

Efficient Deep Learning for Massive MIMO Channel State Estimation



Mason del Rosario
Doctoral Exit Seminar

September 2022

Background

Role of CSI in MIMO

CSI Estimation

Compressed Sensing

Convolutional Neural Networks

Completed Work #1: SphNet

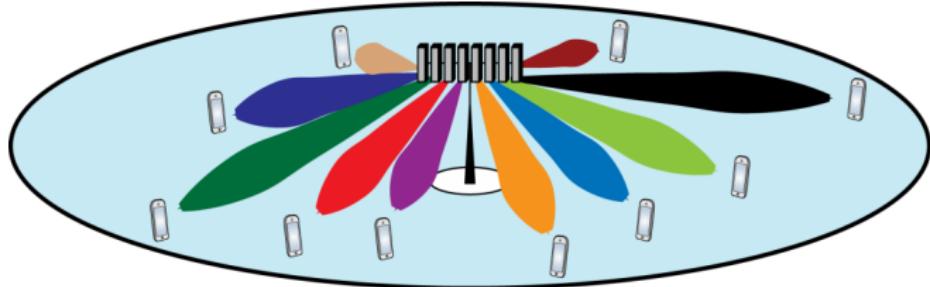
Completed Work #2: MarkovNet

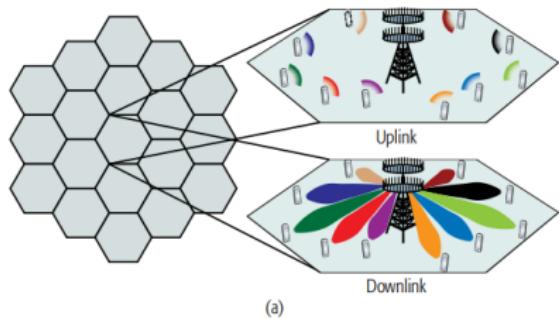
Completed Work #3: Pilots-to-delay Estimator (P2DE) and
Heterogeneous Differential Encoding

Current Work: Pilot Feedback and Model Re-use

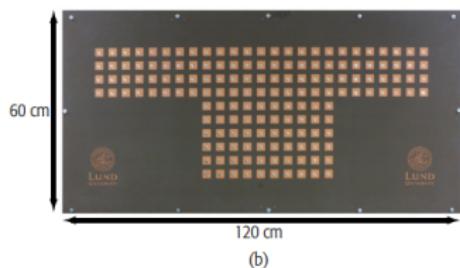
Background

Feedback-based estimation of channel state information in MIMO networks.



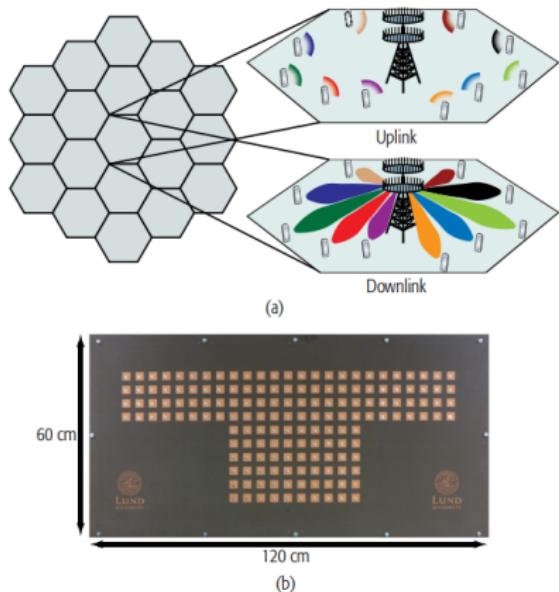


(a)



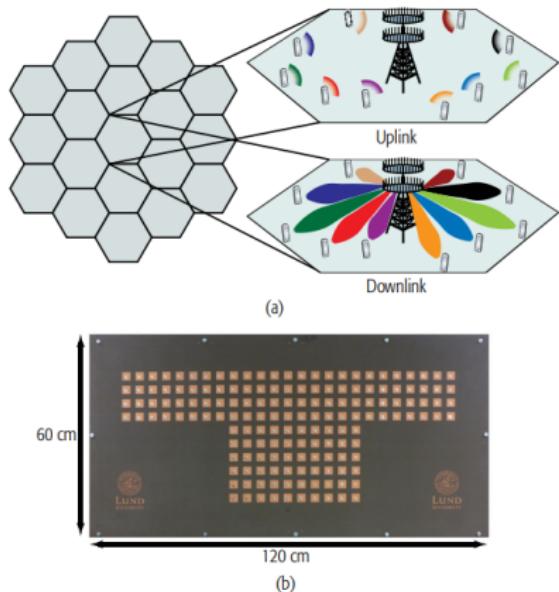
(b)

- ▶ MIMO = Multiple input multiple output



- ▶ MIMO = Multiple input multiple output
- ▶ Massive w.r.t. antenna count, not physical size.

E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten Myths and One Critical Question," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, 2016



- ▶ MIMO = Multiple input multiple output
- ▶ Massive w.r.t. antenna count, not physical size.
- ▶ Spatial diversity → **high throughput.**

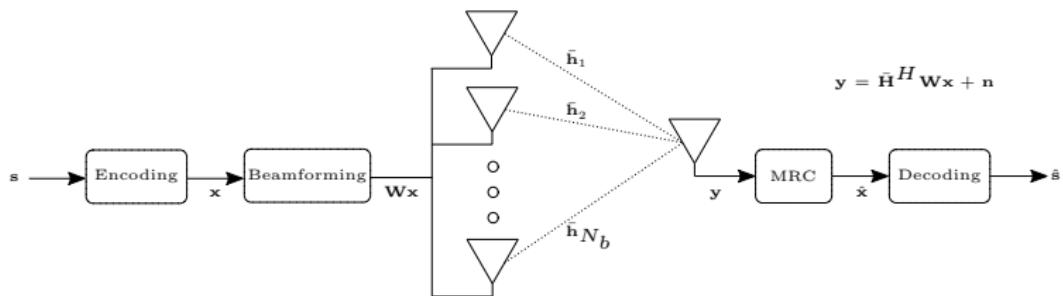


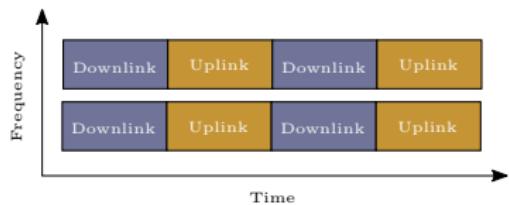
Figure: Multi-antenna transmitter (BS, gNB) and single-antenna user equipment (UE) with relevant system values.

In OFDM, the fading coefficients between Tx/Rx = **Channel State Information (CSI)**, $\bar{\mathbf{H}}$.

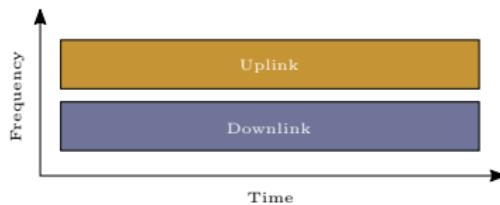
$$\bar{\mathbf{H}} = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,N_f} \\ h_{2,1} & h_{2,2} & \dots & h_{2,N_f} \\ \vdots & \vdots & \vdots & \vdots \\ h_{N_b,1} & h_{N_b,2} & \dots & h_{N_b,N_f} \end{bmatrix} \in \mathbb{C}^{N_b \times N_f}$$

For N_b transmit antennas and N_f subcarriers.

Downlink-uplink reciprocity in TDD, but not in FDD.

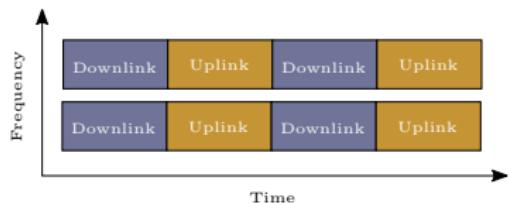


a) Time division duplex (TDD)

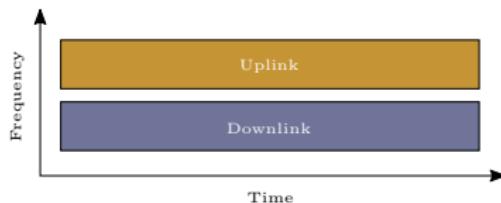


b) Frequency division duplex (FDD)

Downlink-uplink reciprocity in TDD, but not in FDD.



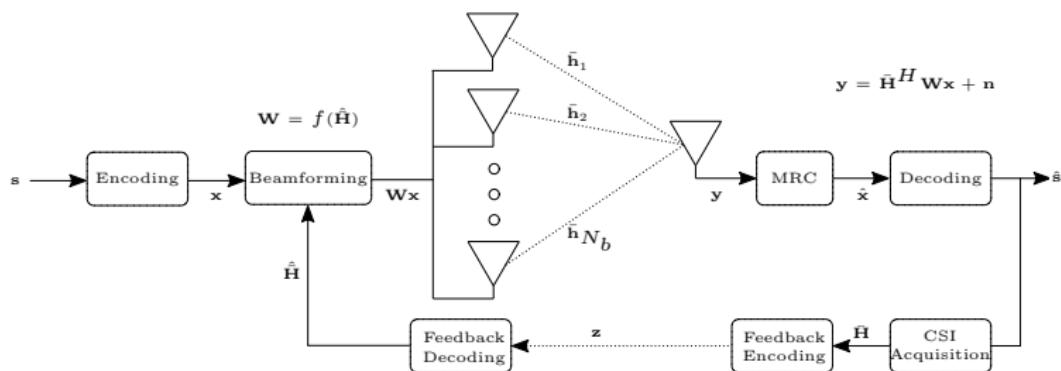
a) Time division duplex (TDD)



b) Frequency division duplex (FDD)

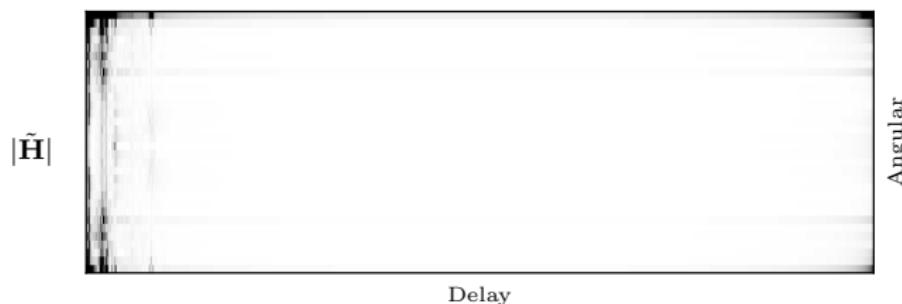
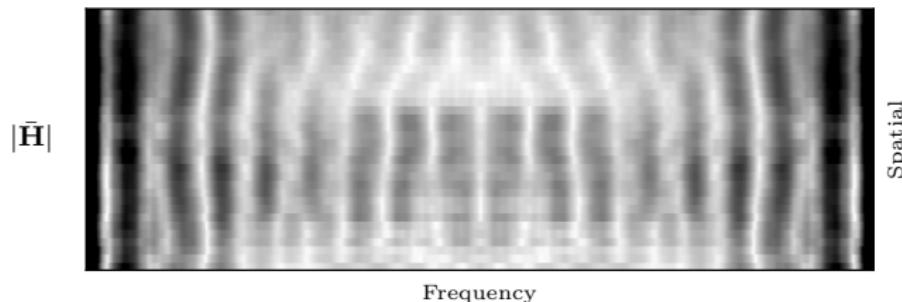
FDD requires feedback for downlink CSI estimation.

Transmitting $\bar{\mathbf{H}}$ is costly. Instead, generate estimates, $\hat{\mathbf{H}}$, based on **compressed feedback**, \mathbf{z} .

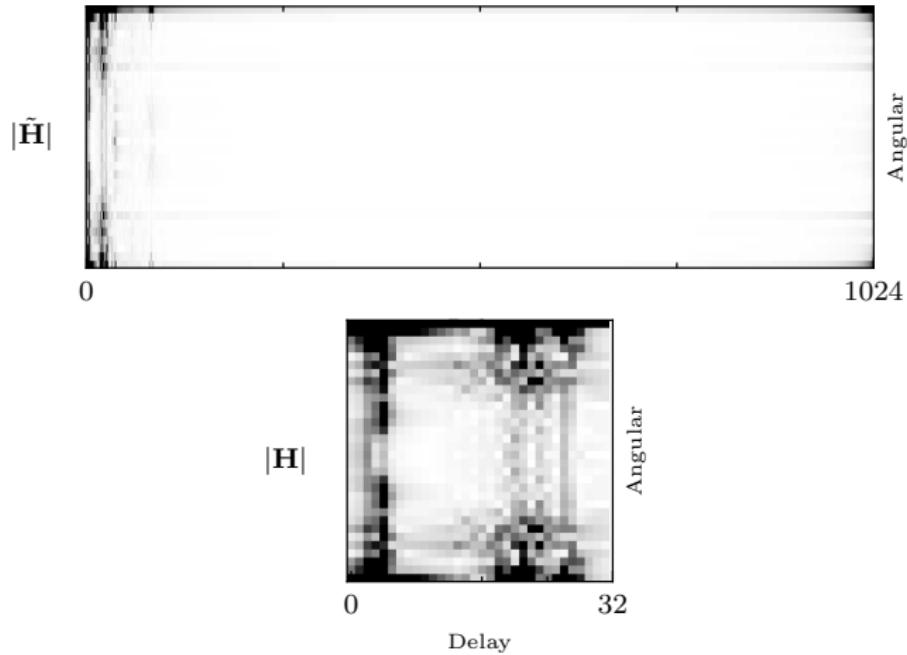


Denote 2D inverse FFT of $\bar{\mathbf{H}}$ as

$$\tilde{\mathbf{H}} = \mathbf{F}^H \bar{\mathbf{H}} \mathbf{F}.$$



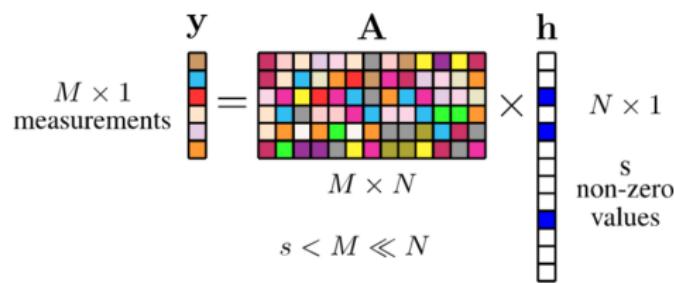
Given sparsity of $\tilde{\mathbf{H}}$, we can encode/decode a truncated version, \mathbf{H} .



1. Compressed Sensing (Conventional)
2. Convolutional Neural Networks (This thesis)

Find low-dimensional basis for sparse data, \mathbf{h} ,

$$\mathbf{y} = \mathbf{A}\mathbf{h} + \mathbf{n}.$$

$$\begin{matrix} \mathbf{y} \\ M \times 1 \\ \text{measurements} \end{matrix} = \begin{matrix} \mathbf{A} \\ M \times N \\ s < M \ll N \end{matrix} \times \begin{matrix} \mathbf{h} \\ N \times 1 \\ s \text{ non-zero values} \end{matrix}$$


CS addresses two major issues:

1. Design of **A** (stochastic or deterministic).

CS addresses two major issues:

1. Design of \mathbf{A} (stochastic or deterministic).
2. Recovery of $\hat{\mathbf{h}}$ given \mathbf{A} and \mathbf{y} , typically via convex optimization on p -norm minimization,

$$\min \|\hat{\mathbf{h}}\|_p \text{ subject to } \|\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}\|_2^2 < \epsilon.$$

CS addresses two major issues:

1. Design of \mathbf{A} (stochastic or deterministic).
2. Recovery of $\hat{\mathbf{h}}$ given \mathbf{A} and \mathbf{y} , typically via convex optimization on p -norm minimization,

$$\min \|\hat{\mathbf{h}}\|_p \text{ subject to } \|\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}\|_2^2 < \epsilon.$$

Problems:

- Recovery algorithms are iterative.

CS addresses two major issues:

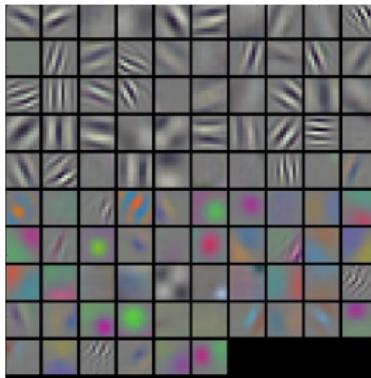
1. Design of \mathbf{A} (stochastic or deterministic).
2. Recovery of $\hat{\mathbf{h}}$ given \mathbf{A} and \mathbf{y} , typically via convex optimization on p -norm minimization,

$$\min \|\hat{\mathbf{h}}\|_p \text{ subject to } \|\mathbf{y} - \mathbf{A}\hat{\mathbf{h}}\|_2^2 < \epsilon.$$

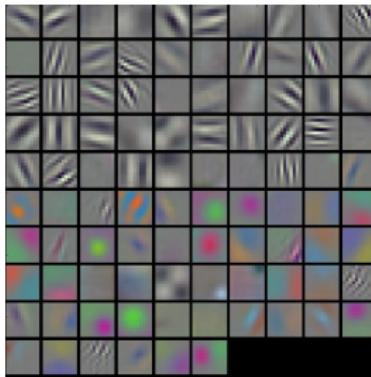
Problems:

- ▶ Recovery algorithms are iterative.
- ▶ Complexity scales with sparsity ($M \propto s$).

- ▶ Layers of trainable linear functions followed by nonlinear ‘activation’ functions.
- ▶ State-of-the art performance in image processing



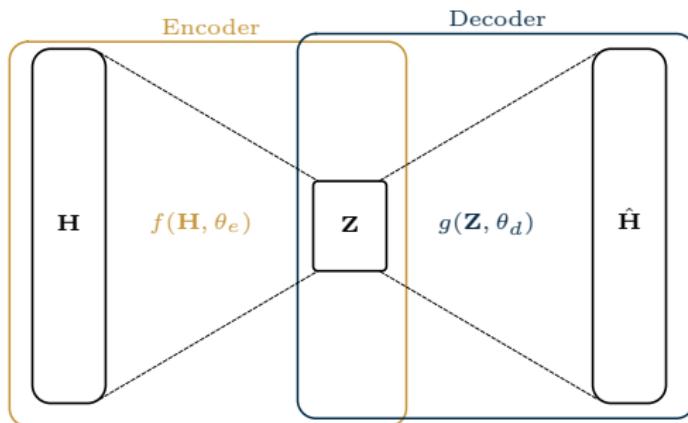
- ▶ Layers of trainable linear functions followed by nonlinear ‘activation’ functions.
- ▶ State-of-the art performance in image processing



- ▶ Instantaneous decoding (non-iterative).

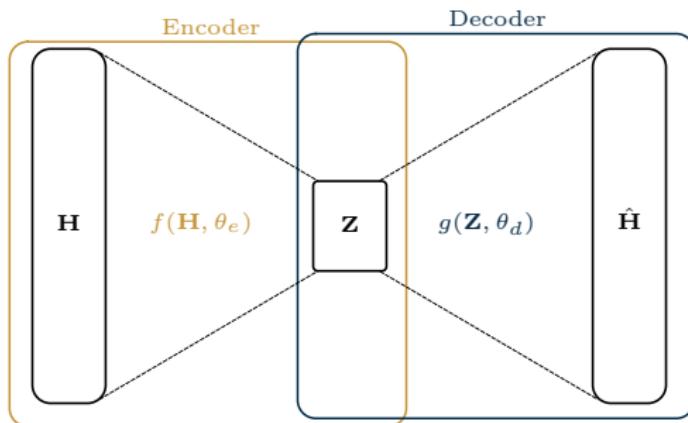
Autoencoder: Estimate $\hat{\mathbf{H}}$, latent code \mathbf{Z} with **compression ratio**,

$$\text{CR} = \frac{\dim(\mathbf{Z})}{\dim(\mathbf{H})} \text{ s.t. } \dim(\mathbf{Z}) < \dim(\mathbf{H}).$$



Autoencoder: Estimate $\hat{\mathbf{H}}$, latent code \mathbf{Z} with **compression ratio**,

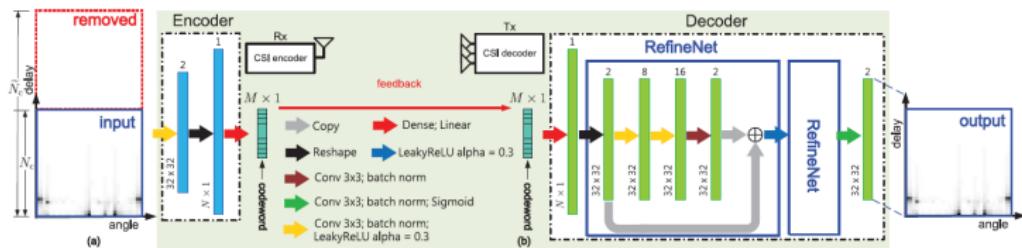
$$\text{CR} = \frac{\dim(\mathbf{Z})}{\dim(\mathbf{H})} \text{ s.t. } \dim(\mathbf{Z}) < \dim(\mathbf{H}).$$



θ_e, θ_d updated to minimize **mean-squared error (MSE)**,

$$\operatorname{argmin}_{\theta_e, \theta_d} \frac{1}{N} \sum_{i=1}^N \|\mathbf{H}_i - g(f(\mathbf{H}_i, \theta_e), \theta_d)\|^2.$$

- CNN autoencoder for learned CSI compression and feedback [3]



CNNs outperform CS at comparable compression ratios.

γ	Methods	Indoor		Outdoor	
		NMSE	ρ	NMSE	ρ
1/4	LASSO	-7.59	0.91	-5.08	0.82
	BM3D-AMP	-4.33	0.80	-1.33	0.52
	TVAL3	-14.87	0.97	-6.90	0.88
	CS-CsiNet	-11.82	0.96	-6.69	0.87
	CsiNet	-17.36	0.99	-8.75	0.91
1/16	LASSO	-2.72	0.70	-1.01	0.46
	BM3D-AMP	0.26	0.16	0.55	0.11
	TVAL3	-2.61	0.66	-0.43	0.45
	CS-CsiNet	-6.09	0.87	-2.51	0.66
	CsiNet	-8.65	0.93	-4.51	0.79
1/32	LASSO	-1.03	0.48	-0.24	0.27
	BM3D-AMP	24.72	0.04	22.66	0.04
	TVAL3	-0.27	0.33	0.46	0.28
	CS-CsiNet	-4.67	0.83	-0.52	0.37
	CsiNet	-6.24	0.89	-2.81	0.67
1/64	LASSO	-0.14	0.22	-0.06	0.12
	BM3D-AMP	0.22	0.04	25.45	0.03
	TVAL3	0.63	0.11	0.76	0.19
	CS-CsiNet	-2.46	0.68	-0.22	0.28
	CsiNet	-5.84	0.87	-1.93	0.59

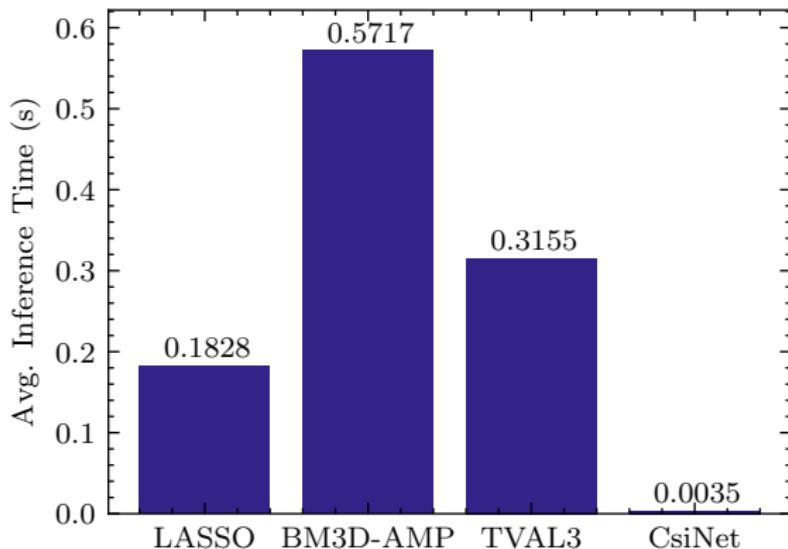


Figure: Average inference time for compressed sensing methods vs. CsiNet.

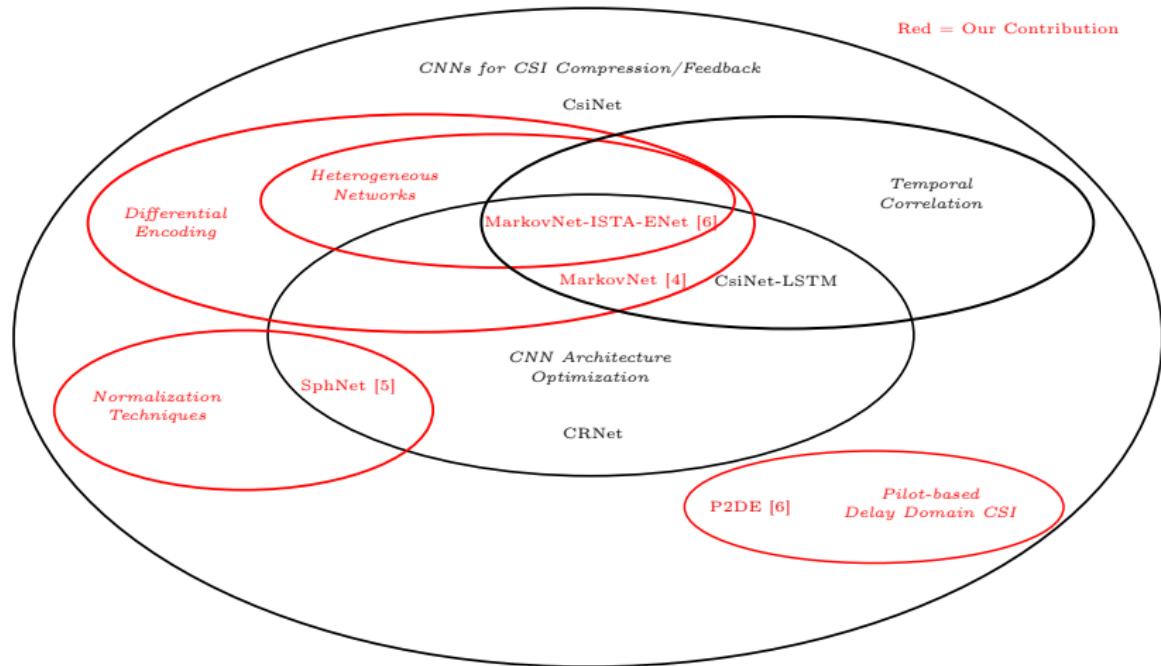
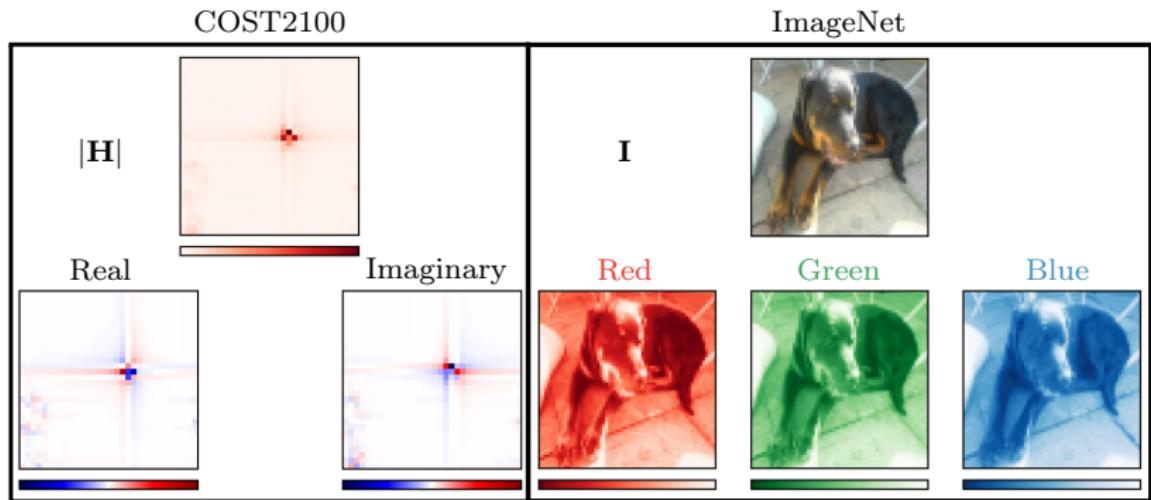
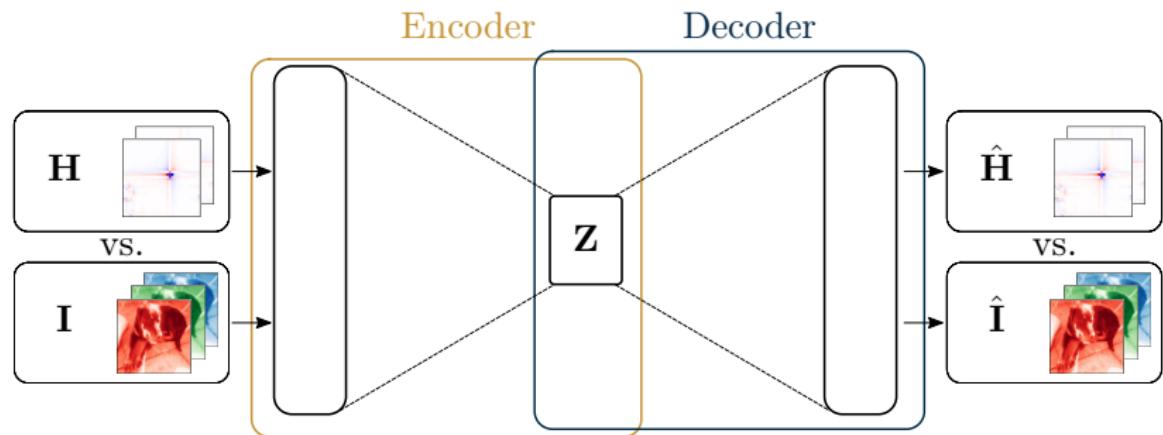


Figure: Areas of *domain knowledge* and corresponding CNNs.

Completed Work #1: SphNet

Power-based normalization for improved CSI reconstruction accuracy.





- ▶ **Minmax normalization** – Find minimum, maximum of channels.

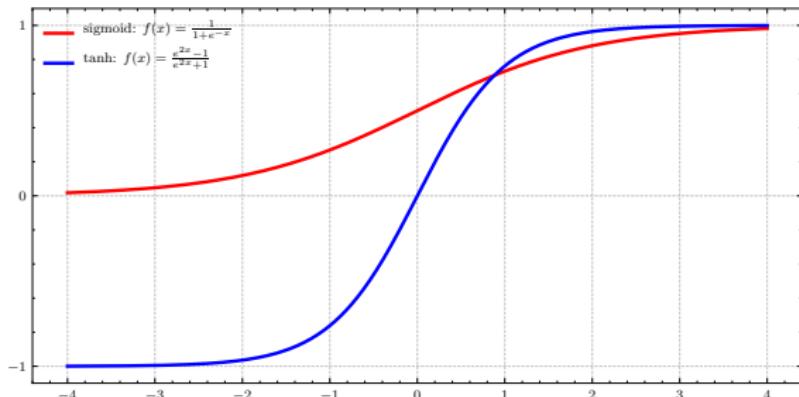
- ▶ **Minmax normalization** – Find minimum, maximum of channels.
- ▶ $H_{n,(i,j)} = (i,j)$ -th element of n -th sample

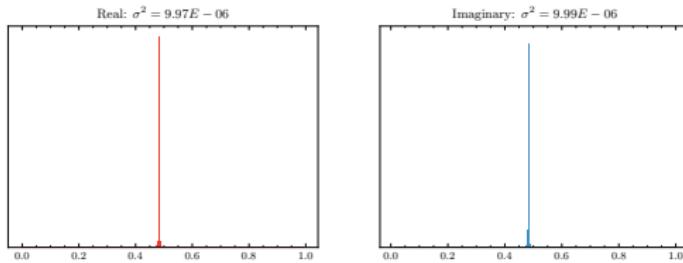
$$H_{\min\max,n,(i,j)} = \frac{H_{n,(i,j)} - H_{\min}}{H_{\max} - H_{\min}} \in [0, 1]$$

- ▶ **Minmax normalization** – Find minimum, maximum of channels.
- ▶ $H_{n,(i,j)} = (i,j)$ -th element of n -th sample

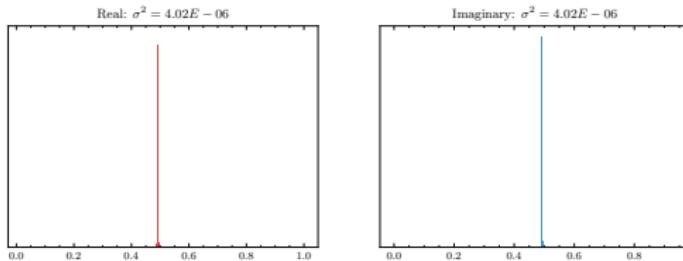
$$H_{\text{minmax},n,(i,j)} = \frac{H_{n,(i,j)} - H_{\min}}{H_{\max} - H_{\min}} \in [0, 1]$$

- ▶ Compatible with common **activation functions** (e.g., tanh, sigmoid)





(a) Indoor



(b) Outdoor

Figure: Distribution/variance of COST2100 real/imaginary channels under minmax normalization ($N = 10^5$).

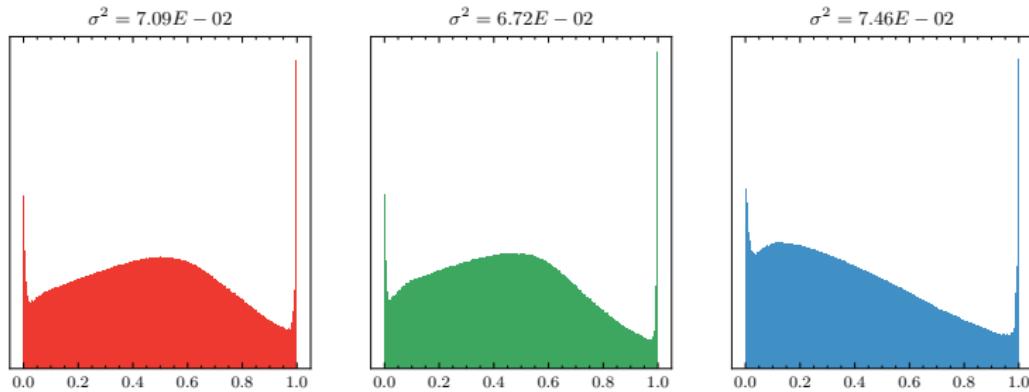


Figure: Distribution and variance of minmax-normalized ImageNet RGB channels ($N = 50000$).

Difference of four orders of magnitude.

Dataset	Env	Channels	Norm	Avg. Variance
ImageNet	-	RGB	Minmax	$7.09E^{-2}$
COST2100	Indoor	Real, Imag	Minmax	$9.98E^{-6}$
COST2100	Outdoor	Real, Imag	Minmax	$4.02E^{-6}$

Table: Minmax normalization applied to COST2100 and ImageNet dataset.

Spherical normalization – scale \mathbf{H} by power. For Frobenius norm $\|\cdot\|$,

$$\check{\mathbf{H}}^n = \frac{\mathbf{H}^n}{\|\mathbf{H}^n\|}. \quad (1)$$

Then apply minmax scaling to the entire dataset.

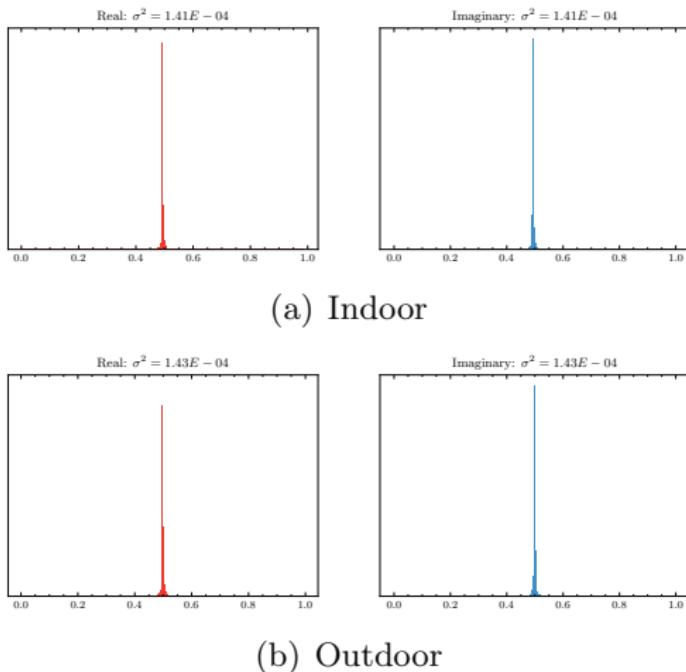


Figure: Distribution/variance of COST2100 real/imaginary channels under spherical normalization ($N = 10^5$).

Difference is now **two orders of magnitude**.

Dataset	Env	Channels	Norm	Avg. Variance
ImageNet	-	RGB	Minmax	$7.09E^{-2}$
COST2100	Indoor	Real, Imag	Spherical	$1.41E^{-4}$
COST2100	Outdoor	Real, Imag	Spherical	$1.43E^{-4}$
COST2100	Indoor	Real, Imag	Minmax	$9.98E^{-6}$
COST2100	Outdoor	Real, Imag	Minmax	$4.02E^{-6}$

Table: Minmax vs. spherical normalization applied to COST2100 datasets compared with ImageNet.

Spherical normalization → MSE equivalent to NMSE.

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N \|\mathbf{H}_k - \hat{\mathbf{H}}_k\|^2, \quad \text{NMSE} = \frac{1}{N} \sum_{k=1}^N \frac{\|\mathbf{H}_k - \hat{\mathbf{H}}_k\|^2}{\|\mathbf{H}_k\|}$$

Spherical normalization → MSE equivalent to NMSE.

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^N \|\mathbf{H}_k - \hat{\mathbf{H}}_k\|^2, \quad \text{NMSE} = \frac{1}{N} \sum_{k=1}^N \frac{\|\mathbf{H}_k - \hat{\mathbf{H}}_k\|^2}{\|\mathbf{H}_k\|}$$

MSE of spherically normalized estimator yields,

$$\begin{aligned}\text{MSE}_{\text{Sph}} &= \frac{1}{N} \sum_{k=1}^N \|\check{\mathbf{H}}_k - \hat{\check{\mathbf{H}}}_k\|^2 \\ &= \frac{1}{N} \sum_{k=1}^N \left\| \frac{\mathbf{H}_k}{\|\mathbf{H}_k\|} - \frac{\hat{\mathbf{H}}_k}{\|\mathbf{H}_k\|} \right\|^2 \\ &= \frac{1}{N} \sum_{k=1}^N \frac{\|\mathbf{H}_k - \hat{\mathbf{H}}_k\|^2}{\|\mathbf{H}_k\|}.\end{aligned}$$

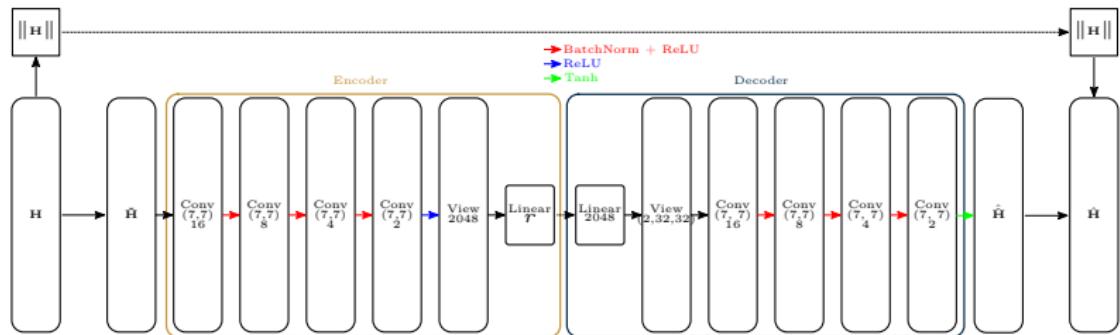


Figure: SphNet – CsiNetPro architecture with Spherical Normalization.

Z. Liu, M. del Rosario, X. Liang, L. Zhang, and Z. Ding, “Spherical Normalization for Learned Compressive Feedback in Massive MIMO CSI Acquisition,” in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020

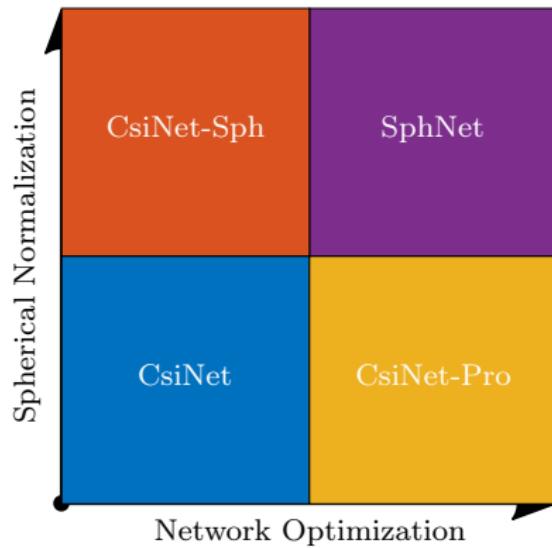


Figure: Illustration of techniques used in different models.

Z. Liu, M. del Rosario, X. Liang, L. Zhang, and Z. Ding, “Spherical Normalization for Learned Compressive Feedback in Massive MIMO CSI Acquisition,” in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020

Table: Parameters for COST2100 model in this work.

Environment	Indoor	Outdoor
Num. BS Antennas (N_b)		32
Num. Subcarriers (N_f)		1024
Truncation Value (R_d)		32
Carrier Frequency	5.3 GHz	300 MHz
UE Mobility	0.001 m/s	1 m/s
UE Starting Position	20×20 m	400×400 m
Num. Channel Samples (N)		10^5
Training/Validation Split		70%/30%
Feedback interval		40 ms

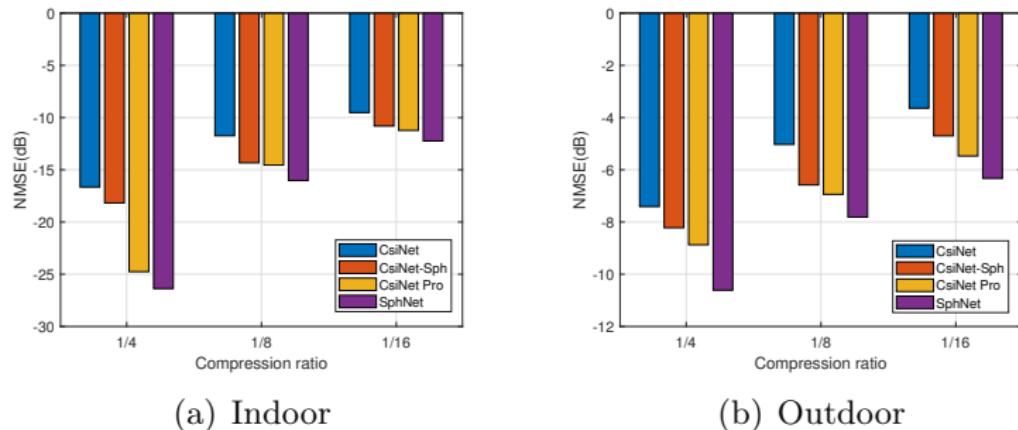


Figure: Sensitivity study for CsiNet-Pro and spherical normalization [5] (lower NMSE is better).

Z. Liu, M. del Rosario, X. Liang, L. Zhang, and Z. Ding, “Spherical Normalization for Learned Compressive Feedback in Massive MIMO CSI Acquisition,” in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020

Completed Work #2: MarkovNet

A deep differential autoencoder for efficient temporal learning.

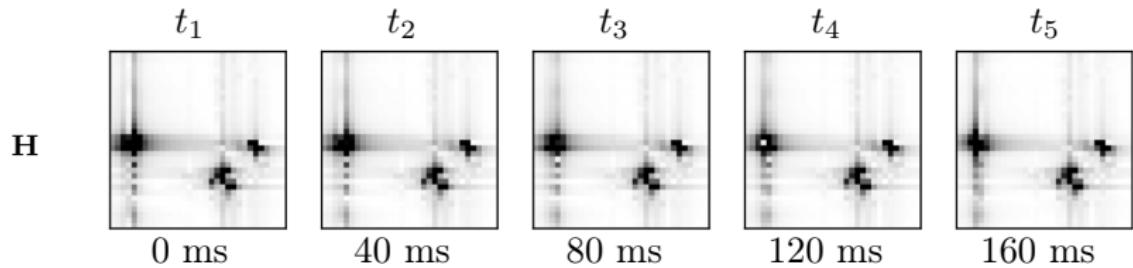


Figure: Ground truth CSI (**H**) for five timeslots (T_1 through T_5) on one outdoor sample from the validation set.

Recurrent neural networks (RNNs) contain trainable long short-term memory (LSTM) cells which learn temporal relationships.

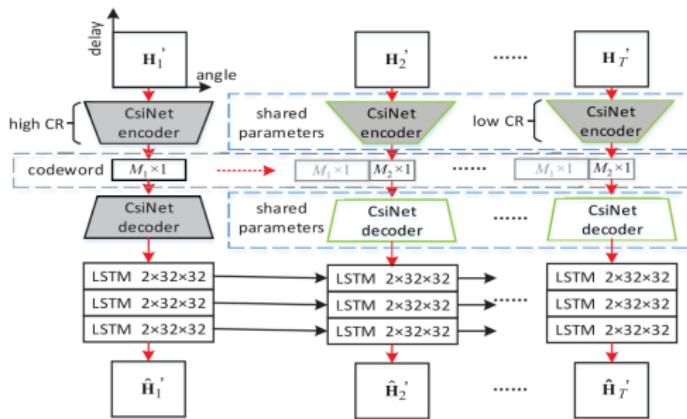


Figure: CsiNet-LSTM network architecture [8].

LSTMs improve NMSE at smaller compression ratios.

	CR	LASSO	BM3D-AMP	TVAL3	CsiNet	CsiNet-LSTM
Indoor	1/16	-2.96	0.25	-3.20	-10.59	-23.06
	1/32	-1.18	20.85	-0.46	-7.35	-22.33
	1/64	-0.18	26.66	0.60	-6.09	-21.24
	ρ	1/16	0.72	0.29	0.73	0.95
		1/32	0.53	0.17	0.45	0.90
		1/64	0.30	0.16	0.24	0.87
Outdoor	1/16	0.2471	0.3454	0.3148	0.0001	0.0003
	1/32	0.2137	0.5556	0.3148	0.0001	0.0003
	1/64	0.2479	0.6047	0.2860	0.0001	0.0003
	NMSE↓	1/16-1/64	94%	105	1.19	42%
						8%
	Outdoor	1/16	-1.09	0.40	-0.53	-3.60
		1/32	-0.27	18.99	0.42	-2.14
		1/64	-0.06	24.42	0.74	-1.65
		ρ	1/16	0.49	0.23	0.46
			1/32	0.32	0.16	0.28
			1/64	0.19	0.16	0.19
	Outdoor	1/16	0.2122	0.4210	0.3145	0.0001
		1/32	0.2409	0.6031	0.2985	0.0001
		1/64	0.0166	0.5980	0.2850	0.0001
		NMSE↓	1/16-1/64	94%	60	2.40
						54%
						10%

Problem: Number of parameters/FLOPs for RNNs is large.

Table: Model size/computational complexity per timeslot for CsiNet-LSTM and CsiNet. M: million.

CR	Parameters		FLOPs	
	CsiNet-LSTM	CsiNet	CsiNet-LSTM	CsiNet
1/4	132.7 M	2.1 M	412.9 M	7.8 M
1/8	123.2 M	1.1 M	410.8 M	5.7 M
1/16	118.5 M	0.5 M	409.8 M	4.7 M
1/32	116.1 M	0.3 M	409.2 M	4.1 M
1/64	115.0 M	0.1 M	409.0 M	3.9 M

For short enough feedback interval, CSI data form a Markov chain,

$$\mathbf{H}_t = \gamma \mathbf{H}_{t-1} + \mathbf{V}_t,$$

with $\gamma \in \mathbb{R}^+$ and i.i.d $\mathbf{V}_t \sim \mathcal{CN}(\mathbf{0}, \Sigma_V)$.

Z. Liu †, M. del Rosario †, and Z. Ding, “A Markovian Model-Driven Deep Learning Framework for Massive MIMO CSI Feedback,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 1214–1228, 2022 († equal contribution)

The ordinary least-squares solution, γ , is given as

$$\gamma = \frac{\text{Trace}(\mathbb{E} [\mathbf{H}_{t-1}^H \mathbf{H}_t])}{\mathbb{E} \|\mathbf{H}_t^H \mathbf{H}_t\|^2}.$$

The ordinary least-squares solution, γ , is given as

$$\gamma = \frac{\text{Trace}(\mathbb{E} [\mathbf{H}_{t-1}^H \mathbf{H}_t])}{\mathbb{E} \|\mathbf{H}_t^H \mathbf{H}_t\|^2}.$$

Utilize estimator, $\hat{\gamma}$, based on the sample statistics,

$$\hat{\gamma} = \frac{\sum_{i=1}^N \text{Trace}([\mathbf{H}_{t-1}^H(i) \mathbf{H}_t(i)])}{\sum_{i=1}^N \|\mathbf{H}_t^H(i) \mathbf{H}_t(i)\|^2},$$

for training set of size N .

Using $\hat{\gamma}$, train encoder on estimation error as

$$\begin{aligned}\mathbf{E}_t &= \mathbf{H}_t - \hat{\gamma} \hat{\mathbf{H}}_{t-1} \\ \mathbf{z}_t &= f_{e,t}(\mathbf{E}_t).\end{aligned}$$

Jointly train a decoder,

$$\begin{aligned}\hat{\mathbf{E}}_t &= f_{d,t}(\mathbf{z}_t) \\ \hat{\mathbf{H}}_t &= \hat{\mathbf{E}}_t + \hat{\gamma} \hat{\mathbf{H}}_{t-1}.\end{aligned}$$

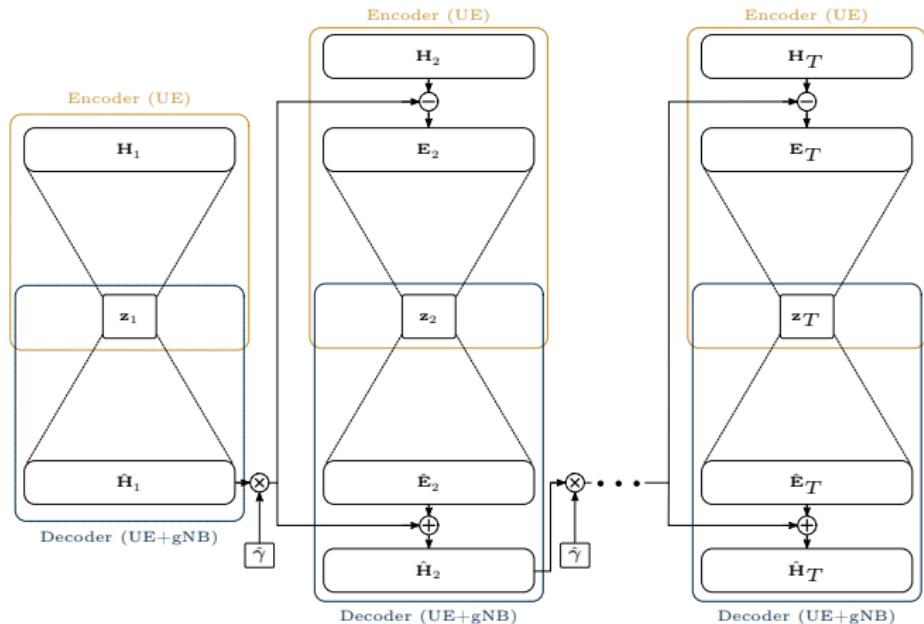


Figure: MarkovNet architecture. Networks at $t \geq 2$ predict estimation error, $\hat{\mathbf{E}}_t$.

MarkovNet Results – NMSE Performance

44

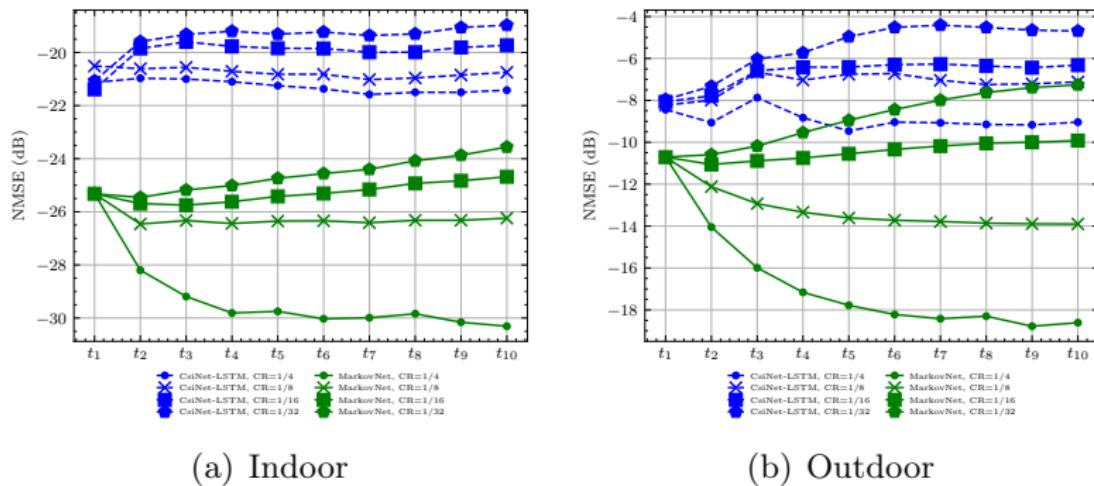


Figure: NMSE (lower is better) comparison of MarkovNet and CsiNet-LSTM at multiple CRs.

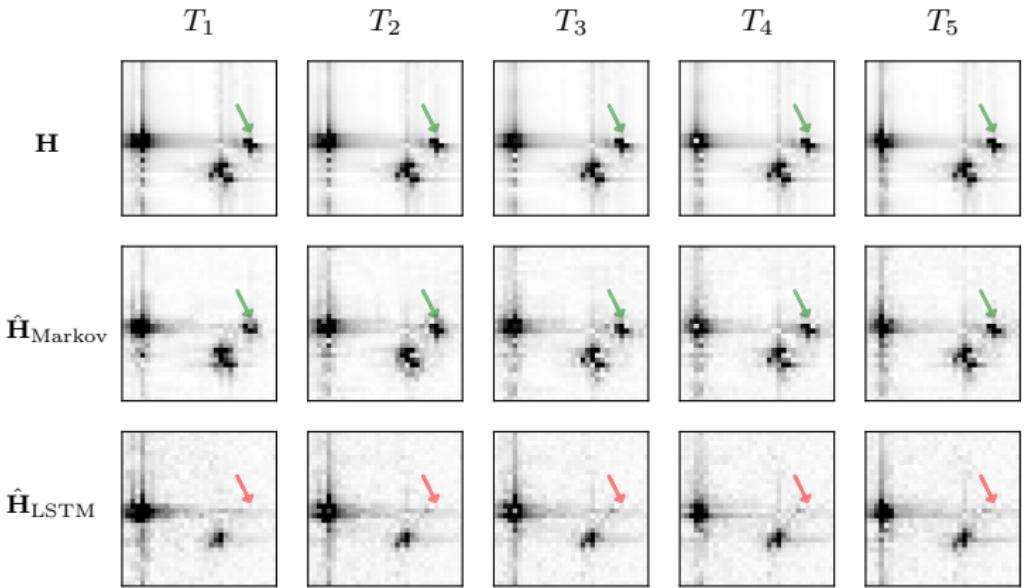


Figure: Ground truth (\mathbf{H}), MarkovNet estimates ($\hat{\mathbf{H}}_{\text{Markov}}$), and CsiNet-LSTM estimates ($\hat{\mathbf{H}}_{\text{LSTM}}$) on from outdoor test set ($\text{CR} = \frac{1}{4}$).

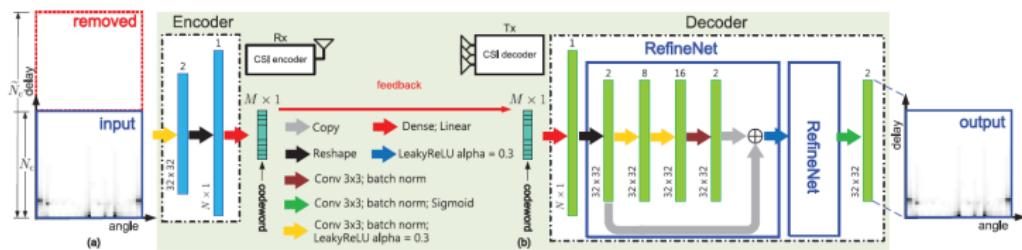
Table: Model size/computational complexity of tested temporal networks (CsiNet-LSTM, MarkovNet) and comparable non-temporal network (CsiNet). M: million.

	Parameters		
	CsiNet-LSTM	MarkovNet	CsiNet
CR=1/4	132.7 M	2.1 M	2.1 M
CR=1/8	123.2 M	1.1 M	1.1 M
CR=1/16	118.5 M	0.5 M	0.5 M
CR=1/32	116.1 M	0.3 M	0.3 M
CR=1/64	115.0 M	0.1 M	0.1 M
	FLOPs		
	CsiNet-LSTM	MarkovNet	CsiNet
CR=1/4	412.9 M	44.5 M	7.8 M
CR=1/8	410.8 M	42.4 M	5.7 M
CR=1/16	409.8 M	41.3 M	4.7 M
CR=1/32	409.2 M	40.8 M	4.1 M
CR=1/64	409.0 M	40.5 M	3.9 M

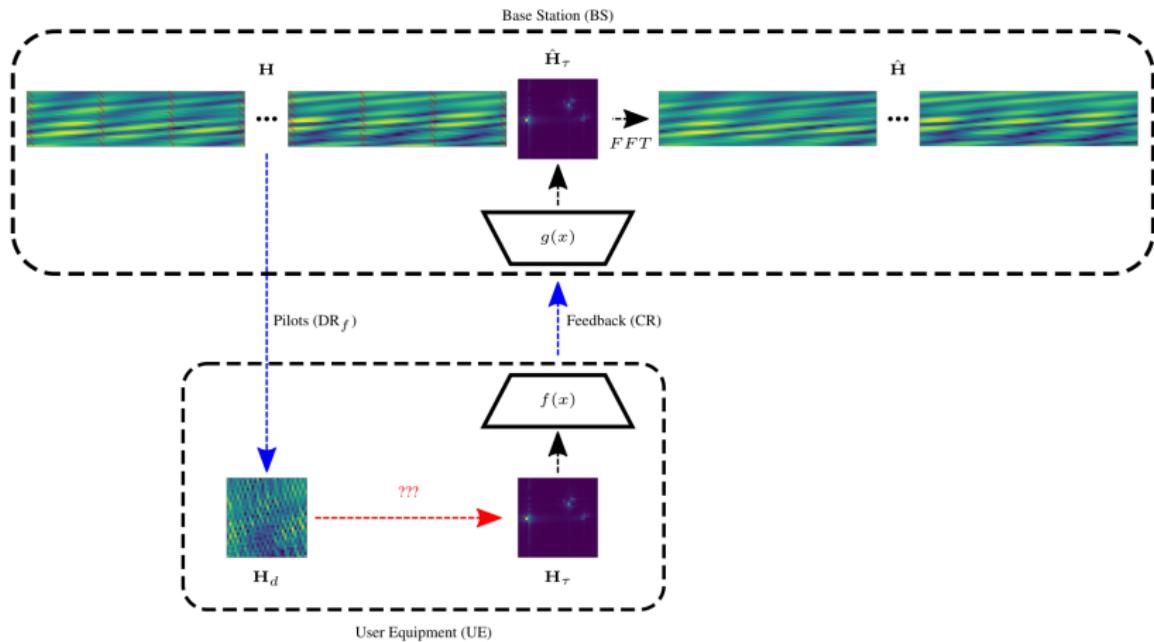
Completed Work #3: Pilots-to-delay Estimator (P2DE) and Heterogeneous Differential Encoding

Acquiring delay domain CSI under practical pilot placement. Improving differential encoding.

Recall: Works in DL-based CSI compression have used delay domain.



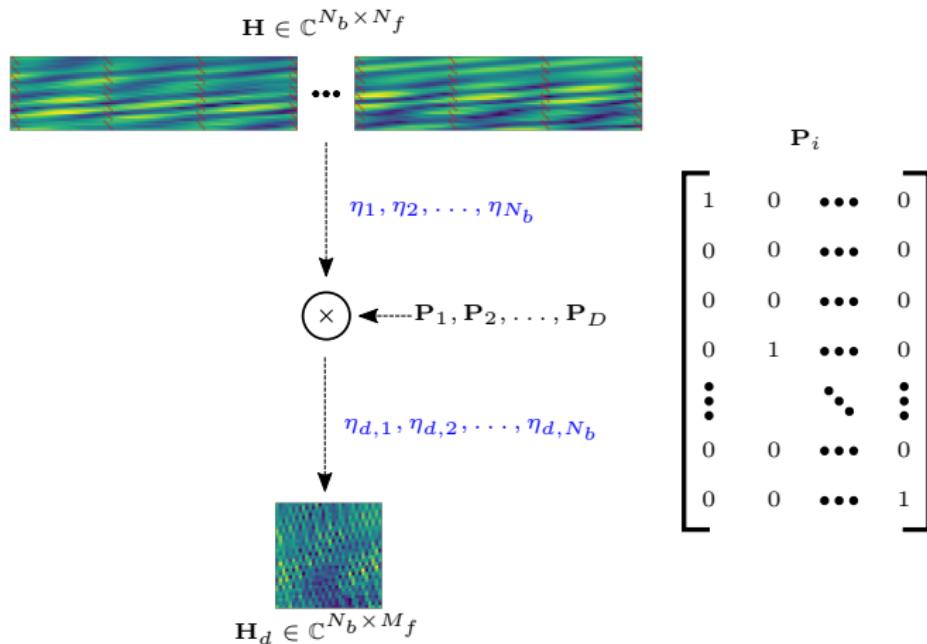
Problem: CSI at UE is based on sparse pilot estimates.



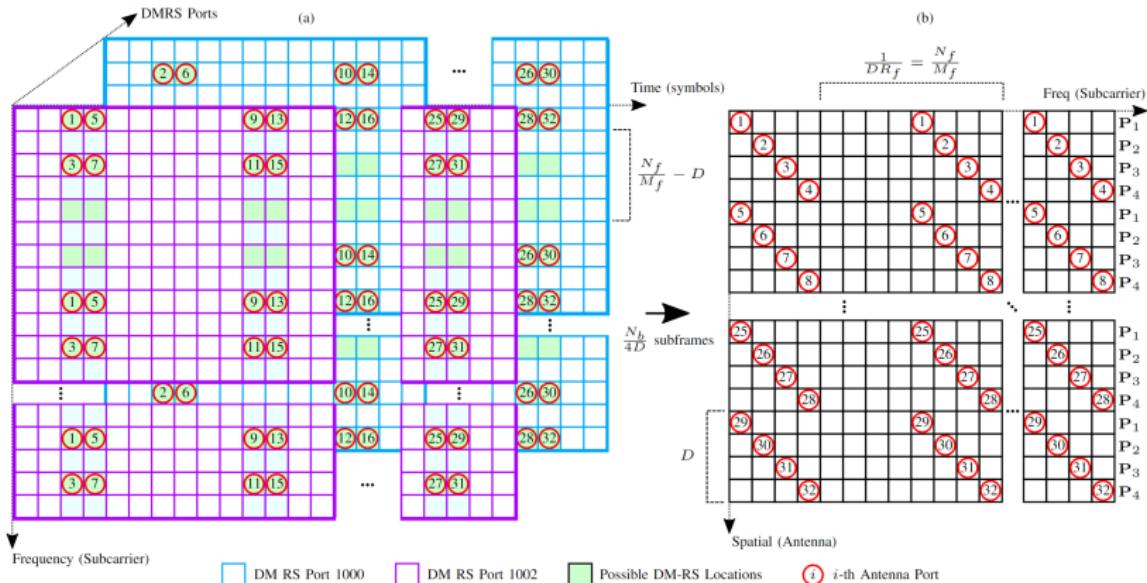
- ▶ Denote $\boldsymbol{\eta}_i \in \mathbb{C}^{N_f}$ as the i -th row of the spatial-frequency matrix \mathbf{H}
- ▶ Denote the downsampled version of $\boldsymbol{\eta}_i$ as $\boldsymbol{\eta}_{d,i} \in \mathbb{C}^{M_f}$ where $M_f << N_f$
- ▶ The spatial-frequency CSI, \mathbf{H} , and its downsampled counterpart, \mathbf{H}_d , can be written as,

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \vdots \\ \boldsymbol{\eta}_{N_b} \end{bmatrix} \in \mathbb{C}^{N_b \times N_f}, \quad \mathbf{H}_d = \begin{bmatrix} \boldsymbol{\eta}_{d,1} \\ \boldsymbol{\eta}_{d,2} \\ \vdots \\ \boldsymbol{\eta}_{d,N_b} \end{bmatrix} \in \mathbb{C}^{N_b \times M_f}. \quad (2)$$

- $M_f = \#$ downsampled subcarriers
- $\text{DR}_f = \frac{M_f}{N_f}$ (frequency downsampling ratio)



- In 5G subframes, downlink pilots are allocated as **Demodulation Reference Signals (DM-RS)**.
- 'Diagonal' pilot pattern in spatial/frequency domain allows faster pilot CSI acquisition (inversely proportional to diagonal size, D)



Denote delay-domain CSI vector, $\tilde{\boldsymbol{\eta}}_i$, which is defined as

$$\tilde{\boldsymbol{\eta}}_i \mathbf{F} = \boldsymbol{\eta}_i, \quad (3)$$

for $\mathbf{F} \in \mathbf{C}^{N_f \times N_f}$ is the discrete Fourier transform (DFT) matrix.

Apply the pilot downsampling matrix \mathbf{P}_i to both sides of (3),

$$\begin{aligned}\tilde{\boldsymbol{\eta}}_i \mathbf{F} \mathbf{P}_i &= \boldsymbol{\eta}_i \mathbf{P}_i \\ \tilde{\boldsymbol{\eta}}_i \mathbf{Q}_i &= \boldsymbol{\eta}_{d,i}\end{aligned}\tag{4}$$

where $\mathbf{Q}_i = \mathbf{F} \mathbf{P}_i \in \mathbb{C}^{N_f \times M_f}$ is the downsampled DFT matrix.

- ▶ **Goal:** Feed back/compress truncated delay domain vectors, $\tilde{\boldsymbol{\eta}}_{c,i} \in \mathbb{C}^{N_t}$.
- ▶ Denote zero-padded vector $\tilde{\boldsymbol{\eta}}_i$ as

$$\tilde{\boldsymbol{\eta}}_i = [\tilde{\boldsymbol{\eta}}_{c,i}, \mathbf{0}_{N_f - N_t}] . \quad (5)$$

- ▶ Based on \mathbf{Q}_i of (4), the delay domain is related to the pilots by the pseudoinverse,

$$\begin{aligned}\tilde{\boldsymbol{\eta}}_i \mathbf{Q}_i \mathbf{Q}_i^T &= \boldsymbol{\eta}_{d,i} \mathbf{Q}_i^T \\ \tilde{\boldsymbol{\eta}}_i &= \boldsymbol{\eta}_{d,i} \mathbf{Q}_i^T (\mathbf{Q}_i \mathbf{Q}_i^T)^{-1} \\ &= \boldsymbol{\eta}_{d,i} \mathbf{Q}_i^\#.\end{aligned} \quad (6)$$

Algorithm 1 outlines the process for acquiring truncated delay domain from sparse/downsampled frequency domain pilots.

Algorithm 1 Pilots-to-delay Estimator (P2DE) for Diagonal Pilot Pattern

Input: P2DE Matrices, $\mathbf{Q}_{c,j}^\#$, $j \in \{1, \dots, D\}$

Input: Pilot spatial-frequency CSI, $\mathbf{H}_d \in \mathbb{C}^{N_b \times M_f}$

Initialize: Spatial-delay CSI, $\tilde{\mathbf{H}}_\tau \in \mathbb{C}^{N_b \times N_t}$

Initialize: Angular-delay CSI estimate, $\mathbf{H}_\tau \in \mathbb{C}^{N_b \times N_t}$

for $i = 1, 2, \dots, N_b$ **do**

Index for j-th pilot matrix

$j = ((i - 1) \bmod D) + 1$

Apply P2D to i-th antenna port

$\eta_{d,i} = \mathbf{H}_d(i, :)$

$\tilde{\mathbf{H}}_\tau(i, :) = \eta_{d,i} \mathbf{Q}_{c,j}^\#$

end for

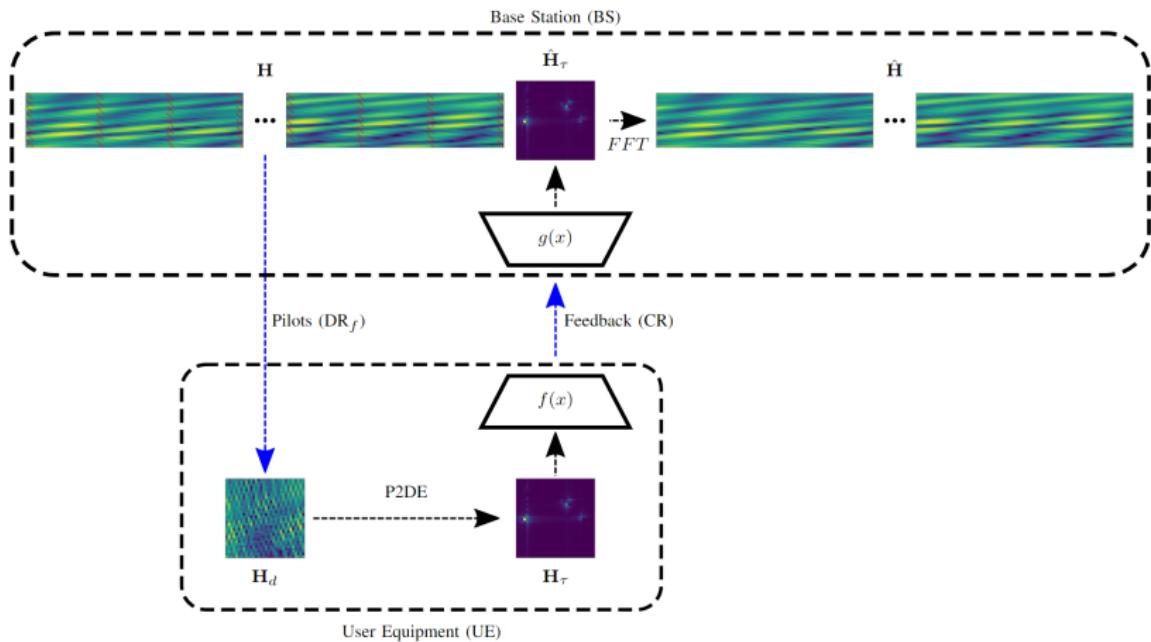
Convert from spatial to angular

$\mathbf{H}_\tau = \mathbf{F}_{N_b} \tilde{\mathbf{H}}_\tau$

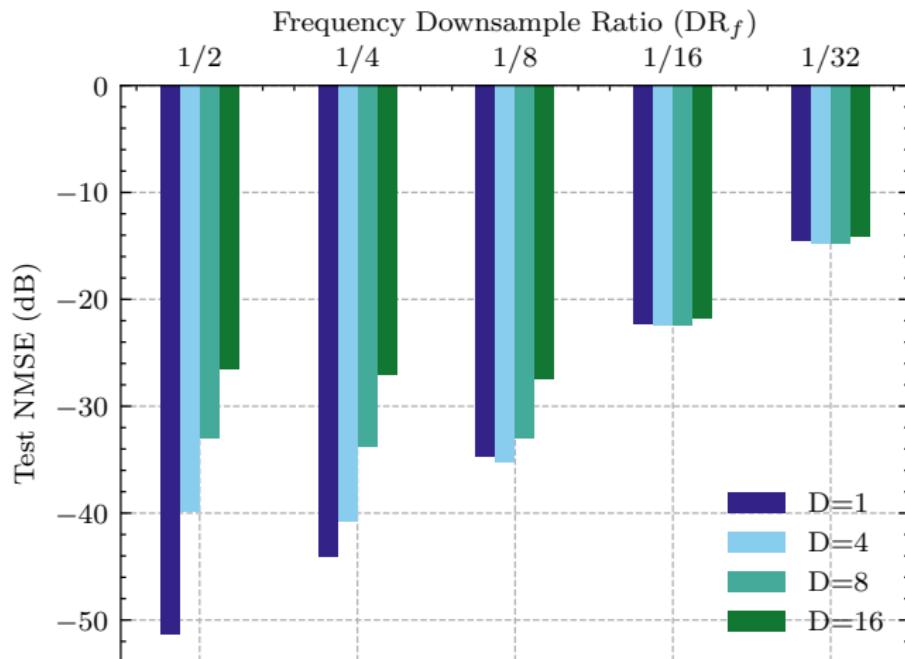
Return \mathbf{H}_τ

Pilots-to-Delay Estimator (P2DE)

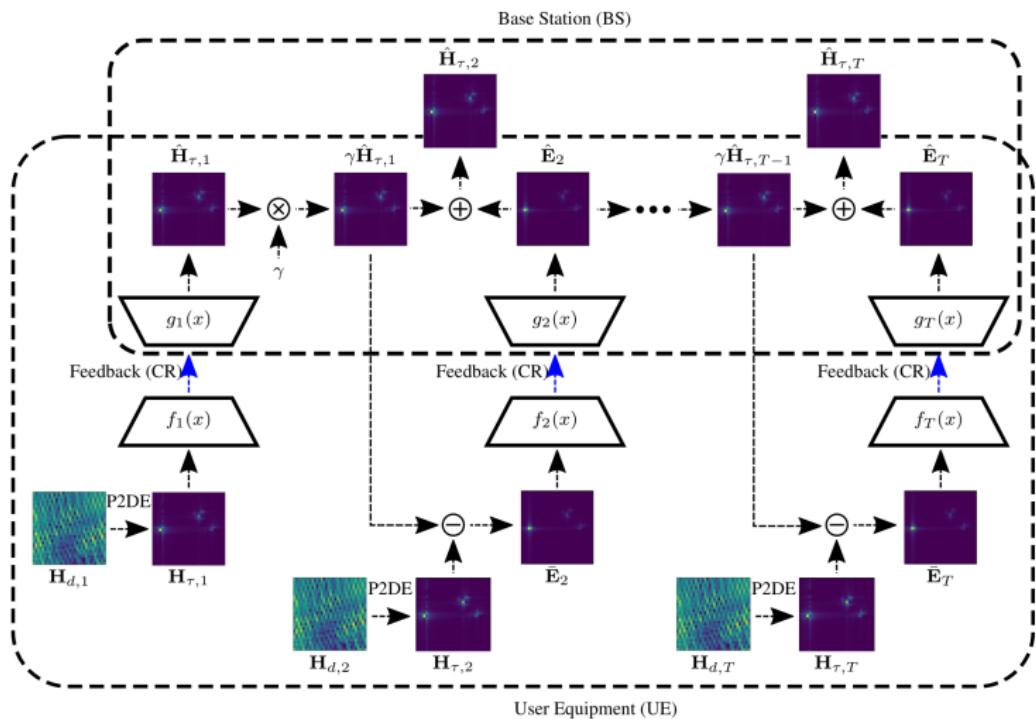
57



Accuracy of the P2DE at the UE (i.e., before compression and feedback) for different DR_f .



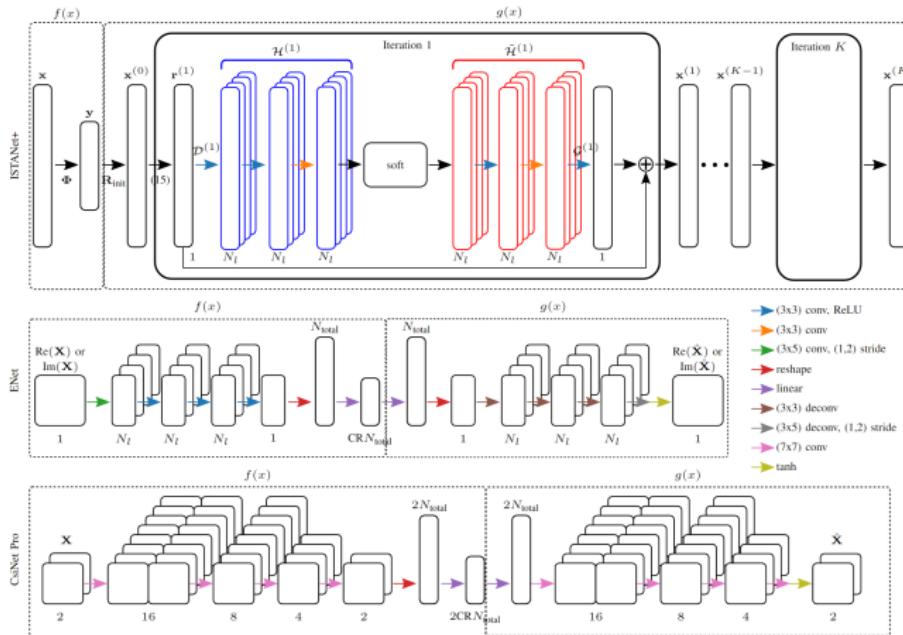
Differential encoding network using P2DE at the UE.



Heterogeneous Differential Encoding

60

- ▶ **Homogeneous:** MarkovNet used the same network at each timeslot (CsiNet Pro).
- ▶ **Heterogeneous:** Use different networks at different timeslots.



- ▶ Single timeslot performance of networks.
- ▶ Deep CS network (ISTANet+) can outperform autoencoder approaches (ENet, CsiNet Pro)
- ▶ ENet can outperform CsiNet Pro

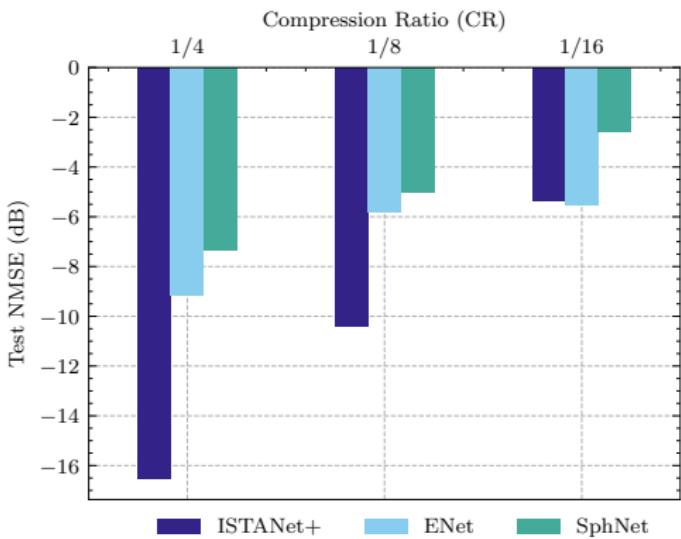


Figure: Performance comparison for different feedback compression networks using P2D estimates ($DF_f = 1/16, D = 4$) for Outdoor COST2100 dataset.

Three different configurations:

- ▶ **MarkovNet-ISTA (MN-I):** *Homogeneous* MarkovNet using ISTANet+ at all timeslots (t_1, t_2, \dots, t_T).
- ▶ **MarkovNet-ENet (MN-E):** *Homogeneous* MarkovNet using ISTANet+ at all timeslots (t_1, t_2, \dots, t_T).
- ▶ **MarkovNet-ISTA-ENet (MN-IE):** *Heterogeneous* MarkovNet using ISTANet+ at t_1 and ENet at all other timeslots (t_2, t_3, \dots, t_T).

- **MN-ISTANet+**: Better performance in first timeslot.
- **MN-ENet**: Better performance in error timeslots.
- **MN-IE**: Enjoys both of the above benefits.

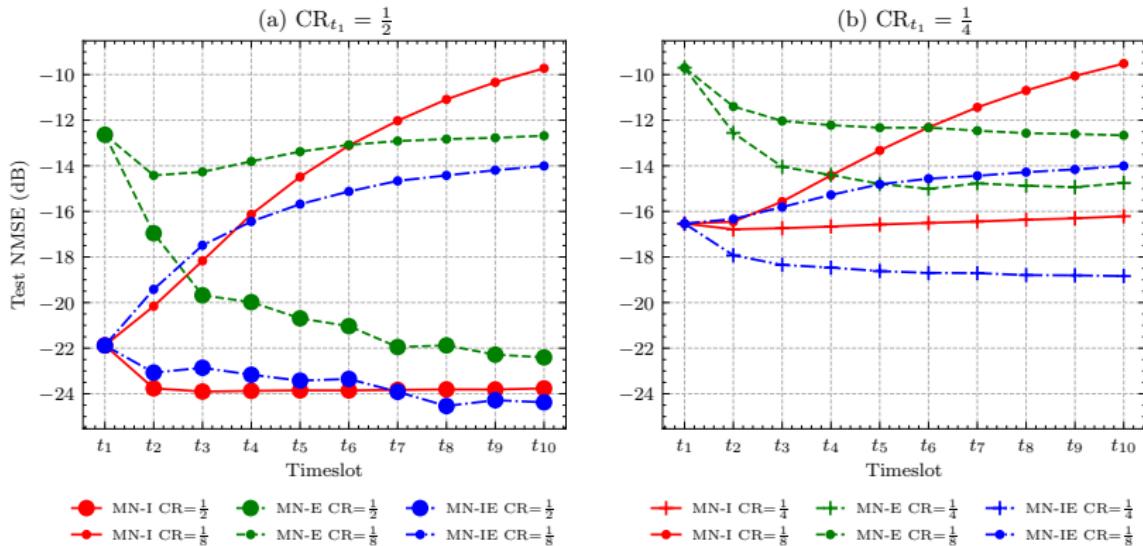


Figure: Differential encoding networks with linear P2DE as input ($M_f = 128$, $DR_f = \frac{1}{8}$, $D = 4$).

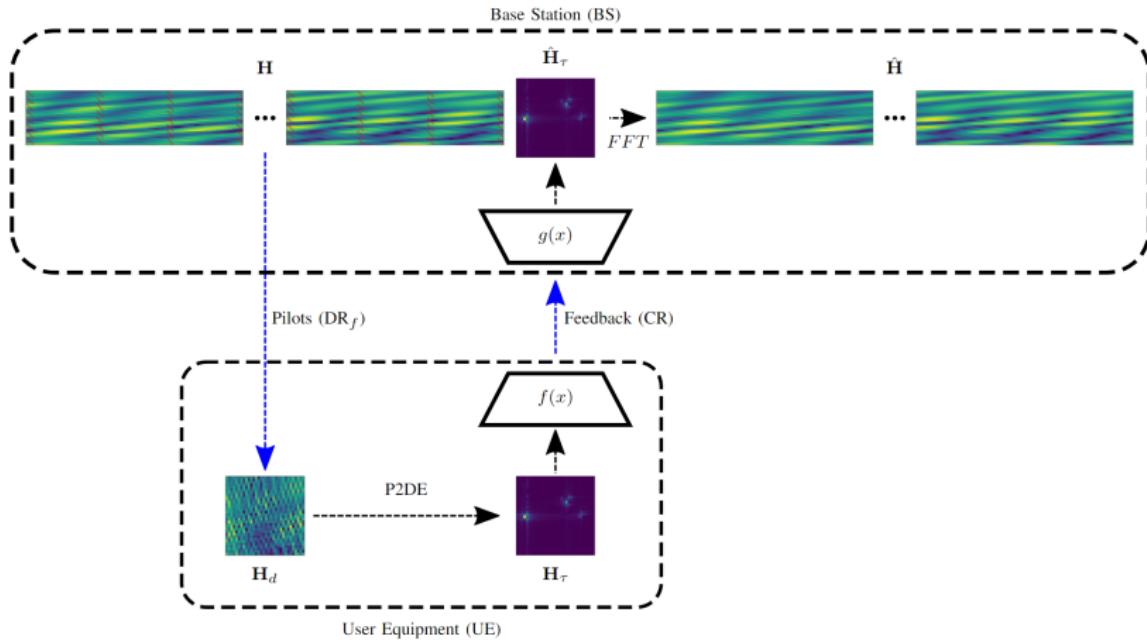
Table: Computational complexity of networks used in this work (lower is better). **Bold face** in a column indicates lowest value for given compression ratio.

		Parameters (M)				FLOPs (M)	
		Trainable		All			
	CR	Enc	Dec	Enc	Dec	Enc	Dec
ISTANet+	1/2	0.00	0.34	2.10	4.54	2.10	393.78
	1/4	0.00	0.34	1.05	2.44	1.05	373.85
	1/8	0.00	0.34	0.52	1.39	0.52	363.89
	1/16	0.00	0.34	0.26	0.87	0.26	358.91
ENet	1/2	0.55	0.55	0.55	0.55	29.98	29.70
	1/4	0.29	0.29	0.29	0.29	29.46	29.18
	1/8	0.16	0.16	0.16	0.16	29.20	28.92
	1/16	0.09	0.09	0.09	0.09	29.07	28.79
CsiNet Pro	1/2	1.06	1.06	1.06	1.06	12.16	12.16
	1/4	0.53	0.53	0.53	0.53	11.11	11.11
	1/8	0.27	0.27	0.27	0.27	10.59	10.59
	1/16	0.14	0.14	0.14	0.14	10.33	10.33

- ▶ Deep CS network (ISTANet+) can provide superior initial estimate with slightly more complexity.
- ▶ Autoencoder network (ENet) can provide good error compression/estimation.
 - ▶ Fewer encoder/decoder parameters
 - ▶ More encoder FLOPs, less decoder FLLOPs

Current Work: Pilot Feedback and Model Re-use

UE-focused reduction of complexity of CSI feedback.



Problem: Encoder computation at (low-resourced) UE. Can we reduce this?

Table: Computational complexity of P2DE for $D = 1$ (diagonal pattern size), $N_f = 1024$ (number of subcarriers), and $N_b = 32$ (# antennas in ULA).

M_f	32	64	128
FLOPs	$1.05 \cdot 10^6$	$2.10 \cdot 10^6$	$4.19 \cdot 10^6$
Parameters	$6.55 \cdot 10^4$	$1.31 \cdot 10^5$	$2.62 \cdot 10^5$

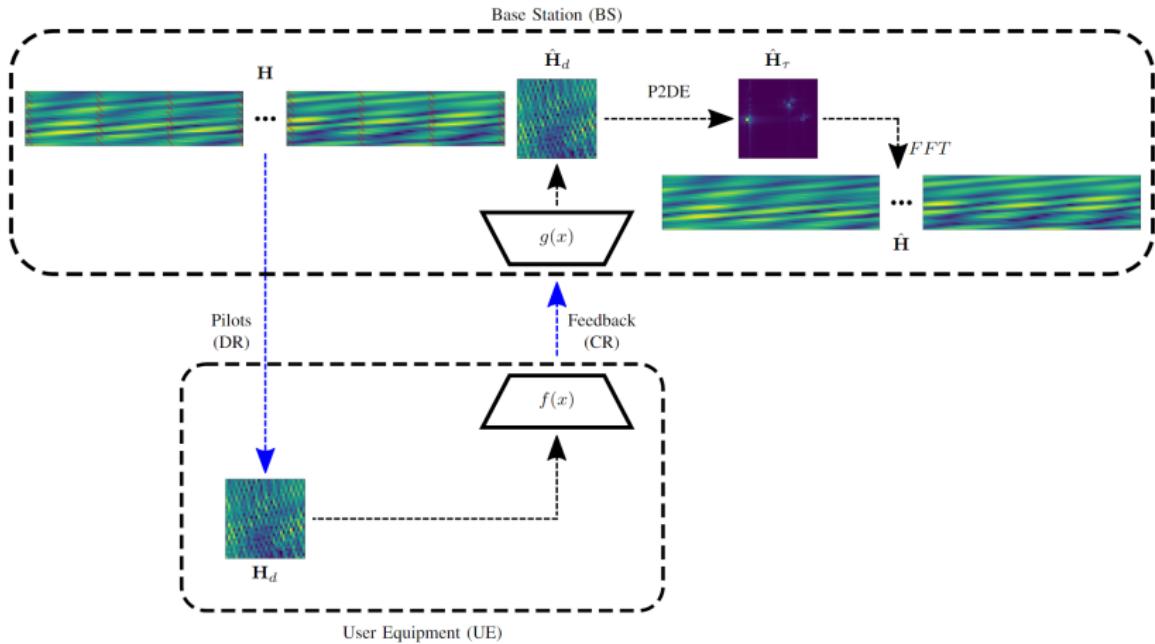


Figure: Compressive CSI estimation based on linear P2D estimator on BS side.

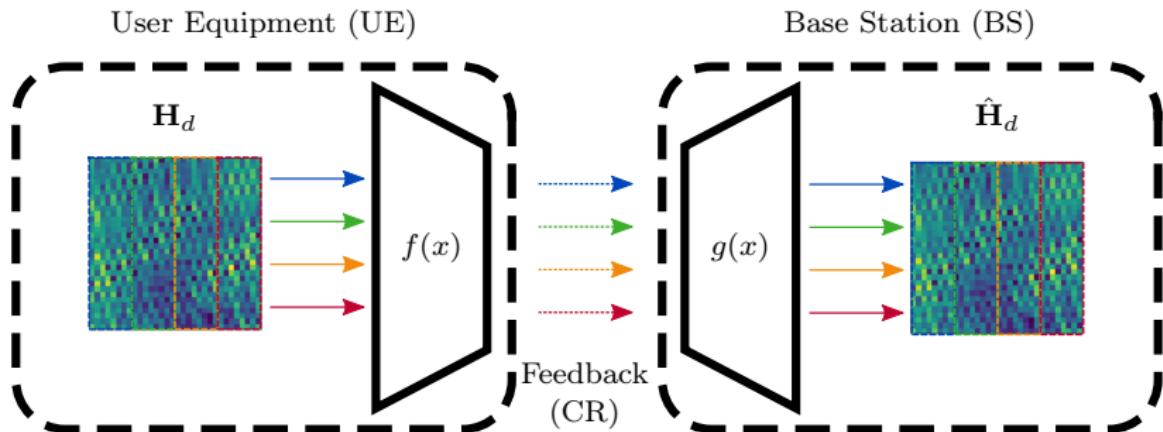
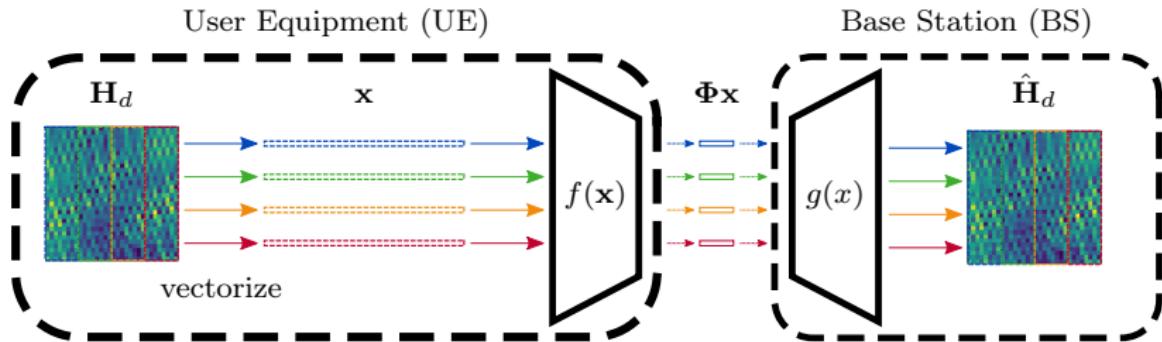


Figure: Compressive CSI estimation with model re-use. The encoder compresses K contiguous pilot subcarriers from the input, resulting in $\frac{M_f}{K}$ payloads of feedback (shown in different colors above).

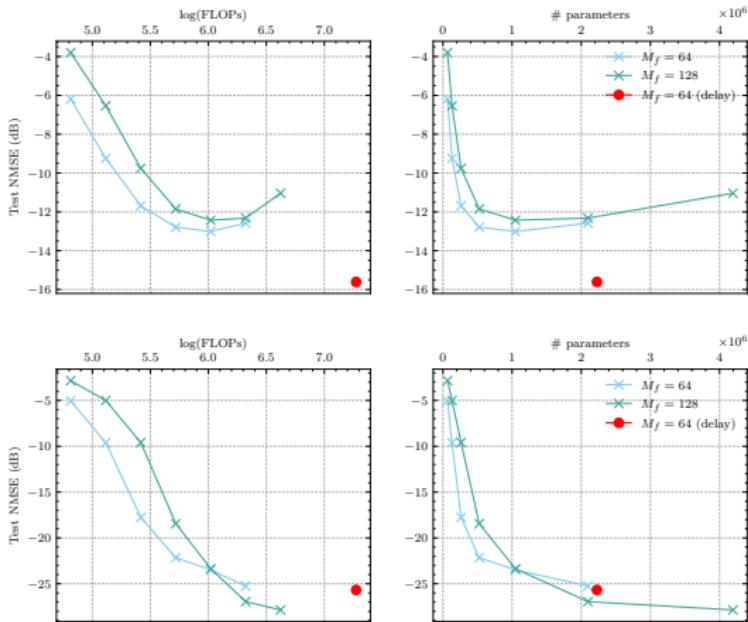


We use ISTANet+ [9], which compresses CSI at UE via,

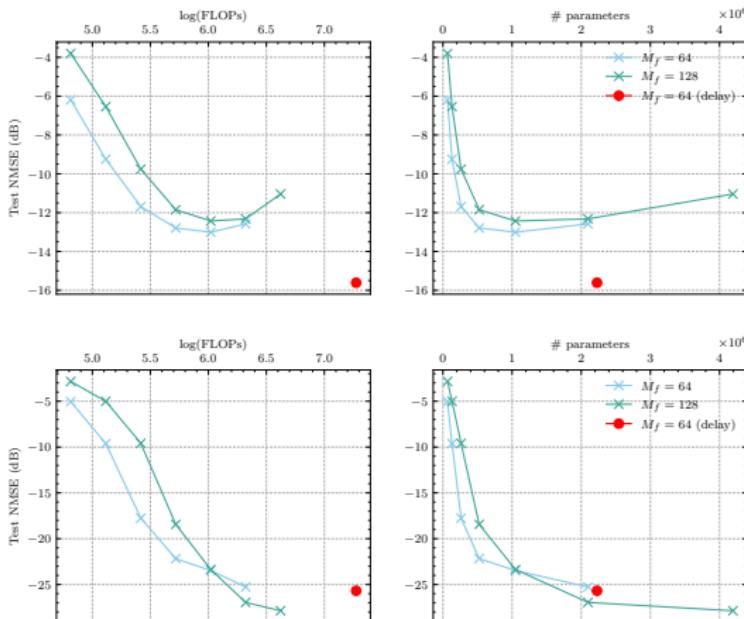
$$f(x) = \Phi \mathbf{x} \quad (7)$$

for $\mathbf{x} \in \mathbb{R}^{N_K}$ and $\Phi \in \mathbb{R}^{\text{CR}N_K \times N_K}$ where $N_K = 2KN_b$

- ▶ Outdoor (top) and Indoor (bottom) results w.r.t. NMSE vs. complexity ($\log(\text{FLOPs})$ on left, parameters on right) using different number of pilot subcarriers (M_f).
- ▶ Compare with P2DE at UE and no model re-use (red).



- ▶ Indoor: Can maintain same NMSE with 10-fold reduction in FLOPs
- ▶ Outdoor: NMSE increase of 2.5dB with 10-fold reduction in FLOPs, 10^6 reduction in parameters



- ▶ Z. Liu, **M. del Rosario**, X. Liang, L. Zhang, and Z. Ding, “Spherical Normalization for Learned Compressive Feedback in Massive MIMO CSI Acquisition,” in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020
- ▶ Z. Liu †, **M. del Rosario** †, and Z. Ding, “A Markovian Model-Driven Deep Learning Framework for Massive MIMO CSI Feedback,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 1214–1228, 2022
- ▶ **M. del Rosario** and Z. Ding, “Learning-Based MIMO Channel Estimation under Spectrum Efficient Pilot Allocation and Feedback,” *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022
- ▶ **M. del Rosario** and Z. Ding, “Direct Pilot Feedback and Model Re-use for Encoder-focused Complexity Reduction in Massive MIMO CSI Feedback,” 2022. "In Preparation."

- ▶ Thesis committee

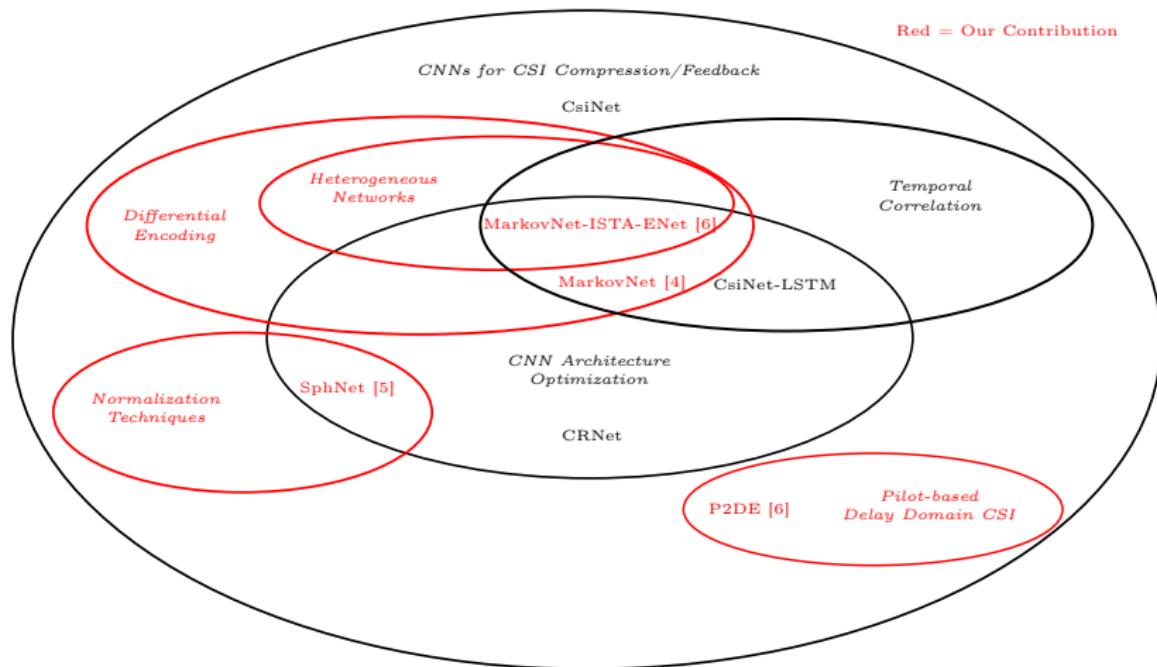
- ▶ Thesis committee
- ▶ Prof. Ding, lab mates, collaborators

- ▶ Thesis committee
- ▶ Prof. Ding, lab mates, collaborators
- ▶ My parents, my brother

- ▶ Thesis committee
- ▶ Prof. Ding, lab mates, collaborators
- ▶ My parents, my brother
- ▶ My SO

Questions?

mdelrosa@ucdavis.edu



References

- [1] E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten Myths and One Critical Question," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, 2016.
- [2] E. Crespo Marques, N. Maciel, L. Naviner, H. Cai, and J. Yang, "A Review of Sparse Recovery Algorithms," *IEEE Access*, vol. 7, pp. 1300–1322, 2019.
- [3] C. Wen, W. Shih, and S. Jin, "Deep Learning for Massive MIMO CSI Feedback," *IEEE Wireless Communications Letters*, vol. 7, pp. 748–751, Oct 2018.
- [4] Z. Liu †, M. del Rosario †, and Z. Ding, "A Markovian Model-Driven Deep Learning Framework for Massive MIMO CSI Feedback," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 1214–1228, 2022.
- [5] Z. Liu, M. del Rosario, X. Liang, L. Zhang, and Z. Ding, "Spherical Normalization for Learned Compressive Feedback in Massive MIMO CSI Acquisition," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, 2020.
- [6] M. del Rosario and Z. Ding, "Learning-Based MIMO Channel Estimation under Spectrum Efficient Pilot Allocation and Feedback," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2022.
- [7] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. D. Doncker, "The COST 2100 MIMO Channel Model," *IEEE Wireless Communications*, vol. 19, pp. 92–99, December 2012.
- [8] T. Wang, C. Wen, S. Jin, and G. Y. Li, "Deep Learning-Based CSI Feedback Approach for Time-Varying Massive MIMO Channels," *IEEE Wireless Comm. Letters*, vol. 8, pp. 416–419, April 2019.
- [9] J. Zhang and B. Ghanem, "ISTA-Net: Interpretable Optimization-inspired Deep Network for Image Compressive Sensing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1828–1837, 2018.
- [10] M. del Rosario and Z. Ding, "Direct Pilot Feedback and Model Re-use for Encoder-focused Complexity Reduction in Massive MIMO CSI Feedback," 2022.

"In Preparation.".

Appendix

Metrics used:

- ▶ Normalized Mean-squared Error

$$\text{NMSE} = \frac{1}{N} \sum_i^N \frac{\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|^2}{\|\mathbf{H}_i\|^2}$$

- ▶ Cosine Similarity

$$\rho = \frac{1}{NN_f} \sum_{i=1}^N \sum_{m=1}^{N_f} \frac{|\hat{\mathbf{h}}_{i,m}^H \bar{\mathbf{h}}_{i,m}|}{\|\hat{\mathbf{h}}_{i,m}\| \|\bar{\mathbf{h}}_{i,m}\|},$$

SphNet (and benchmark networks)

- ▶ **Epochs:** 1000
- ▶ **Optimizer:** Adam with learning rate 10^{-3}

MarkovNet

- ▶ **Epochs (t_1):** 1000
- ▶ **Epochs (t_2, \dots, t_T):** 150
- ▶ **Optimizer:** Adam with learning rate 10^{-3}
- ▶ Each timeslot is initialized with weights from previous timeslot.

CsiNet-LSTM

- ▶ **Epochs:** 1000 (pretraining CsiNet), 500 (CsiNet-LSTM)
- ▶ **Optimizer:** Adam with learning rate 10^{-3}

D-AMP = Denoising approximate message passing. Initialize $x^0 = \mathbf{0}$, and alternate between:

$$x^{t+1} = D_{\hat{\sigma}^t}(x^t + \mathbf{A}^* z^t)$$

$$z^t = y - \mathbf{A}x^t + z^{t-1} \frac{\text{div} D_{\hat{\sigma}^{t-1}}(x^{t-1} + \mathbf{A}^* z^{t-1})}{m}$$

where $\hat{\sigma}^t = \text{Var}(x^t + \mathbf{A}^* z^t)$, $D_{\hat{\sigma}_t}$ = denoising algorithm.

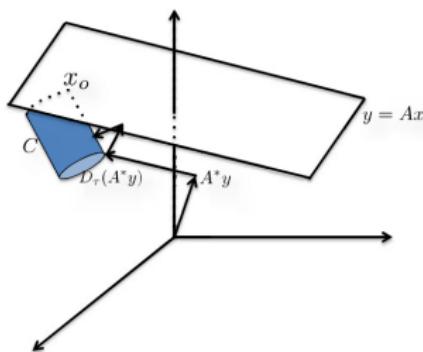


Figure: Subspaces of interest in D-AMP.

BM3D-AMP = D-AMP with *block matching 3D collaborative filtering (BM3D)*.

- ▶ Combination of non-local means (NLM) and wavelet thresholding.
- ▶ Procedure:
 1. Compare patches of pixels in images
 2. Group similar patches
 3. 2D (DCT or Bior Wavelet) + 1D Haar wavelet transforms on group
 4. Shrink coefficients in groups ($N \rightarrow M$)
 5. Perform inverse transform by 1) hard thresholding and 2) Wiener filter ($M \rightarrow N$)

Given mean μ , standard deviation σ w.r.t \mathbf{H} ,

$$H_{\text{tanh}}(i, j) = \tanh\left(\frac{H(i, j) - \mu}{2\nu\sigma}\right) + 1.$$

Scale parameter ν chosen by designer.

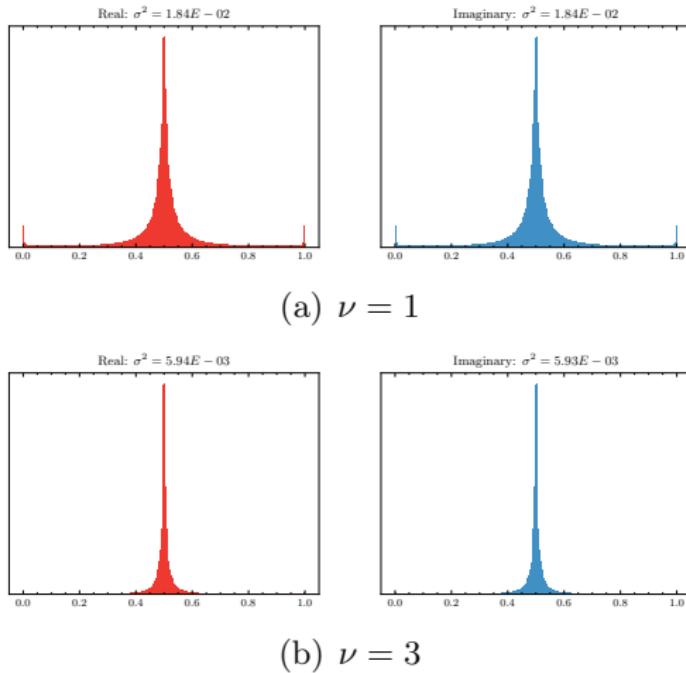


Figure: Distribution/variance of indoor COST2100 real/imaginary channels under tanh normalization ($N = 9.910^5$).

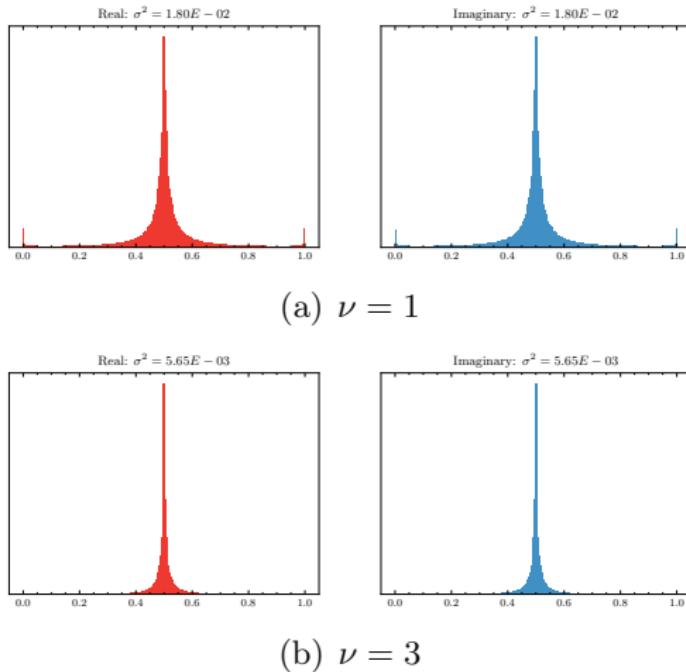


Figure: Distribution/variance of outdoor COST2100 real/imaginary channels under tanh normalization ($N = 10^5$).

Rather than scalar $\hat{\gamma} \in \mathbb{R}^+$, we can derive a multivariate p -step predictor, $\mathbf{W}_1, \dots, \mathbf{W}_p$. Given p prior CSI samples, the mean-square optimal predictor \hat{H}_t is a linear combination of these the prior CSI samples,

$$\hat{\mathbf{H}}_t = \mathbf{H}_{t-1}\mathbf{W}_1 + \cdots + \mathbf{H}_{t-p}\mathbf{W}_p + \mathbf{E}_t. \quad (8)$$

Error terms are uncorrelated with the CSI samples (i.e. $\mathbf{H}_{t-i}^H \mathbf{E}_t = 0$ for all $i \in [0, \dots, p]$), and we pre-multiply by \mathbf{H}_{t-i}^H ,

$$\begin{aligned}\mathbf{H}_{t-i}^H \hat{\mathbf{H}}_t &= \mathbf{H}_{t-i}^H \mathbf{H}_{t-1} \mathbf{W}_1 + \cdots + \mathbf{H}_{t-i}^H \mathbf{H}_{t-p} \mathbf{W}_p + \mathbf{H}_{t-i}^H \mathbf{E}_t \\ &= \mathbf{H}_{t-i}^H \mathbf{H}_{t-1} \mathbf{W}_1 + \cdots + \mathbf{H}_{t-i}^H \mathbf{H}_{t-p} \mathbf{W}_p.\end{aligned}\tag{9}$$

Denote the correlation matrix $\mathbf{R}_i = \mathbb{E}[\mathbf{H}_{t-i}^H \mathbf{H}_t]$. Presume CSI matrices arise from a stationary process, implying the following properties:

1. $\mathbf{R}_i = \mathbb{E}[\mathbf{H}_{t-i}^H \mathbf{H}_t] = \mathbb{E}[\mathbf{H}_t^H \mathbf{H}_{t+i}]$
2. $\mathbf{R}_i = \mathbf{R}_{-i}^H$

Taking the expectation, write (9) as a linear combination of \mathbf{R} ,

$$\mathbf{R}_{i+1} = \mathbf{R}_i \mathbf{W}_1 + \cdots + \mathbf{R}_{i-p+1} \mathbf{W}_p.$$

For p CSI samples, write a system of p equations, admitting the following,

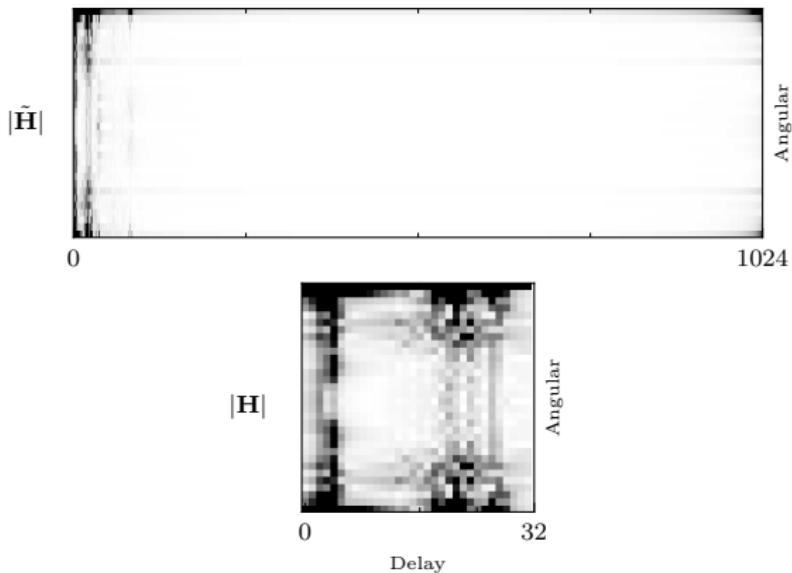
$$\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_p \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1^H & \cdots & \mathbf{R}_{p-1}^H \\ \mathbf{R}_1 & \mathbf{R}_0 & \cdots & \mathbf{R}_{p-2}^H \\ \vdots & & \ddots & \vdots \\ \mathbf{R}_{p-1} & \mathbf{R}_{p-2} & \cdots & \mathbf{R}_0 \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \cdots \\ \mathbf{W}_p \end{bmatrix}.$$

Solving for the coefficient matrices admits the solution

$$\begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_p \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1^H & \cdots & \mathbf{R}_{p-1}^H \\ \mathbf{R}_1 & \mathbf{R}_0 & \cdots & \mathbf{R}_{p-2}^H \\ \vdots & & \ddots & \vdots \\ \mathbf{R}_{p-1} & \mathbf{R}_{p-2} & \cdots & \mathbf{R}_0 \end{bmatrix}^+ \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_p \end{bmatrix}, \quad (10)$$

where $[\cdot]^+$ denotes the Moore-Penrose pseudoinverse.

$$\text{NMSE}_{\text{all}} = \frac{1}{N} \sum_i^N \frac{\|\tilde{\mathbf{H}}_i - \hat{\mathbf{H}}_i\|^2}{\|\tilde{\mathbf{H}}_i\|^2}, \quad \text{NMSE}_{\text{truncate}} = \frac{1}{N} \sum_i^N \frac{\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|^2}{\|\mathbf{H}_i\|^2},$$



		MarkovNet		CsiNet-LSTM	
Env	CR	NMSE _{truncate}	NMSE _{all}	NMSE _{truncate}	NMSE _{all}
Indoor	$\frac{1}{4}$	-29.26	-20.81	-21.28	-18.4
	$\frac{1}{8}$	-26.25	-20.26	-20.76	-18.12
	$\frac{1}{16}$	-25.27	-19.99	-19.96	-17.67
	$\frac{1}{32}$	-24.62	-19.78	-19.41	-17.34
Outdoor	$\frac{1}{4}$	-16.8	-12.4	-8.89	-7.99
	$\frac{1}{8}$	-13.19	-10.86	-7.17	-6.60
	$\frac{1}{16}$	-10.45	-9.13	-6.65	-6.15
	$\frac{1}{32}$	-8.87	-7.92	-5.33	-4.99

Table: NMSE of truncated vs. full CSI matrices.

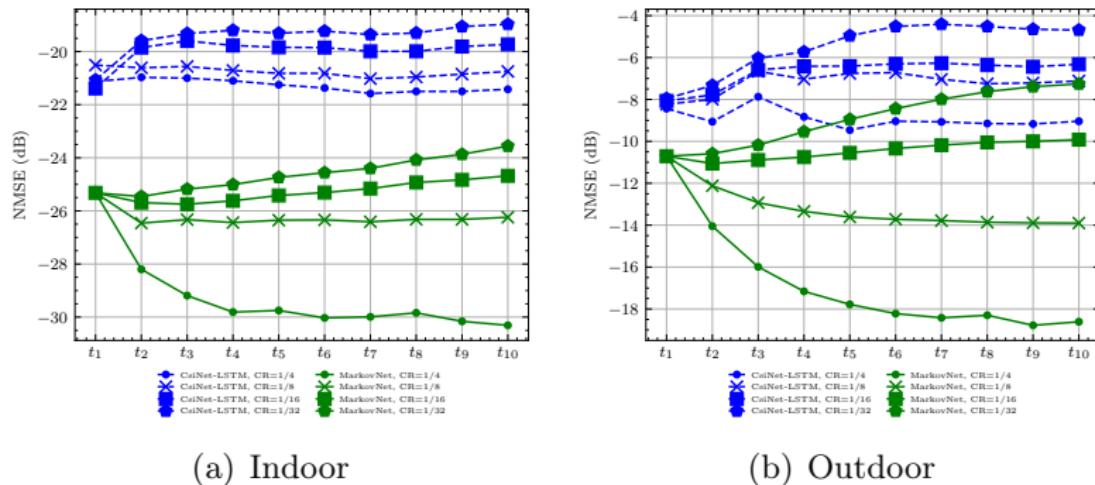


Figure: NMSE_{truncated} comparison of MarkovNet and CsiNet-LSTM at various compression ratios (CR).

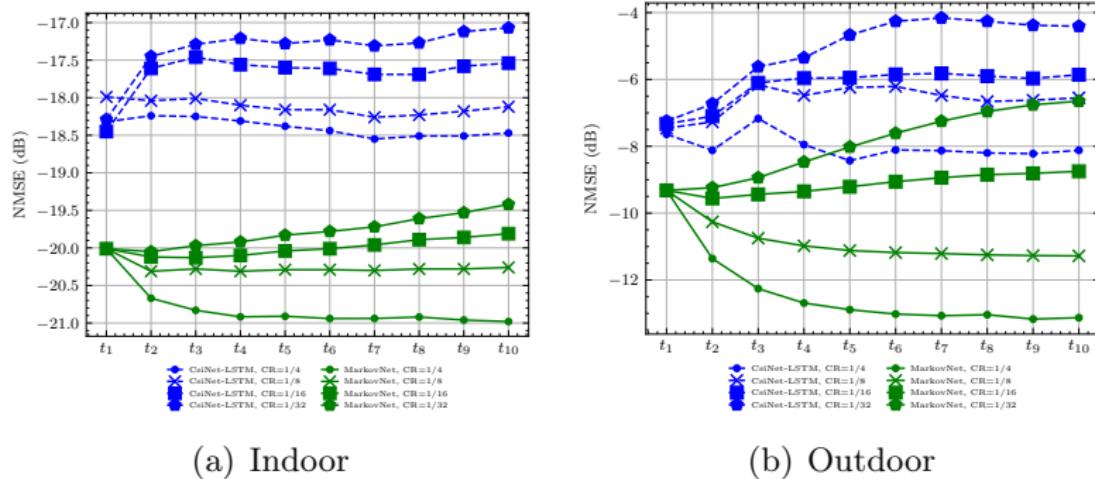


Figure: NMSE_{all} comparison of MarkovNet and CsiNet-LSTM at various compression ratios (CR).

TO-DO

To simulate pilot estimation error, we use additive Gaussian noise,

$$\hat{\mathbf{H}}_d = \mathbf{H}_d + \mathbf{N}_d$$

where $\mathbf{N}_d(i, j) \sim \mathcal{N}(0, \sigma^2)$ for $i \in [1, 2, \dots, N_b], j \in [1, 2, \dots, M_f]$. We show the accuracy of the P2DE for different values of σ^2 ($D = 4, \text{DR}_f = \frac{1}{32}$).

