

Efficient Deep Learning for Massive MIMO Channel State Estimation



Mason del Rosario
Doctoral Qualifying Examination

May 2021

Background

Role of CSI in MIMO

CSI Estimation

Convolutional Neural Networks

Prior Work #1: SphNet

Spherical Normalization

Prior Work #2: MarkovNet

Differential Encoding

Proposed Work: CsiNet-SoftQuant

Soft-to-Hard Vector Quantization

Background

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

Massive MIMO uses numerous antennas to endow transceivers with spatial diversity.

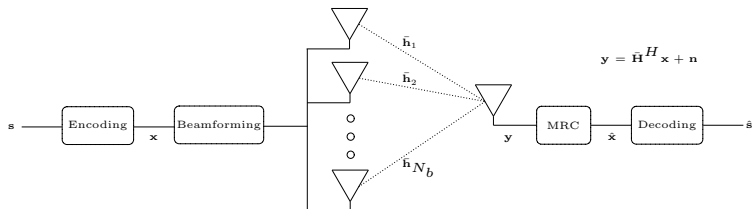
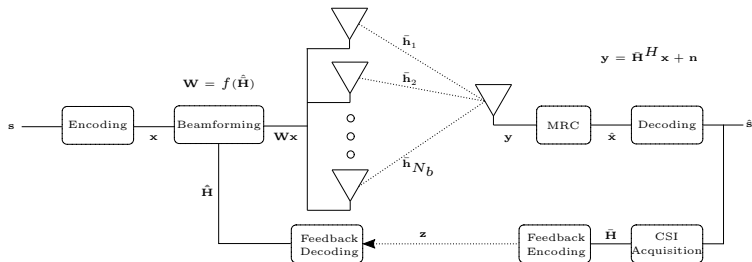


Figure: Example multi-antenna transmitter (BS, gNB) and single-antenna user equipment (UE) and relevant system values.

In OFDM, the fading coefficients between the Tx/Rx antennas constitute **Channel State Information (CSI)**, $\bar{\mathbf{H}}$. For n_T transmit antennas and n_f subcarriers,

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

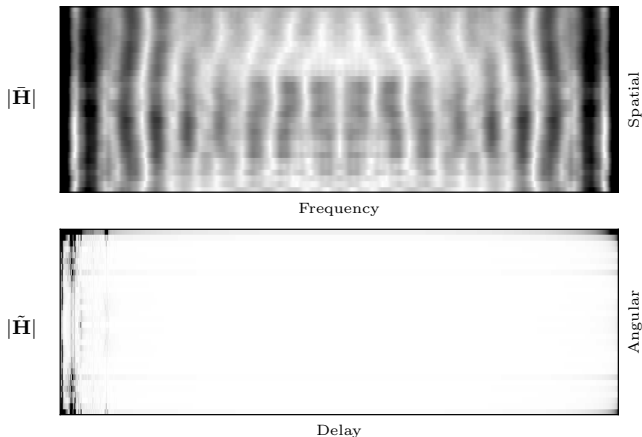
However, transmitting $\bar{\mathbf{H}}$ is costly. Instead, generate **CSI Estimates**, $\hat{\bar{\mathbf{H}}}$, based on **compressed feedback**, \mathbf{z} .



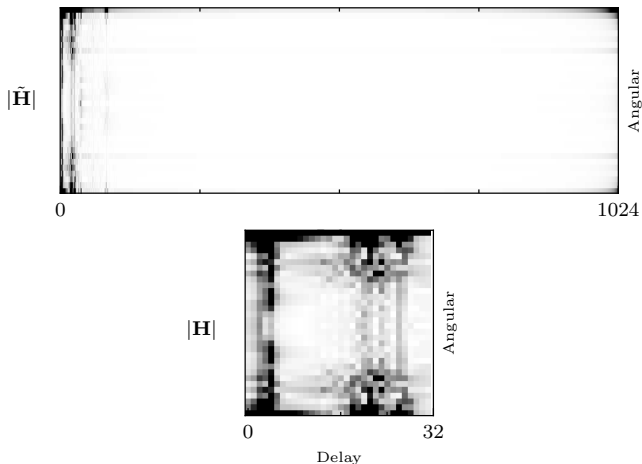
Denote the 2D inverse FFT of $\bar{\mathbf{H}}$ as

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

While $\bar{\mathbf{H}}$ is used for beamforming, \mathbf{H} is more amenable to compression.



Given the sparsity of \mathbf{H} (angular-delay domain), we choose to encode a truncated \mathbf{H} .



1. Compressed Sensing
2. Convolutional Neural Networks

Find a low-dimensional representation of sparse data, \mathbf{h} , by a linear transform, \mathbf{A} ,

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

where \mathbf{h} is a vectorized CSI measurement, \mathbf{A} is the measurement matrix, and \mathbf{n} is an additive noise vector.

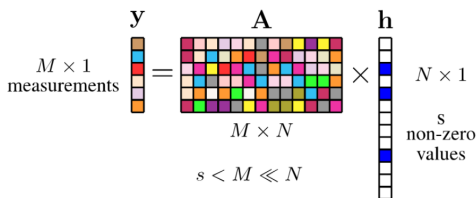


Figure: Compressed sensing via random measurement matrix \mathbf{A} (from [1]).

CS theory relies on the following assumptions:

1. The sparsity of the signal to be estimated must meet a certain level.
2. The **Restricted Isometry Criterion (RIC)** must be met. For $\delta \in [0, 1]$,

$$(1 - \delta)\|\mathbf{h}\|_F^2 \leq \|\mathbf{A}\mathbf{h}\|_F^2 \leq (1 + \delta)\|\mathbf{h}\|_F^2$$

Generally, CS approaches address two major issues:

1. The design of the measurement matrix, \mathbf{A} (stochastic or deterministic).
2. The recovery of $\hat{\mathbf{h}}$ given \mathbf{A} and \mathbf{y} (e.g., Matching Pursuit, Orthogonal Matching Pursuit [2]).

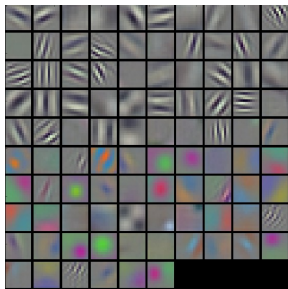
Generally, CS approaches address two major issues:

1. The design of the measurement matrix, \mathbf{A} (stochastic or deterministic).
2. The recovery of $\hat{\mathbf{h}}$ given \mathbf{A} and \mathbf{y} (e.g., Matching Pursuit, Orthogonal Matching Pursuit [2]).

Problem: Recovery algorithms are iterative; complexity scales with M .

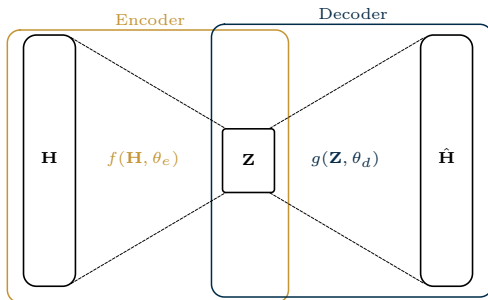
E. C. Marques, N. Maciel, L. A. B. Naviner, H. Cai, and J. Yang, “Compressed Sensing for Wideband HF Channel Estimation,” in *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pp. 1–5, 2018

- ▶ CNNs = state-of-the art performance in image processing
- ▶ Multiple layers of trainable linear functions followed nonlinear ‘activation’ functions.
- ▶ No assumptions on sparsity, RIC. Instantaneous decoding.



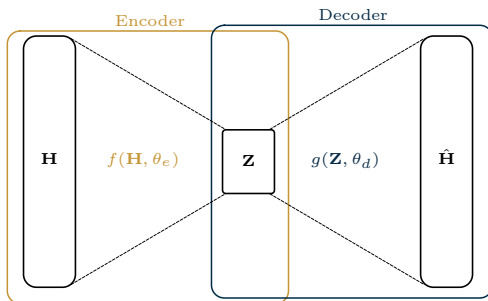
An autoencoder learns a latent code \mathbf{Z} with **compression ratio**,

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



An autoencoder learns a latent code \mathbf{Z} with **compression ratio**,

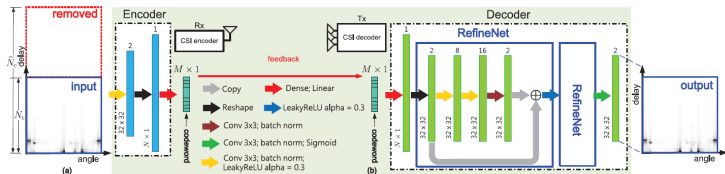
$$\text{CR} = \frac{\dim(\mathbf{Z})}{\dim(\mathbf{H})} \text{ s.t. } \dim(\mathbf{Z}) < \dim(\mathbf{H}).$$



The encoder/decoder parameters θ_e, θ_d are updated via a stochastic optimizer to minimize the **mean-squared error**,

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

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



Metrics used are:

► **Normalized Mean-squared Error**

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

► **Cosine Similarity**

$$\rho = \frac{1}{NN_f} \sum_{t=1}^N \sum_{n=1}^{N_f} \frac{|\hat{\mathbf{h}}_m^H \bar{\mathbf{h}}_m|}{\|\hat{\mathbf{h}}_m\|_F \|\bar{\mathbf{h}}_m\|_F},$$

Metrics used are:

► **Normalized Mean-squared Error**

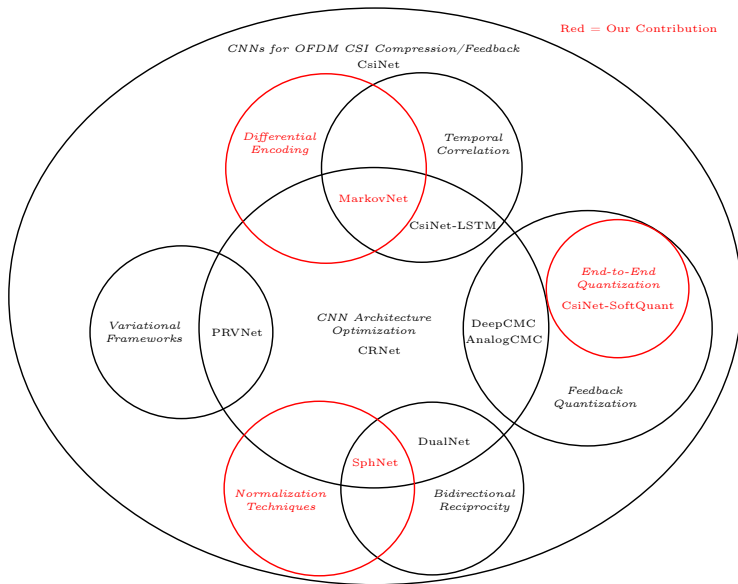
$$\text{NMSE} = \frac{1}{N} \sum_i \frac{\|\mathbf{H}_i - \hat{\mathbf{H}}_i\|_F^2}{\|\mathbf{H}_i\|_F^2}$$

► **Cosine Similarity**

$$\rho = \frac{1}{NN_f} \sum_{t=1}^N \sum_{n=1}^{N_f} \frac{|\hat{\mathbf{h}}_m^H \bar{\mathbf{h}}_m|}{\|\hat{\mathbf{h}}_m\|_F \|\bar{\mathbf{h}}_m\|_F},$$

CNN-based approaches outperform CS-based approaches 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

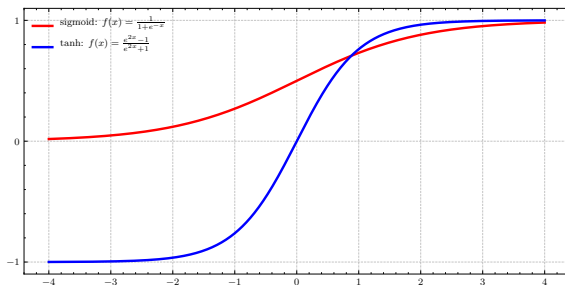


Prior Work #1: SphNet

Power-based normalization for improved CSI reconstruction accuracy.

- ▶ **Minmax scaling** for typical works
- ▶ Take extrema (H_{\min} , H_{\max}) of real and imaginary channels.
- ▶ $H_{n,(i,j)} = (i,j)$ -th element of n -th sample

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



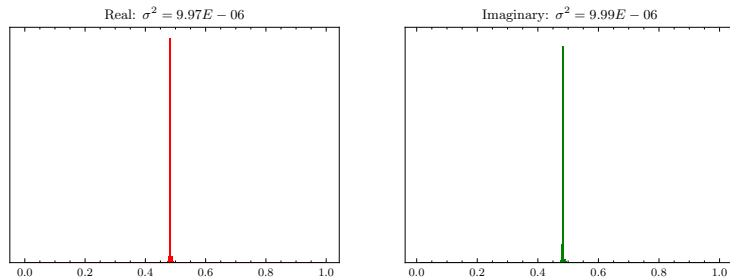


Figure: Distribution and variance of minmax-normalized COST2100 real/imaginary channels ($N = 99000$) images.

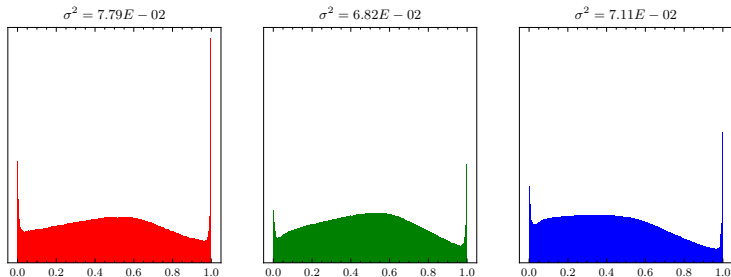


Figure: Distribution and variance of minmax-normalized ImageNet color channels ($N = 50000$) images.

Difference of **four orders of magnitude**.

Dataset	Channels	Normalization	Avg. Variance
ImageNet	RGB	Minmax	<u>$7.24E^{-2}$</u>
COST2100	Real, Imag	Minmax	<u>$9.98E^{-6}$</u>

Table: Minmax normalization applied to COST2100 and ImageNet dataset.

Spherical normalization – scale each channel sample by its 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.

Resulting dataset under spherical normalization exhibits a larger variance than minmax scaling.

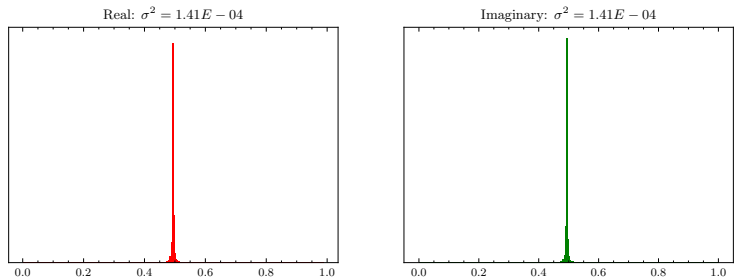


Figure: Distribution/variance of COST2100 real/imaginary channels under spherical normalization ($N = 99000$) images.

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

Dataset	Channels	Normalization	Avg. Variance
ImageNet	RGB	Minmax	<u>$7.24E^{-2}$</u>
COST2100	Real, Imag	Spherical	<u>$1.41E^{-4}$</u>
COST2100	Real, Imag	Minmax	$9.98E^{-6}$

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

Under spherical normalization, MSE loss becomes equivalent to NMSE.
Recall the definitions,

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

Under spherical normalization, MSE loss becomes equivalent to NMSE.
Recall the definitions,

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

The MSE of the spherically normalized estimator is equivalent to the NMSE of the regular estimator, i.e.

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

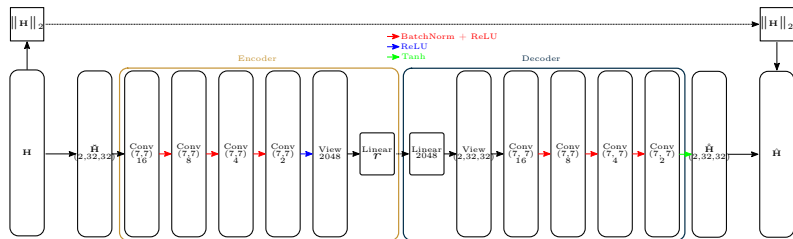


Figure: SphNet – CsiNetPro architecture with Spherical Normalization.

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

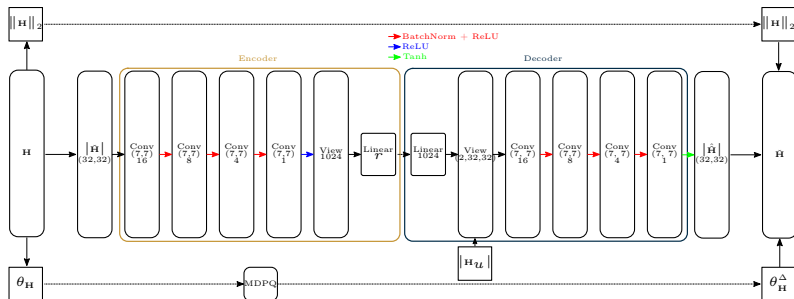


Figure: DualNet-Sph – CsiNetPro architecture with Spherical Normalization and Bidirectional Reciprocity.

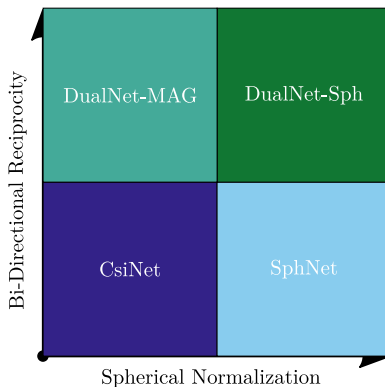


Figure: Illustration of techniques used in different models.

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. gNB Antennas (N_T)	32	
Num. Subcarriers (N_f)	1024	
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	

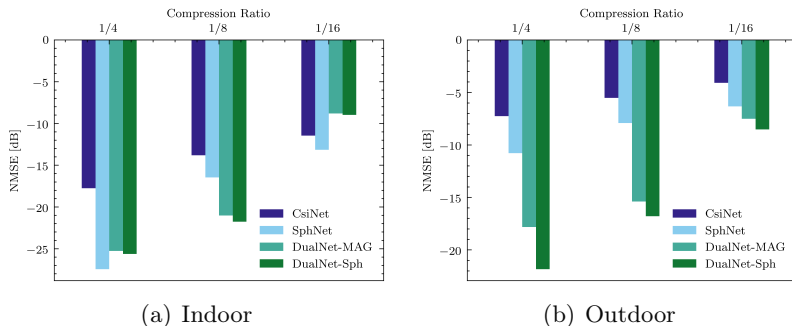


Figure: NMSE (lower is better) comparison of bidirectional reciprocity and spherical normalization against CsiNet for increasing compression ratio [4]

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

Prior Work #2: MarkovNet

A deep differential autoencoder for efficient temporal learning.

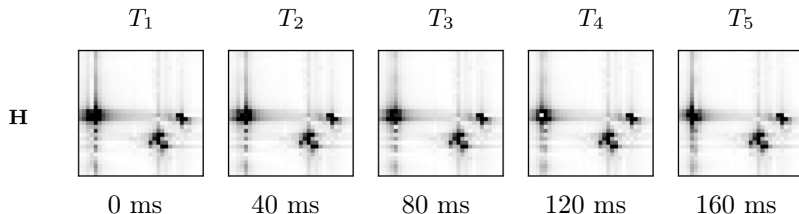


Figure: Ground truth CSI (\mathbf{H}) for five timeslots (T_1 through T_5) on one outdoor sample from the validation set.

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

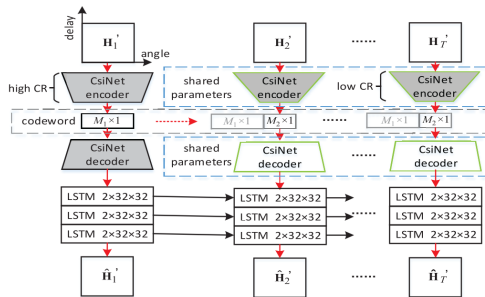


Figure: CsiNet-LSTM network architecture [6].

LSTMs improve NMSE at smaller compression ratios.

		CR	LASSO	BM3D-AMP	TVAL3	CsiNet	CsiNet-LSTM
Indoor	NMSE	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	0.99
		1/32	0.53	0.17	0.45	0.90	0.99
		1/64	0.30	0.16	0.24	0.87	0.99
	runtime	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	NMSE	1/16	-1.09	0.40	-0.53	-3.60	-9.86
		1/32	-0.27	18.99	0.42	-2.14	-9.18
		1/64	-0.06	24.42	0.74	-1.65	-8.83
	ρ	1/16	0.49	0.23	0.46	0.75	0.95
		1/32	0.32	0.16	0.28	0.63	0.94
		1/64	0.19	0.16	0.19	0.58	0.93
	runtime	1/16	0.2122	0.4210	0.3145	0.0001	0.0003
		1/32	0.2409	0.6031	0.2985	0.0001	0.0003
		1/64	0.0166	0.5980	0.2850	0.0001	0.0003
	NMSE↓	1/16-1/64	94%	60	2.40	54%	10%

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

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

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

	Parameters		FLOPs	
CR	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

Instead of learning a temporal dependency across multiple timeslots, we proposed a **one-step differential encoder**.

For short enough feedback intervals between t and $t - 1$, we view CSI data as a Markov chain, i.e.

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

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

Z. Liu †, M. del Rosario †, and Z. Ding, “A Markovian Model-Driven Deep Learning Framework for Massive MIMO CSI Feedback,” *arXiv e-prints*, Sept.

2020. Submitted to IEEE Transactions on Wireless Communications († 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

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

We utilize the estimator, $\hat{\gamma}$, based on the second-order statistics of the CSI matrices,

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

for training set of size N .

With the one-step estimator $\hat{\gamma}$, we propose train an encoder for the estimation error as

$$\mathbf{z}_t = f_{e,t}(\mathbf{H}_t - \hat{\gamma}\hat{\mathbf{H}}_{t-1}),$$

and we jointly train a decoder,

$$\hat{\mathbf{H}}_t = f_{d,t}(\mathbf{z}_t) + \hat{\gamma}\hat{\mathbf{H}}_{t-1}$$

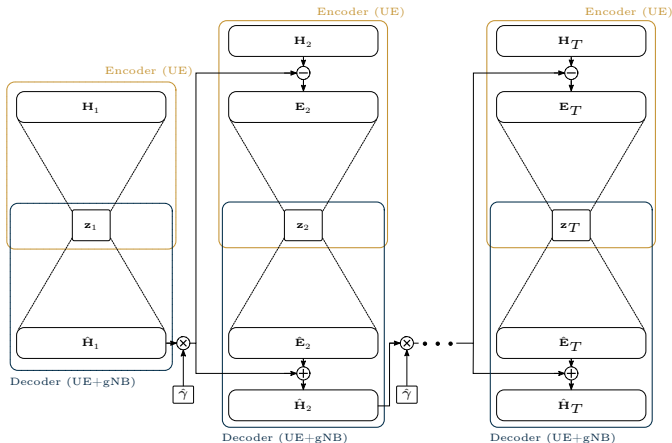
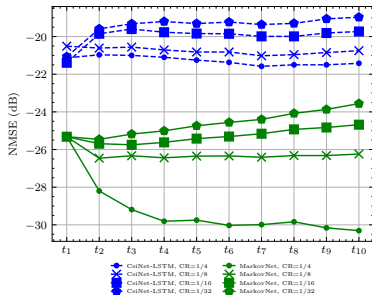
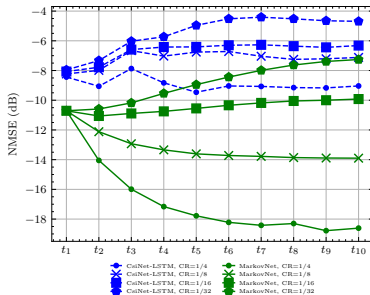


Figure: Abstract architecture for MarkovNet. Networks at $t \geq 2$ are trained to predict the estimation error, E_t .



(a) Indoor



(b) Outdoor

Figure: NMSE comparison of MarkovNet and CsiNet-LSTM at various compression ratios (CR).

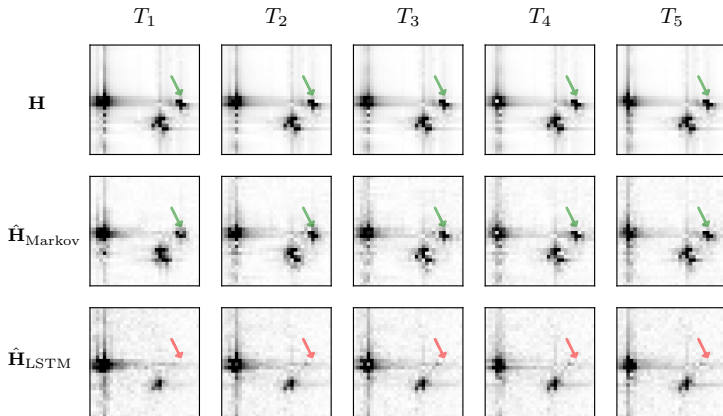


Figure: Ground truth CSI (\mathbf{H}), MarkovNet estimates ($\hat{\mathbf{H}}_{\text{Markov}}$), and CsiNet-LSTM estimates ($\hat{\mathbf{H}}_{\text{LSTM}}$) for five timeslots (T_1 through T_5) on one outdoor sample from the test set (both networks at $\text{CR} = \frac{1}{4}$).

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

Proposed Work: CsiNet-SoftQuant

An end-to-end trained autoencoder with learned feedback quantization.

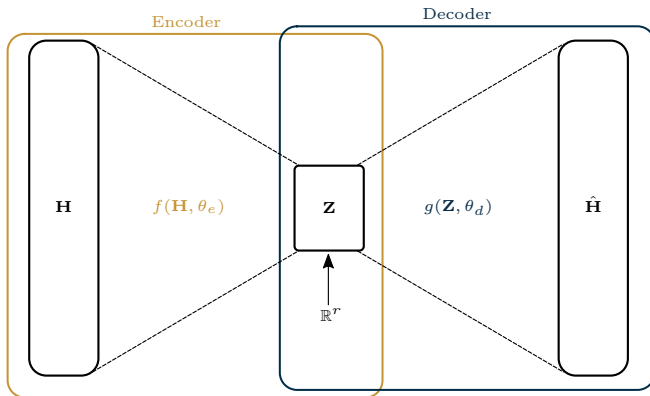


Figure: Autoencoder architecture with r -dimensional real-valued latent feedback elements.

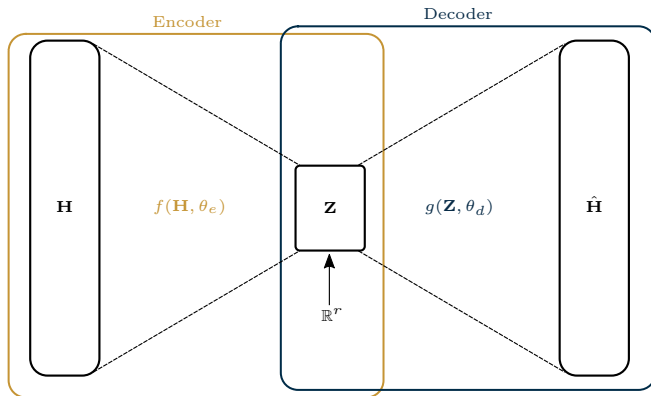


Figure: Autoencoder architecture with r -dimensional real-valued latent feedback elements.

Problem: Feedback elements must be discrete-valued. How to quantize?

Network with uniform quantization and arithmetic encoding of latent vectors.

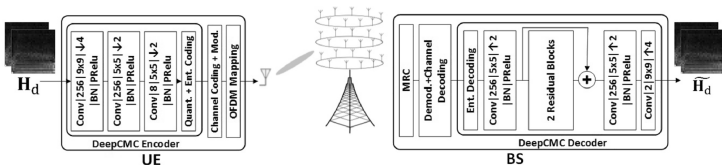


Figure: Architecture for DeepCMC [8]. Network uses entropy encoding of uniform quantized feedback elements to minimize bit rate.

Network with uniform quantization and arithmetic encoding of latent vectors.

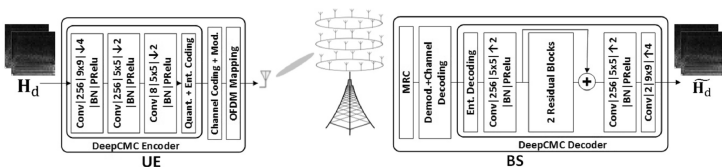


Figure: Architecture for DeepCMC [8]. Network uses entropy encoding of uniform quantized feedback elements to minimize bit rate.

Is fixed quantization scheme optimal?

We propose to use soft-to-hard vector quantization (SHVQ) [9]. Define the m -dimensional codebook of size L as $\mathbf{C} \in \mathbb{R}^{m \times L}$. The soft vector assignments of the j -th latent vector $\tilde{\mathbf{z}}_j$ can be written as,

$$\phi(\tilde{\mathbf{z}}_j) = \left[\frac{\exp(-\sigma \|\tilde{\mathbf{z}}_j - \mathbf{c}_\ell\|^2)}{\sum_{i=1}^L \exp(-\sigma \|\tilde{\mathbf{z}}_j - \mathbf{c}_i\|^2)} \right]_{\ell \in [L]} \in \mathbb{R}^L, \quad (2)$$

which is referred to as the ‘softmax’ function. σ is a *temperature* or *annealing* parameter which controls the degree of quantization,

$$\lim_{\sigma \rightarrow \infty} \phi(\tilde{\mathbf{z}}_j) = \text{onehot}(\tilde{\mathbf{z}}_j) = \begin{cases} 1 & \ell = \underset{\ell}{\operatorname{argmax}} \phi(\tilde{\mathbf{z}}_j)[\ell] \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, “Soft-to-hard Vector Quantization for End-to-end Learning Compressible Representations,” *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. NIPS, pp. 1142–1152, 2017

The soft assignments ϕ admit probability masses over the codewords,

$$q_j = \phi(\tilde{\mathbf{z}}_j).$$

Based on finite samples, we define the histogram probability estimates p_j

$$p_j = \frac{|\{e_l(\mathbf{z}_i) | l \in [m], i \in [N], e_l(\mathbf{z}_i) = j\}|}{mN}.$$

Our target for the rate loss is the crossentropy between p_j and q_j term,

$$H(\phi) := H(p, q) = - \sum_{j=1}^L p_j \log q_j = H(p) + D_{\text{KL}}(p \| q).$$

Loss function for soft quantization = regularized rate-distortion

$$\operatorname{argmin}_{\theta_e, \theta_d, \mathbf{C}} \overbrace{L_d(\mathbf{H}, \hat{\mathbf{H}})}^{\text{distortion}} + \lambda \overbrace{L_{\ell^2}(\theta_e, \theta_d, \mathbf{C})}^{\ell_2 \text{ penalty}} + \beta \overbrace{L_r(\theta_e, \mathbf{C})}^{\text{rate}} \quad (4)$$

Where the different loss terms are

Term	Definition
$L_d(\mathbf{H}, \hat{\mathbf{H}})$	$\frac{1}{N} \sum_{i=1}^N \ \mathbf{H}_i - g(Q(f(\mathbf{H}_i, \theta_e), \mathbf{C}), \theta_d)\ ^2$
$L_{\ell^2}(\theta_e, \theta_d, \mathbf{C})$	$\ \theta_e\ ^2 + \ \theta_d\ ^2 + \ \mathbf{C}\ ^2$
$L_r(\theta_e, \mathbf{C})$	$mH(\phi)$

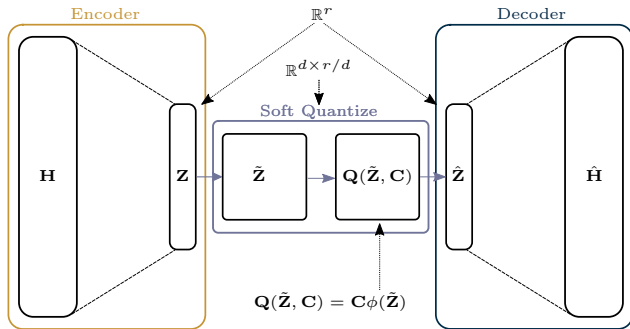


Figure: Abstract architecture for CsiNet-SoftQuant [8]. SoftQuantize layer ($\mathbf{Q}(\tilde{\mathbf{Z}})$) is a continuous, softmax-based relaxation of a d -dimensional quantization of the latent layer \mathbf{Z} .

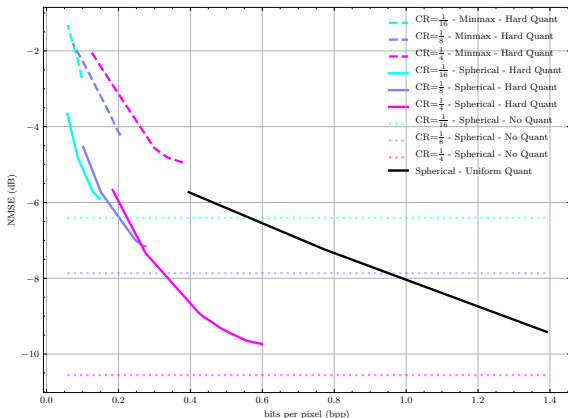


Figure: Rate distortion of CsiNet-SoftQuant under both minmax (dotted line) and spherical (solid line) normalization using: $L = 1024$ centers, $d = 4$. Hard quantization performance shown for each CR.

Given angular-delay domain CSI data, \mathbf{H} , assume i.i.d. $\mathbf{H}_{(i,j)}$ for i -th (j -th) row (col).

Given angular-delay domain CSI data, \mathbf{H} , assume i.i.d. $\mathbf{H}_{(i,j)}$ for i -th (j -th) row (col).

Denote the quantized CSI matrix, \mathbf{H}^Δ , quantized with b bits. The entropy of the (i, j) -th element is

$$H(\mathbf{H}_{(i,j)}^\Delta) = - \sum_k^{2^b} p(\mathbf{H}_{(i,j)}^\Delta = k) \log p(\mathbf{H}_{(i,j)}^\Delta = k),$$

where $p(\mathbf{H}_{(i,j)}^\Delta = k)$ can be obtained as a histogram estimate over the entire dataset.

Given angular-delay domain CSI data, \mathbf{H} , assume i.i.d. $\mathbf{H}_{(i,j)}$ for i -th (j -th) row (col).

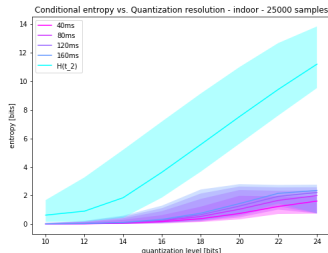
Denote the quantized CSI matrix, \mathbf{H}^Δ , quantized with b bits. The entropy of the (i, j) -th element is

$$H(\mathbf{H}_{(i,j)}^\Delta) = - \sum_k^{2^b} p(\mathbf{H}_{(i,j)}^\Delta = k) \log p(\mathbf{H}_{(i,j)}^\Delta = k),$$

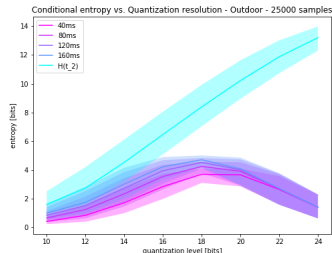
where $p(\mathbf{H}_{(i,j)}^\Delta = k)$ can be obtained as a histogram estimate over the entire dataset.

A conservative upper bound on the entropy of the full CSI matrix is

$$H(\mathbf{H}^\Delta) = \frac{1}{R_d n_T} \sum_i^{R_d} \sum_j^{n_T} H(\mathbf{H}_{(i,j)}^\Delta).$$



(a) Indoor



(b) Outdoor

Figure: Mean entropy/conditional entropy estimates $H(\mathbf{H}^\Delta)$ with 95% c.i. for quantized i.i.d COST2100 elements vs. quantization level (bits).

Again, assume i.i.d. $\mathbf{H}_{(i,j)}$ for i -th (j -th) row (col).

Again, assume i.i.d. $\mathbf{H}_{(i,j)}$ for i -th (j -th) row (col).

The differential entropy of the (i, j) -th element is

$$\hat{h}(\mathbf{H}_{(i,j)}) = - \int p(\mathbf{H}(i, j) = k) \log p(\mathbf{H}_{(i,j)} = k) dk,$$

Again, assume i.i.d. $\mathbf{H}_{(i,j)}$ for i -th (j -th) row (col).

The differential entropy of the (i, j) -th element is

$$\hat{h}(\mathbf{H}_{(i,j)}) = - \int p(\mathbf{H}(i, j) = k) \log p(\mathbf{H}_{(i,j)} = k) dk,$$

$p(\mathbf{H}_{(i,j)})$ is difficult to obtain. Instead resort to Kozachenko–Leonenko (KL) estimator [10] for each element in \mathbf{H} and average over the elements,

$$\hat{h}(\mathbf{H}) = \frac{1}{R_d n_T} \sum_i^{R_d} \sum_j^{n_T} \hat{h}(\mathbf{H}_{(i,j)}),$$

for KL estimator \hat{h} .

L. Kozachenko and N. N. Leonenko, “Sample Estimate of the Entropy of a Random Vector,” *Problemy Peredachi Informatsii*, vol. 23, no. 2, pp. 9–16, 1987

Based on Theorem 8.3.1 from Cover [11], – for sufficiently small quantization interval $\Delta = \frac{1}{2^n}$, the entropy of a quantized random variable is related to its differential entropy as,

$$H(\mathbf{H}^\Delta) = h(\mathbf{H}) + n,$$

for n -bit quantization. Thus, the differential entropy estimator admits an estimate for the entropy of the quantized CSI, $\hat{\mathbf{H}}^\Delta$.

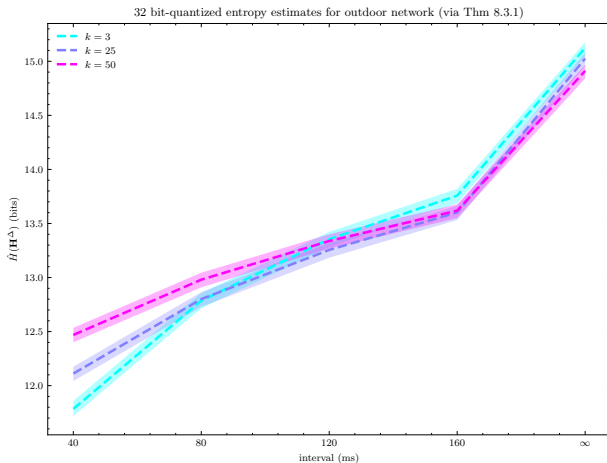


Figure: Mean entropy/conditional entropy estimates $\hat{H}(\mathbf{H}^\Delta) = \hat{h}(\mathbf{H}) + n$ with 95% c.i. for quantized i.i.d COST2100 elements vs. quantization level (bits).

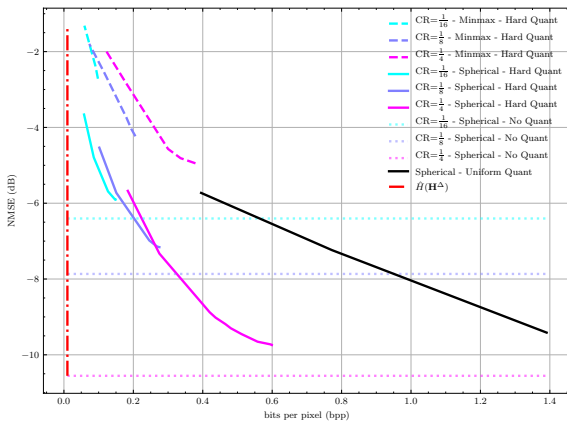


Figure: Rate distortion of CsiNet-SoftQuant on outdoor dataset with entropy bound based on differential entropy estimate.

Questions?

`mdelrosa@ucdavis.edu`

- [1] 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.
- [2] E. C. Marques, N. Maciel, L. A. B. Naviner, H. Cai, and J. Yang, "Compressed Sensing for Wideband HF Channel Estimation," in *2018 4th International Conference on Frontiers of Signal Processing (ICFSP)*, pp. 1–5, 2018.
- [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, 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.
- [5] Z. Liu, L. Zhang, and Z. Ding, "Exploiting Bi-Directional Channel Reciprocity in Deep Learning for Low Rate Massive MIMO CSI Feedback," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 889–892, 2019.
- [6] 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.
- [7] Z. Liu [†], M. del Rosario [†], and Z. Ding, "A Markovian Model-Driven Deep Learning Framework for Massive MIMO CSI Feedback," *arXiv e-prints*, Sept. 2020.
Submitted to IEEE Transactions on Wireless Communications.
- [8] Q. Yang, M. B. Mashhadi, and D. Gündüz, "Deep Convolutional Compression For Massive MIMO CSI Feedback," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2019.
- [9] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, "Soft-to-hard Vector Quantization for End-to-end Learning Compressible Representations," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. NIPS, pp. 1142–1152, 2017.
- [10] L. Kozachenko and N. N. Leonenko, "Sample Estimate of the Entropy of a Random Vector," *Problemy Peredachi Informatsii*, vol. 23, no. 2, pp. 9–16, 1987.

Appendix

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,

$$\hat{\mathbf{H}}_t = \mathbf{H}_{t-1} \mathbf{W}_1 + \dots + \mathbf{H}_{t-p} \mathbf{W}_p + \mathbf{E}_t. \quad (5)$$

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 ,

$$\begin{aligned}\mathbf{H}_{t-i}^H \hat{\mathbf{H}}_t &= \mathbf{H}_{t-i}^H \mathbf{H}_{t-1} \mathbf{W}_1 + \dots + \mathbf{H}_{t-i}^H \mathbf{H}_{t-p} \mathbf{W}_p + \mathbf{H}_{t-i}^H \mathbf{E}_t \\ &= \mathbf{H}_{t-i}^H \mathbf{H}_{t-1} \mathbf{W}_1 + \dots + \mathbf{H}_{t-i}^H \mathbf{H}_{t-p} \mathbf{W}_p.\end{aligned}\tag{6}$$

Denote the correlation matrix $\mathbf{R}_i = \mathbb{E}[\mathbf{H}_{t-i}^H \mathbf{H}_t]$. We presume the CSI matrices are generated by a stationary process, and consequently, they have 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, we can write (6) as a linear combination of correlation matrices,

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

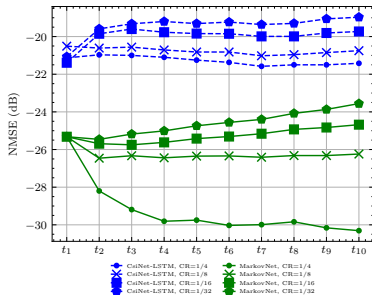
For p CSI samples, we can write a system of p equations, which admits the following,

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

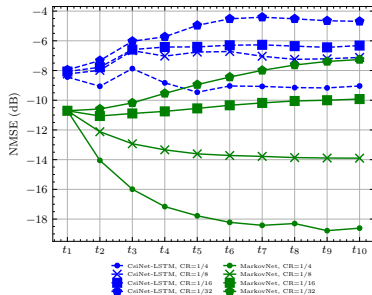
Solving for the coefficient matrices admits the solution

$$\begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \dots \\ \mathbf{W}_p \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 & \mathbf{R}_1^H & \dots & \mathbf{R}_{p-1}^H \\ \mathbf{R}_1 & \mathbf{R}_0 & \dots & \mathbf{R}_{p-2}^H \\ \vdots & & \ddots & \vdots \\ \mathbf{R}_{p-1} & \mathbf{R}_{p-2} & \dots & \mathbf{R}_0 \end{bmatrix}^+ \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \dots \\ \mathbf{R}_p \end{bmatrix}, \quad (7)$$

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

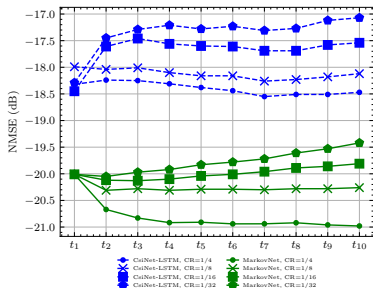


(a) Indoor

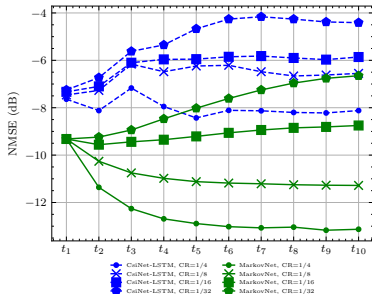


(b) Outdoor

Figure: $\text{NMSE}_{\text{truncated}}$ comparison of MarkovNet and CsiNet-LSTM at various compression ratios (CR).



(a) Indoor



(b) Outdoor

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

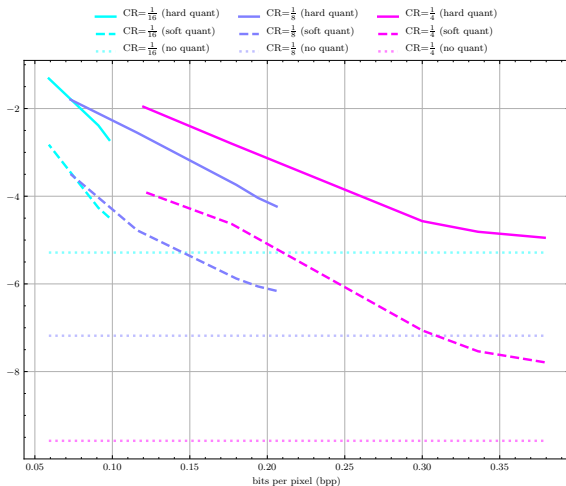


Figure: Rate distortion of CsiNet-SoftQuant under minmax normalization using: $L = 1024$ centers, $d = 4$.

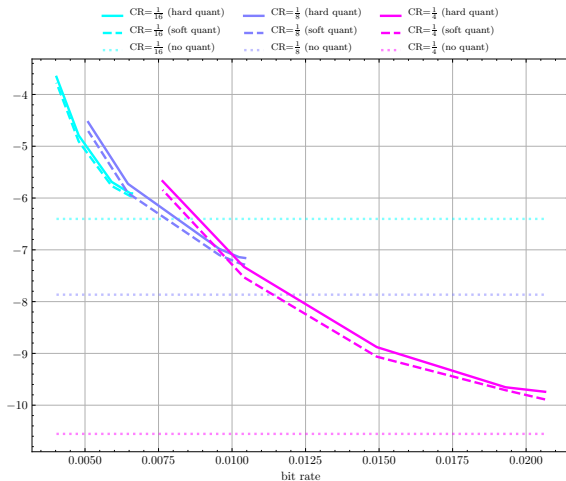


Figure: Rate distortion of CsiNet-SoftQuant using: $L = 1024$ centers, $CR = \frac{1}{4}$, $d = 4$. Bit rates are realized under arithmetic coding of quantized features.

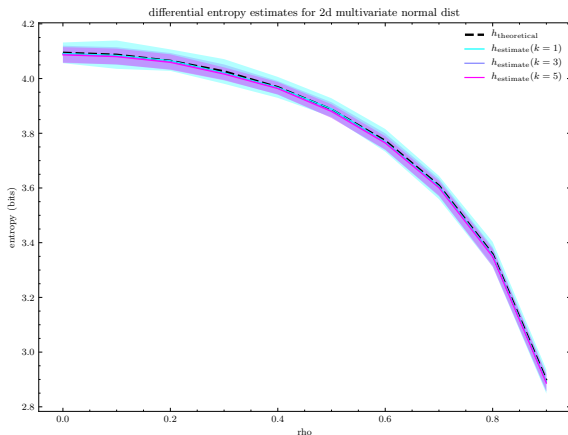


Figure: Differential entropy and estimates for 2d multivariate normal distribution. Estimates are based on the KL estimator [10] using the NPEET library [12].