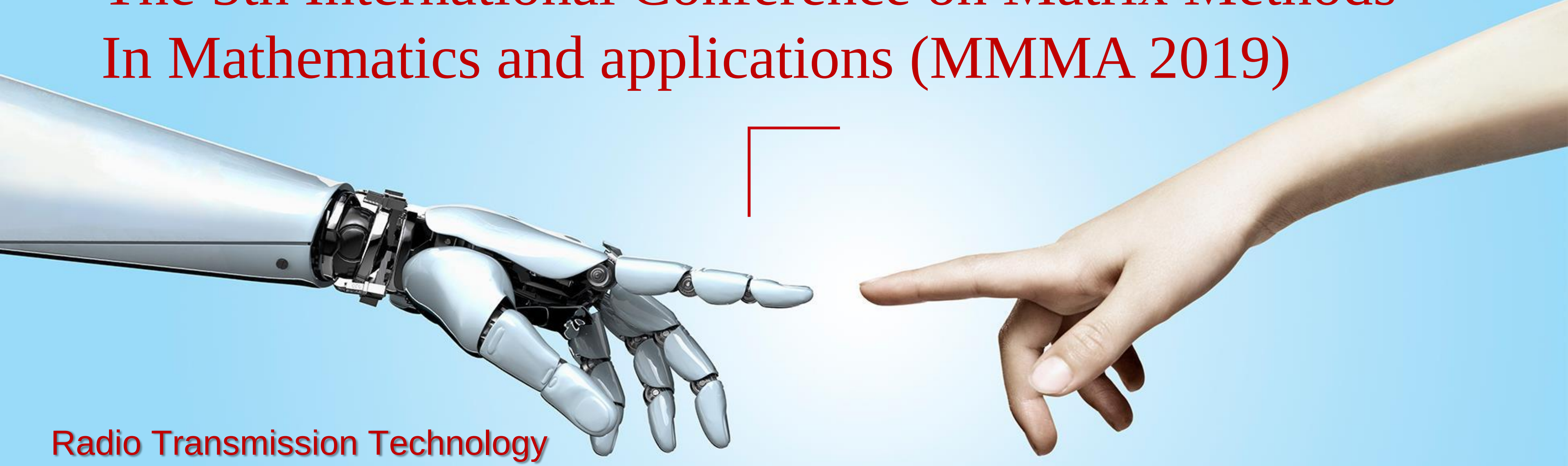
August/ 2019

The 5th International Conference on Matrix Methods In Mathematics and applications (MMMA 2019)



Radio Transmission Technology Algorithmic Laboratory (RTT Lab.)

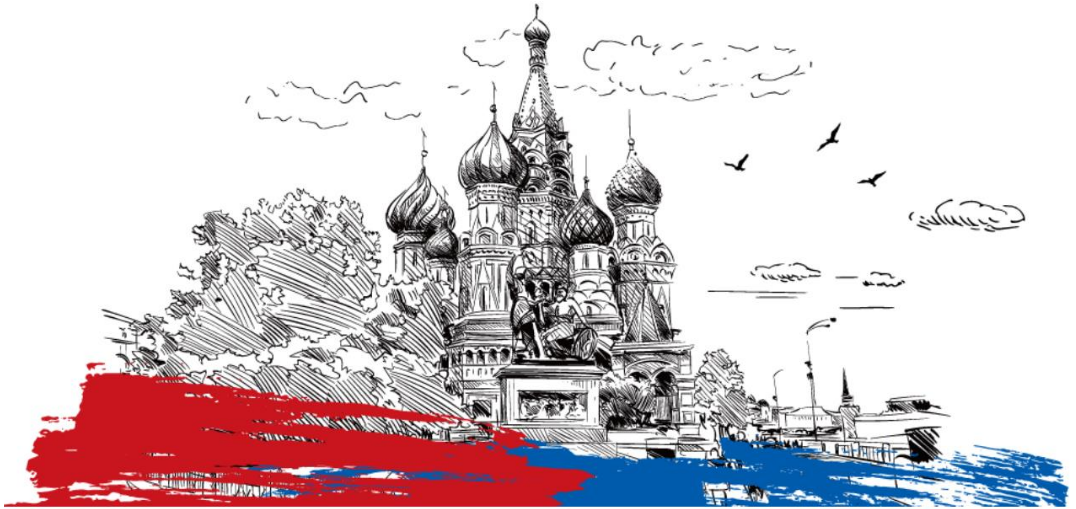
Usatyuk Vasily[usatyuk.vasily@huawei.com]

Moscow Research Center

The 5th International Conference on Matrix Methods in Mathematics and applications (MMMA 2019)

19th-23th Aug 2019

Moscow, Russia



Radio Transmission Technology Algorithmic Laboratory (RTT Lab.)

Moscow Research Center

Anything

Modern ~~Signal~~ Processing from Mathematics point of view

Estimate, Forecast, Compress, Control

We doing Multi-objective optimization ...

by Machine Learning(Data Driven based GPM, DNN ...), Topology Embedding, Linear Integer Programming, Probabilistically relaxation, apply different regularizations and preconditions, solve ODE/PDE...

to make smooth enough manifold (topological complexes)

Anything

Modern ~~Signal~~ Processing from Physics point of view

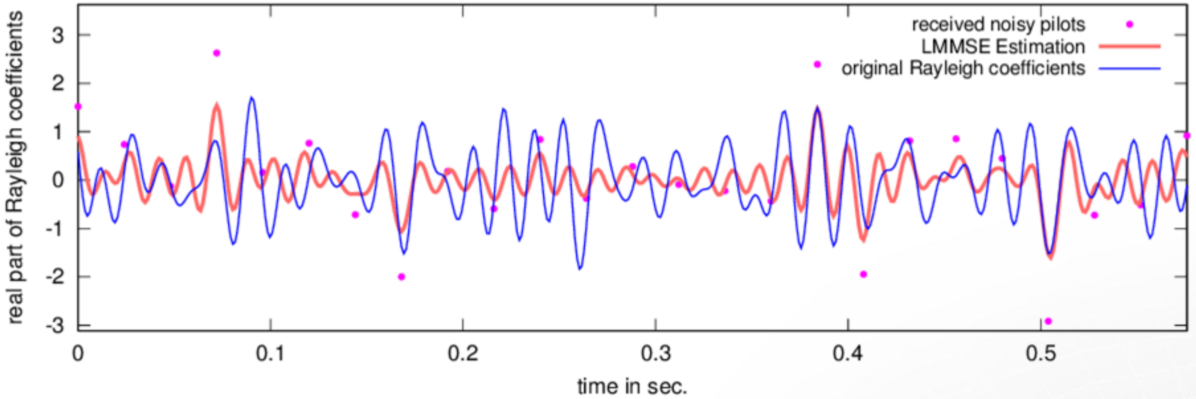
Estimate, Forecast, Compress, Control

We exploit Physical cost functions curvature property (Quantization and Metric)

at symmetric (locality) high temperature (high SNR) phase and
non-symmetric* (action-at-a-distance) low temperature (low Signal-to-Noise Ratio) phase

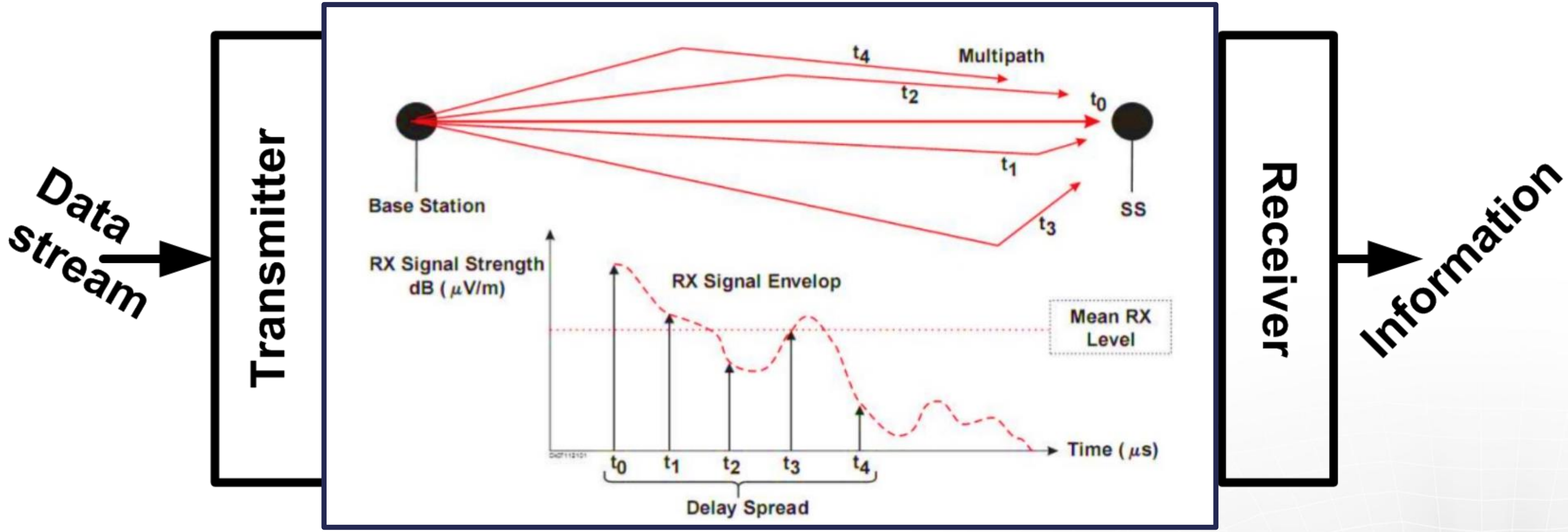
because we know Physical law cause symmetry
the topology of such symmetry with a sufficient level of entropy (admissible error level)
provide to us smooth enough manifold (topological complexes)

*- such non-symmetric phase still admits symmetry if intrinsic entropy allows take Lie bracket, not violate the uncertainty principle



Deep Neural Network Channel Estimation

Multiple Input – Multiple Output Communications



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

$$\mathbf{x} \in \mathbb{C}^{M\times 1}, \mathbf{y} \in \mathbb{C}^{N\times 1}, \mathbf{H} = \mathbf{R}_{TX}^{1/2}\mathcal{N}(0, \mathbf{I}) \in \mathbb{C}^{N\times M}$$

To measure H, *we use souding* x_p *by sending pilots sequence* $\quad y_p = Hx_p + n$

Tradition methods for channel estimation

$$y_k = x_k H_k + n_{\ k},$$

y_k-received noisy signal, x_k $-pilot\ signal,$ H_k-channel transfer function, $n - noise$

Least−Squares (LS) channel estimation:

$$H_k^{LS} = x_k^{\ -1} y_{\ k}$$

Tradition methods for channel estimation

$$y_k = x_k H_k + n_k,$$

where y_k-received noisy signal, x_k $-pilot\ signal,$
H_k-channel transfer function, $n - noise$

Least−Squares (LS) channel estimation:

$$H_k^{LS} = x_k{}^{-1} y_k$$

Linear Minimum Mean Square Error Wiener filter channel estimation (Soft Window):

$$H_k^{MMSE} = R_{H\tilde{H}}(R_{HH} + \sigma^2 I)^{-1} H_k^{LS}$$

Where σ^2 - noise variance, R_{HH}- the autocorrelation matrix of H_k^{LS},
$R_{H\tilde{H}}$−frequency dom. crooss−corelation matrix

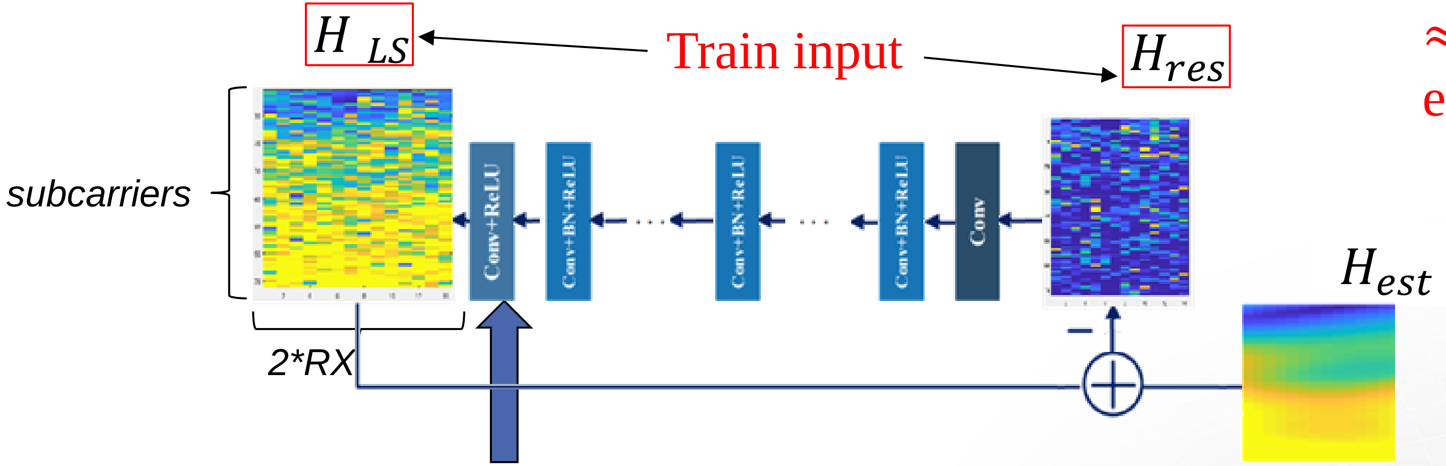
$$R_{H\tilde{H}}(k,l) = E\left[h_{k,l}\tilde{h}_{k',l'}\right] = r_f[k-k']\, r_t[l-l']$$

$$r_f[k] = (1 + j2\pi k \triangle f)^{-1}, \quad r_t[l] = J_0(2\pi f_D t l),$$

J_0- zero-th order Bessel function, f_D- maximum Doppler shift at delay t

Denoising Deep Convolutional Neural Network(dnCNN) for uplink channel estimation

Residual Learning Train(backpropagation):



$\approx 4 \times 10^4$ label data enough to train:

H_{LS}, H_{res}

One denoising stack contain 3 real value layers:
conv2d (Conv2D), Batch_normalization, activation_ (ReLU)

$$H_{res} = H_{LS} - H_{ideal};$$

H_{res}-residual noise, H_{ideal}- ideal channel est
H_{LS}- LS channel estimation
H_{est}-dnCNN denoised channel estimation

dnCNN for uplink channel estimation

Denoising 1st layer

Denoising 15 layer

1	imageinput 12x128x1 images with 'zerocenter' normalization	Image Input	12×128×1	-
2	conv_1 128 3x3x1 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	12×128×128	Weights 3×3×1×128 Bias 1×1×128
3	relu_1 ReLU	ReLU	12×128×128	-
4	conv_2 128 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	12×128×128	Weights 3×3×128×128 Bias 1×1×128
5	batchnorm_1 Batch normalization with 128 channels	Batch Normalization	12×128×128	Offset 1×1×128 Scale 1×1×128
6	relu_2 ReLU	ReLU	12×128×128	-
7	conv_3 128 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	12×128×128	Weights 3×3×128×128 Bias 1×1×128
8	batchnorm_2 Batch normalization with 128 channels	Batch Normalization	12×128×128	Offset 1×1×128 Scale 1×1×128
9	relu_3 ReLU	ReLU	12×128×128	-

. . .

46	conv_16 128 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	12×128×128	Weights 3×3×128×128 Bias 1×1×128
47	batchnorm_15 Batch normalization with 128 channels	Batch Normalization	12×128×128	Offset 1×1×128 Scale 1×1×128
48	relu_16 ReLU	ReLU	12×128×128	-
49	conv_17 1 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	12×128×1	Weights 3×3×128 Bias 1×1
50	regressionoutput mean-squared-error	Regression Output	-	-

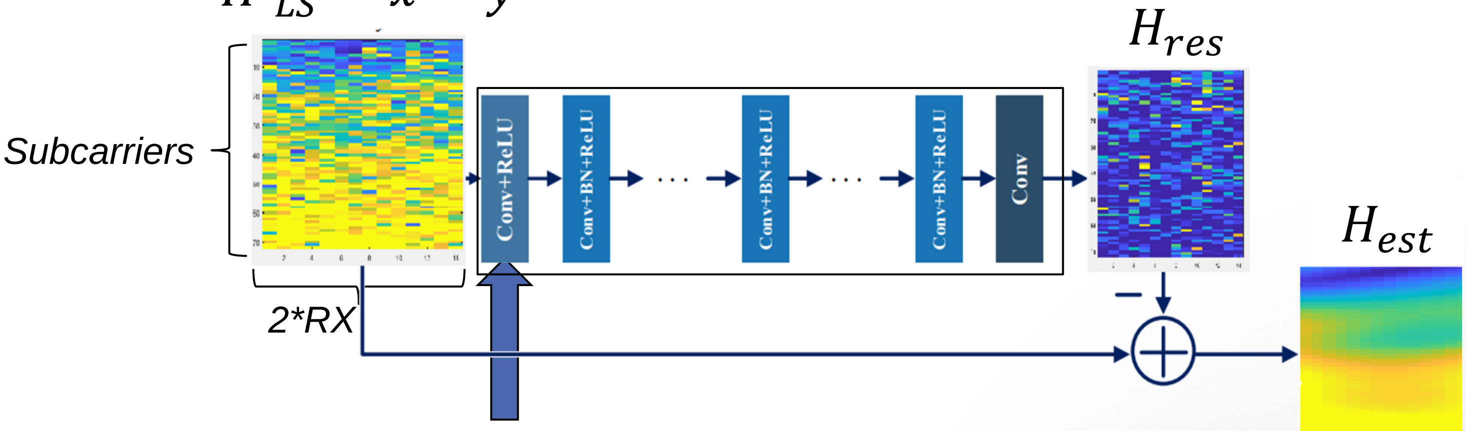
Complexity $O(d \cdot N^3)$ –inference(forward-propagation)

d –hidden layers number (17), N – trainable parameters $(\approx 6 \times 10^5)$

Denoising Deep Convolutional Neural Network(dnCNN) for uplink channel estimation

Residual learning Inference (forward propagation) :

$$H_{LS} = x^{-1}y$$

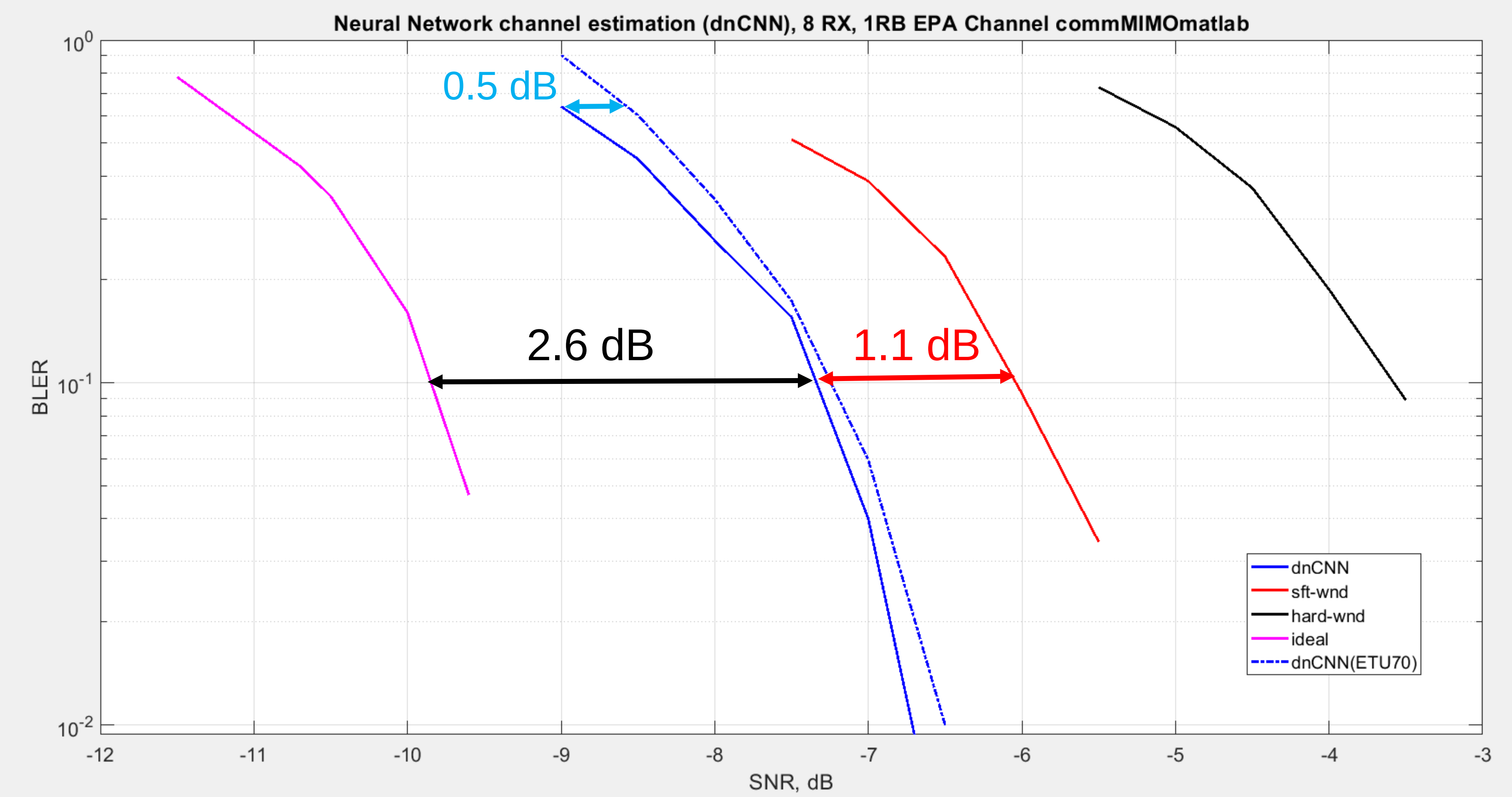


One denoising stack contain 3 real value layers:
conv2d (Conv2D), Batch_normalization, activation_ (ReLU)

$$H_{res} = dnCNN.inference(H_{LS});$$
$$H_{est}=H_{LS}-H_{res}.$$

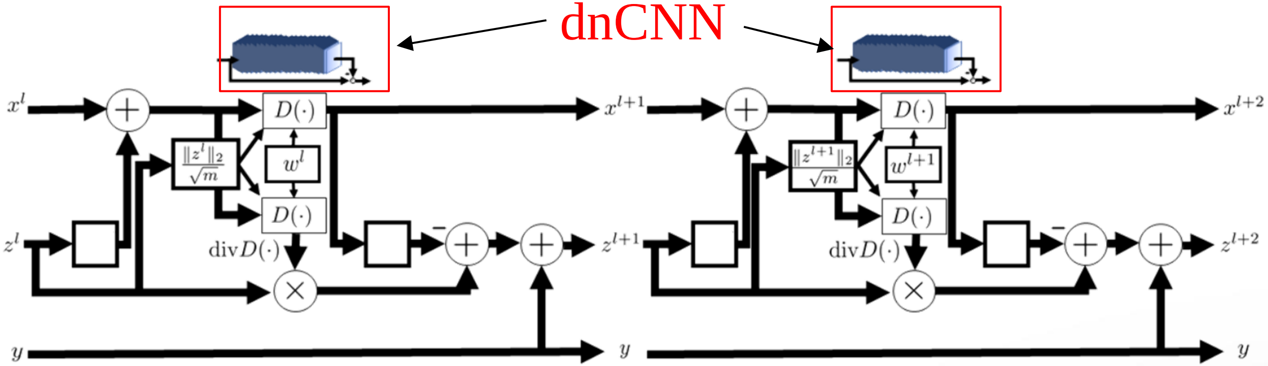
H_{res}-residual noise
H_{LS}- Leas square (LS) channel estimation
H_{est}-dnCNN denoised channel estimation

Generalization power of Deep Neural Network 12x16 image



dnCNN Trained on ETU70 channel work under EPA5

Further improvement: *Combine Deep denoising CNN and compressed sensing*:*



$l \leq 10$

Represented two layers of the LDAMP neural network. When used with the DnCNN denoiser, each denoiser block is a 16 to 20 stack of convolutional-layer neural network (Conv2D 3x3x64, Batch Normalization, Activation(ReLu))

$$argmin_x\|y - Hx\|_2^2 : z^t = y - Hx^t + b^t$$

$$b^t = \frac{z^{t-1} divD_{\sigma^{t-1}}(x^{t-1} + H^T z^{t-1})}{m}$$

$$\sigma^t = \frac{\|z^t\|}{\sqrt{m}}, x^{t+1} = D_{\sigma^{t-1}}(x^t + H^T z^t)$$

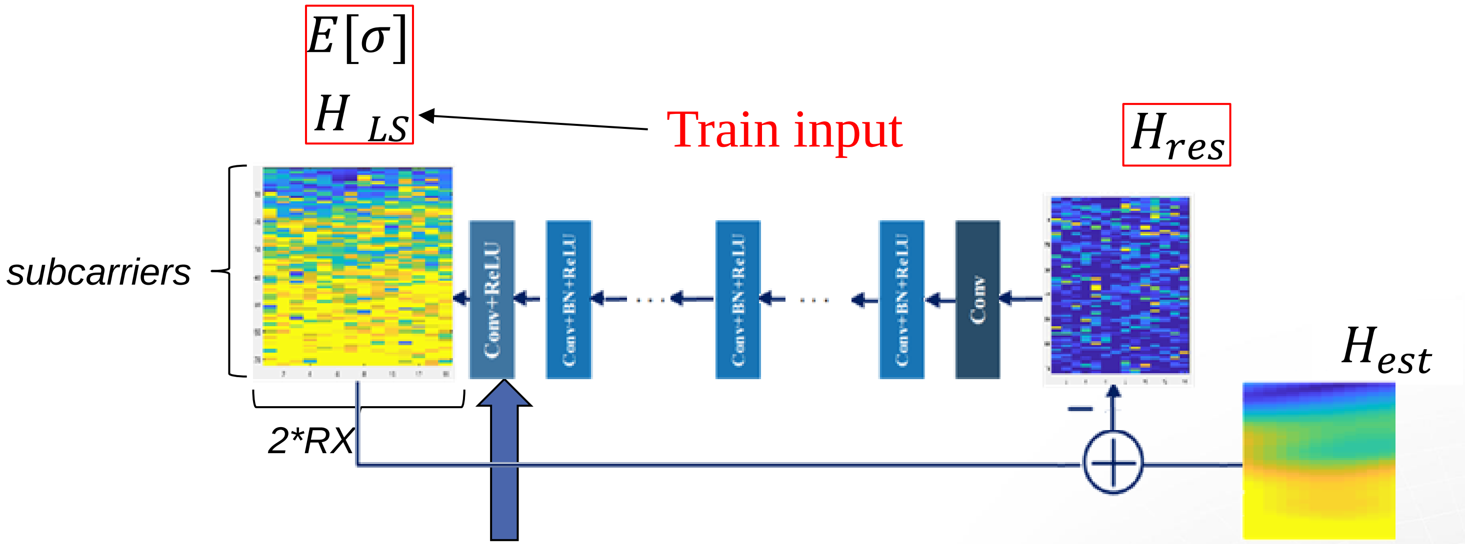
b^t – "*onsager*“ *term to decouple errors*, t- iteration,

$D_{\sigma^{t-1}}$- denoiser, σ^t-standard deviation of effective noise,

$divD(x) = \sum_{i=1}^{m} \frac{\partial D(x)}{\partial x_i}$ – divergence is the sum of the partial derivatives with respect to each element of x

Metzler C.A., Mousavi A, Richard Baraniuk. "Learned D-AMP: Principled Neural Network based Compressive Image Recovery." Advances in Neural Information Processing Systems. 2017. “From denoising to compressed sensing”.

What if we don't have H_{ideal}



One denoising stack contain 3 real value layers:
conv2d (Conv2D), Batch_normalization, activation_ (ReLU)

How to get $H_{res} = H_{LS} - H_{ideal}$;

H_{res}-residual noise, H_{ideal}- ideal channel est

H_{LS}- LS channel estimation

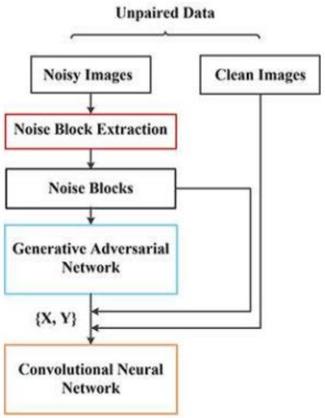
H_{est}-dnCNN denoised channel estimation

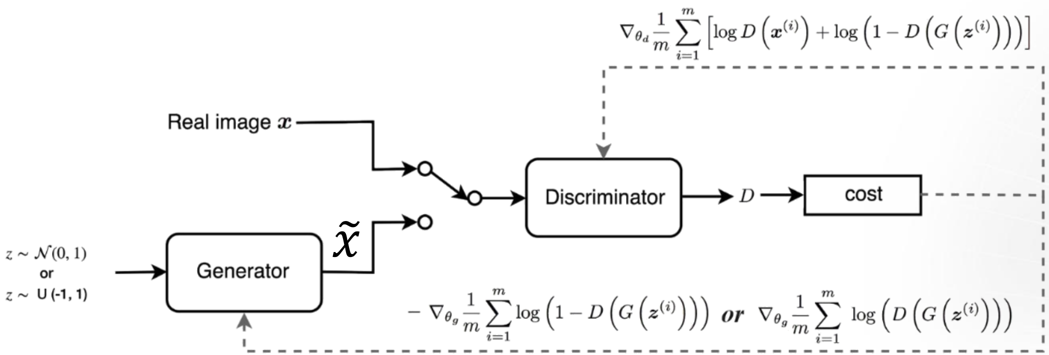
What if we don't have H_{ideal}

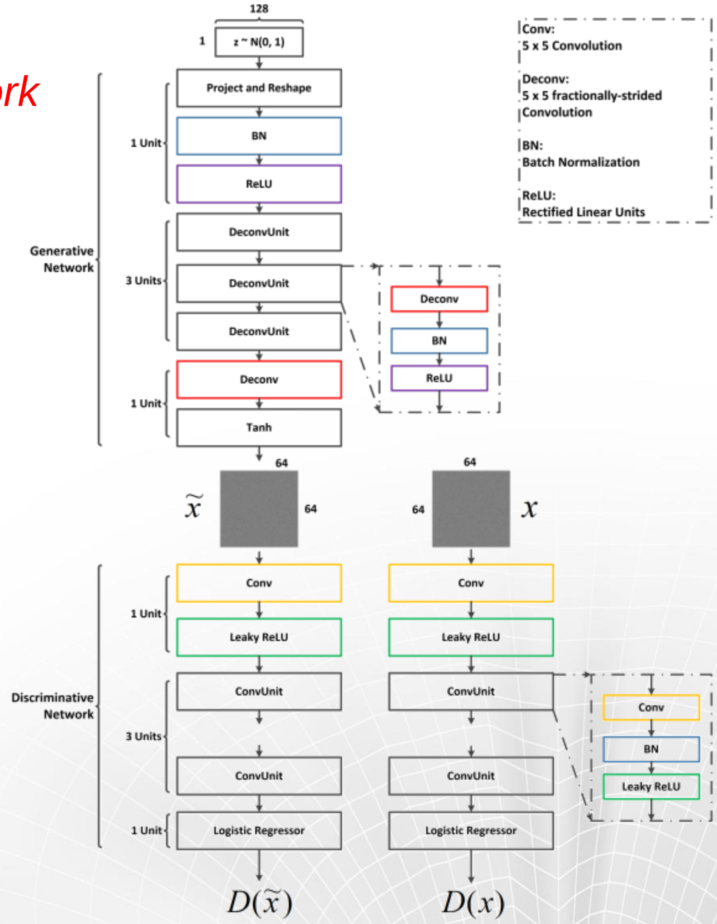
Fortunately Human math skills stronger than NN

Bayesian Nonparametric models(BNM): Variational Auto-Encoders, Generative adversarial network

$E[\sigma]$ is available.
BNM can construct paired training data from the given noisy channel H_{LS}.
We train GAN to generate H_{res}.
After use GAN's label data to train dnCNN.





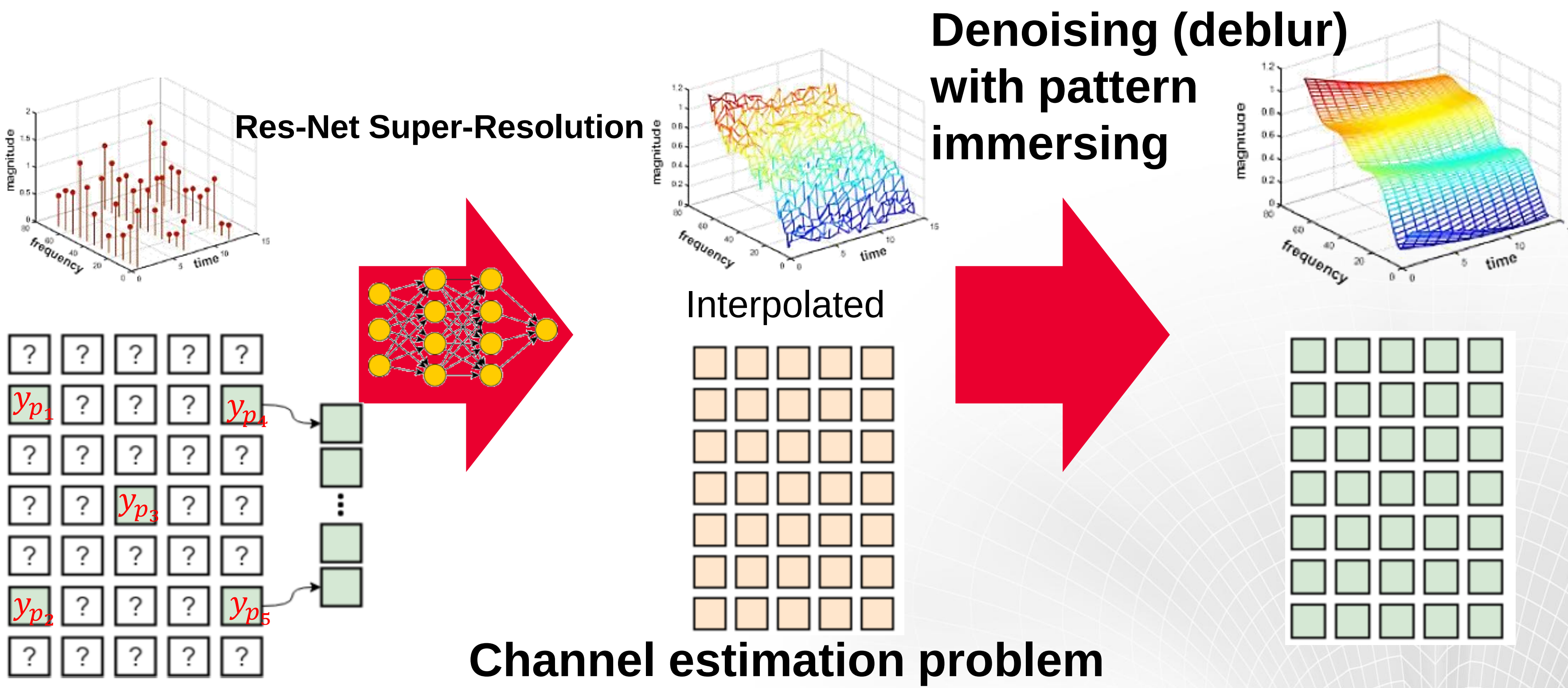


$\tilde{x}$ noise image generated by generative network
x noise image extracted from noisy image

J. Chen, H. Chao and M. Yang, "Image Blind Denoising with Generative Adversarial Network Based Noise Modeling," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 3155-3164.

Sounding resource not enough this is why we doesn't know full y_p

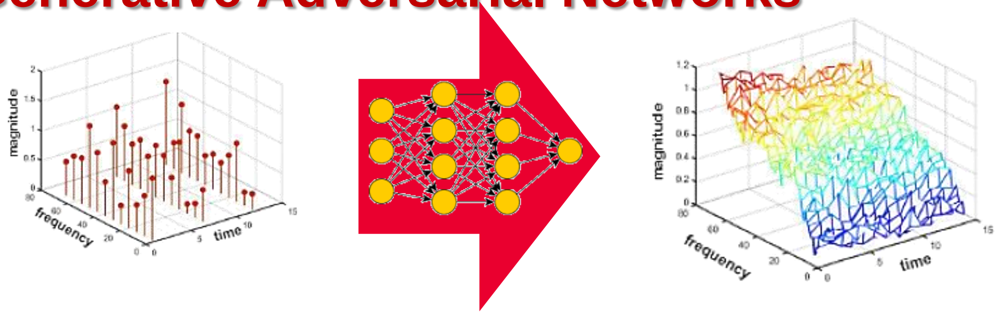
$$y_p = Hx_p + n$$



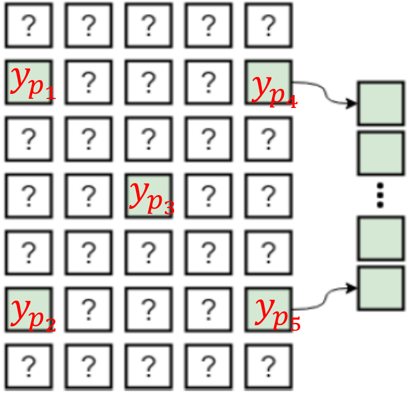
Channel estimation problem

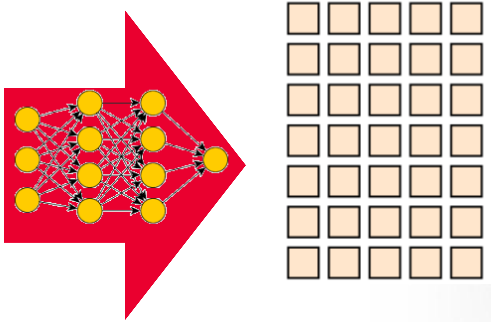
Make unsupervised channel matrix estimation using Enhanced Super-Resolution(SR) Generative Adversarial Networks

ESRGAN paying **attention** to the EM-wave **quantization pattern (texture)**



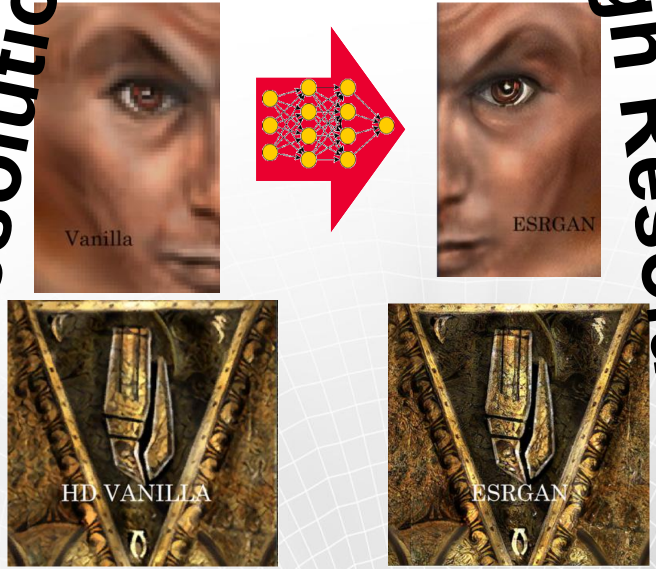
Super-Resolution alg.





Pilots y_p

Low Resolution



High Resolution

Open Problems of Deep Neural Network Channel Estimation

1. **DNN too huge. E.x.** monstrous **transformer.Symmetry phase** solutions like **Bayesian approach-Replace one horse by ensemble of chickens!? “Seymour Cray”**

2. VAE

Bayesian inference

Physical(DiffEq)/IT* inspired curvature score-GAN!

ESR/Cycle/W GAN

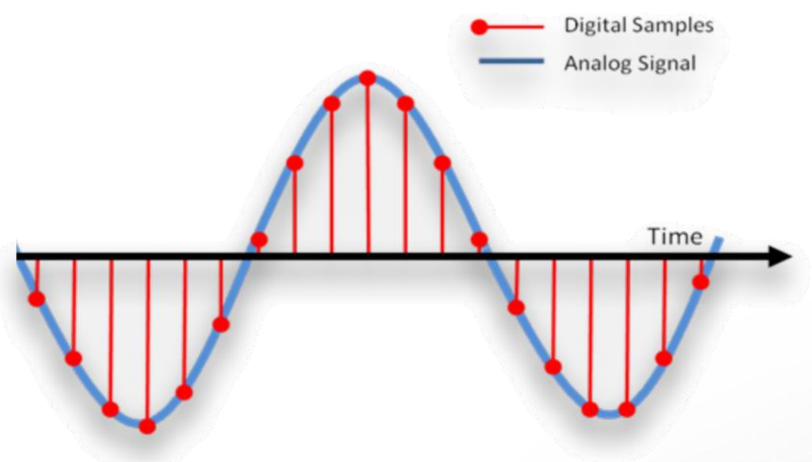
first steps in such direction

pure GAN

Local Nash Equilibrium C/D net/s and G nets.

3. New Reinforcement Learning “beyond Bellman’s equation”, independence characteristic of Markov chains is inherent only in hyperbolic space(sparse or approx. sparse like LDPC).

*Information Theory, Shannon theory look like simplest example



Sub-Nyquist Sampling

hyperbolic space – expander (LDPC Sparse dictionary)

way from binary to continue value Hamiltonian symmetry phase

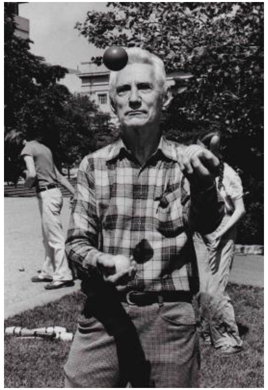
 Pawart S. and Ramchandran K. A robust sub-linear time R-FFAST algorithm for computing a sparse DFT

Whittaker-Nyquist-Kotelnikov-Shannon Sampling Theorem

Whittaker 1915

Nyquist 1928

Kotelnikov 1933



Shannon1949

Communication in the Presence of Noise

CLAUDE E. SHANNON, MEMBER, IRE

Theorem 1: If a function $f(t)$ contains no frequencies higher than W cps, it is completely determined by giving its ordinates at a series of points spaced $1/2\,W$ seconds apart.

pointwise sampling!

...

Mathematically, this process can be described as follows. Let x_n be the nth sample. Then the function $f(t)$ is represented by

$$\boxed{f(t) = \sum_{n=-\infty}^{\infty} x_n \frac{\sin \pi(2Wt - n)}{\pi(2Wt - n)}.} \qquad (7)$$

interpolation

Shannon, Claude E. (January 1949). "Communication in the presence of noise". Proceedings of the Institute of Radio Engineers. ***37*** *(1): 10–21.*

Aliasing

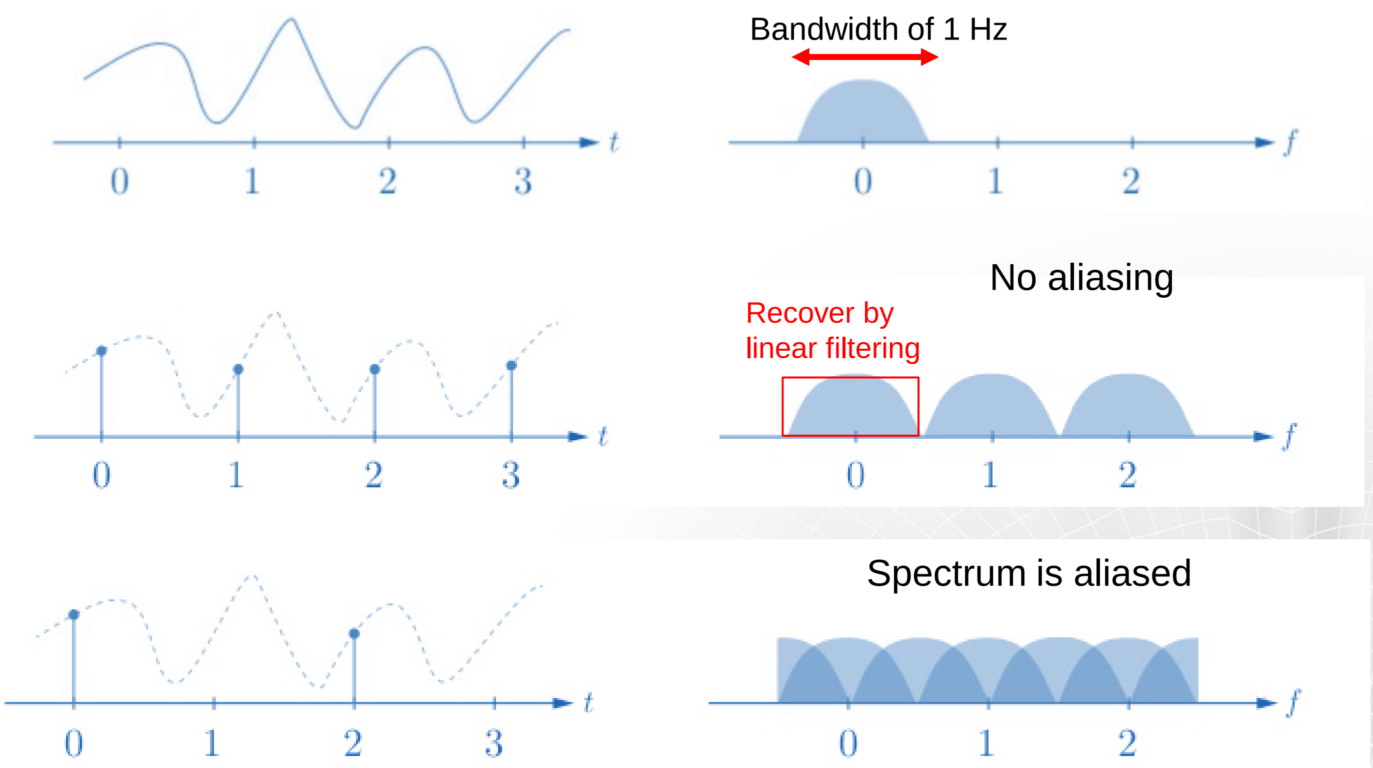
Time domain

Frequency domain

Input signal

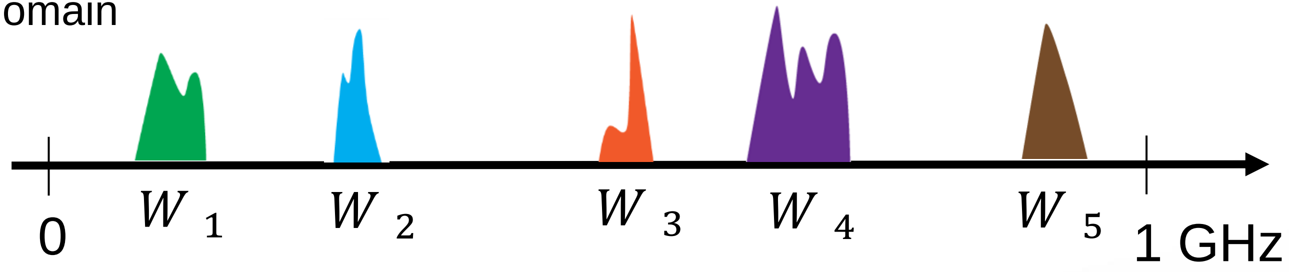
Sampling at rate 1

Sampling at rate 1/2



What if the spectrum is sparsely occupied?

Frequency domain



$$f_{occ} = \sum_{i=1}^{5} W_i = 100 \text{ MHz}$$

$$A \int_{\mathbb{R}^d} |f(x)|^2\, dx \leqslant \sum_{\lambda \in \Lambda} |f(\lambda)|^2 \leqslant B \int_{\mathbb{R}^d} |f(x)|^2\, dx$$

for all band-limited functions $f \in L^2(\mathbb{R}^d)$ with $\operatorname{supp} \hat{f} \subseteq \Omega \subseteq \mathbb{R}^d$ implies a lower estimate for the density of $\Lambda \subseteq \mathbb{R}^d$ by the measure of the spectrum Ω. Dually, if the interpolation problem $f(\lambda_n) = a_n$ for $n \in \mathbb{Z}$, has a band-limited solution $f \in L^2(\mathbb{R}^d)$, $\operatorname{supp} \hat{f} \subseteq \Omega$, for all sequences $(a_n) \in l^2$, then the density of Λ is bounded above the measure of Ω. These theorems provided a solution to a conjecture of A. Beurling concerning the balayage of measures **[2, 3]**, but they are now commonly seen as a precise mathematical formulation of the Nyquist density **[11]**.

Landau, 1967 Average sampling rate (uniform or otherwise) must be twice the *occupied* bandwidth of the signal, assuming *frequency support* known

Frequency support - a priori known what portion of the spectrum was occupied

H. J. Landau, “Necessary density conditions for sampling and interpolation of certain entire functions,” Bell Lab. Acta Math., vol. 117, pp. 37–52, Feb. 1967.

Mathematical description: Learning polynomials problem

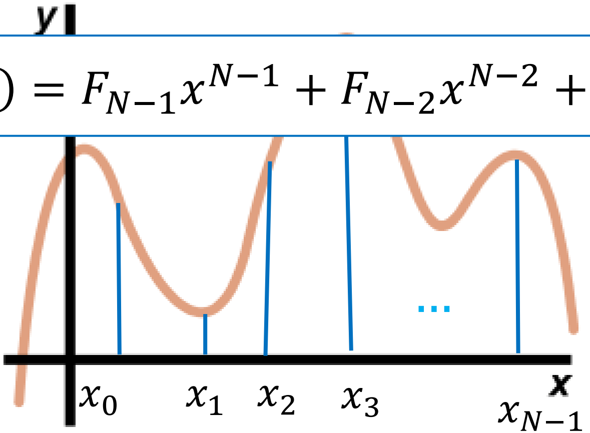
$$Given\ f(x) = \sum_{n=0}^{N-1} F_n x^n$$

$$Find\ coefficients\ \{F_n\}_{n=0}^{N-1}$$

N coefficient from which K is non-zero

K sublinear in N $(K/N \to 0)$

$$f(x) = F_{N-1}x^{N-1} + F_{N-2}x^{N-2} + \cdots + F_0$$



$$\begin{pmatrix} f(x_0) \\ f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-1}) \end{pmatrix} = \begin{pmatrix} 1 & x_0 & \cdots & x_0^{N-1} \\ 1 & x_0 & \cdots & x_1^{N-1} \\ 1 & x_0 & \cdots & x_3^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{19} & \cdots & x_{N-1}^{N-1} \end{pmatrix} \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{pmatrix}$$

One of solution Schumacher Fast Sparse ***FFT***

$$x = e^{i\frac{2\pi}{N}m}$$

Sample complexity: K

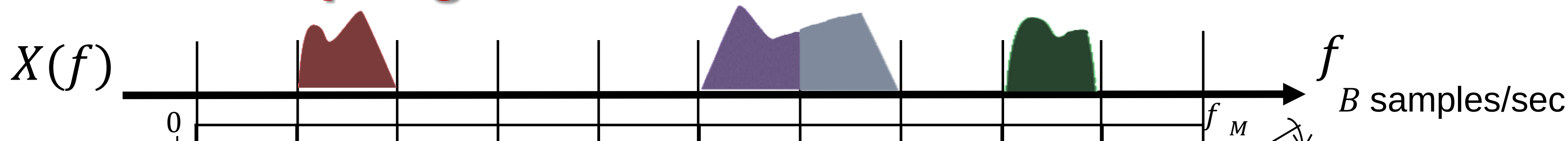
Computation cost: $O(K \log N)$

We not know accurate location of K nonzero coefficient, noise free

J. Schumacher and M. Püschel, "High-performance sparse fast Fourier transforms," *2014 IEEE Workshop on Signal Processing Systems (SiPS)*, Belfast, 2014, pp. 1-6.

Filter bank for Sampling

N Bands



m channels/filters
N bands

channels

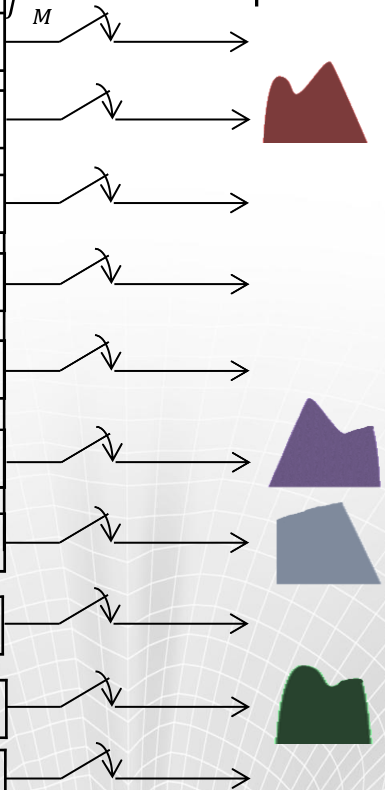
$x(t)$

Overall
Sampling rate

$$f_M = N \frac{f_M}{N}$$

f_M Nyquist rate for $x(t)$

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1



Spectrum-blind sampling

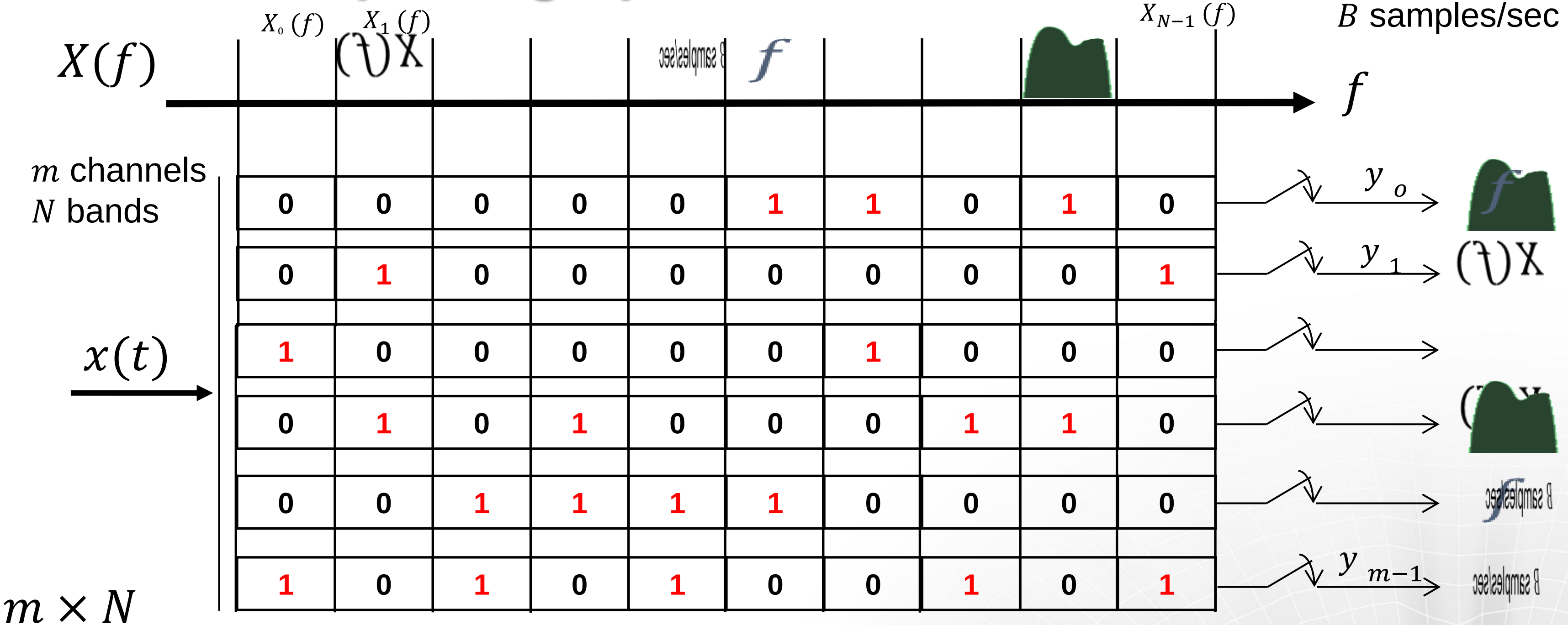
Any bandlimited signal $x(t) \in \mathbb{C}$ whose spectrum has occupancy f_{occ} can be sampled asymptotically at rate $f_s = 2f_{occ}$ by a randomized "*sparse-graph-coded filter bank*" with probability 1 using $O(f_{occ})$ operations per unit time.

Computational cost $O(f_{occ})$ independent of bandwidth

Can be made robust to sampling noise

O. Öçal, X. Li and K. Ramchandran, "A sparse-graph-coded filter bank approach to minimum-rate spectrum-blind sampling," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, 2016, pp. 6365-6369.

Sparse graph coded filter bank bands



0	0	0	0	0	1	1	0	1	0
0	1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	1	0	0	0
0	1	0	1	0	0	0	1	1	0
0	0	1	1	1	1	0	0	0	0
1	0	1	0	1	0	0	1	0	1

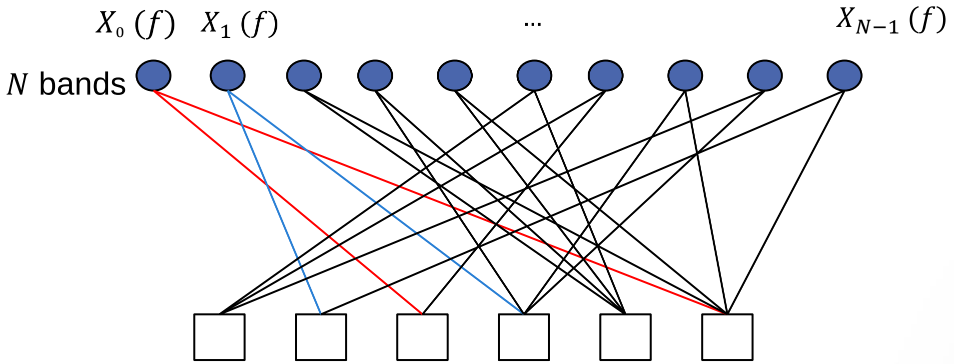
$$Y\left(e^{j2\pi f}\right) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} X(Bf), \quad where\ X(f) = \begin{pmatrix} X_0(f) \\ \vdots \\ X_{N-1}(f) \end{pmatrix}$$

Factor Graph representation of sparse graph coded filter bank

$$Y\left(e^{j2\pi f}\right) = \begin{array}{cc} & \begin{array}{cccccccccc} X_0(f) & X_1(f) & & & & & & & & \end{array} \\ m = 6 \text{ channels} & \underbrace{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}}_{N = 10 \text{ bands}} \end{array} X(Bf)$$

Parity-check matrix

Variable nodes



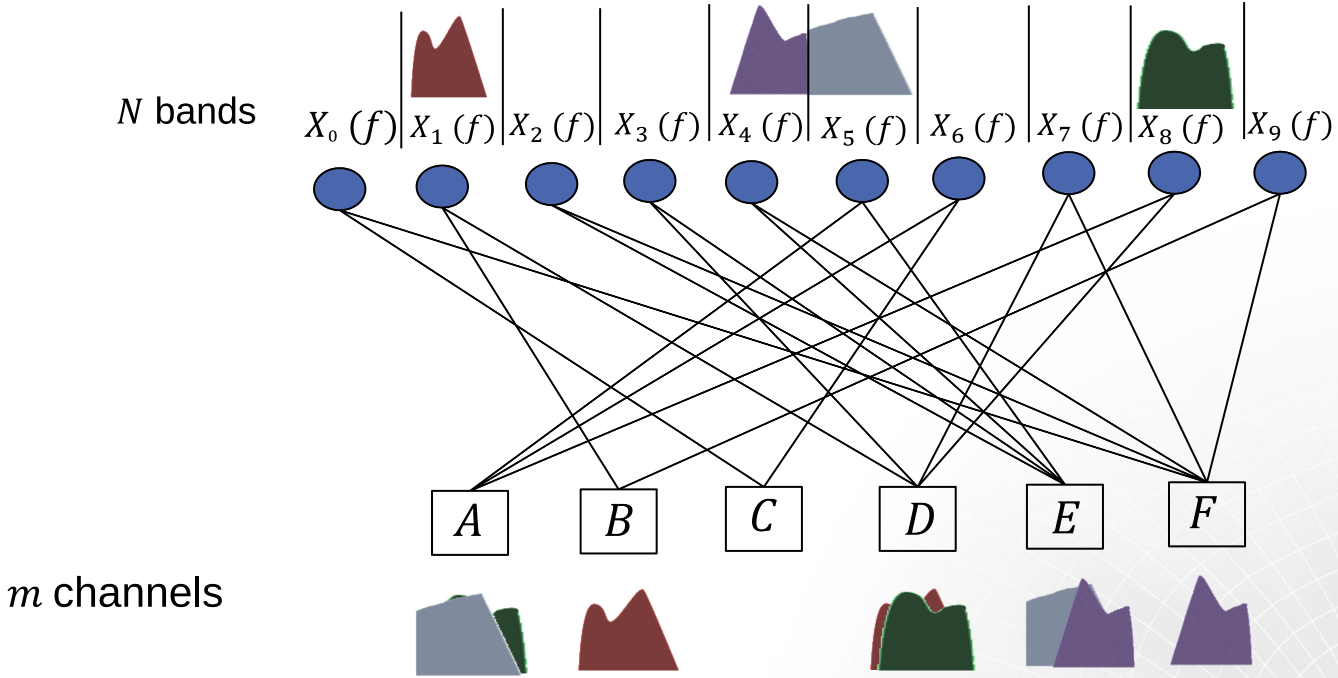
m channels (factors)

Check nodes

Bipartite Factor Graph (Tanner Graph)

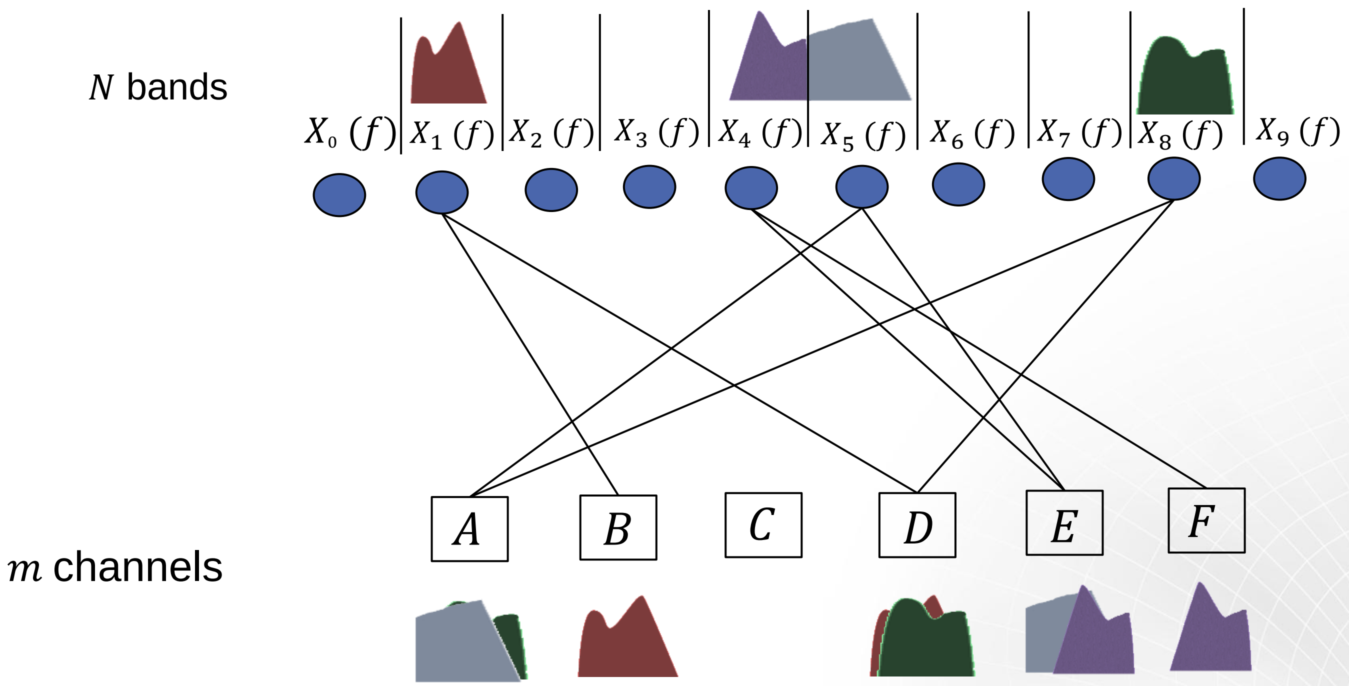
R. Tanner, "A recursive approach to low complexity codes," in *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533-547, September 1981.

Factor Graph representation of sparse graph coded filter bank

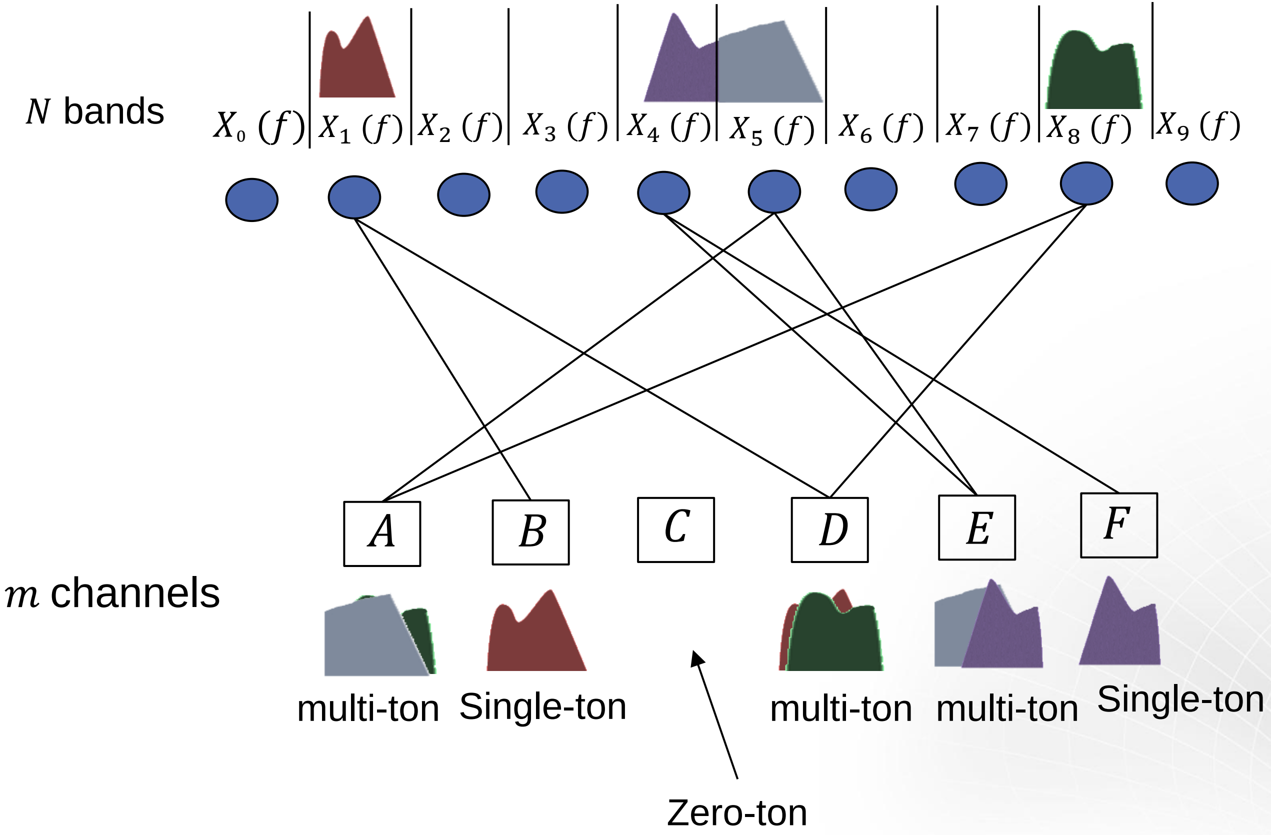


Decode Sparse graph coded filter bank to resolve aliasing

First remove edges which connect to non-active band according current sample

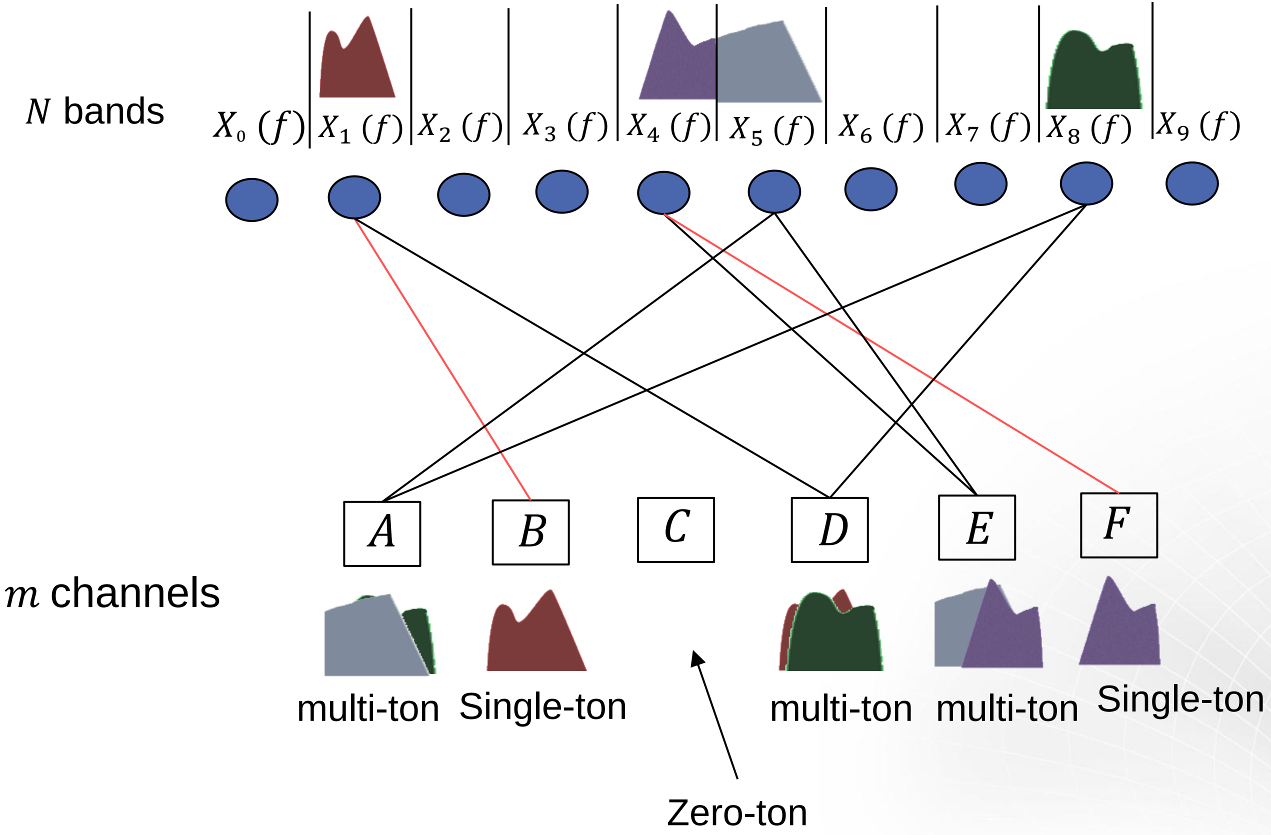


Peeling Decoder of Sparse graph coded filter bank



Zero-ton: no signal
Single-ton: no aliasing
Multi-ton: aliasing

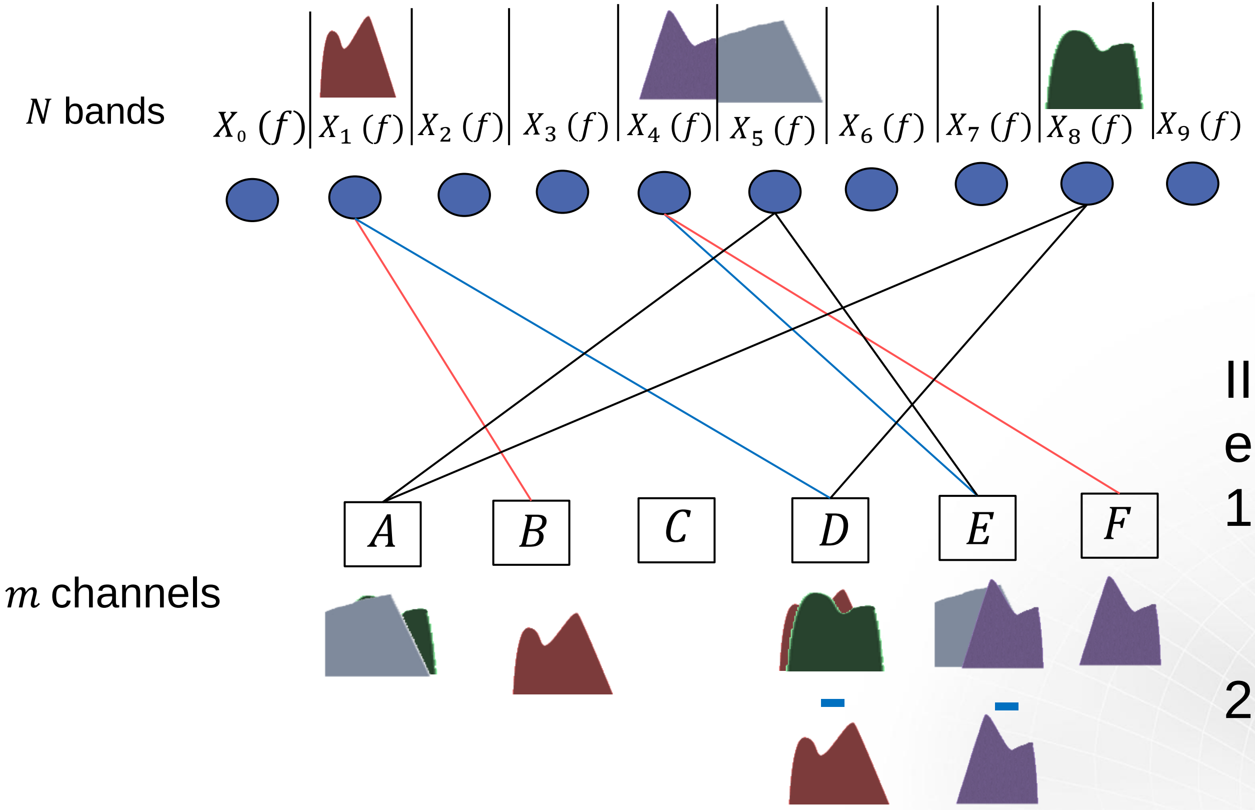
Peeling Decoder example



I.identifies which channels have no aliasing and maps them to which bands they came from:

1. Channel $B = X_1$
2. Channel $F = X_4$

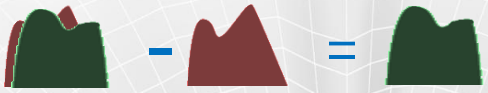
Peeling Decoder example



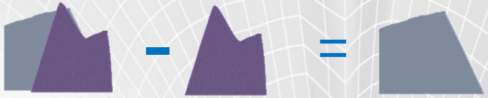
1. Channel $B = X_1$
2. Channel $F = X_4$

II.Peeling “-” single-ton from edge connected multi-ton:

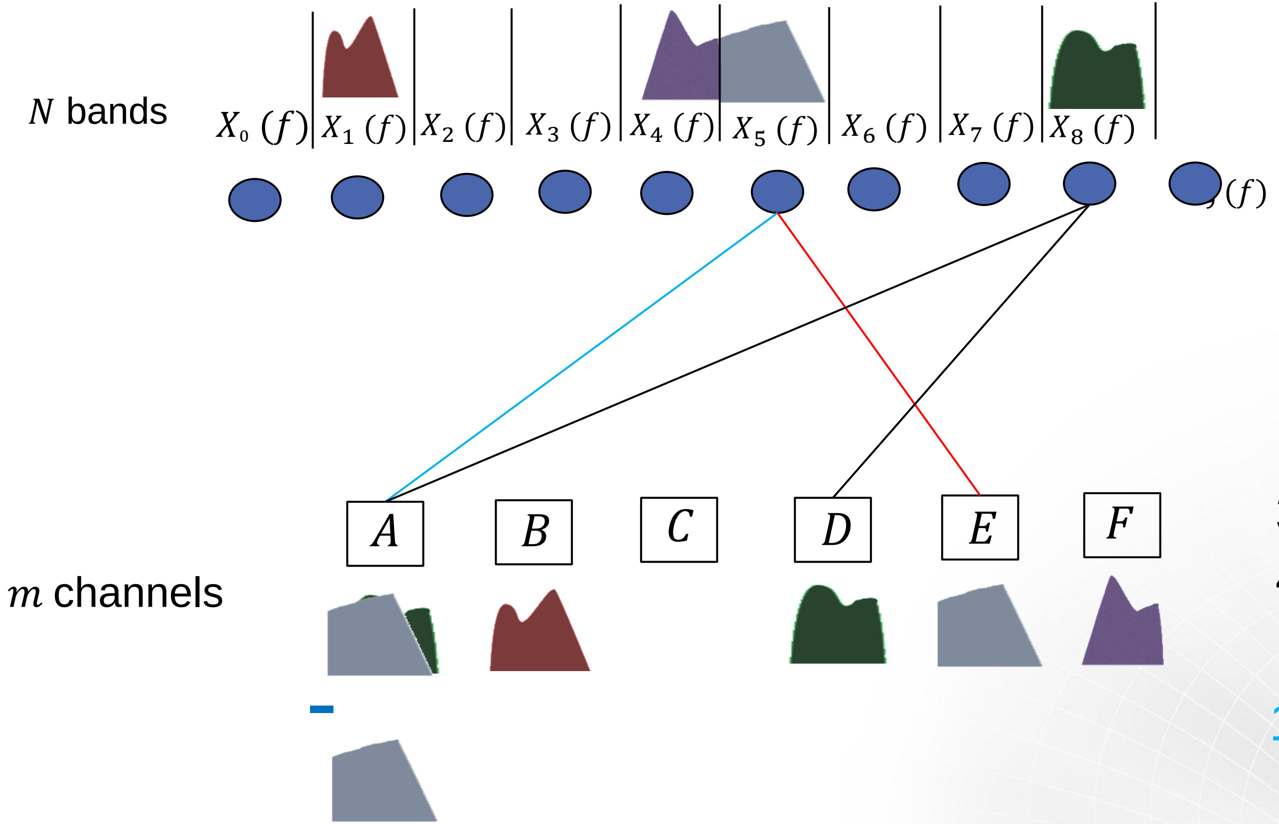
1.Channel D – Channel B



2. Channel E – Channel F



Peeling Decoder example



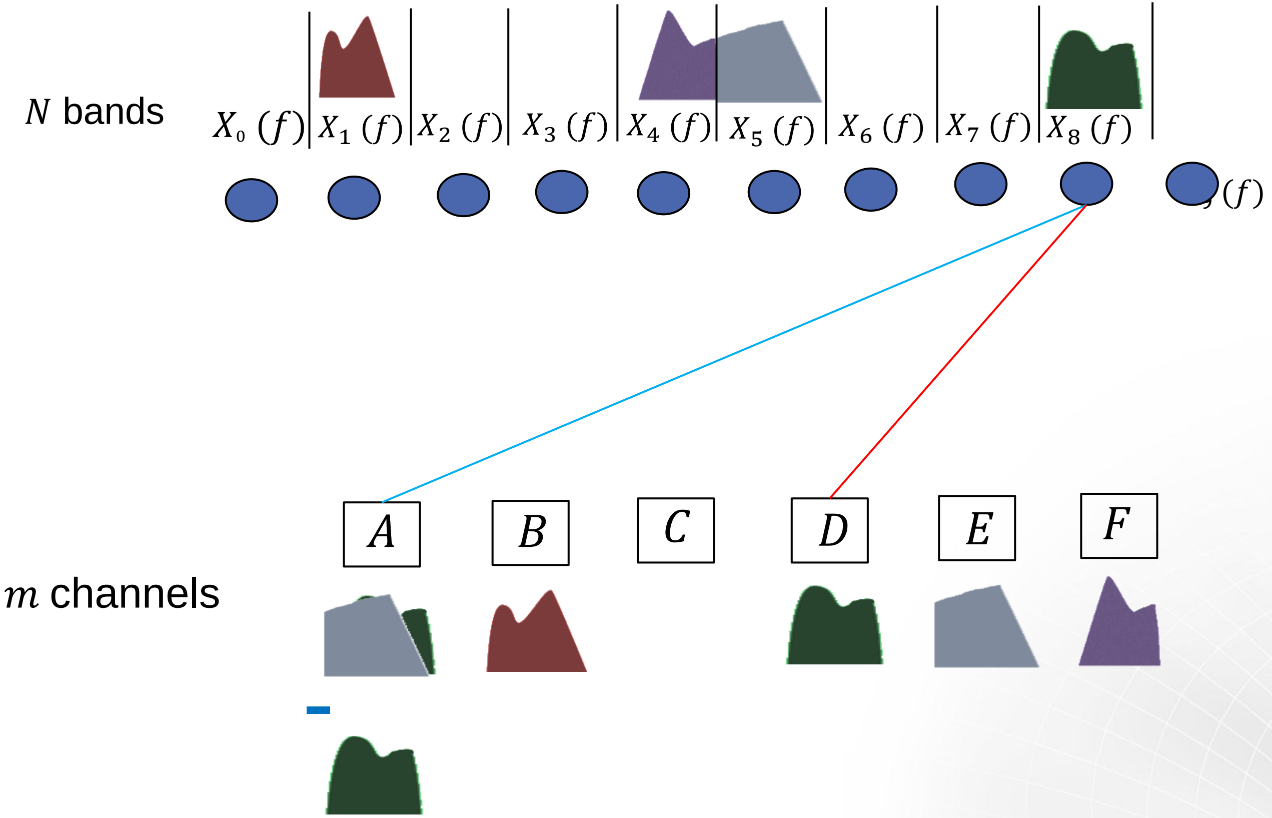
1. Channel B = X_1
2. Channel F =X_4

identifies which channels have no aliasing and maps them to which bands they came from

3. Channel D = X_8
4. Channel E = X_5

1. Channel A – Channel E

Peeling Decoder example



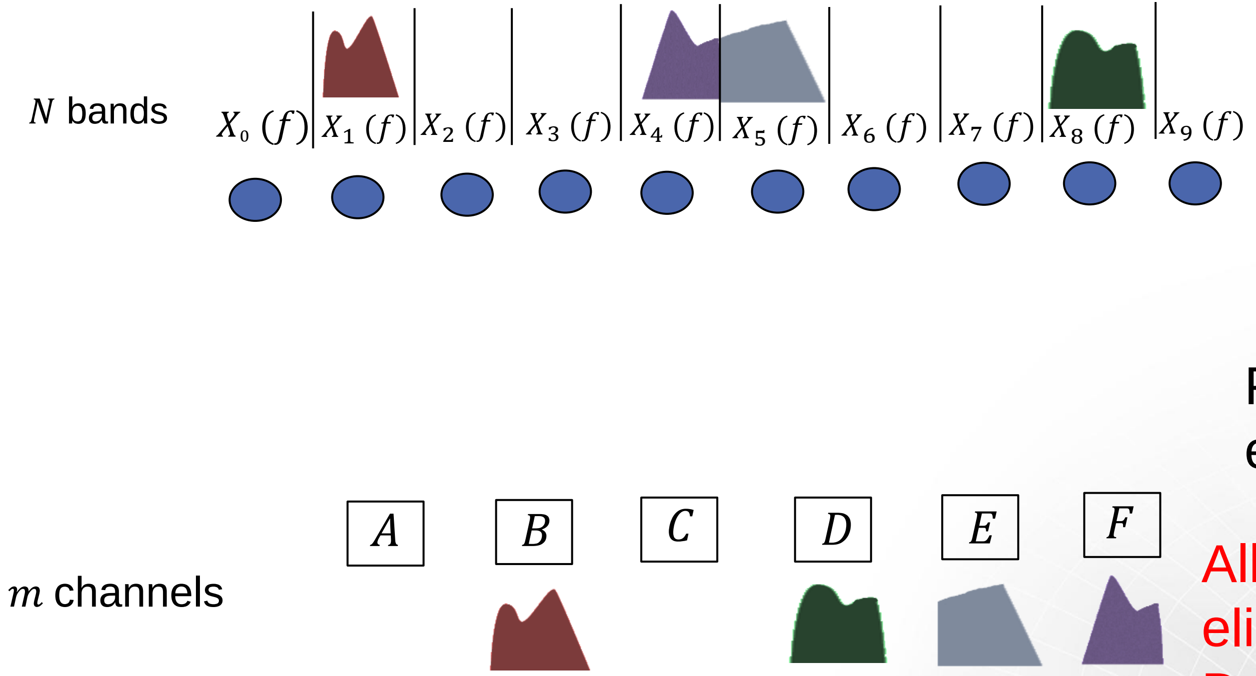
1. Channel B = X_1
2. Channel F = X_4

identifies which channels have no aliasing and maps them to which bands they came from

3. Channel D = X_8
4. Channel E = X_5

1. Channel A – Channel E
2. Channel A – Channel D

Peeling Decoder example



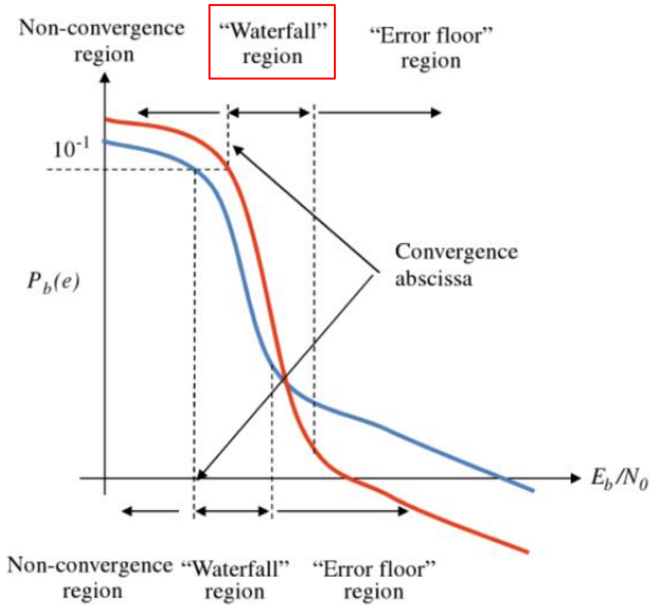
1. Channel B = X_1
2. Channel F =X_4
3. Channel D = X_8
4. Channel E = X_5

Peel "-" single-ton from edge connected multi-ton

All 1 degree nodes eliminated, all edges peeled
Decoding Finished
Aliasing Resolve

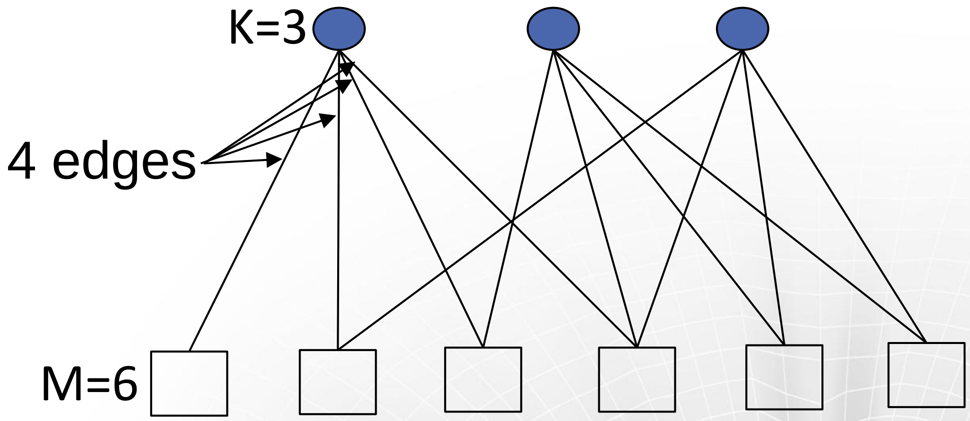
How to construct Sparse graph coded filter bank?

Density evolution: Asymptotically optimal($N \to \infty$, H random, without Automorphism): Construct filter bank codes with best waterfall.



Example

Regular LDPC (K,M) with $\deg(\lambda) = 3$,



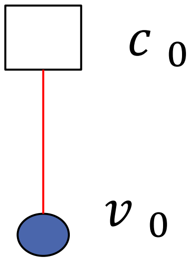
Every Variable node connected to $\deg(\lambda)$+1 factor(check) nodes

λ=0,1,2,3

$|\lambda|$=4

How to construct Sparse graph coded filter bank?

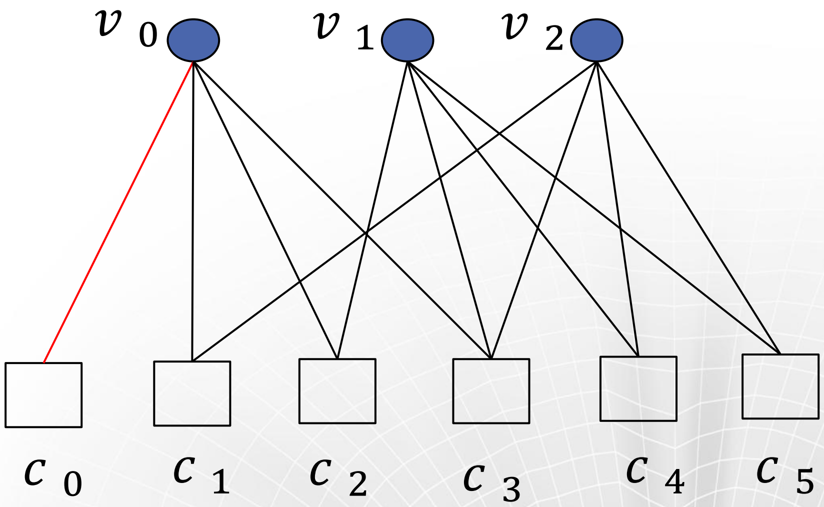
Asymptotically optimal: Density evolution



Pick an arbitrary edge in the graph (c, v)

Example

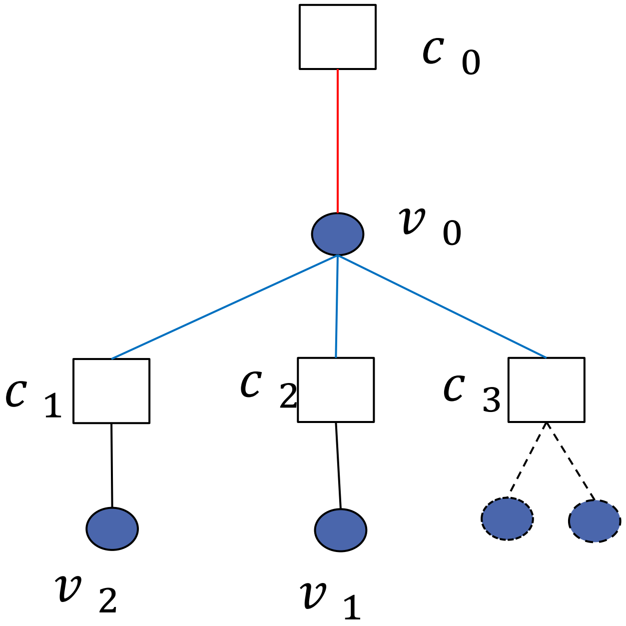
Regular LDPC (3,6) with $\deg(\lambda) = 3$,



every variable node (v_i) connected to $\deg(\lambda)$+1 factor(check) nodes (c_i)

How to construct Sparse graph coded filter bank?

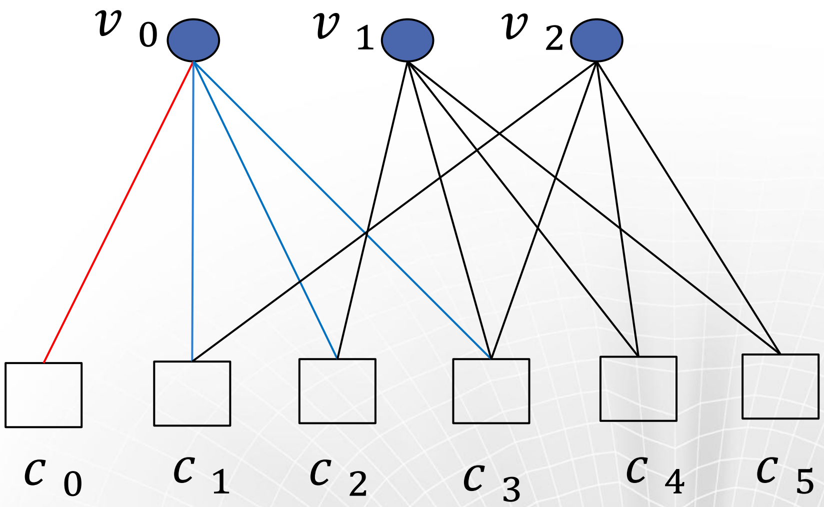
Asymptotically optimal: Density evolution



Examine its directed neighborhood at depth-2ℓ

Example

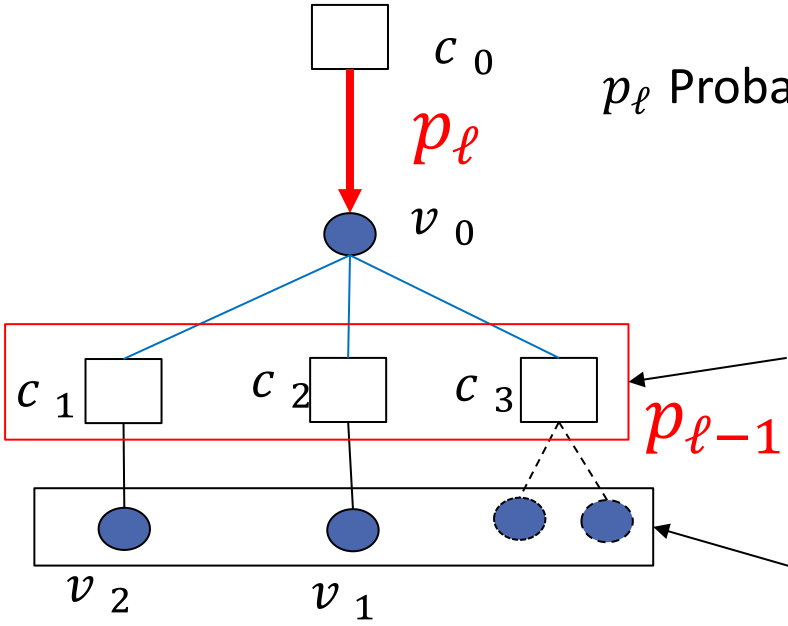
Regular LDPC (3,6) with $\deg(\lambda) = 3$,



every variable node (v_i) connected to $\deg(\lambda)$+1 factor(check) nodes (c_i)

How to construct Sparse graph coded filter bank?

Asymptotically optimal: Density evolution



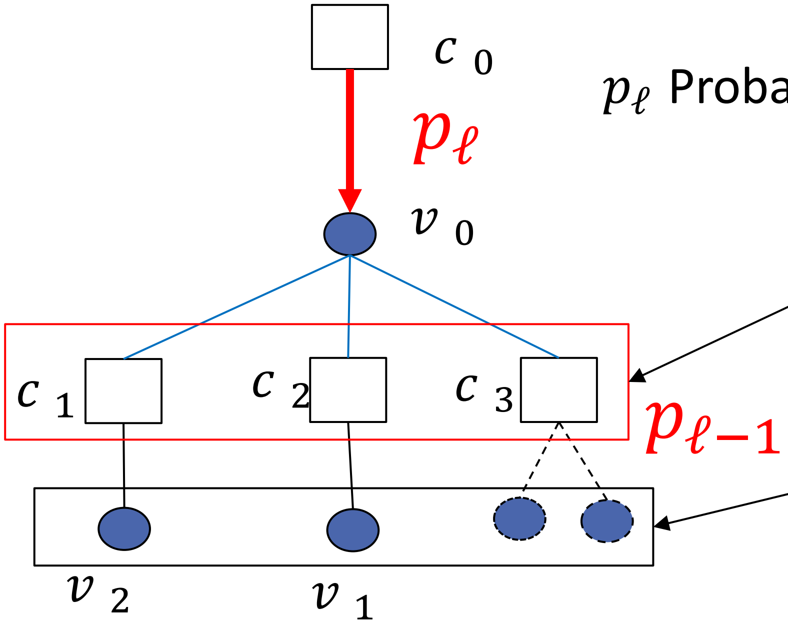
p_ℓ Probability of being present at depth 2ℓ

The variable v_0 can be resolved if any of these check nodes can be resolved

A check nodes is resolved from below if all of the variable nodes connected to it from below are resolved

How to construct Sparse graph coded filter bank?

Asymptotically optimal: Density evolution



p_ℓ Probability of being present at depth 2ℓ

The variable v_0 can be resolved if any of these check nodes can be resolved

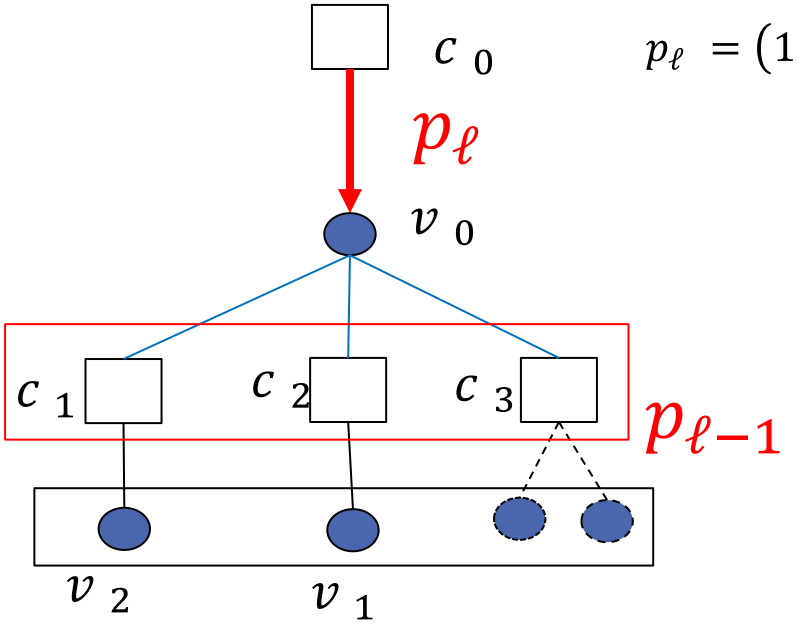
A check nodes is resolved from below if all of the variable nodes connected to it from below are resolved

Power 2 is because the check node c_3 has 2 variable nodes as children

$$p_\ell = \left(1-(1-p_{\ell-1})^1\right) * \left(1-(1-p_{\ell-1})^1\right) * \left(1-(1-p_{\ell-1})^2\right)$$

How to construct Sparse graph coded filter bank?

Asymptotically optimal: Density evolution



$$p_\ell = \left(1-(1-p_{\ell-1})^{\boxed{1}}\right) * \left(1-(1-p_{\ell-1})^{\boxed{1}}\right) * \left(1-(1-p_{\ell-1})^{\boxed{2}}\right)$$

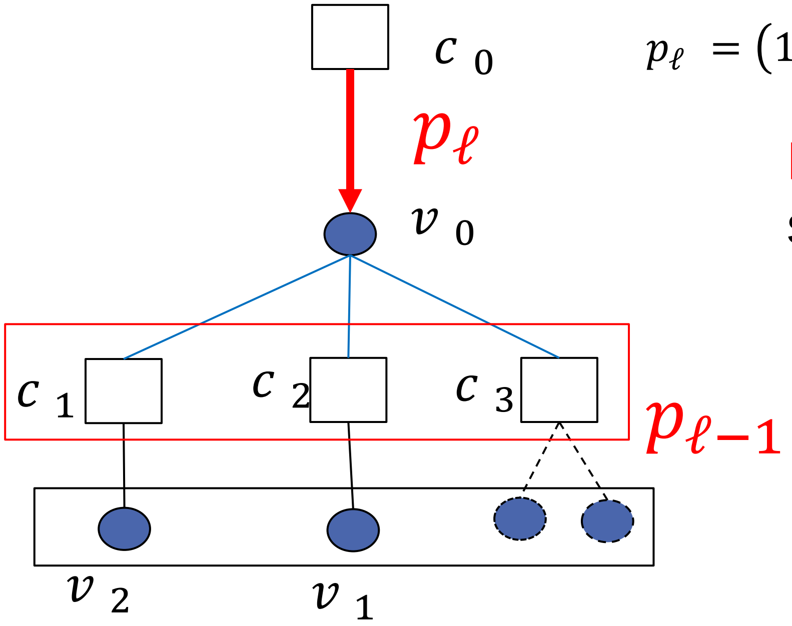
Number of children of check nodes has Poisson distribution with mean $\frac{K(\deg(\lambda)+1)}{M}$

Regular(K, M) graph constructed such: every variable node (v_i) connected to $\deg(\lambda)$+1 factor(check) nodes (c_i) chosen uniformly at random

$$p_\ell(c_i) = e^{-\frac{K(\deg(\lambda)+1)}{M} p_{\ell-1}}$$

How to construct Sparse graph coded filter bank?

Asymptotically optimal: Density evolution

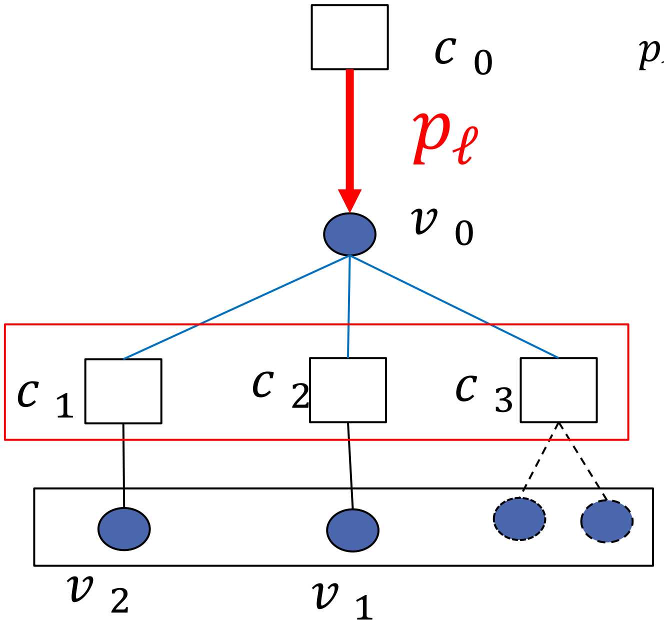


$$p_\ell = \left(1-(1-p_{\ell-1})^{1}\right) * \left(1-(1-p_{\ell-1})^{1}\right) * \left(1-(1-p_{\ell-1})^{2}\right)$$

Let $\ell \to \infty$, we want to choice $K, M, \deg(\lambda)$ such that $p_\ell \to 0$

$$p_\ell(c_i) = e^{-\frac{K(\deg(\lambda)+1)}{M} p_{\ell-1}}$$

How to construct Sparse graph coded filter bank?



$$p_\ell = \left(1-(1-p_{\ell-1})^1\right) * \left(1-(1-p_{\ell-1})^1\right) * \left(1-(1-p_{\ell-1})^2\right)$$

Let $\ell \to \infty$, we want to choice $K, M, \deg(\lambda)$ such that $p_\ell \to 0$

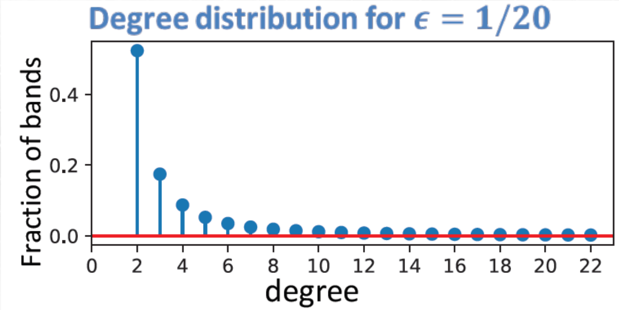
$$p_\ell(c_i) = e^{-\frac{K(\deg(\lambda)+1)}{M} p_{\ell-1}}$$

Asymptotically optimal solution:

$M=(1+\epsilon)K$, $D = \deg(\lambda)+1>1/\epsilon$

Variable Node distribution $P(j) = \frac{D+1}{D}\frac{1}{j(j-1)}, j = 2, \dots, D+1$

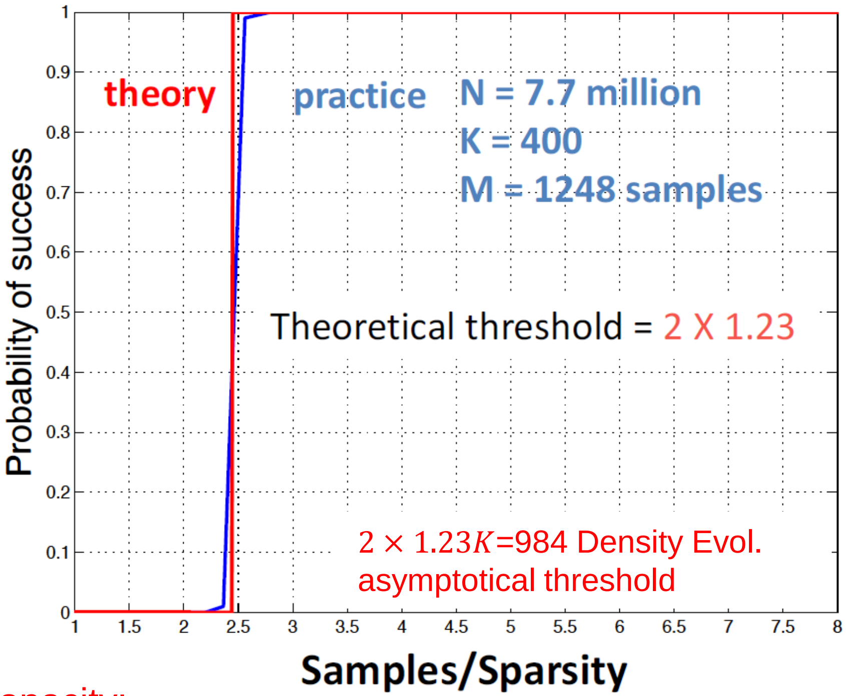
$$p_\ell = \frac{1}{H(D)} \sum_{j=2}^{D+1} \frac{1}{j-1}\left(1 - e^{-\frac{d}{1+\epsilon} p_{\ell-1}}\right)$$

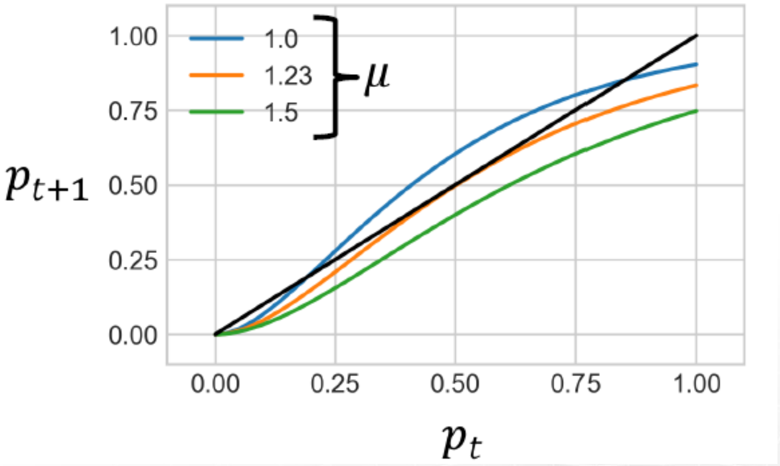


Compute asymptotical efficient sampling

M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," in *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 585-598, Feb 2001. First solution Soliton degree distribution

How to construct Sparse graph coded filter bank?





Fraction of non-zero coefficients not recovered at time t

Capacity:
M=2×K
=800

N signal length,

K- non zero coefficient

$$p_{t+1}(c_i) = \left(1 - e^{-\frac{3p_t}{\mu}}\right)^2$$

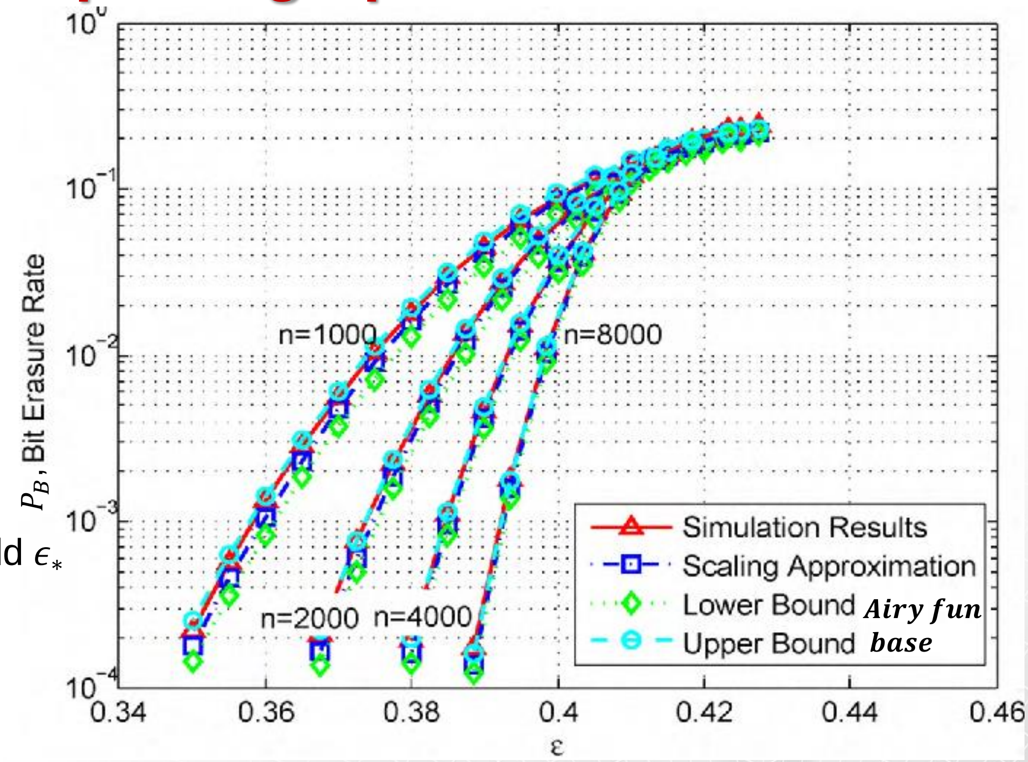
X. Li, S. Pawar and K. Ramchandran, "Sub-linear time compressed sensing using sparse-graph codes," *2015 IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, 2015, pp. 1645-1649.

How to construct finite-length Sparse graph coded filter bank?

$$\mathrm{E}[P_B] = Q\left(\frac{\sqrt{N}(\epsilon_* - \epsilon)}{\alpha}\right) + o(1)$$

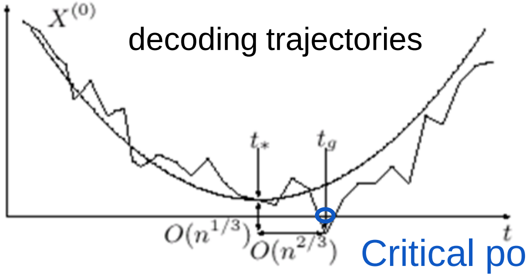
More accurate if add shift β

$$\mathrm{E}[P_B] = Q\left(\frac{\sqrt{N}\left(\epsilon_* - \beta N^{-\frac{2}{3}} - \epsilon\right)}{\alpha}\right) + O\left(N^{-\frac{1}{3}}\right)$$



N – *length of code*
ϵ_*-Density evolution threshold
ϵ-erasure probability (BEC)

α – *scale*, β – *shift* from Density Evolution threshold ϵ_* estimated by Covariance Evolution



Critical point

Two side Brownian motion with parabolic drift of remained ratio of degree one check nodes at Peeling decoder around solution of Density Evolution (DE). Several critical points may exist

A. H. Djahanshahi, P. H. Siegel and L. B. Milstein, "On unequal error protection of finite-length LDPC codes over BECs: A scaling approach," *2009 IEEE International Symposium on Information Theory*, Seoul, 2009, pp. 1824-1828.

How to construct Finite-length Sparse graph coded filter bank?

Covariance Evolution

degree 1 check nodes

$c_1(\tau) = E\,[r_0(\tau)]$ $\delta_1(\tau) = Var[c_1(\tau)]$

VN/CNs at time τ with the trajectory converging to differential equations:

$$\frac{\partial \hat{v}_d\,(\tau)}{\partial \tau} = f(\triangle \mathrm{V}_d(\tau))$$
$$= E[\triangle\, \mathrm{V}_d(\tau) | \{\hat{v}_d(\tau), \hat{r}_c(\tau)\}_{d\in\mathcal{F}_v, c\in\bar{\mathcal{F}}_c}]$$
$$\frac{\partial \hat{r}_c\,(\tau)}{\partial \tau} = f(\triangle \mathrm{R}_c(\tau))$$
$$= E[\triangle\, \mathrm{R}_c(\tau) | \{\hat{v}_d(\tau), \hat{r}_c(\tau)\}_{d\in\mathcal{F}_v, c\in\bar{\mathcal{F}}_c}]$$

where

$$\triangle\, \mathrm{V}_d(\tau) = \mathrm{V}_d\left(\tau + \tfrac{1}{N}\right) - \mathrm{V}_d(\tau)$$
$$\triangle\, \mathrm{R}_c(\tau) = \mathrm{R}_c\left(\tau + \tfrac{1}{N}\right) - \mathrm{R}_c(\tau)$$

Covariance of a state space variables follow:

$$\frac{\partial \delta_{c,c'}\,(\tau)}{\partial \tau} = Cov[\Delta \mathrm{R}_c, \Delta \mathrm{R}_{c'} | \mathfrak{E}(\tau)]$$
$$+\sum_{u\in\bar{\mathcal{F}}_c} \delta_{c,u}(\tau) \left.\frac{\partial f\left(\triangle \mathrm{R}_{c'}(\tau)\right)}{\partial r_u}\right| \mathfrak{E}(\tau)$$
$$+\sum_{u\in\bar{\mathcal{F}}_c} \delta_{c',u}(\tau) \left.\frac{\partial f(\triangle \mathrm{R}_c(\tau))}{\partial r_u}\right| \mathfrak{E}(\tau)$$
$$+\sum_{d\in\mathcal{F}_v} \delta_{c,d}(\tau) \left.\frac{\partial f\left(\triangle \mathrm{R}_{c'}(\tau)\right)}{\partial r_d}\right| \mathfrak{E}(\tau)$$
$$+\sum_{d\in\mathcal{F}_v} \delta_{c',u}(\tau) \left.\frac{\partial f(\triangle \mathrm{R}_c(\tau))}{\partial r_d}\right| \mathfrak{E}(\tau)$$

$$\boxed{\tau \doteq \frac{\ell}{N},\ r_c \doteq \frac{R_c(\tau)}{N}, v_d(\tau) = \frac{V_d(\tau)}{N}}$$

Rescaled to pass continuous limit for the diff. eq.

$\mathcal{F}_v(\mathcal{F}_c)$- set of variable(check) node types, $V_d(\tau)(R_c(\tau))$-number of VNs (CNs) of type c_d at time τ, ℓ- iteration of peeling decoder, N- length , $\mathfrak{E}(\tau)$-expected degree distribution at time τ: $\mathfrak{E}(\tau) = \{\hat{v}_d\,(\tau), \hat{r}_c\,(\tau)\}_{d\in\mathcal{F}_v, c\in\bar{\mathcal{F}}_c}$

A. Amraoui, A. Montanari, T. Richardson and R. Urbanke, "Finite-Length Scaling for Iteratively Decoded LDPC Ensembles," in *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 473-498, Feb. 2009.

How to Generalizing Covariance Evolution from the BEC?

- No obvious incremental form (diff. Eq.)
- No state space characterization of failure.
- No clear finite dimensional state space.
- Not clear what the right coordinates are for the general case (Capacity?).

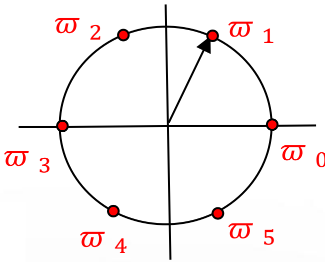
Stanford
Institut des hautes études scientifiques
Ecole polytechnique fédérale de Lausanne

For BSC Exist EXIT-chart like and GEXIT solutions.

What if we have cycles in graph (N is short) or/and Noise

Automorphism Group Decoder*. Use DFT shift to form cycle code and differ value subsample. Create new graphs of same signal allow to broke aliasing (correlation, cycles).

DFT form cycle code, under multiplied, e.x. $\omega = e^{i\frac{2\pi}{20}}$.
Using cyclic shift by ϖ together with DFT different size (overlap signal).

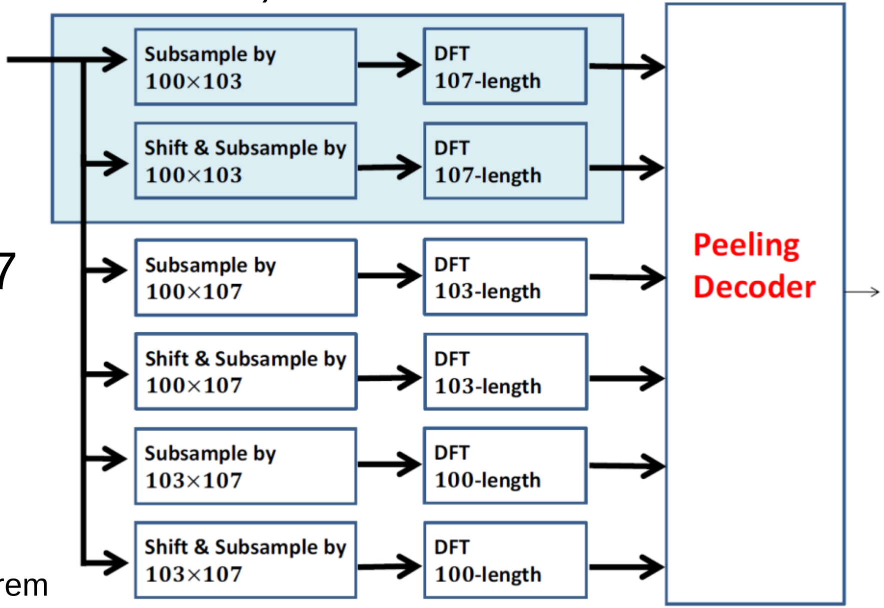


FFAST** (FAST Fourier Aliasing-based Sparse Transform)

Noiseless: For **K** sublinear in **N**
4K samples, $\boldsymbol{O(K\log K)}$
Example:

Signal length N=100×103×107

K≈ 200 not zero coefficients
M ≈616 samples require



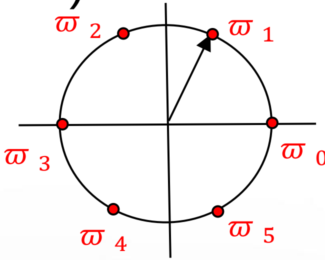
uniform filter-les sub-sampling
patterns guided by the Chinese Remainder Theorem

*F. J. MacWilliams. Permutation decoding of systematic codes. Bell System Tech. J., 43:485–505, 1964.
**S. Pawar, K. Ramchandran, "FFAST: An Algorithm for Computing an Exactly k-Sparse DFT in O(klog k)Time," ISIT'18 pp. 429-450

What if we have cycles in graph (N is short) or/and Noise

1. Automorphism Group Decoder*: use shift to form cycle code and differ sample factor to make new graph and broke cycle (make closer to ML calculation).

DFT form cycle code, under multiplied $\omega = e^{i\frac{2\pi}{20}}$ Using cyclic shift by ϖ together with DFT different size (overlap)



FFAST** (FAST Fourier Aliasing-based Sparse Transform

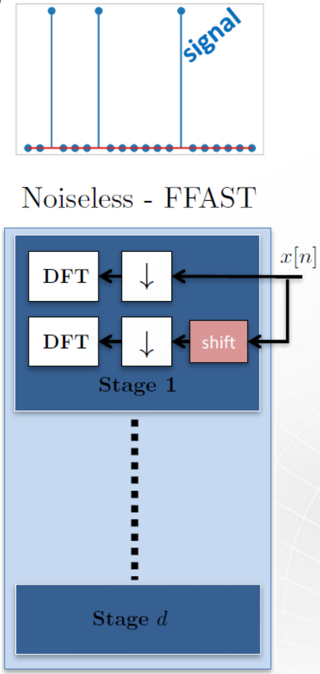
Noiseless: For **K** sublinear in **N**

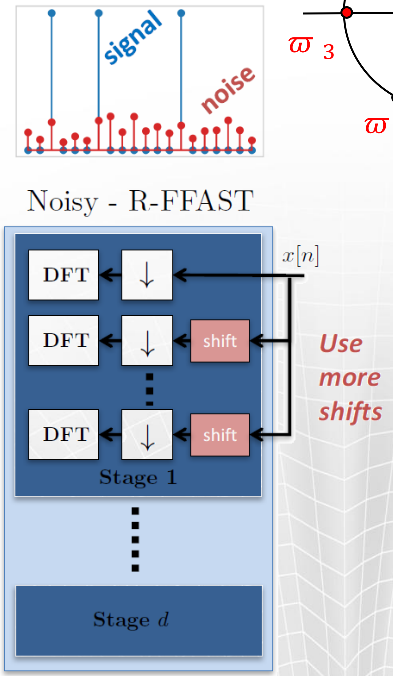
4K samples, $\boldsymbol{O(KlogK)}$

Robust to noise:

$\boldsymbol{O(Klog^{4/3}K)}$ sample

for $\boldsymbol{O(Klog^{\frac{7}{3}}N)}$ time





Decoder fail due to cycles
fails to recover 8/17000 non-zero DFT coefficients due to an 8-cycle in the alias-code graph

*F. J. MacWilliams. Permutation decoding of systematic codes. Bell System Tech. J., 43:485–505, 1964.
**S. Pawar, K. Ramchandran, "FFAST: An Algorithm for Computing an Exactly k-Sparse DFT in O(klog k)Time," ISIT'18 pp. 429-450

What if we have cycles in graph (N is short) or/and Noise

Interaction Screening* (decorrelate variables nodes participated in cycle)

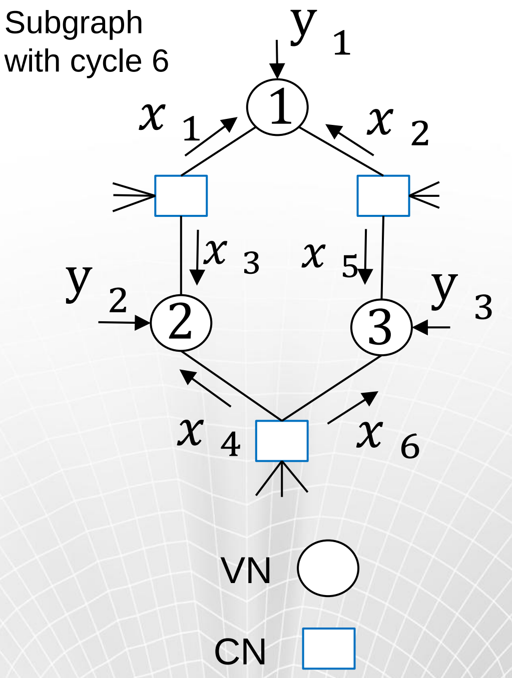
Opposing "force" can* decorrelate *spin moment (ISING model). Same work for factor graph.

$x_1, x_2, x_3, x_4, x_5, x_6$ messages

NMS decoder with scale factor α: $Ax = y,$

$$\begin{cases} x_1 = (y_2 + x_4)/\alpha \\ x_2 = (y_3 + x_6)/\alpha \\ x_3 = (y_1 + x_2)/\alpha \\ x_4 = (y_3 + x_5)/\alpha \\ x_5 = (y_1 + x_1)/\alpha \\ x_6 = (y_2 + x_3)/\alpha \end{cases}$$

$$A = \begin{pmatrix} \alpha & 0 & 0 & -1 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & -1 \\ 0 & -1 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & -1 & 0 \\ -1 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & -1 & 0 & 0 & \alpha \end{pmatrix}$$



Choice of α effect on converge

For $\alpha > 1, A$ – full rank and system have unique solution

***0 codeword**, $x_i \gg 0$*

M. Chertkov et al. Interaction screening: efficient and sample-optimal learning of ising models. NIPS'16 pp. 2603-2611

R. M. Tanner, et al "Improved min-sum decoding algorithms for irregular LDPC codes," *ISIT 2005*., pp. 449-453.

What if we have cycles in graph (N is short) or/and Noise

Interaction Screening* (decorrelate variables nodes participated in cycle)

Opposing "force" can* decorrelate *spin moment (ISING model). Same work for factor graph.

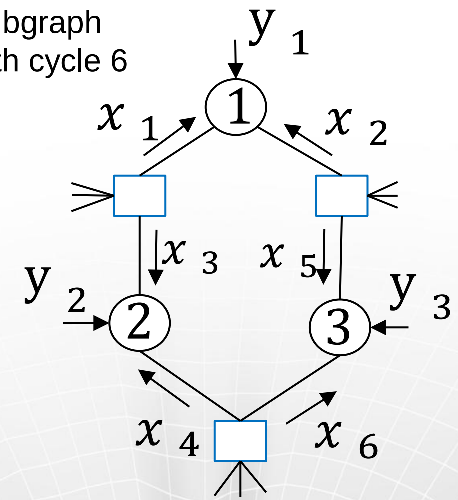
$x_1, x_2, x_3, x_4, x_5, x_6$ messages

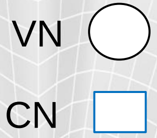
NMS decoder: $Ax = y,$

$$\begin{cases} x_1 = (y_2 + x_4)/\alpha \\ x_2 = (y_3 + x_6)/\alpha \\ x_3 = (y_1 + x_2)/\alpha \\ x_4 = (y_3 + x_5)/\alpha \\ x_5 = (y_1 + x_1)/\alpha \\ x_6 = (y_2 + x_3)/\alpha \end{cases}$$

$$A = \begin{pmatrix} \alpha & 0 & 0 & -1 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & -1 \\ 0 & -1 & \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & -1 & 0 \\ -1 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & -1 & 0 & 0 & \alpha \end{pmatrix}$$

Subgraph with cycle 6





$If\ x_1 + x_2 + y_1 < 0$

$VN\ 1,\ erroneously\ decoded$ no matter how reliable other messages are.

$\alpha = 1\ (MS, BP), A$ not full rank

decoding can not reach a steady-state

For $\alpha > 1, A$ – full rank and system have unique solution

0 codeword, $x_i \gg 0$

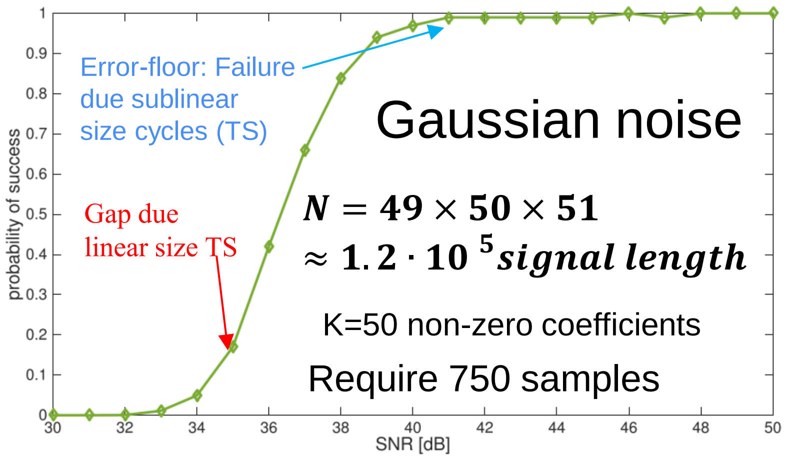
M. Chertkov et al. Interaction screening: efficient and sample-optimal learning of ising models. NIPS'16 pp. 2603-2611

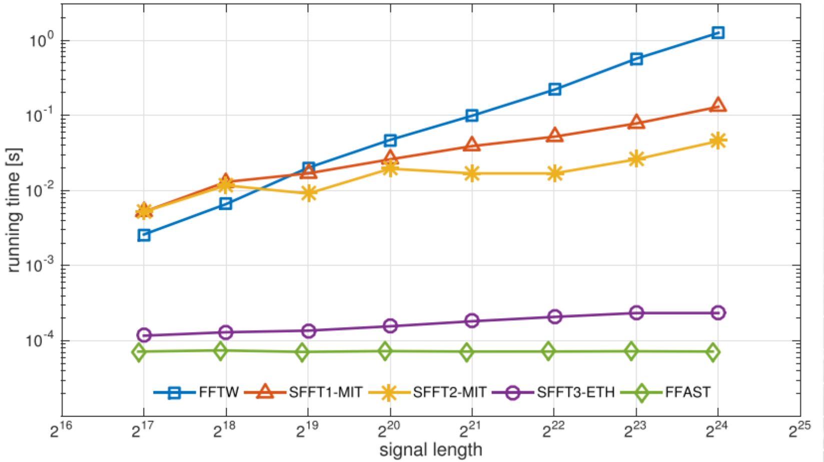
R. M. Tanner, et al "Improved min-sum decoding algorithms for irregular LDPC codes," *ISIT 2005*., pp. 449-453.

Open problem related to sub-Nyquist Sampling

1. *Finite-length Analysis of decoding probabilities (Covariance evolution);*
2. *Optimal CRT based ensemble for capacity-archieving family, best scale factor among all ensembles of codes;*

3. Optimization of peeling decoder scheduler for improve converge and performance. Polar Sparse-filter-bank codes for Sub-Nyquist Sampling?





SNR=39 dB too high for many applications

Signal length large (latency), Run-time/accuracy advantage - on long length

7 dB error-floor

Thank You