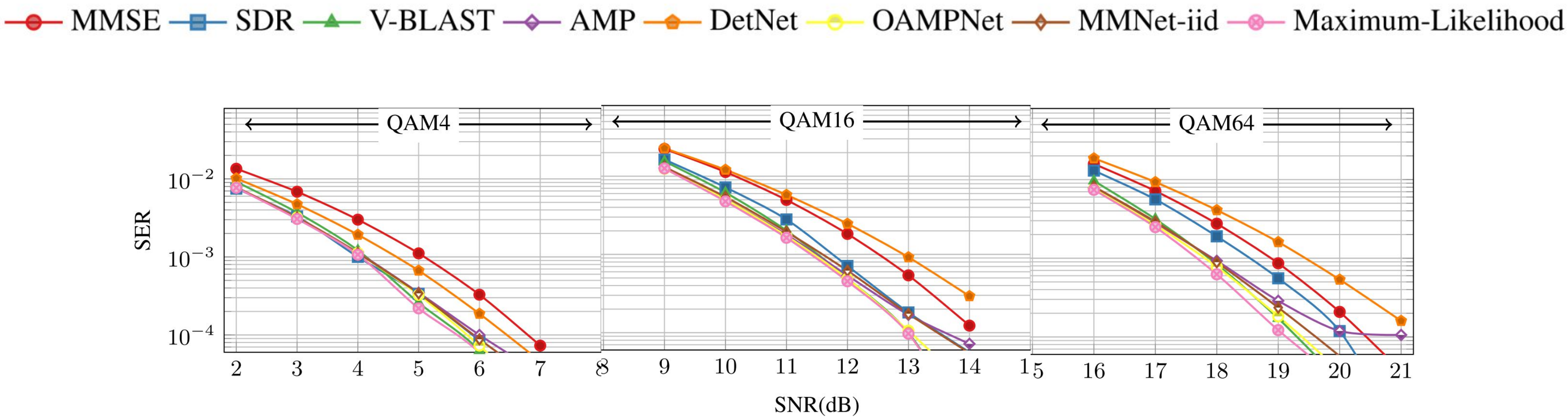
Fig. 12. Number of multiplication operations per signal detection for different algorithms on QAM16 with $N_r = 64$ receive antennas in 3GPP MIMO model. Detection with MMNet, including its online training process, requires fewer multiplication operations than detection with pre-trained DetNet and OAMPNet models.

Result – i.i.d. Gaussian Channels



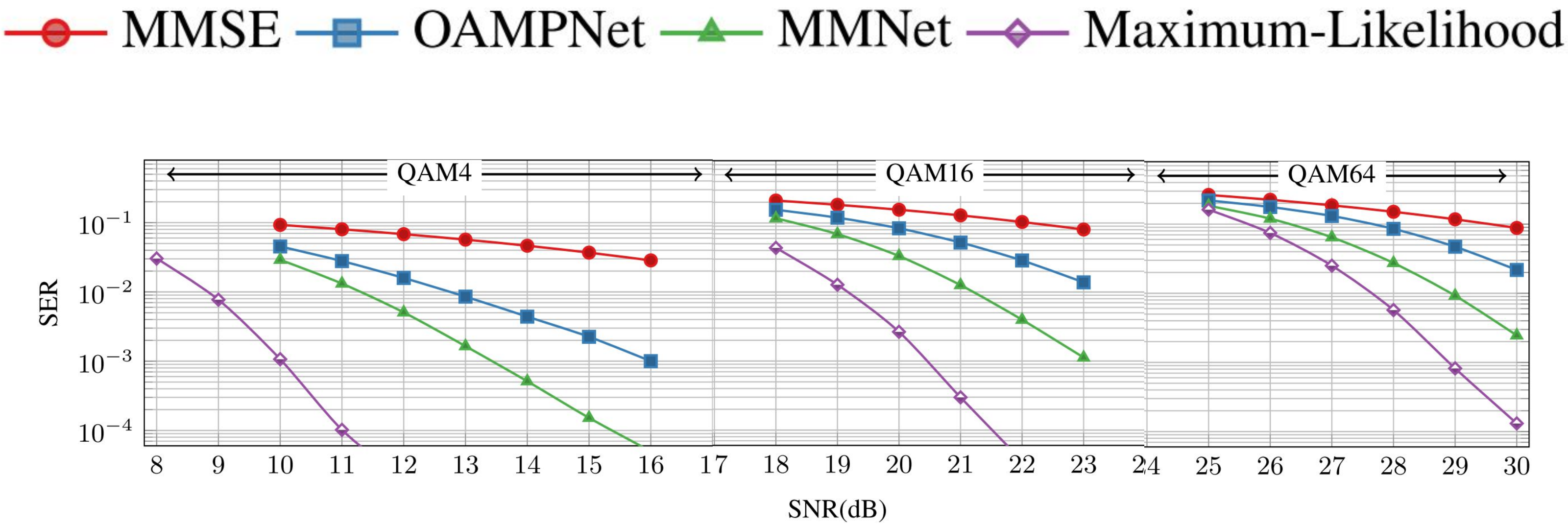
32 Tx, 64 Rx

Result – i.i.d. Gaussian Channels



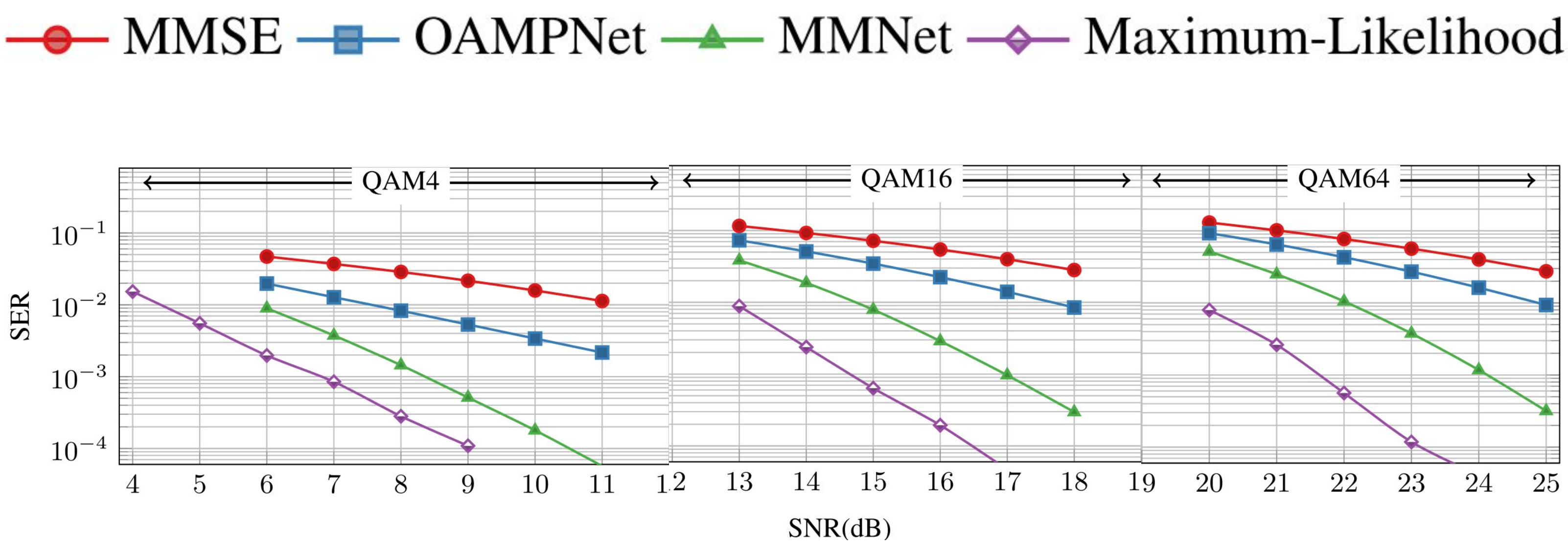
16 Tx, 64 Rx

Result – Realistic Channel



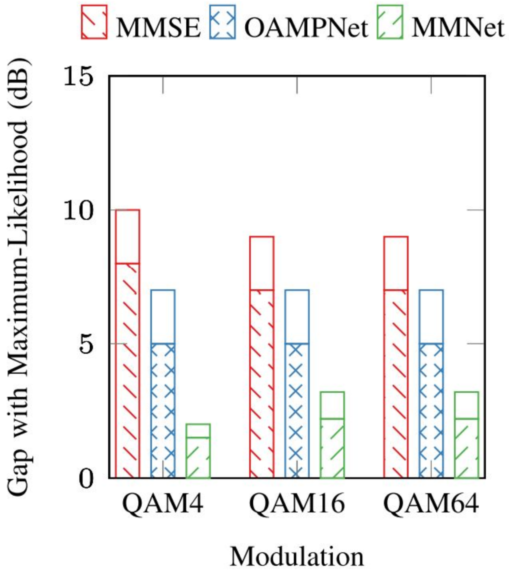
32 Tx, 64 Rx

Result – Realistic Channel

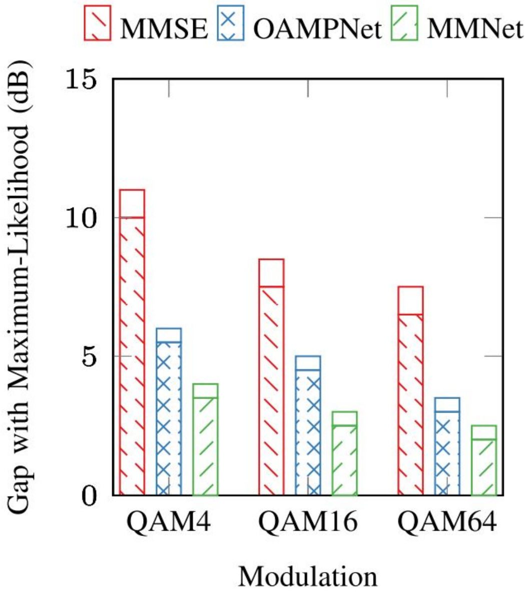


16 Tx, 64 Rx

SNR Gap from the Maximum Likelihood



(a) 3GPP MIMO $N_t = 16$



(b) 3GPP MIMO $N_t = 32$

To achieve a 10^-3 SER, how much more SNR the methods require, compared with ML

Robustness from Channel Estimation Errors

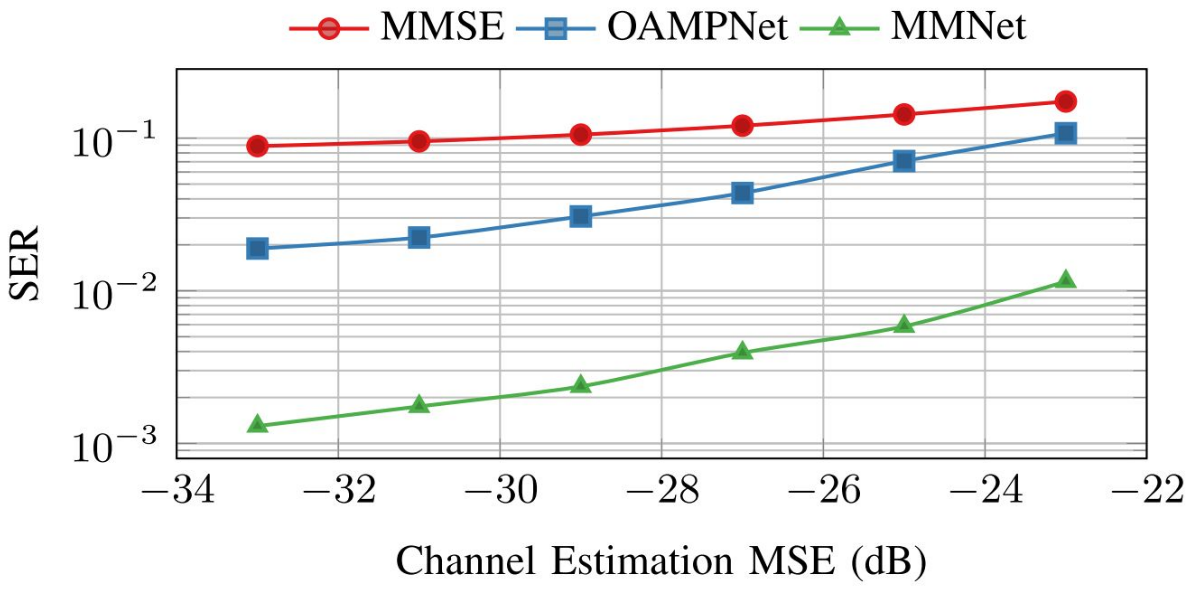


Fig. 5. SER for QAM16 versus channel estimation MSE.

Why MMNet Works

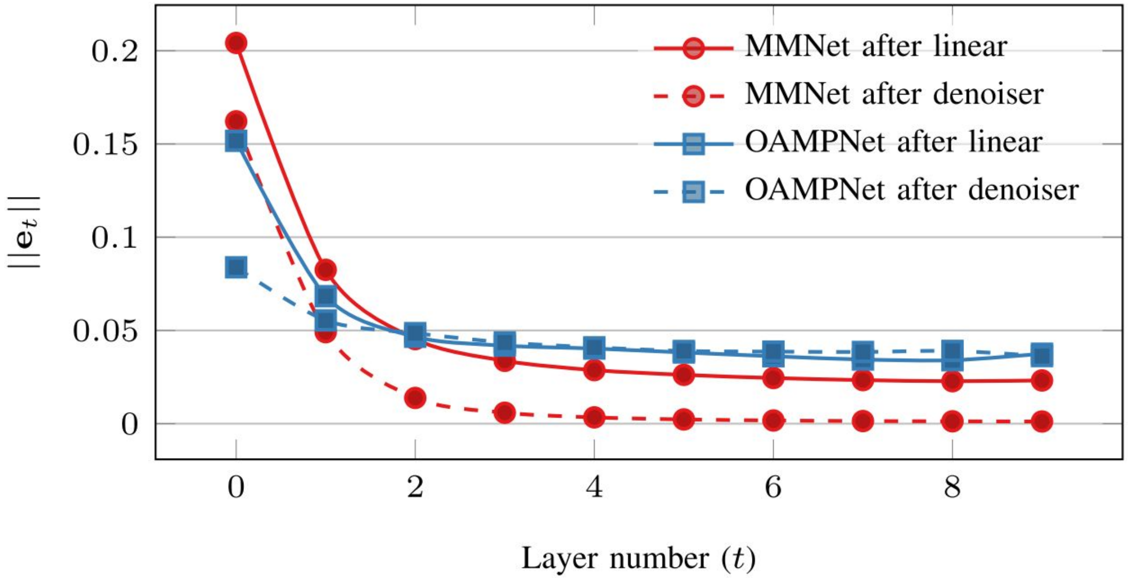


Fig. 6. Noise power after the linear and denoiser stages at different layers of OAMPNet and MMNet. The OAMPNet denoisers become ineffective after the third layer on 3GPP MIMO channels.

Opinion

This paper is well-developed in structure and the resource (GPU) it requires is well-fit in this generation (compared with the next generation Quantum methods).

The online training method considers both effectiveness and the actual deployment computational burden, which is good.

The structure of this paper is a bit of weird – they put the online training section way behind, which could help us understand if it’s in earlier section.

Discussion

https://app.perusall.com/courses/cos597s_f2024-advanced-topics-in-computer-science-recent-advances-in-wireless-networks/mmnet?assignmentId=fM76uikFDdRRvsZJj&part=1