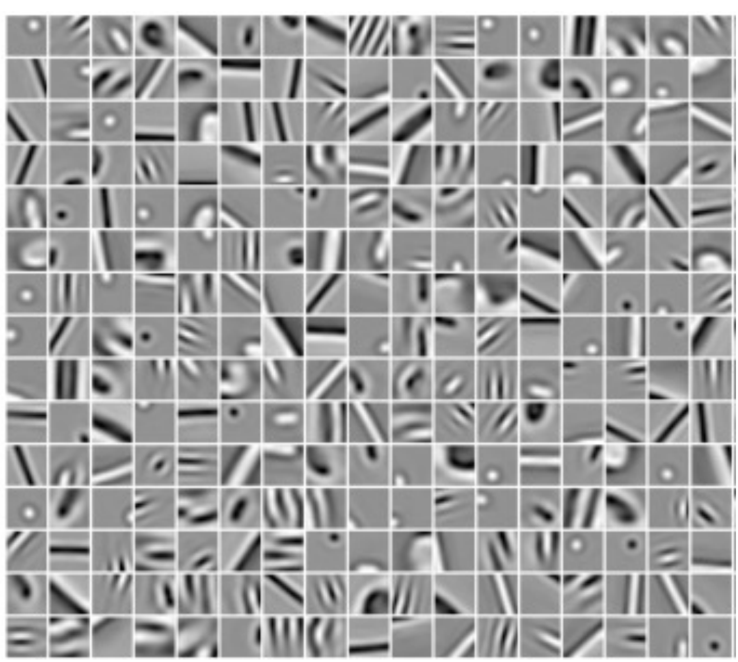
Dictionary learning in geoscience

Michael Bianco
UCSD Noise Lab, Scripps Institution of Oceanography
noiselab.ucsd.edu

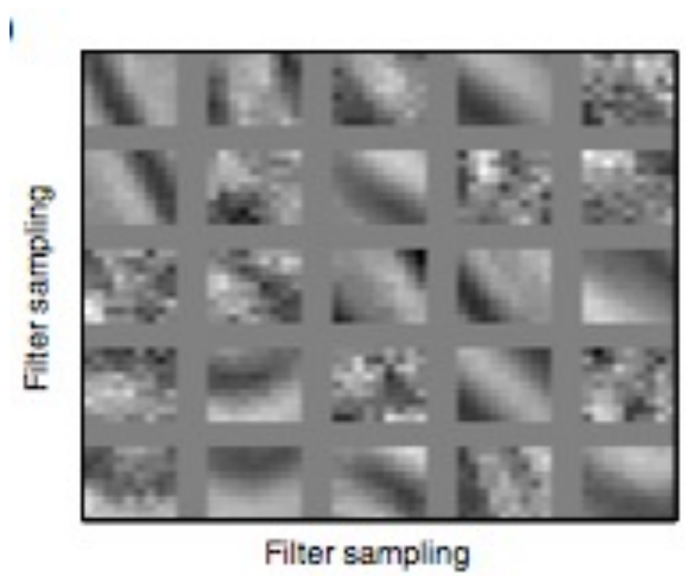
5/9/18

Dictionary learning

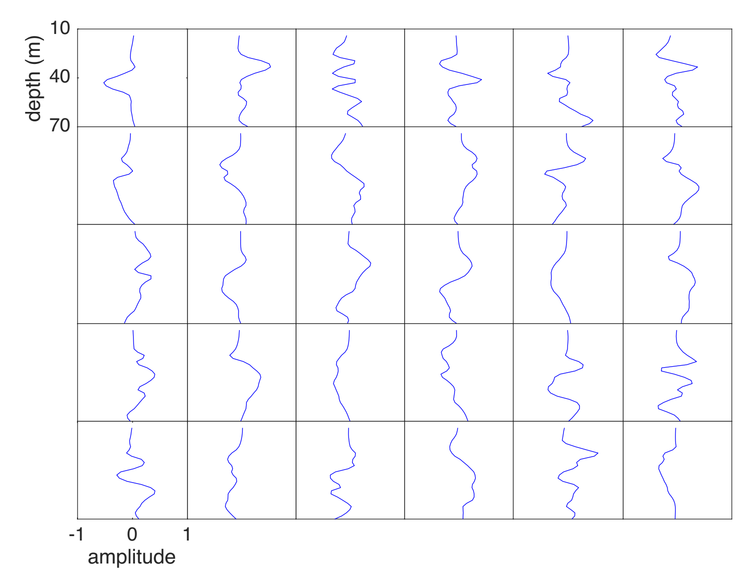
- Means of estimating sparse causes for given classes of signals, e.g. natural images, audio
- Originated in neuroscience to estimate structure of V1 visual cortex cells from natural images
- Useful for regularization of general image denoising inverse problem, but only recent applications in the geosciences
    - Seismic survey image denoising
    - Dictionary learning of ocean sound speed profiles (SSPs)



Olshausen 2009

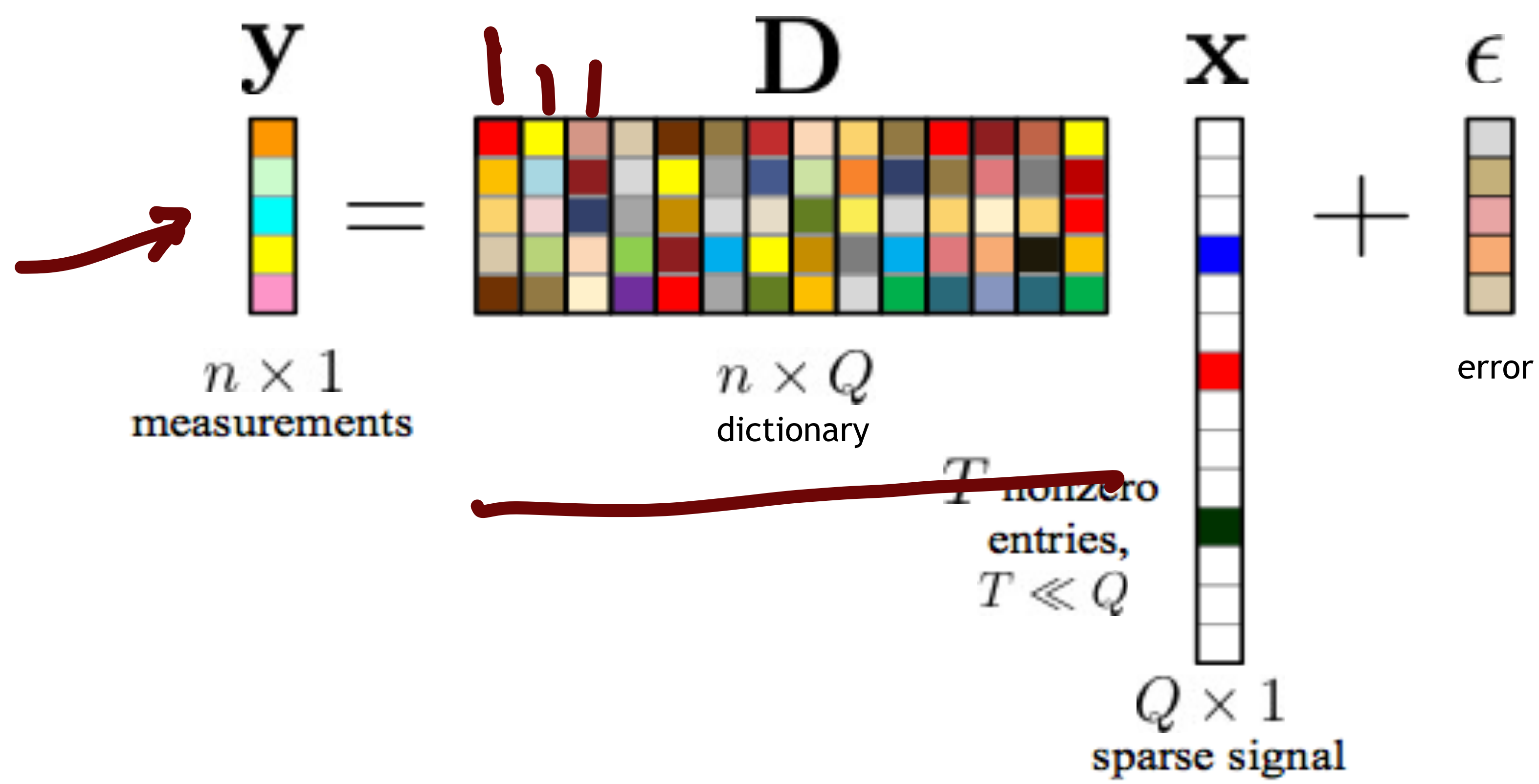


Beckouche 2014



Bianco and Gerstoft 2017

Background: sparse modeling of arbitrary signal **y**

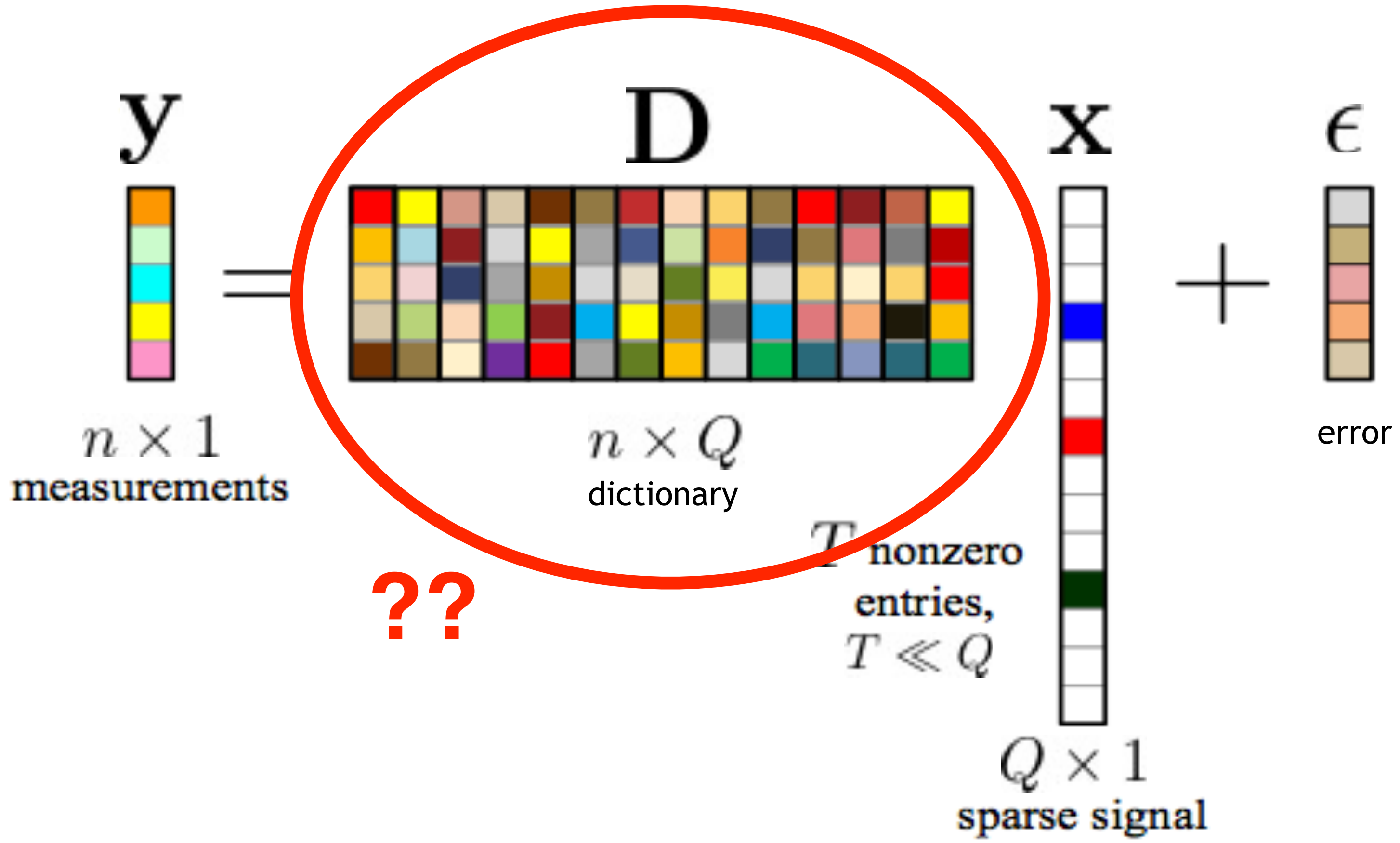


- Measurement vector **y** is expressed as sparse linear combination of columns or "atoms" from dictionary **D**
- **y** could be (for example) segments of speech or vectorized 2D image patches
- Dictionary atoms represent elemental patterns that generate **y,** e.g. wavelets or learned from the data using dictionary learning
- **x** is estimated using sparsity inducing constraint, example "ℓ_0 -norm" regularization:

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq T$$

ℓ_0- norm "counts" # non-zero coefficients

Background: sparse modeling of arbitrary signal **y**



- Measurement vector **y** is expressed as sparse linear combination of columns or "atoms" from dictionary **D**
- **y** could be (for example) segments of speech or vectorized 2D image patches
- Dictionary atoms represent elemental patterns that generate **y,** e.g. wavelets or learned from the data using dictionary learning
- **x** is estimated using sparsity inducing constraint, example "ℓ_0-norm" regularization:

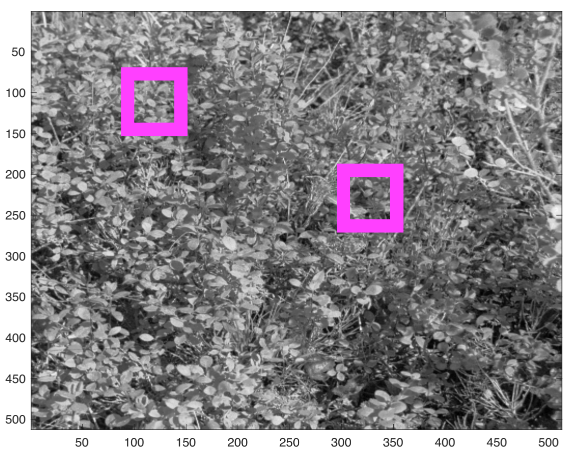
$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq T$$

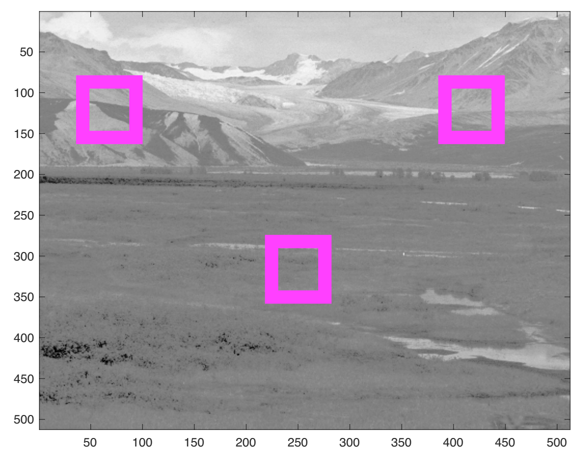
ℓ_0- norm "counts" # non-zero coefficients

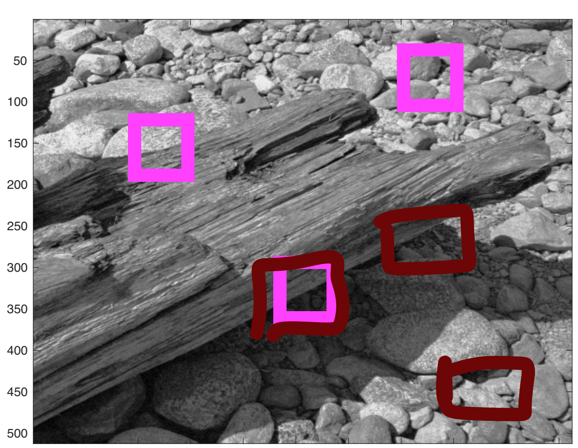
Background: sparsity and dictionary learning

- Dictionary learning obtains "optimal" sparse modeling dictionaries directly from data
- Dictionary learning was developed in neuroscience (a.k.a. sparse coding) to help understand mammalian visual cortex structure
- Assumes (1) Redundancy in data: image patches are repetitions of a smaller set of elemental shapes; and (2) Sparsity: each patch is represented with few atoms from dictionary

"Natural" images, patches shown in **magenta**

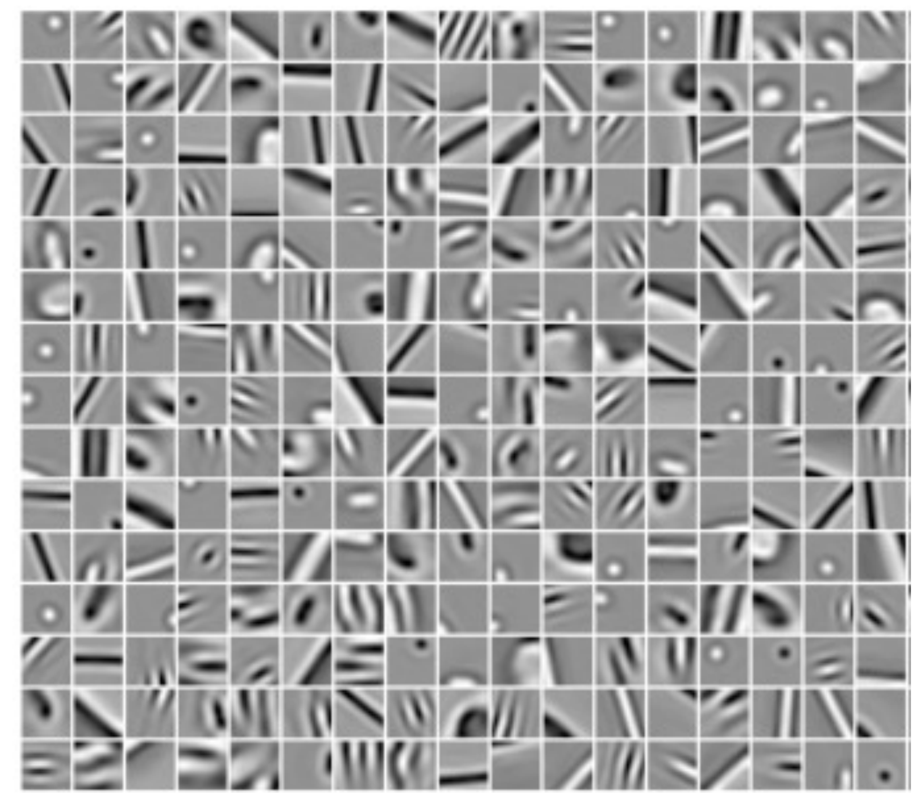






- Each patch is signal $\mathbf{y}_i$
- Set of all patches $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_I]$

Learn dictionary **D** describing $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_I]$



Olshausen 2009

Sparse model for patch $\mathbf{y}_i$ composed of few atoms from **D**

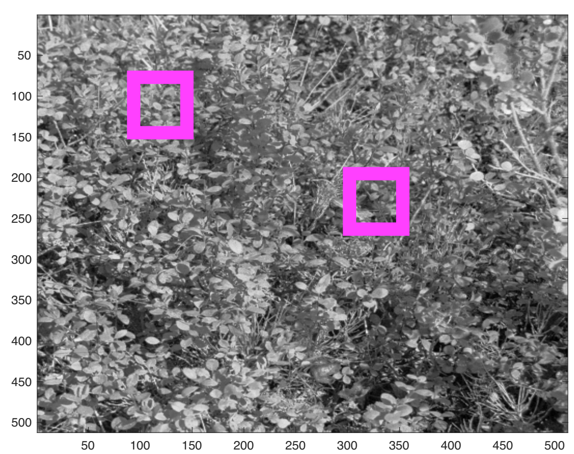
$$\widehat{\mathbf{x}}_i = \arg\min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T$$

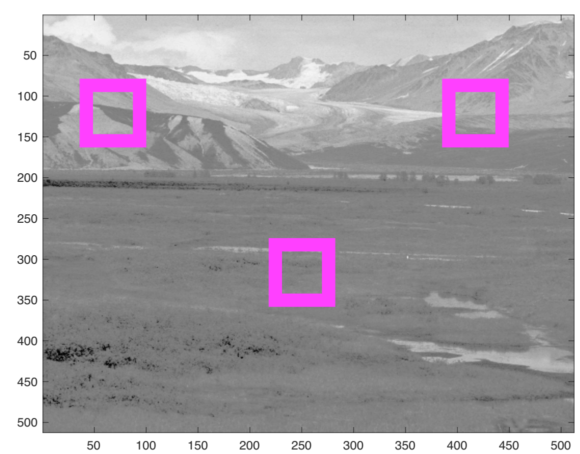
"counts nonzero"

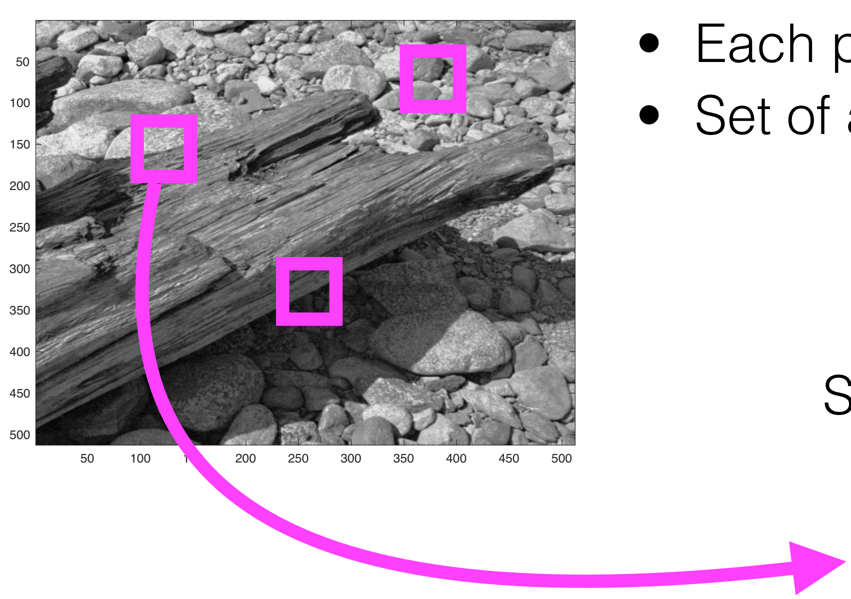
Background: sparsity and dictionary learning

- Dictionary learning obtains "optimal" sparse modeling dictionaries directly from data
- Dictionary learning was developed in neuroscience (a.k.a. sparse coding) to help understand mammalian visual cortex structure
- Assumes (1) Redundancy in data: image patches are repetitions of a smaller set of elemental shapes; and (2) Sparsity: each patch is represented with few atoms from dictionary

"Natural" images, patches shown in **magenta**

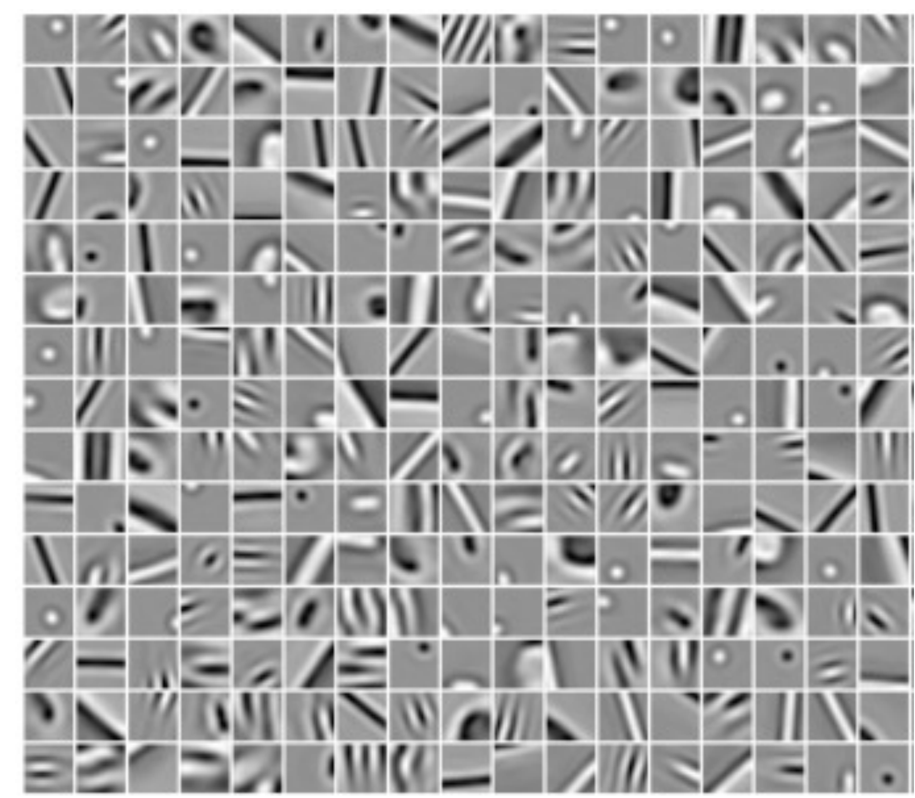






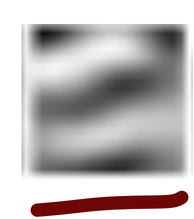
- Each patch is signal $\mathbf{y}_i$
- Set of all patches $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_I]$

Learn dictionary **D** describing $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_I]$



Olshausen 2009

Sparse model for patch $\mathbf{y}_i$ composed of few atoms from **D**

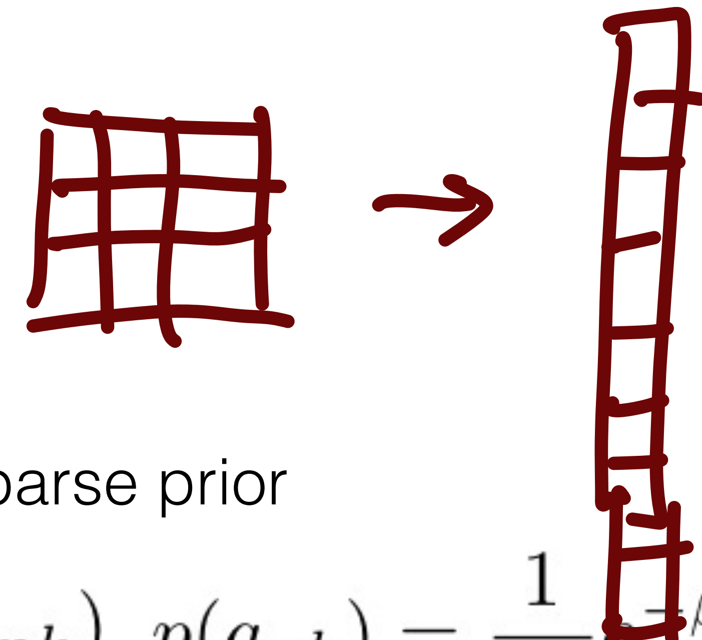
$\mathbf{y} =$  $=$  $x_1 +$  $x_2 + \ldots$

Olshausen and Field 1997: image model with sparse prior

Assume that each image patch described by linear system

$y \in R^k, k=64$

$$\mathbf{y}_k = \sum_n a_{nk}\phi_n = \mathbf{\Phi}\mathbf{a}_k \qquad \mathbf{y}_k = \mathbf{\Phi}\mathbf{a}_k + \mathbf{n}$$



Goal: estimate bases $\mathbf{\Phi}$ from observations $\mathbf{y}_k$

Probability of image patch arising from bases phi is

$$p(\mathbf{y}_k|\mathbf{\Phi}) = \int p(\mathbf{y}_k|\mathbf{a}_k, \mathbf{\Phi})p(\mathbf{a}_k)d\mathbf{a}_k$$, with

Likelihood

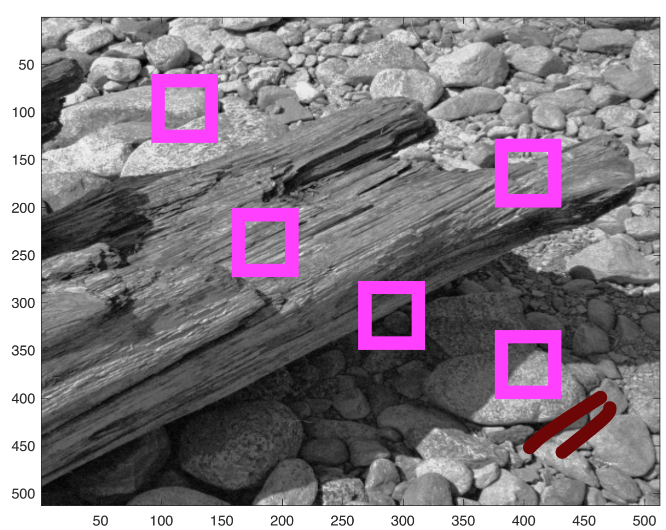
$$p(\mathbf{y}_k|\mathbf{a}_k, \mathbf{\Phi}) = \frac{1}{Z_\sigma} e^{\frac{-\|\mathbf{y}_k - \mathbf{\Phi}\mathbf{a}_k\|_2^2}{2\sigma^2}}$$

Independent, sparse prior

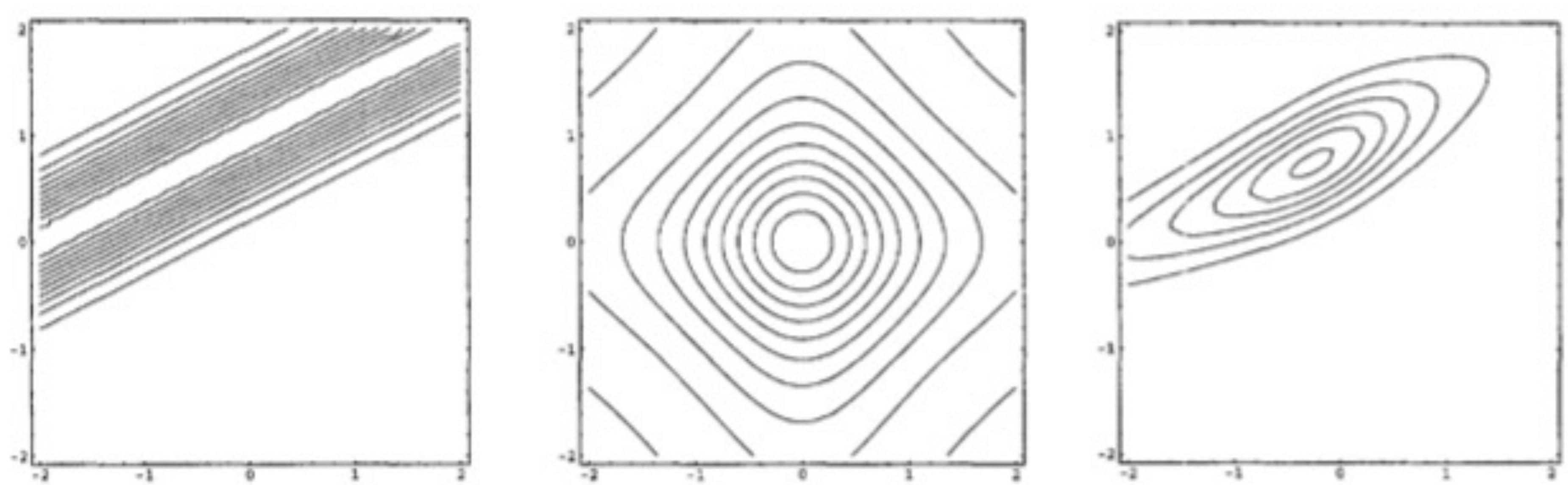
$$p(\mathbf{a}_k) = \prod p(a_{nk}) \quad p(a_{nk}) = \frac{1}{Z_\beta} e^{-\beta S(a_{nk})}$$

$$p(\mathbf{y}_k, \mathbf{a}_k|\mathbf{\Phi}) = p(\mathbf{y}_k|\mathbf{a}_k, \mathbf{\Phi})p(\mathbf{a}_k)$$

Image patches $\mathbf{y}_k$



$p(\mathbf{y}_k|\mathbf{a}_k, \mathbf{\Phi})$ $\qquad$ $p(\mathbf{a}_k)$



Likelihood $\qquad$ Prior $\qquad$ Posterior

Olshausen and Field 1997- sparse prior induces sparse coefficients

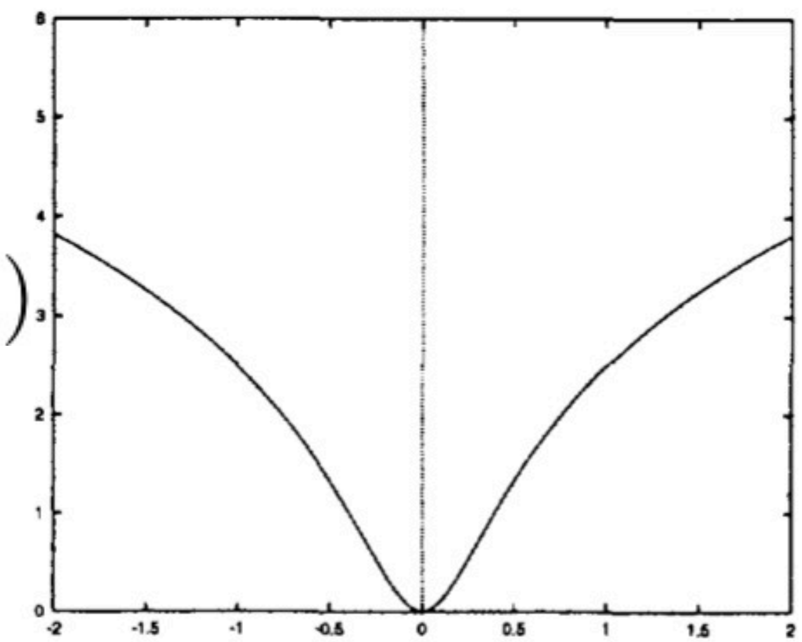
Sparsity inducing prior

$$p(a_{nk}) = \frac{1}{Z_\beta} e^{-\beta S(a_{nk})}$$

$$S(a_{nk}) = \ln(1 + a_{nk})^2$$

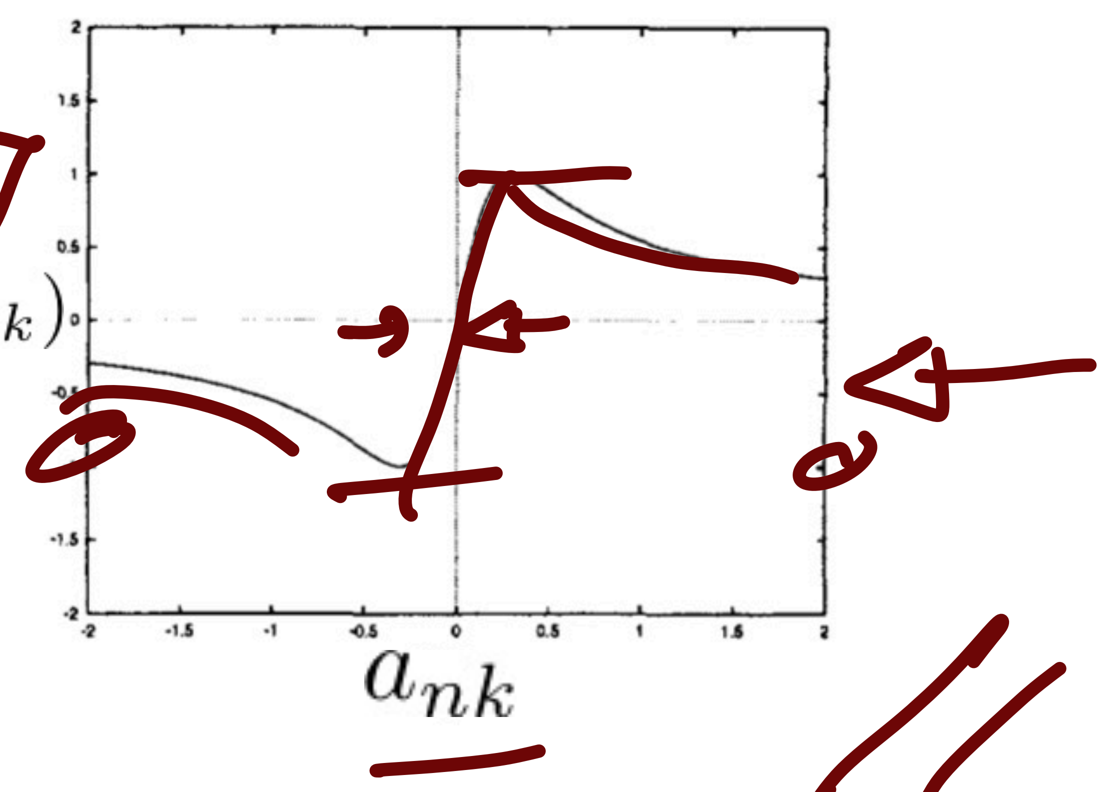
"Cauchy distribution"

$S(a_{nk})$



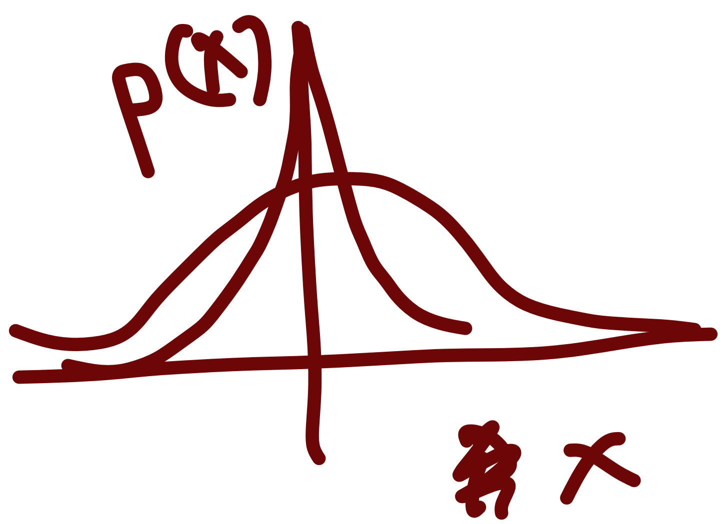
a_{nk}

$S'(a_{nk})$



a_{nk}

Derivative of prior induces sparsity in solution, as we'll see…

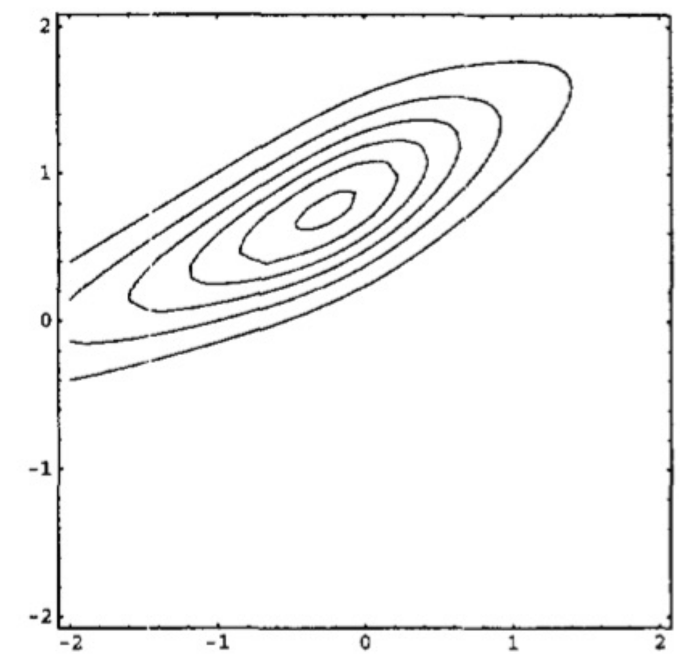


Olshausen and Field 1997 - derivation of Error function

Learn basis functions $\mathbf{\Phi}$ by minimizing Kullback-Leibler (KL) divergence between true images and those reproduced by model

$$KL = \int p^*(\mathbf{y}_k) \ln \frac{p^*(\mathbf{y}_k)}{p(\mathbf{y}_k|\mathbf{\Phi})} d\mathbf{y}_k$$

$$p(\mathbf{y}_k, \mathbf{a}_k|\mathbf{\Phi}) = p(\mathbf{y}_k|\mathbf{a}_k, \mathbf{\Phi})p(\mathbf{a}_k)$$



Since $p^*(\mathbf{y}_k)$ is fixed, KL is minimized by maximizing log-likelihood (or minimizing negative log-likelihood) of image patches generated from model, hence

$$\{\widehat{\mathbf{\Phi}}, \widehat{\mathbf{a}}_k\} = \arg\min_{\mathbf{\Phi}} \left[\min_{\mathbf{a}_k} E(\mathbf{y}_k, \mathbf{a}_k|\mathbf{\Phi})\right]$$

$$E(\mathbf{y}_k, \mathbf{a}_k|\mathbf{\Phi}) = -\ln p(\mathbf{y}_k|\mathbf{a}_k, \mathbf{\Phi})p(\mathbf{a}_k)$$

Given: $p(\mathbf{y}_k|\mathbf{a}_k, \mathbf{\Phi}) = \frac{1}{Z_\sigma} e^{\frac{-\|\mathbf{y}_k - \mathbf{\Phi}\mathbf{a}_k\|_2^2}{2\sigma^2}}$ $\quad p(\mathbf{a}_k) = \prod p(a_{nk})$ $\quad p(a_{nk}) = \frac{1}{Z_\beta} e^{-\beta S(a_{nk})}$

$$E(\mathbf{y}_k, \mathbf{a}_k|\mathbf{\Phi}) = \|\mathbf{y}_k - \mathbf{\Phi}\mathbf{a}_k\|_2^2 + \lambda \sum_n S(a_{nk})$$

Olshausen and Field 1997 - derivation of Error function cont'd

Learn basis functions $\mathbf{\Phi}$ by minimizing Kullback-Leibler (KL) divergence between true images and those reproduced by model

$$KL = \int p^*(\mathbf{y}_k) \ln \frac{p^*(\mathbf{y}_k)}{p(\mathbf{y}_k|\mathbf{\Phi})} d\mathbf{y}_k$$

Min KL

$$= \int p^*(y) [\ln p^*(y) - \ln p(y|\phi)] \, dy$$

$$= \cancel{\int p^*(y) \ln p^*(y) \, dy} - \int p^*(y) \ln p(y|\phi) \, dy$$

$$\min KL = \min \text{''} = \max \underbrace{\int p^*(y) \ln p(y|\phi) \, dy}$$

$$\max \langle \ln p(y|\phi) \rangle$$

Olshausen and Field 1997 - derivation of Error function cont'd

Learn basis functions $\mathbf{\Phi}$ by minimizing Kullback-Leibler (KL) divergence between true images and those reproduced by model

$$\{\widehat{\mathbf{\Phi}}, \widehat{\mathbf{a}}_k\} = \arg\min_{\mathbf{\Phi}} \left[\min_{\mathbf{a}_k} E(\mathbf{y}_k, \mathbf{a}_k | \mathbf{\Phi}) \right]$$

$$E(\mathbf{y}_k, \mathbf{a}_k | \mathbf{\Phi}) = -\ln p(\mathbf{y}_k | \mathbf{a}_k, \mathbf{\Phi}) p(\mathbf{a}_k)$$

Given: $p(\mathbf{y}_k | \mathbf{a}_k, \mathbf{\Phi}) = \frac{1}{Z_\sigma} e^{\frac{-\|\mathbf{y}_k - \mathbf{\Phi}\mathbf{a}_k\|_2^2}{2\sigma^2}}$ $\quad p(\mathbf{a}_k) = \prod p(a_{nk})$ $\quad p(a_{nk}) = \frac{1}{Z_\beta} e^{-\beta S(a_{nk})}$

$$p(y|\phi) = \int_{-\infty}^{\infty} p(y|a,\phi)\, p(a)\, da.$$

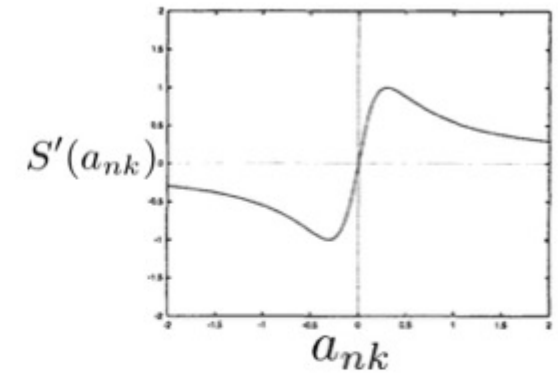
$$\hat{\phi} = \arg\max_{\phi} < \max_{a} \ln p(y|a,\phi)\, p(a)$$

Obtain: $E(\mathbf{y}_k, \mathbf{a}_k | \mathbf{\Phi}) = \|\mathbf{y}_k - \mathbf{\Phi}\mathbf{a}_k\|_2^2 + \lambda \sum_n S(a_{nk})$

Olshausen and Field 1997 - gradients for network model

Rewriting Error function, take derivatives to find gradient

$$E(\mathbf{y}_k, \mathbf{a}|\mathbf{\Phi}) = \sum_m (y_{mk} - \sum_n \phi_{mn} a_{nk})^2 + \lambda \sum_n S(a_{nk})$$



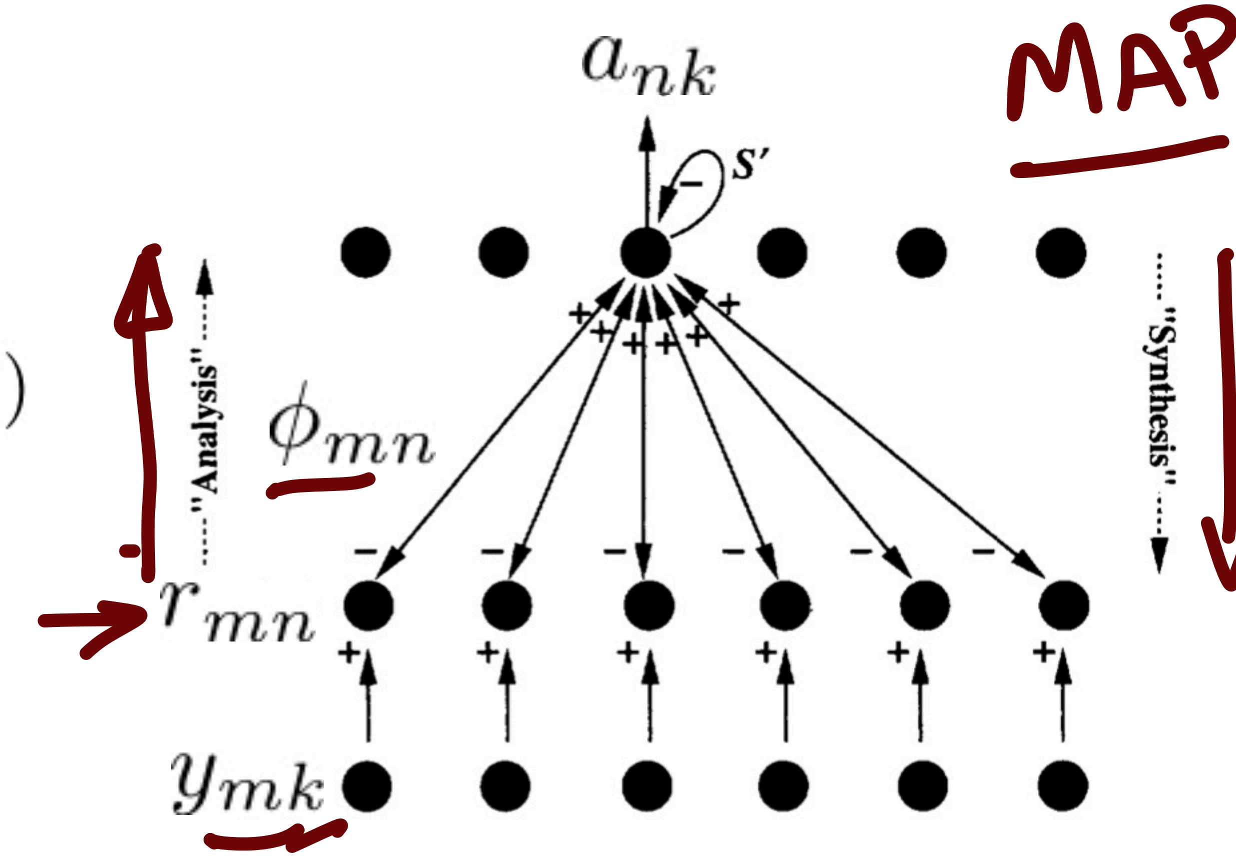
$$\{\widehat{\mathbf{\Phi}}, \widehat{\mathbf{a}}_k\} = \arg\min_{\mathbf{\Phi}} \big[\min_{\mathbf{a}_k} E(\mathbf{y}_k, \mathbf{a}|\mathbf{\Phi})\big]$$

MAP

Update to a_{nk} with network (inner loop)

$$\dot{a}_{nk} = -\frac{dE}{da_{nk}} = \sum_m \phi_{mn} r_{mn} - \lambda S'(a_{nk})$$

with $\quad r_{mn} = y_{mk} - \sum_n \phi_{mn} a_{nk}$



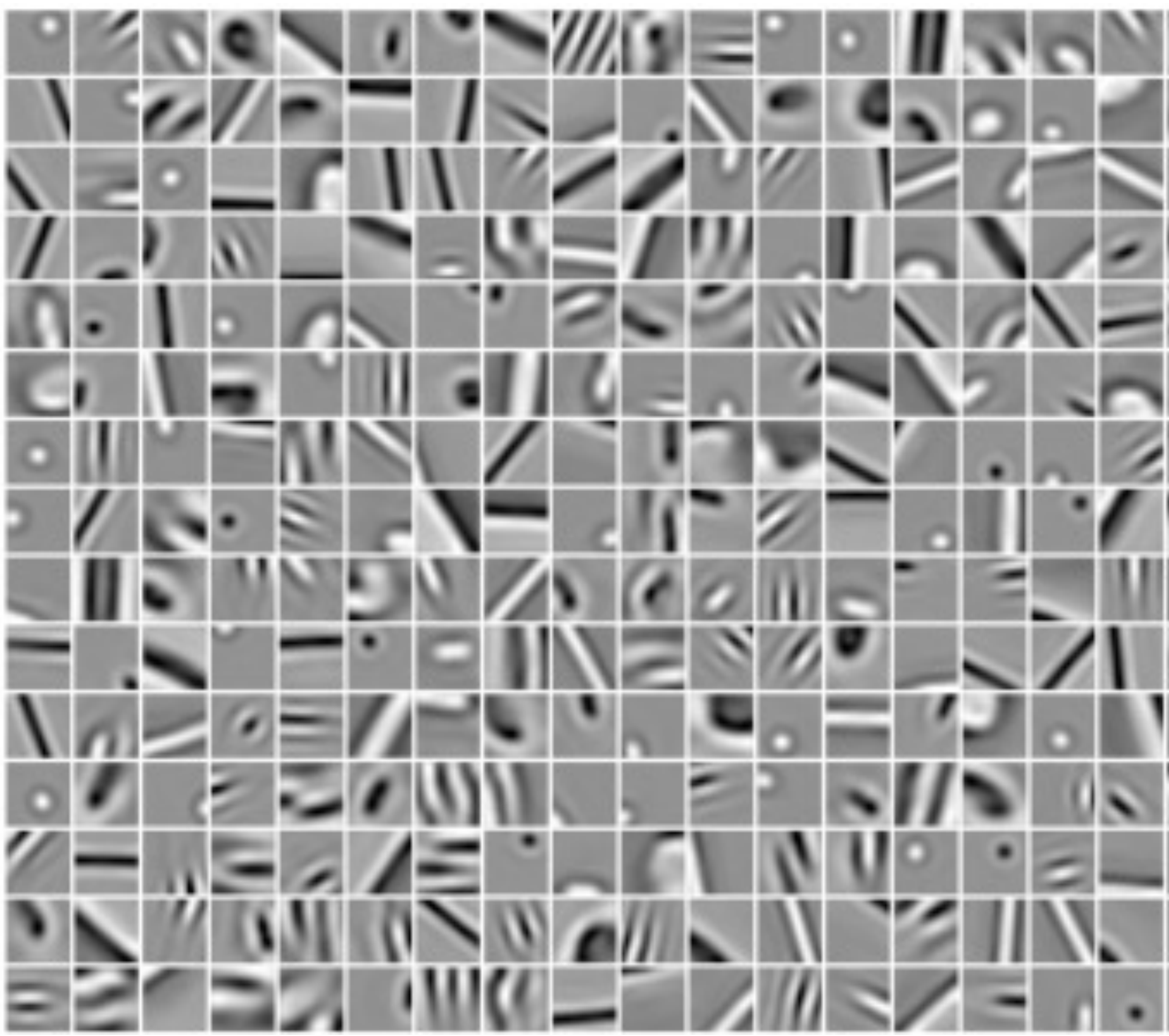
Update to ϕ_{mn} with gradient descent (outer loop)

$$\Delta\phi_{mn} = \eta < a_{nk} r_{mn} >$$

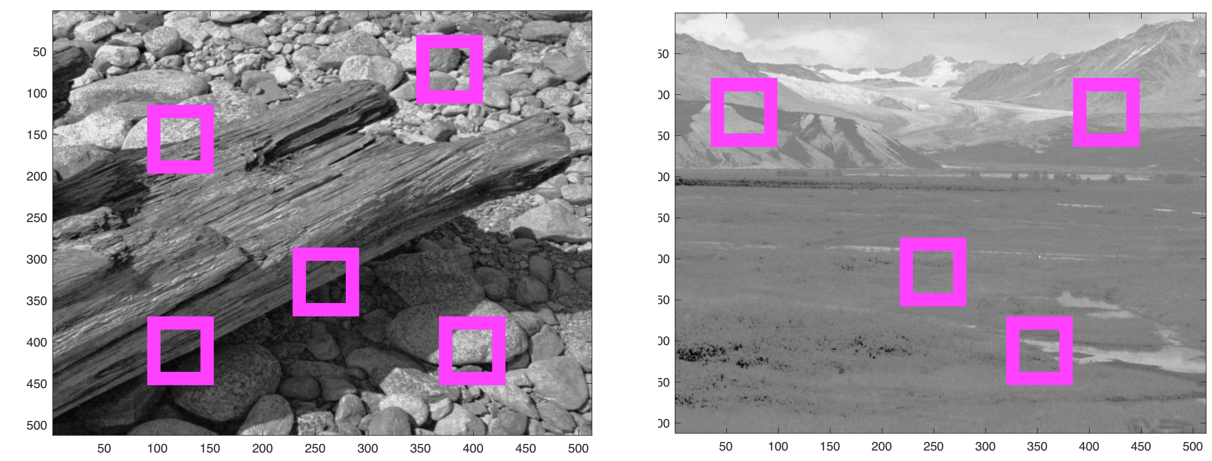
"Hebbian" update

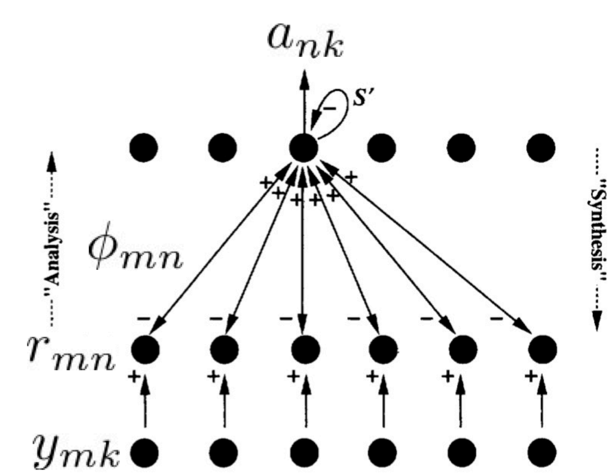
From Olshausen '97 method, obtain dictionary atoms that resemble cells from mammalian visual cortex

Dictionary elements ϕ_k



Natural image patches





"Hebbian" update

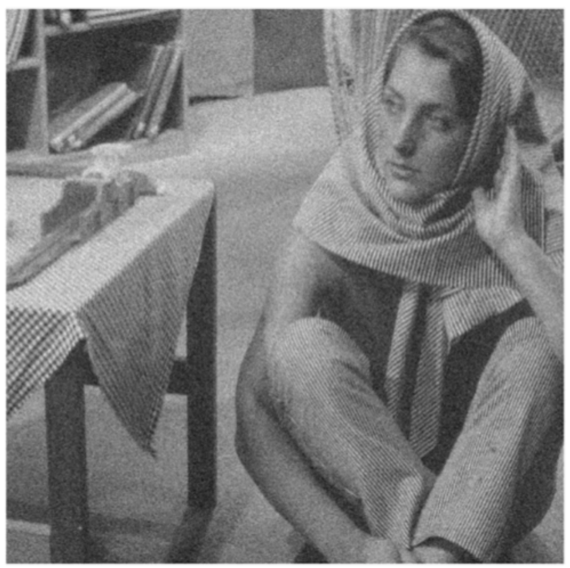
Nice to have atoms like cells, but what else is dictionary learning useful for?

Nice to have atoms like cells, but what else is dictionary learning useful for?

Image restoration tasks

Denoising

Noisy Image (22.1307 dB, σ=20)



Denoised Image Using Adaptive Dictionary (30.8295 dB)

Elad 2006

Inpainting (a.k.a. matrix completion)

Mairal 2009

Olshausen and Field 1997 - gradients for network model

"cauchy"

Can be rephrased with Laplacian prior

$$\widehat{\mathbf{\Phi}} = \arg\min_{\mathbf{\Phi}} \sum_k \min_{\mathbf{a}_k} \{ \|\mathbf{\Phi}\mathbf{a}_k - \mathbf{y}_k\|_2^2 + \lambda \|\mathbf{a}_k\|_1 \}$$

"Laplacian"

Coefficients calculated using gradient descent, then dictionary updated by

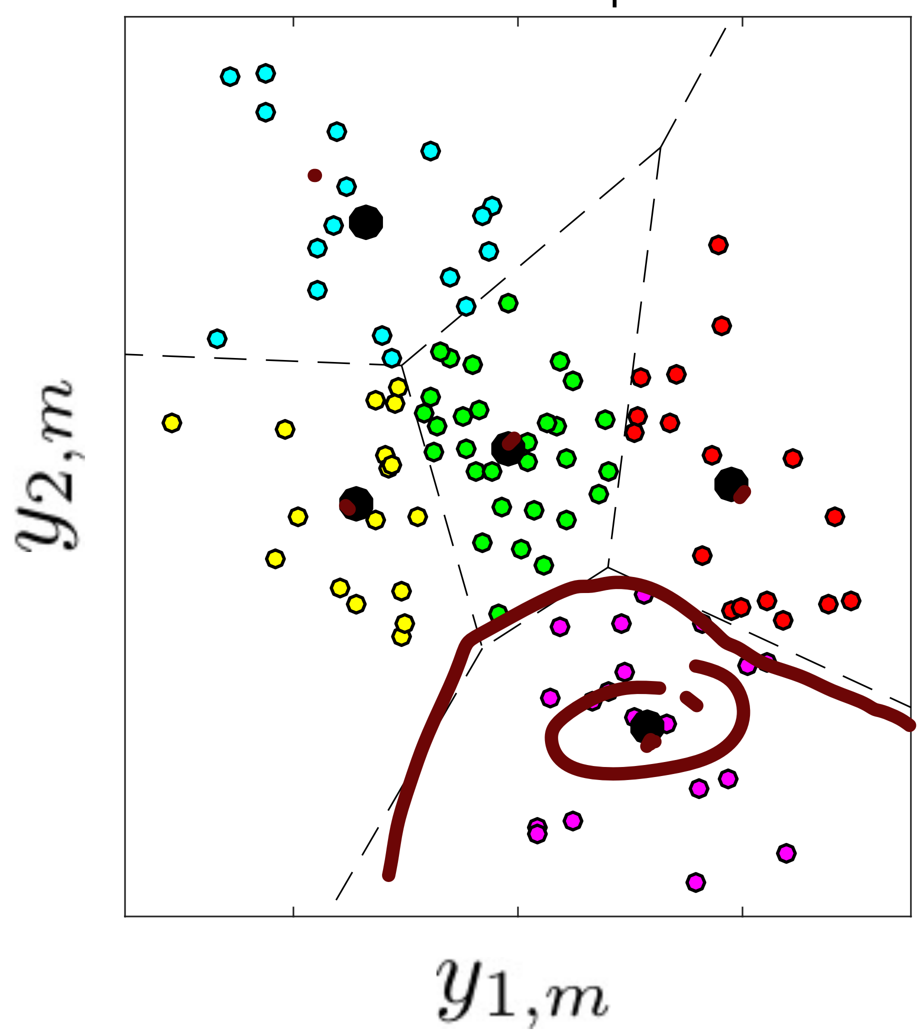
$$\mathbf{\Phi}^{(i+1)} = \mathbf{\Phi}^{(i)} - \eta \sum_k \left(\mathbf{\Phi}^{(i)} \mathbf{a}_k - \mathbf{y}_k \right) \mathbf{a}_k^T$$

This idea of iterative refinement is familiar: solving for coefficients, then updating basis functions

K-means.

Vector Quantization and K-means

2D example



Vector quantization (VQ): means of compressing a set of data observations $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ using a nearest neighbor metric with codebook $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]$

$$R_n = \{i \mid \forall_{l \neq n}, \|\mathbf{y}_i - \mathbf{c}_n\|_2 < \|\mathbf{y}_i - \mathbf{c}_l\|_2\}$$

$$S_n(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} \in R_n \\ 0 & \text{otherwise,} \end{cases} \quad \widehat{\mathbf{y}}_m = \sum_{i=1}^{N} S_i(\mathbf{y}_m)\mathbf{c}_i$$

K-means: finds optimal codebook for VQ

Given: training vectors $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_M] \in \mathbb{R}^{K \times M}$

Initialize: index $i = 0$, codebook $\mathbf{C}^0 = [\mathbf{c}_1^0, \ldots, \mathbf{c}_N^0] \in \mathbb{R}^{K \times N}$, MSE^0

I: Update codebook

1. Partition $\mathbf{Y}$ into N regions $(R_1, \ldots, R_N)$ by
   $R_n = \{i \mid \forall_{l \neq n}, \|\mathbf{y}_i - \mathbf{c}_n^i\|_2 < \|\mathbf{y}_i - \mathbf{c}_l^i\|_2\}$
2. Make code vectors centroids of $\mathbf{y}_j$ in partitions R_n
   $\mathbf{c}_n^{i+1} = \frac{1}{|R_n^i|} \sum_{j \in R_n^i} \mathbf{y}_j$

II. Check error

1. Calculate MSE^{i+1} from updated codebook $\mathbf{C}^{i+1}$
2. If $|\mathrm{MSE}^{i+1} - \mathrm{MSE}^i| < \eta$
   $i = i + 1$, return to I
   else

end

Relationship to sparse coding

$T = 1$

Sparse processor

$$\hat{\mathbf{x}}_m = \arg\min_{\mathbf{x}_m} \|\mathbf{y}_m - \mathbf{Q}\mathbf{x}_m\|_2 \text{ subject to } \|\mathbf{x}_m\|_0 \leq T$$

VQ operators

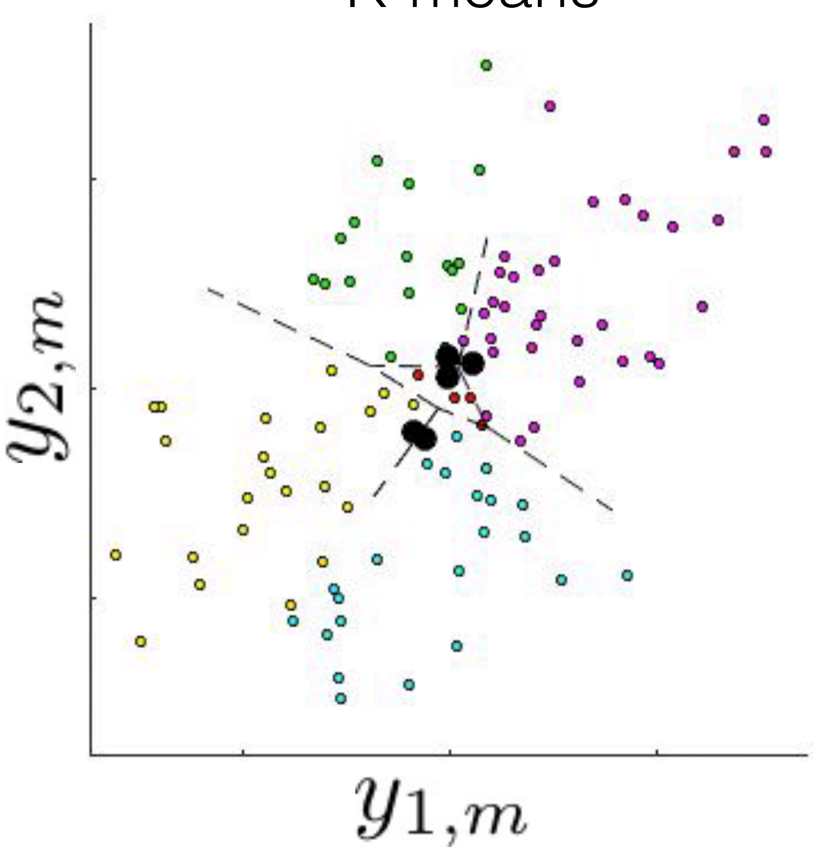
$$R_n = \{i \mid \forall_{l \neq n}, \|\mathbf{y}_i - \mathbf{c}_n\|_2 < \|\mathbf{y}_i - \mathbf{c}_l\|_2\}$$

$$\hat{\mathbf{y}}_m = \sum_{i=1}^{N} S_i(\mathbf{y}_m)\mathbf{c}_i \qquad S_n(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} \in R_n \\ 0 & \text{otherwise,} \end{cases}$$

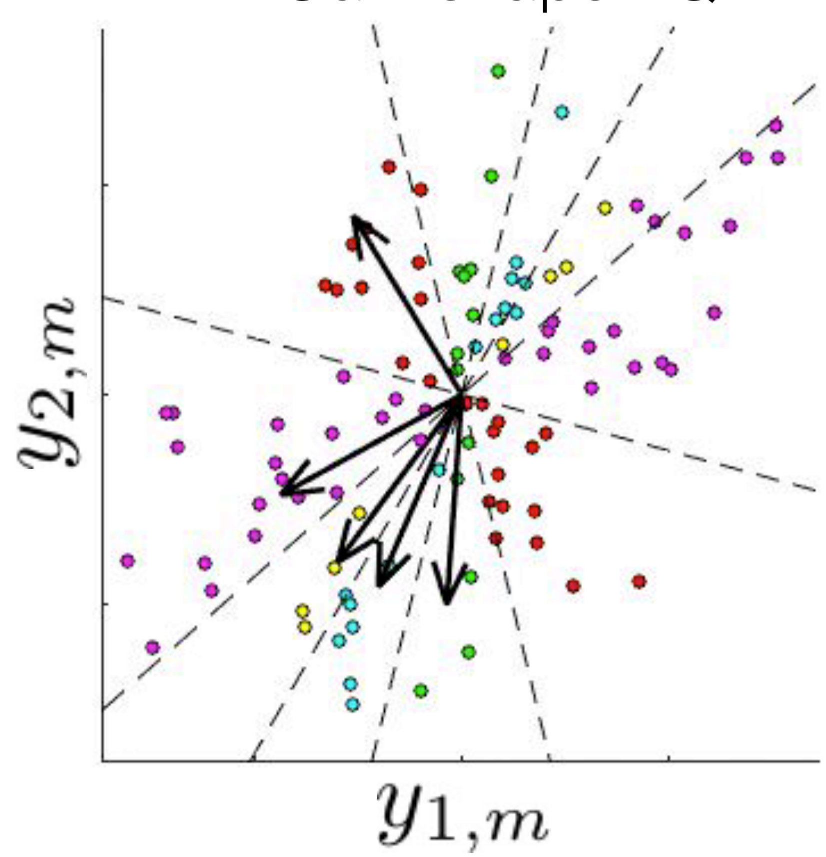
Dictionary learning objective

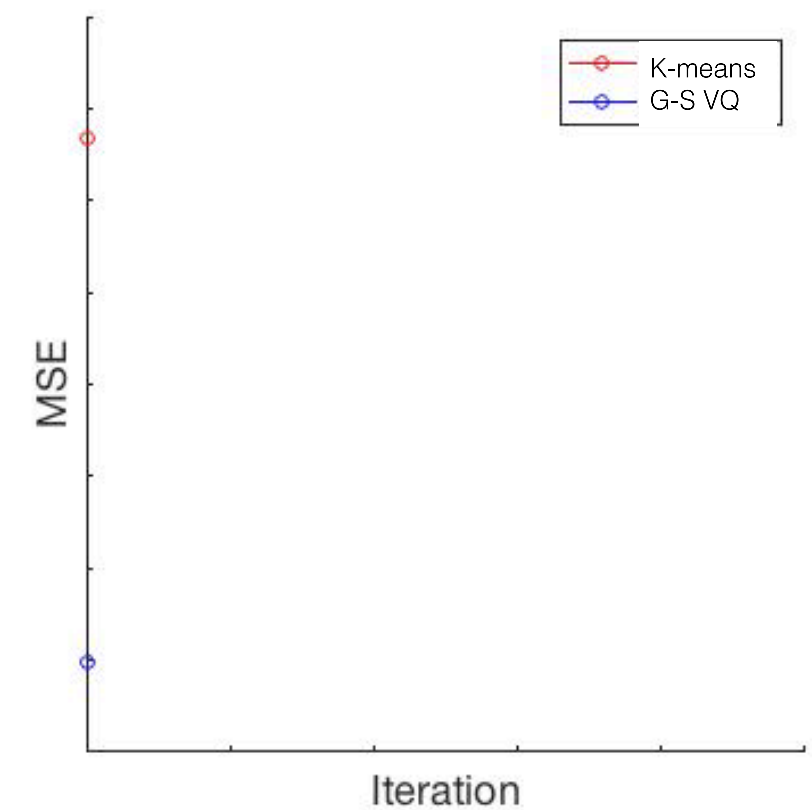
$$\min_{\mathbf{Q}}\{\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{Q}\mathbf{X}\|_F^2 \text{ subject to } \forall m, \ \|\mathbf{x}_m\|_0 \leq T\}$$

K-means



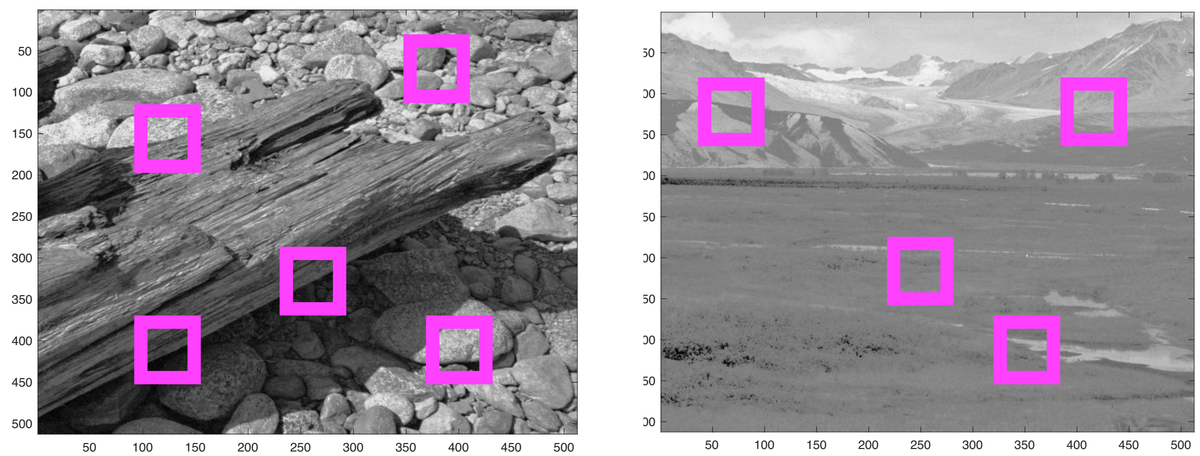
Gain-shape VQ



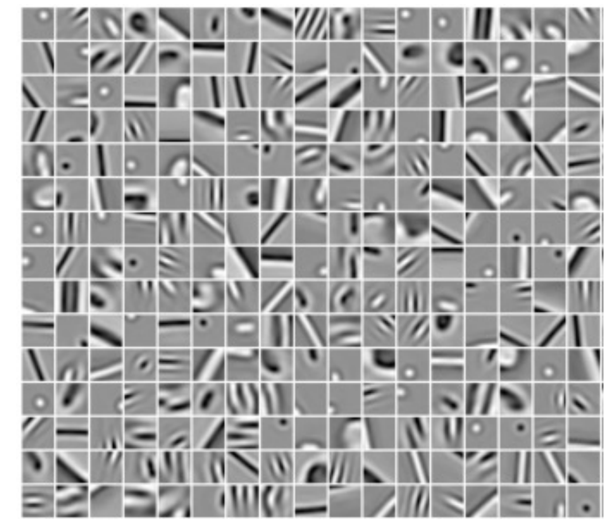


Background: a basic dictionary learning framework

Given set of patches $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_I]$, learn dictionary **D** describing them



Patches shown in **magenta**



Dictionary **D**

Dictionary learning objective

$$\min_{\mathbf{D}}\{\min_{\mathbf{X}}\|\mathbf{Y} - \mathbf{DX}\|_F^2 \ \text{ subject to } \ \|\mathbf{x}_i\|_0 \leq T \ \forall \ i\}$$

Objective solved as simple optimization problem "alt. min."

1. Solve for sparse coefficients $\mathbf{X} = [\widehat{\mathbf{x}}_{\ell_0,1}, \ldots \widehat{\mathbf{x}}_{\ell_0,I}]$ using sparse solver
2. Solve for dictionary **D** using sparse coefficients from step (1)….. repeat until convergence

MOD algorithm: Extending K-means to dictionary learning problem

Method of Optimal Directions (MOD) [Engan 2000]

$$\min_{\mathbf{Q}}\{\min_{\mathbf{X}}\|\mathbf{Y}-\mathbf{Q}\mathbf{X}\|_F^2 \;\text{ subject to }\; \forall m,\; \|\mathbf{x}_m\|_0 \le T\}$$

MOD algorithm:

1. COEFFICIENTS: Solve for coefficients **X=[x_1…x_i]** for fixed **Q** using orthogonal matching pursuit (OMP)
2. DICTIONARY UPDATE: Solve for dictionary **Q=[q_1…q_i]**, by inverting the coefficient matrix **X**, and normalizing dictionary entries to have unit norm.

$$\widehat{\mathbf{Q}} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$$

"pseudo inverse"

…. repeat until convergence

Simple and flexible but, a few drawbacks:

- computationally expensive to invert coefficient matrix **X**
- since keeping coefficients in **X** fixed during dictionary update, slow convergence

$$QX = Y$$
$$\rightarrow QXX^T = YX^T$$
$$\rightarrow Q\,\cancel{X} = YX^T(XX^T)^{-1}$$

K-SVD algorithm

K-SVD [Aharon 2006]: Learn optimal dictionary for sparse representation of data

$$\min_{\mathbf{Q}}\{\min_{\mathbf{X}}\|\mathbf{Y}-\mathbf{QX}\|_F^2 \ \text{ subject to } \ \forall m,\ \|\mathbf{x}_m\|_0 \le T\}$$

K-SVD algorithm:

1. Solve for coefficients **X=[x_1…x_i]** for fixed **Q** using OMP
2. Solve (1) for dictionary **Q=[q_1…q_i]**, updating both **Q** and **X** from the SVD of representation error

$$\|\mathbf{Y}-\mathbf{QX}\|_F = \left\|\underbrace{\left(\mathbf{Y}-\sum_{j\neq k}\mathbf{q}_j\mathbf{x}_T^j\right)}_{}-\mathbf{q}_k\mathbf{x}_T^k\right\|_F$$

SVD

$$= \|\mathbf{E}_k - \mathbf{q}_k\mathbf{x}_T^k\|$$

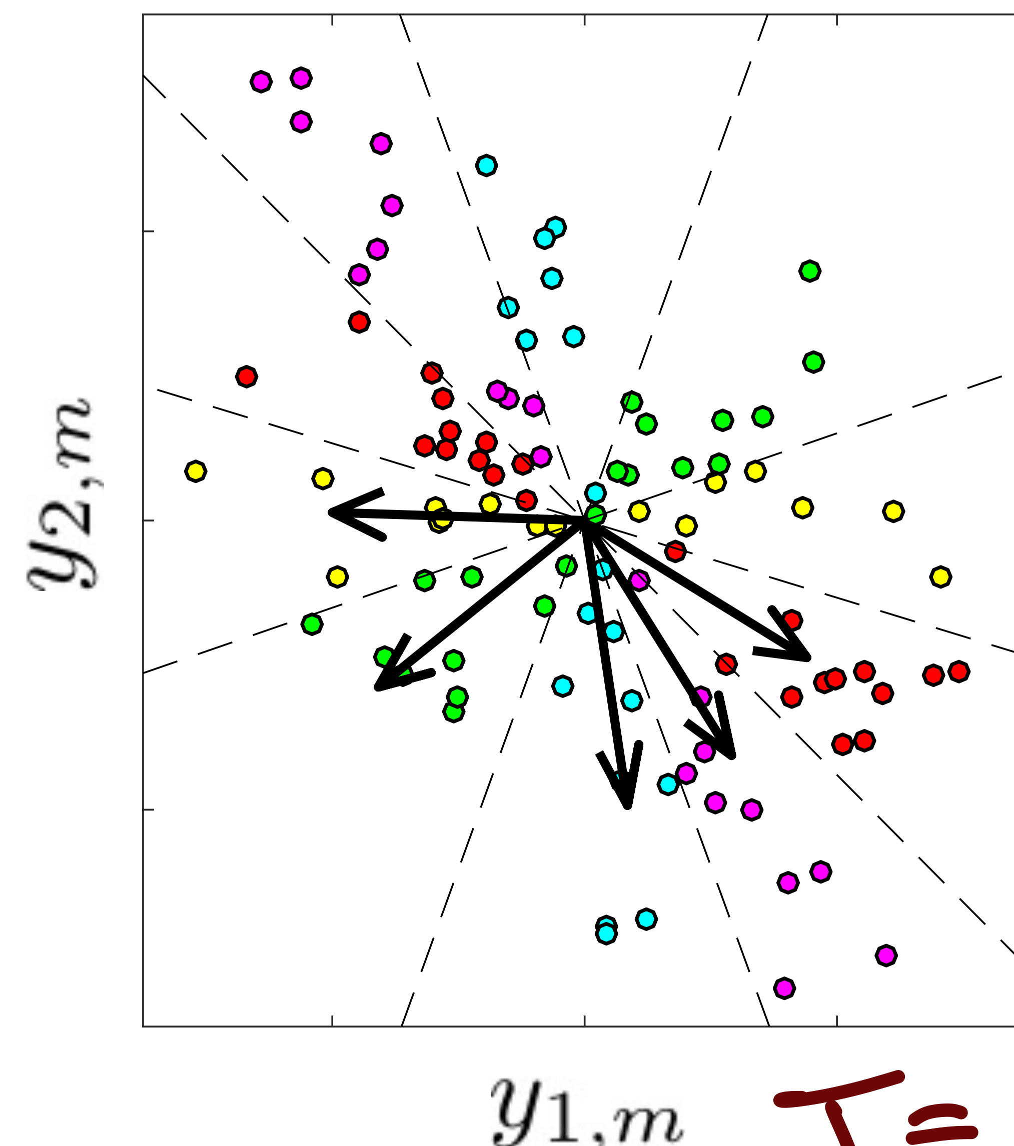
update **q_k, x_k** by SVD

$$\mathbf{E}_k^e = \mathbf{USV}^T$$

$$\mathbf{q}_k = \mathbf{U}(:,1), \mathbf{x}_T^k = \mathbf{V}(:,1)\mathbf{S}(1,1)$$

…. repeat until convergence

2D example



T=1

Image restoration tasks

Denoising

Noisy Image (22.1307 dB, σ=20)

Denoised Image Using
Adaptive Dictionary (30.8295 dB)

64pix

x_i

256

~60k

256

Elad 2006

$$\min_{\mathbf{D}\in\mathcal{C},\mathbf{A}\in\mathbb{R}^{p\times n}} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{\alpha}_i\|_0 \leq s$$

Inpainting (a.k.a. matrix completion)

Mairal 2009

$$\min_{\mathbf{D}\in\mathcal{C},\mathbf{A}\in\mathbb{R}^{p\times n}} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\mathbf{M}_i(\mathbf{y}_i-\mathbf{D}\boldsymbol{\alpha}_i)\|_2^2 + \lambda\psi(\boldsymbol{\alpha}_i),$$

Image restoration tasks

Denoising: learning from noisy image patches for specific image

Noisy Image (22.1307 dB, σ=20)

Denoised Image Using
Adaptive Dictionary (30.8295 dB)

Elad 2006

$$\min_{\mathbf{D}\in\mathcal{C},\mathbf{A}\in\mathbb{R}^{p\times n}} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \ \text{ s.t. } \ \|\boldsymbol{\alpha}_i\|_0 \le s$$

$\{\hat\alpha_{ij}, \hat X, \hat D\}$

$= \min \|X - Y\|_2^2 + \sum_i \|R_i X - D\alpha_i\| + \lambda \|\alpha_i\|_0$

Solved using block-coordinate descent algorithm (also two steps):

60K

(1) $$\hat{\boldsymbol{\alpha}}_{ij} = \arg\min_{\boldsymbol{\alpha}} \mu_{ij}\|\boldsymbol{\alpha}\|_0 + \|\mathbf{D}\boldsymbol{\alpha} - \mathbf{x}_{ij}\|_2^2$$

$\Rightarrow \{\hat D, \hat\alpha_{ij}\} = \min_D \{\min_\alpha$

(2) $$\hat{\mathbf{X}} = \arg\min_{\mathbf{x}} \lambda\|\mathbf{X} - \mathbf{Y}\|_2^2 + \sum_{ij}\|\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij} - \mathbf{R}_{ij}\mathbf{X}\|_2^2$$

$$\hat{\mathbf{X}} = \left(\cancel{\lambda\mathbf{I}} + \sum_{ij}\mathbf{R}_{ij}^T\mathbf{R}_{ij}\right)^{-1}\left(\cancel{\lambda\mathbf{Y}} + \sum_{ij}\mathbf{R}_{ij}^T\mathbf{D}\hat{\boldsymbol{\alpha}}_{ij}\right)$$

Why not just use neural networks?

Burger 2012: Multi-layer perceptron competes with state of art denoising algorithms, using 362 million training samples (~one month of GPU time)

… at least in geoscience (seimsics, ocean acoustics) we rarely have this much training data

Adaptive image denoising-like | **MLP-like**

Handcrafted | **Blackbox**

Adhere to existing algorithm architecture, few learnable parameters

Deviate from existing algorithm design, many learnable parameters

Pros:
- Likely more generalizable
- Less training data needed
- Natural initialization (from standard algorithm settings)

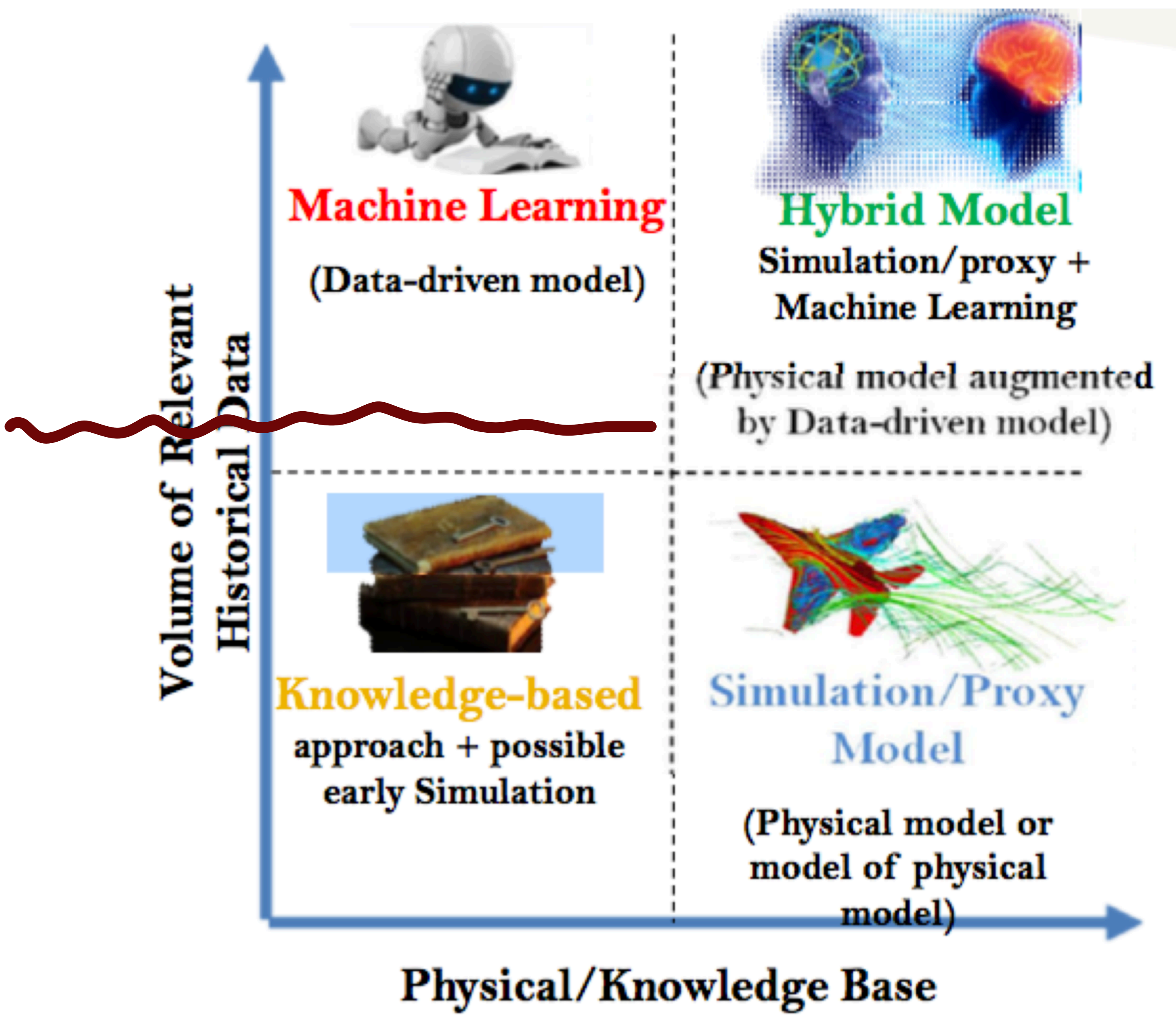
Cons:
- Reduced chance for optimal performance

Pros:
- Increased chance of optimal performance given sufficient training data

Cons:
- Maybe less generalizable
- More training data needed
- No algorithm to potentially guide initialization

Wipf 2018

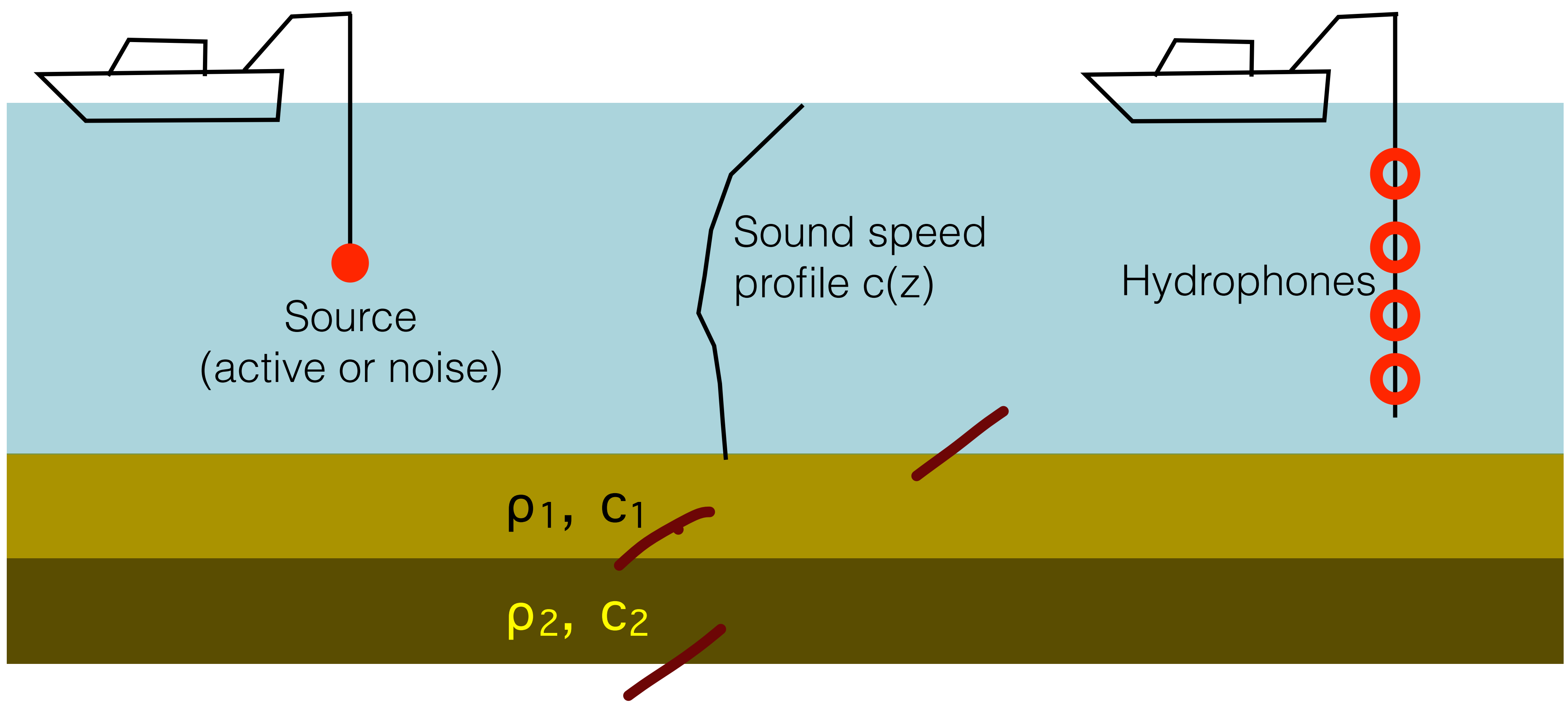
Why not just use neural networks? (cont'd)



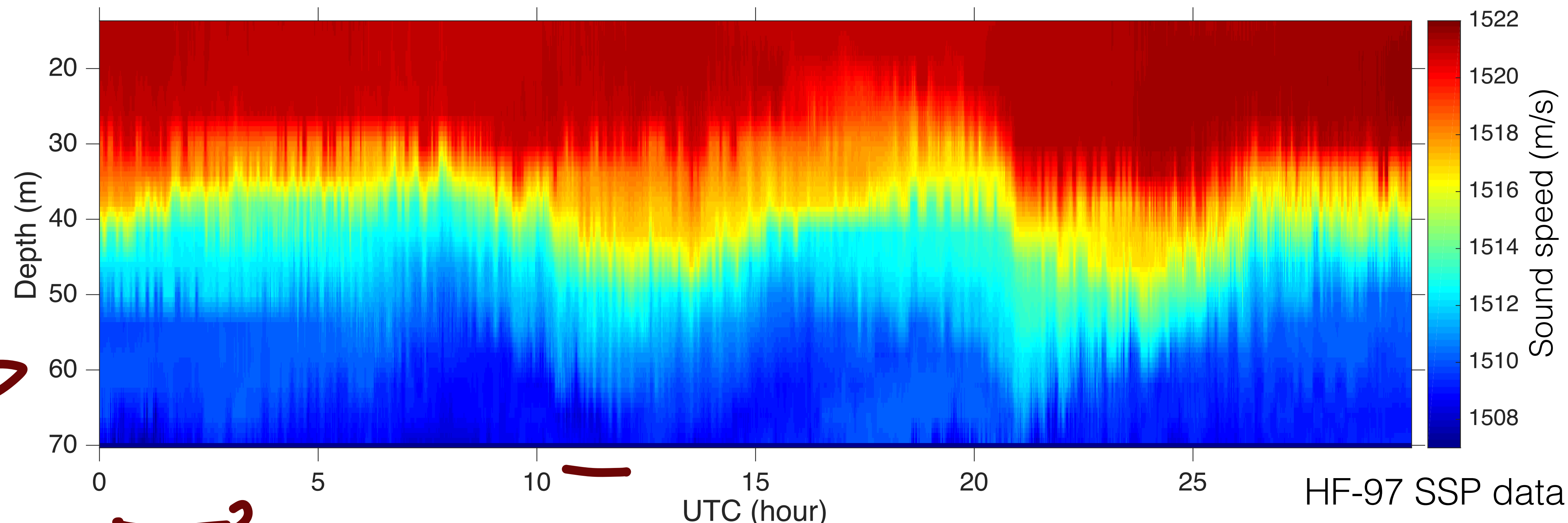
Dictionary learning of ocean sound speed profiles

Bianco and Gerstoft 2017

- Acoustic observations from ocean contain information about ocean environment
- The inversion of environment parameters is limited by physics and signal processing assumptions



Sound speed profiles

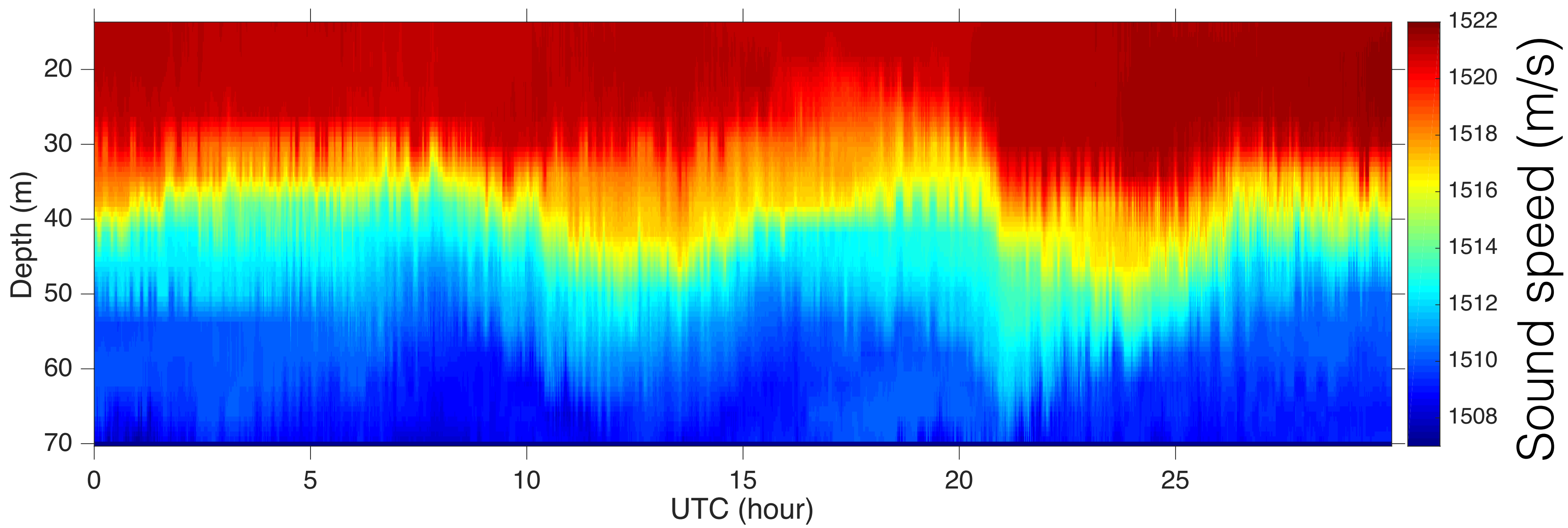


HF-97 SSP data

- Sound speed profiles (SSPs) in the ocean are often highly variable with fine scale fluctuations
- Acoustic inversion of SSPs is ill-posed and traditionally regularized using EOFs (=PCA in this case)
- Dictionaries obtained via unsupervised learning may provide better representation of SSP dynamics

Dictionary learning of sound speed profiles

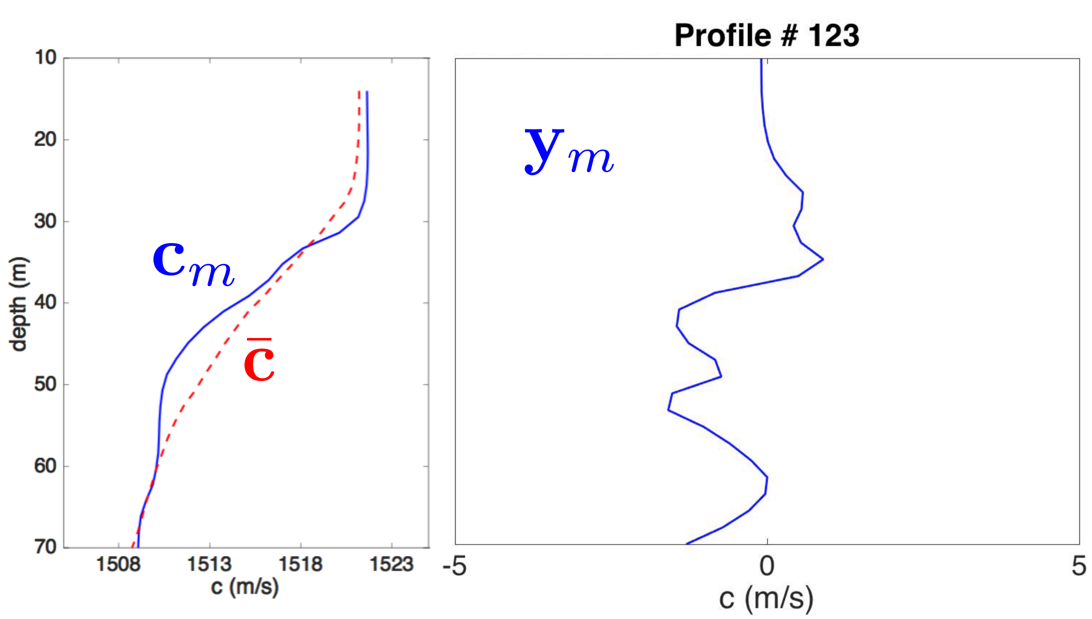
Bianco and Gerstoft JASA 2017 (published)



HF-97 Experiment

- 30 hours of SSP data
- Used 1000 profiles for dictionary learning
- K = 30 point SSP's (interpolated from 15 measurements)

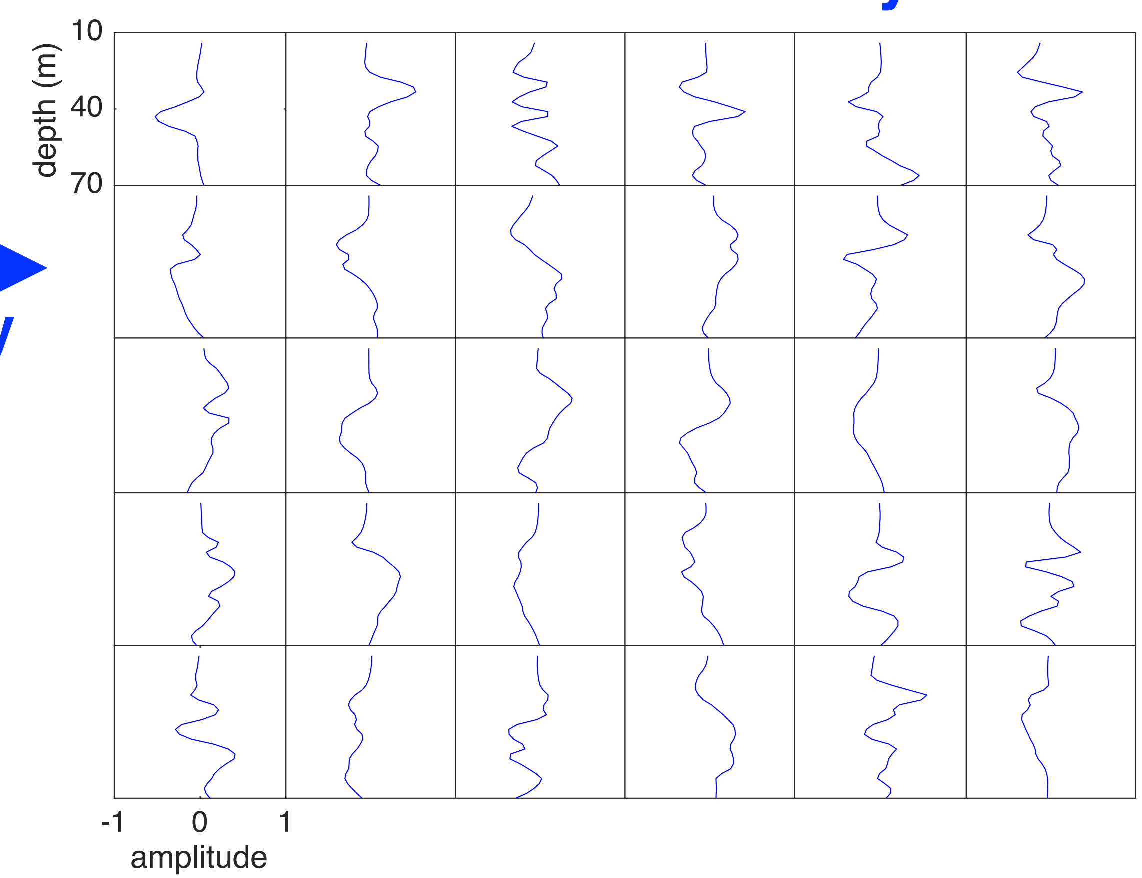
SSP Variability



$$\mathbf{y}_m = \mathbf{c}_m - \bar{\mathbf{c}}$$

Dictionary Learning

'Learned Dictionary'



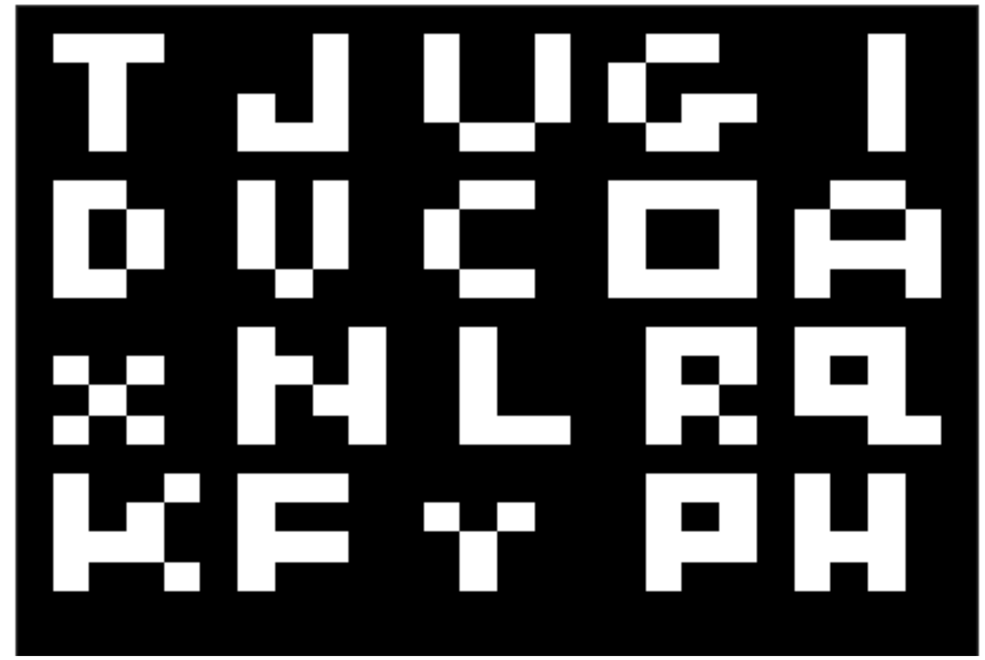
Example: Denoising alphabet with K-SVD algorithm

20 True alphabet

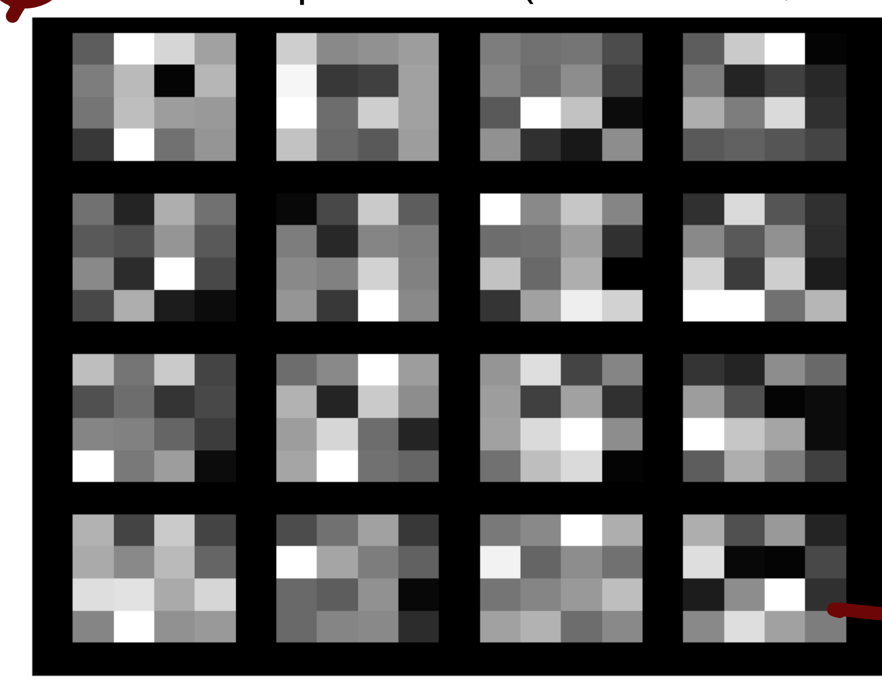
16  y_i

20

20 Recovered alphabet (no noise, K-SVD)

 Φ

16 Recovered alphabet (no noise, PCA)

Φ  16

20 Recovered alphabet (noise std = .5, K-SVD)

$\Phi \in {}^{M\times N}$

$M<N$

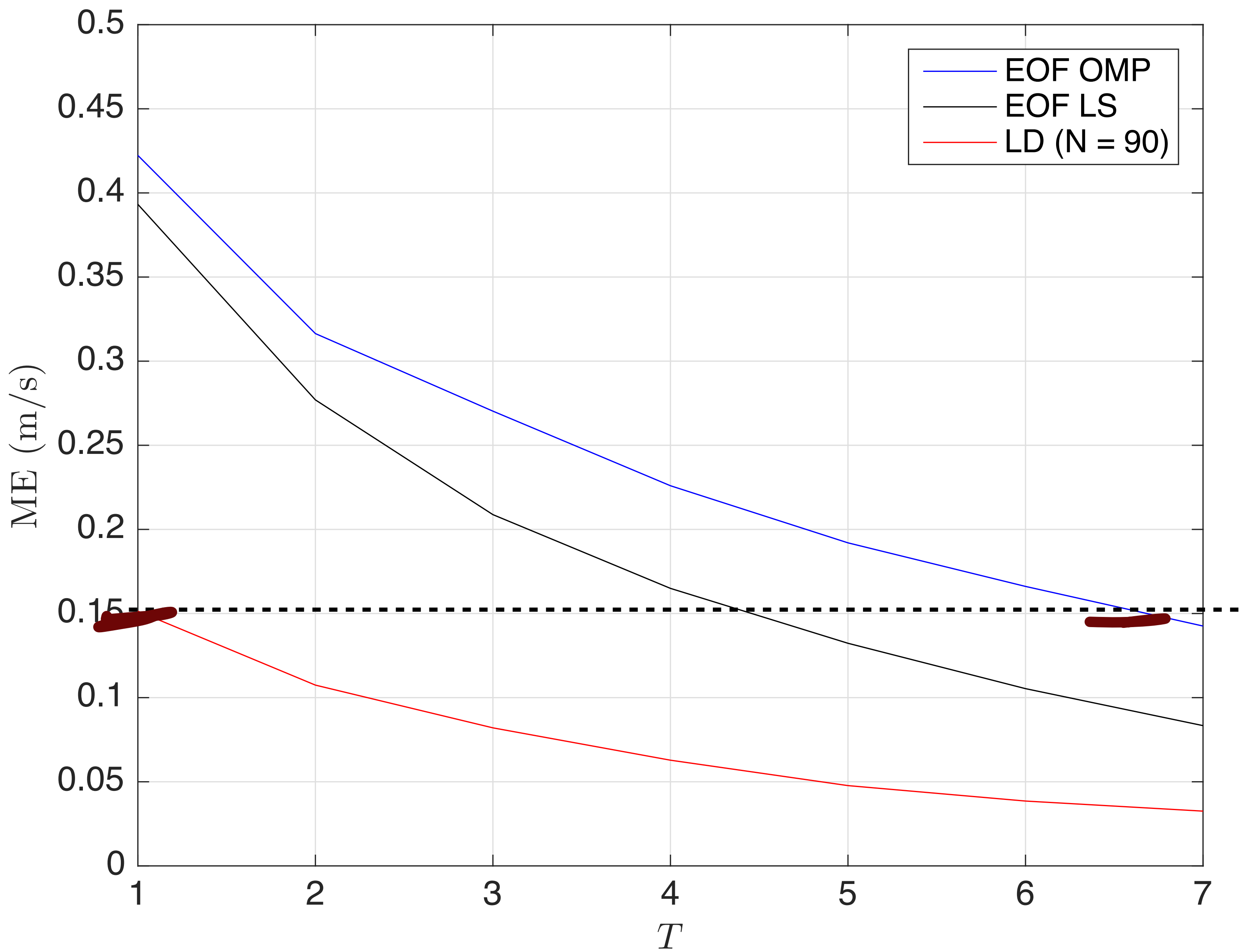
1

$Y = [y_1 \cdots y_I]$

$\|Y - \Phi X\|_2^2 \le \epsilon \quad \|x_i\| = T \; \forall i$

SSP reconstruction error using Dictionary Learning

Based on 1000 profiles from HF-97



LS: Least squares
OMP: Sparse processor

Mean Error (ME):

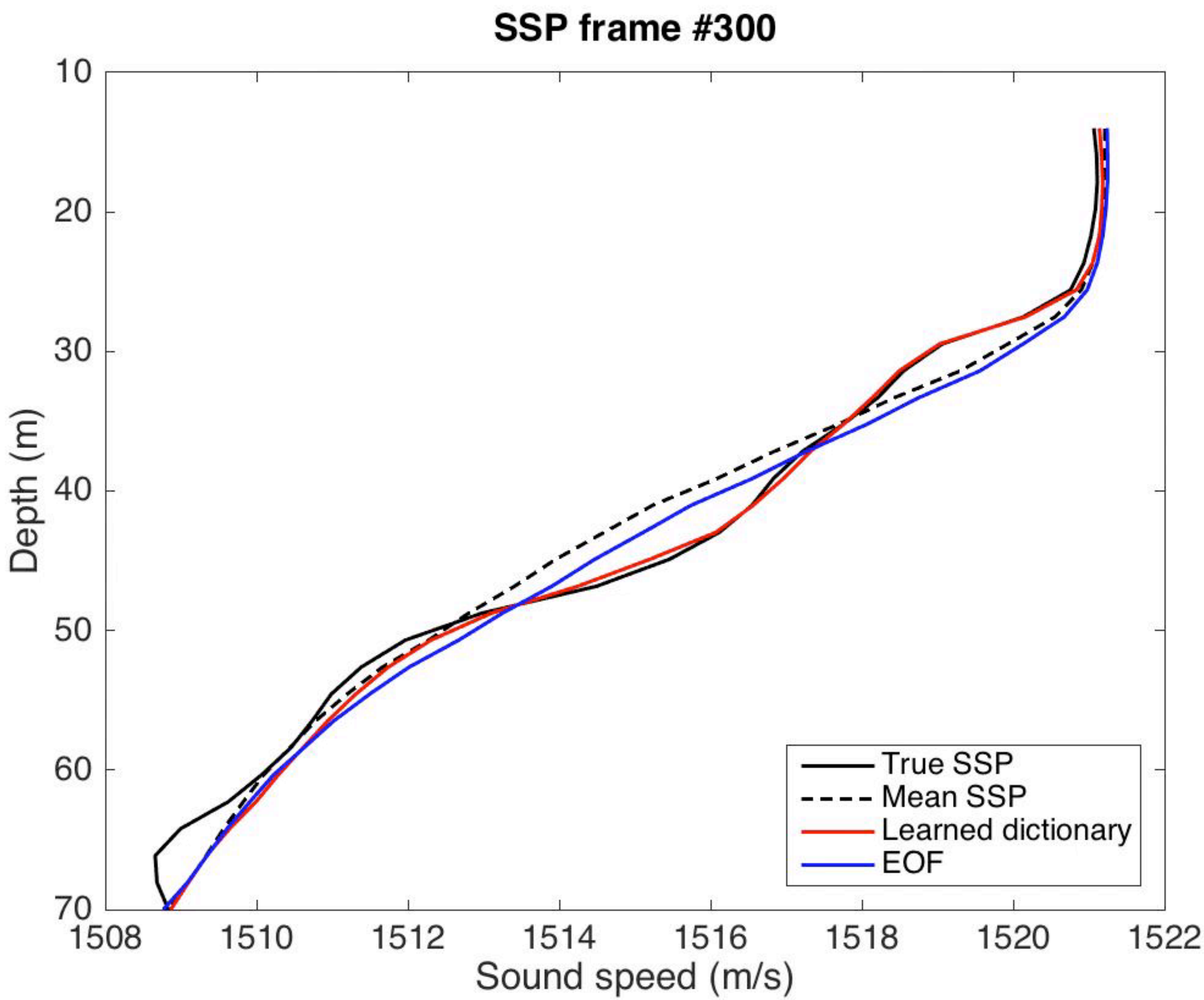
$$\mathrm{ME} = \frac{1}{KM}\|\mathbf{Y} - \widehat{\mathbf{Y}}\|_1$$

"compression tech."

- One entry from Learned Dictionary fits SSP data better than 6 EOFs
- Learned dictionary (LD) reconstruction error less than 50% of EOF error

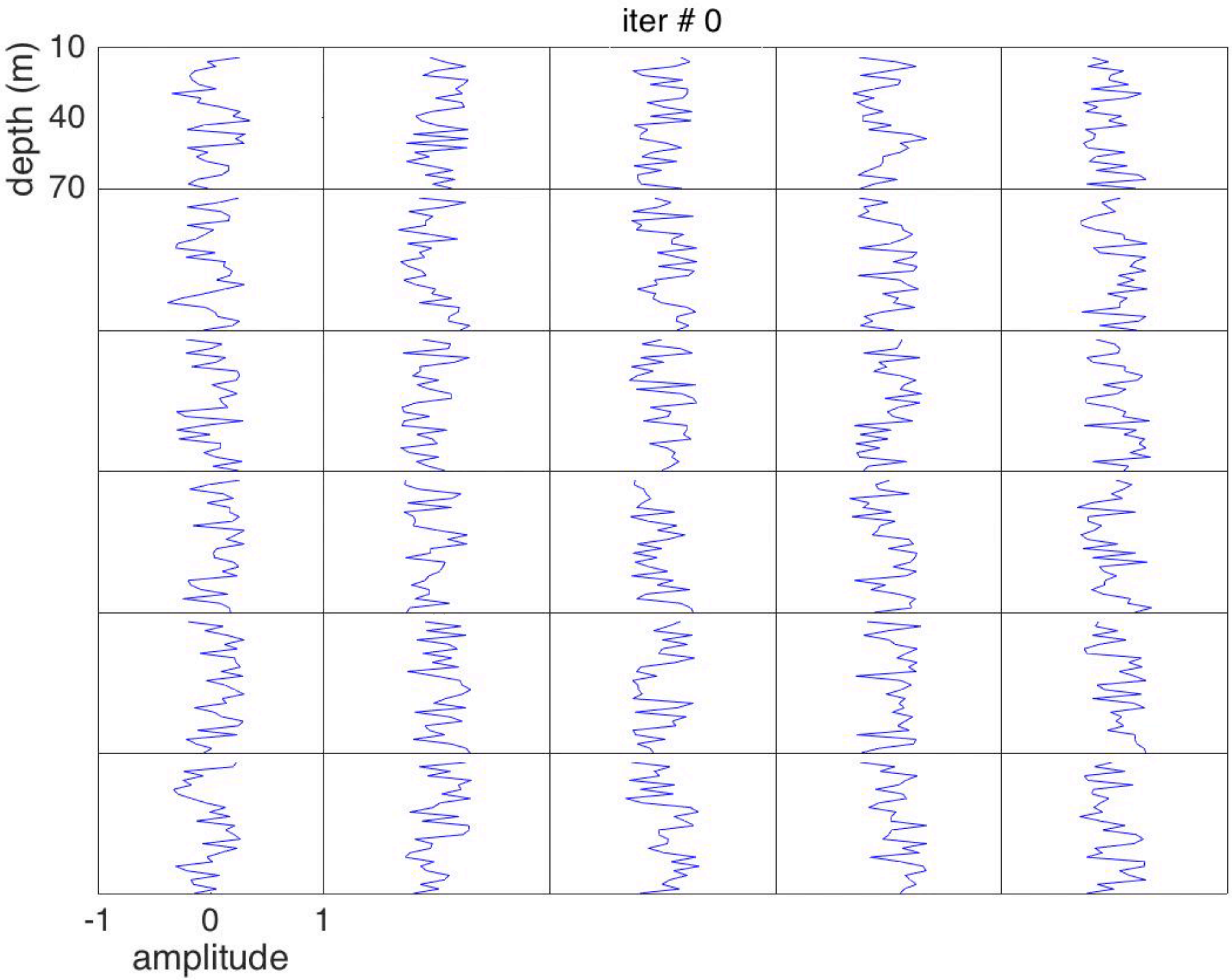
SSP reconstruction using Dictionary Learning

HF-97: One coefficient from Learned Dictionary vs. One EOF coefficient



Learning dictionary from HF-97 SSP variation

Q random initialized, converges within 15 iterations



LD solution space much smaller than EOFs

Inversion for SSP:

Assuming a potentially non-linear mapping:

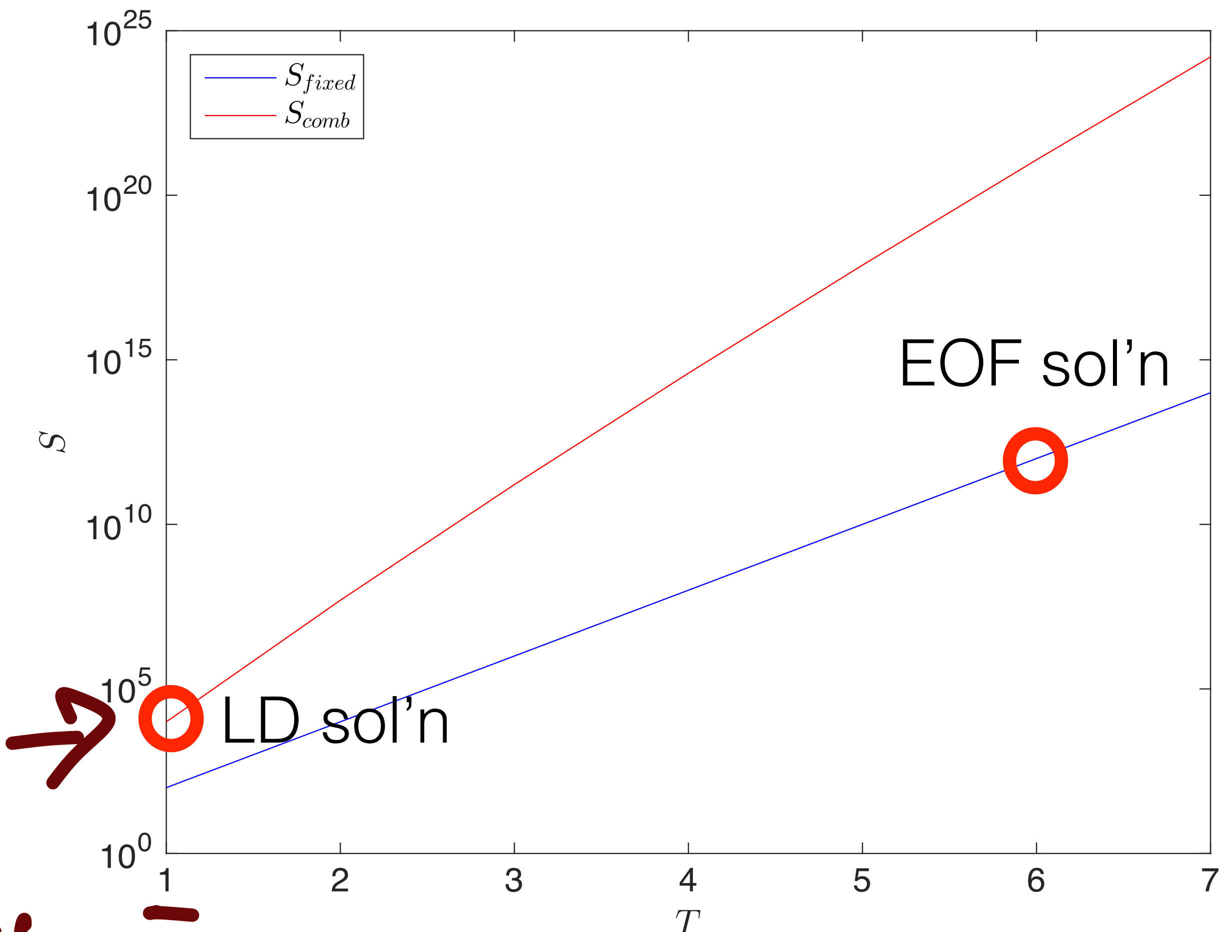
- EOF solution: T leading order coefficients (fixed indices)

$$S_{\text{fixed}} = H^T \quad 6 \qquad 100^6$$

- LD solution: T-non-zero coefficients (combinatorial indices)

$$S_{\text{comb}} = H^T \frac{N!}{T!(N-T)!}$$

$\|x\|_0$ – "combinatorial"



- Since 6 EOFs or 1 LD entry required, if coefficients discretized in H=100 coefficients number of possible solutions are

EOFs: $S_{fixed} = 10^{12}$ solutions LD: $S_{comb} = 10^4$ solutions

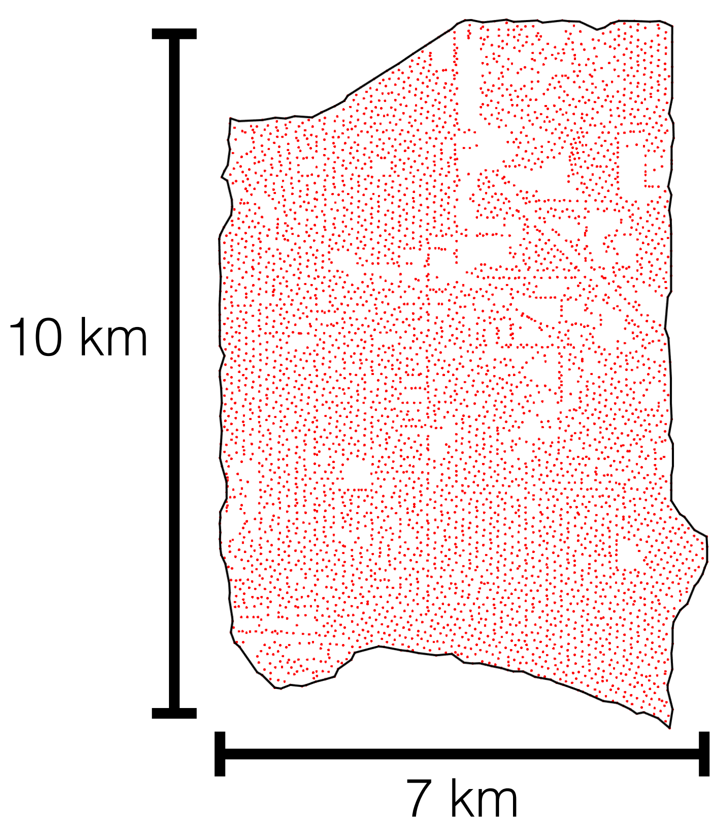
10^{12}

Dictionary learning in travel time tomography

Bianco and Gerstoft 2018

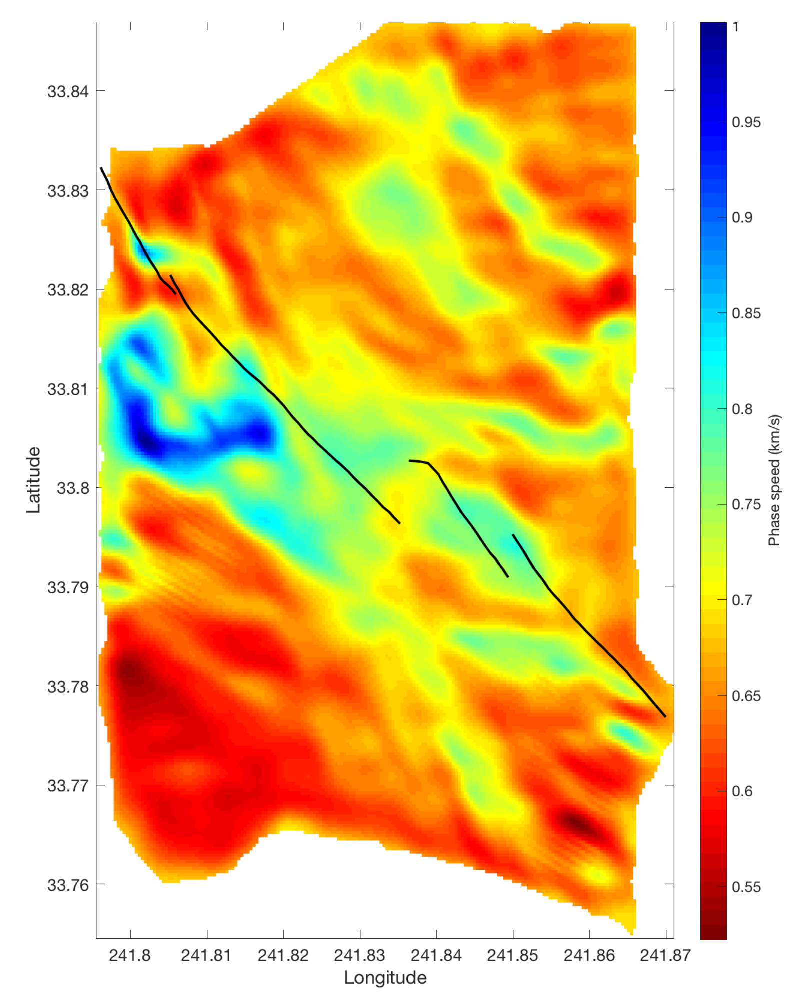
- The Earth contains both smooth and discontinuous variations in slowness (e.g. Moho, faults) at multiple spatial scales
- Most existing travel time inversion methods are ad hoc: regularize inversion assuming exclusively smooth or discontinuous slownesses
- Propose locally-sparse 2D travel time tomography (LST) method with three main ingredients:
    - Sparsity constraint on slowness patches
    - Dictionary learning (unsupervised machine learning)
    - Damped least squares regularization on overall slowness map

Long Beach seismic array
(5200 stations, ~14 million rays)



LST →

1 Hz Rayleigh wave phase speed map



Consider simple travel time model

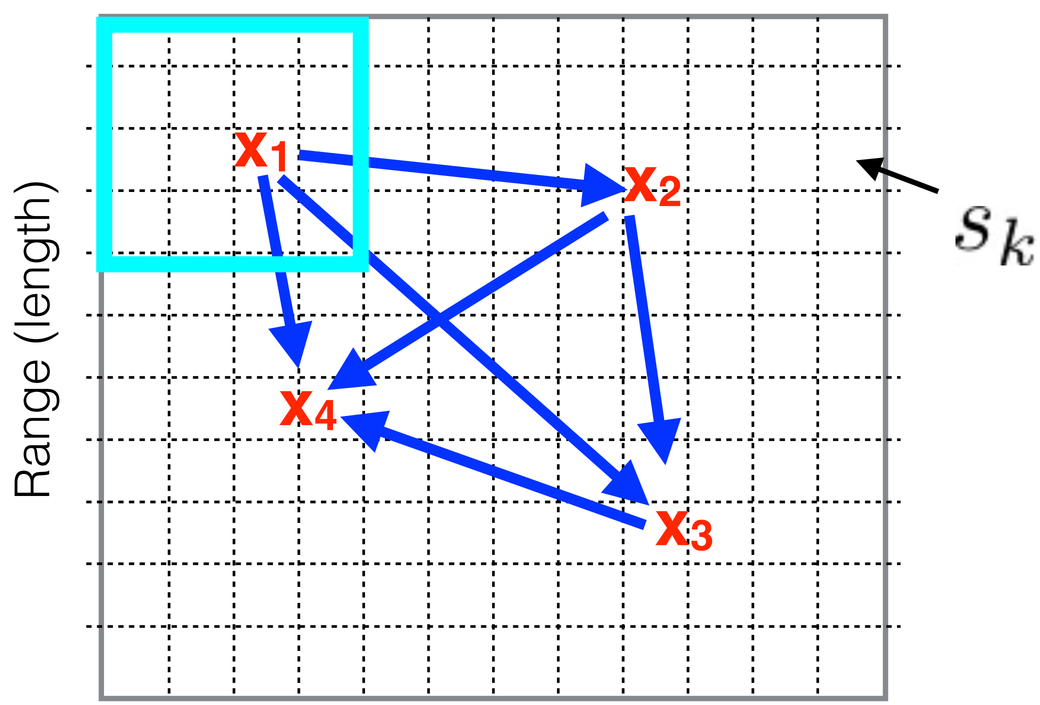
For slowness field, get travel time:

$$t = \frac{d}{c} = ds$$

$c =$ wave speed

$s =$ slowness

2D map slowness map



For straight-rays get simple formulation:

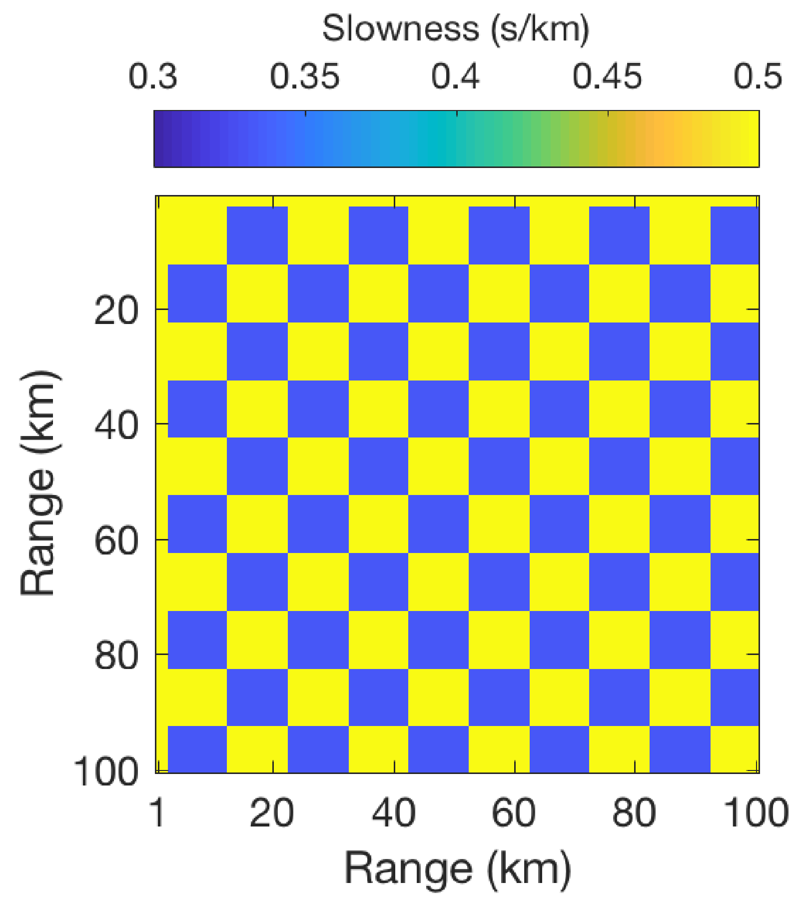
$$\begin{bmatrix} t_{12} \\ \vdots \\ t_{ij} \end{bmatrix} = \begin{bmatrix} \delta r_{12,1} & \dots & \delta r_{12,k} \\ \vdots & \ddots & \vdots \\ \delta r_{12,k} & \dots & \delta r_{ij,k} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_k \end{bmatrix} \longrightarrow \boxed{\mathbf{t} = \mathbf{A}\mathbf{s}}$$

"tomography matrix" $\mathbf{A}$

- Propose LST tomography ingredients:
    - Sparsity constraint on slowness patches
    - Dictionary learning (unsupervised machine learning)
    - Damped least squares regularization on overall slowness map

Proposed locally-sparse tomography (LST) basics

Synthetic "checkerboard" slowness example



LST approach three ingredients: classified as **local** and **global** models

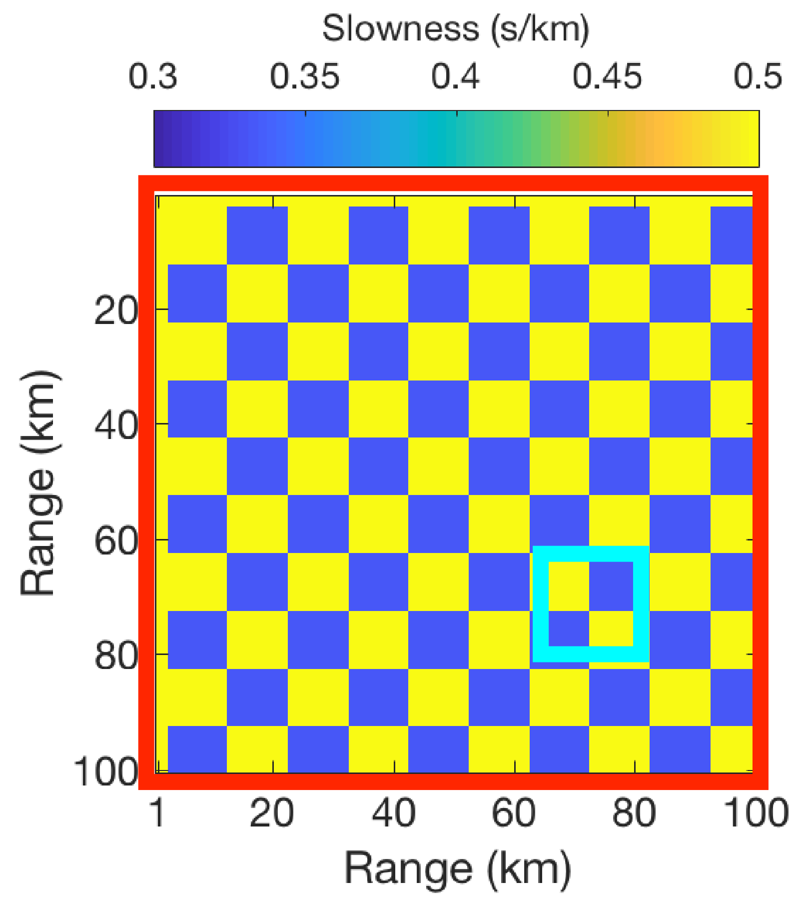
"Local" model	1.	Sparsity constraint on slowness patches
	2.	Dictionary learning (unsupervised machine learning)
"Global" model	3.	Damped least squares regularization on overall slowness map

"Local" model: Models small-scale features as patches

"Global" model: Models larger-scale features with damped least squares

Proposed locally-sparse tomography (LST) basics

Synthetic "checkerboard" slowness example



LST approach three ingredients: classified as **local** and **global** models

"Local" model	1.	Sparsity constraint on slowness patches
	2.	Dictionary learning (unsupervised machine learning)
"Global" model	3.	Damped least squares regularization on overall slowness map

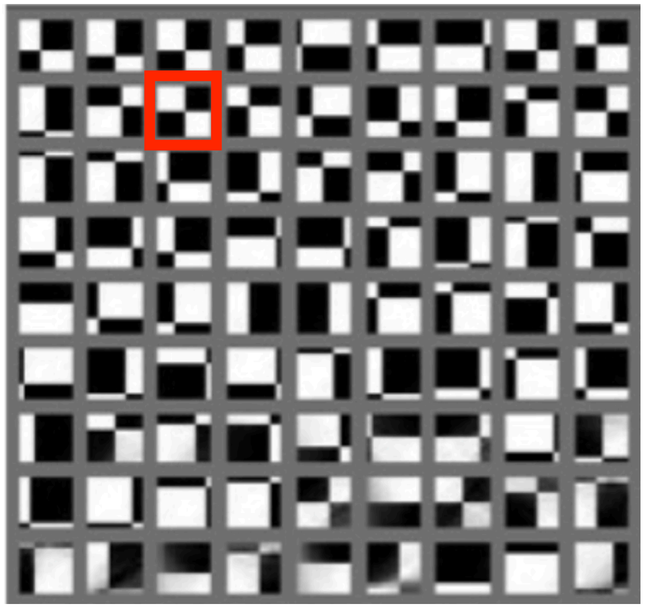
"Local" model: Models small-scale features as patches

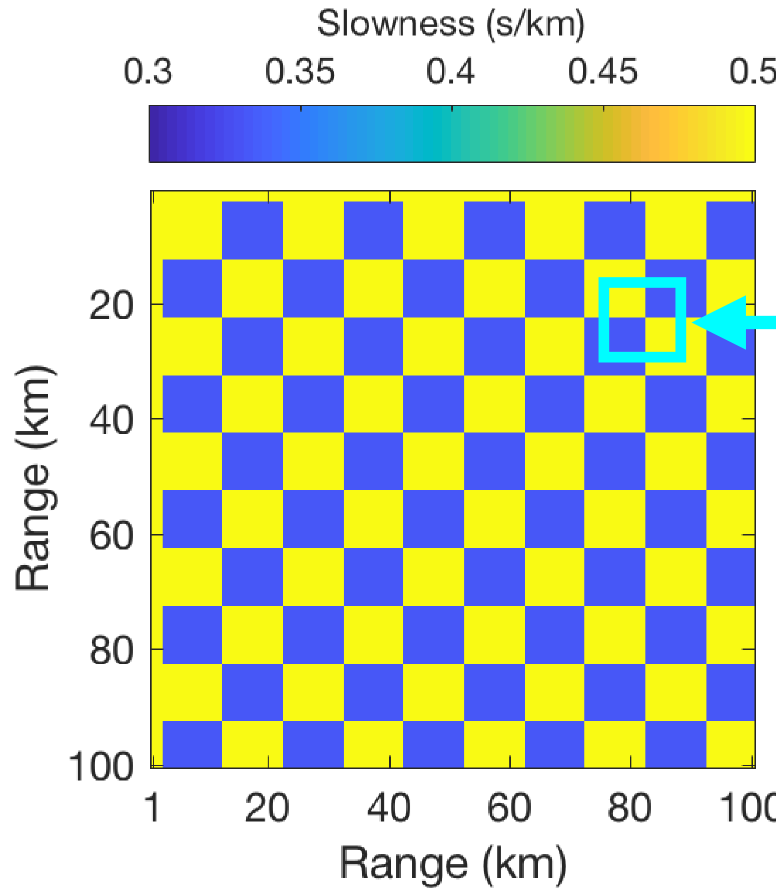
"Global" model: Models larger-scale features with damped least squares

Local model: slowness patches related to dictionary entries

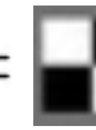
Dictionary $\mathbf{D}$

Slowness





$$\mathbf{y} = \mathbf{R}_i\mathbf{s} = \mathbf{D}\mathbf{x}_i$$

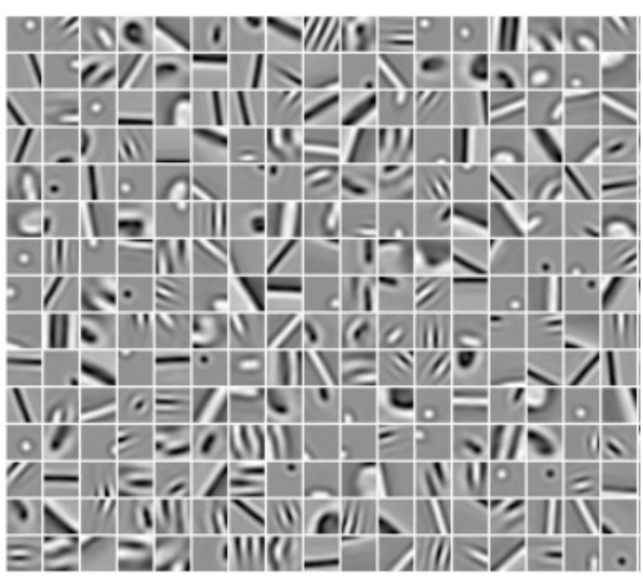
$\mathbf{R}_i\mathbf{s} =$  x_i

10x10 pixel patches

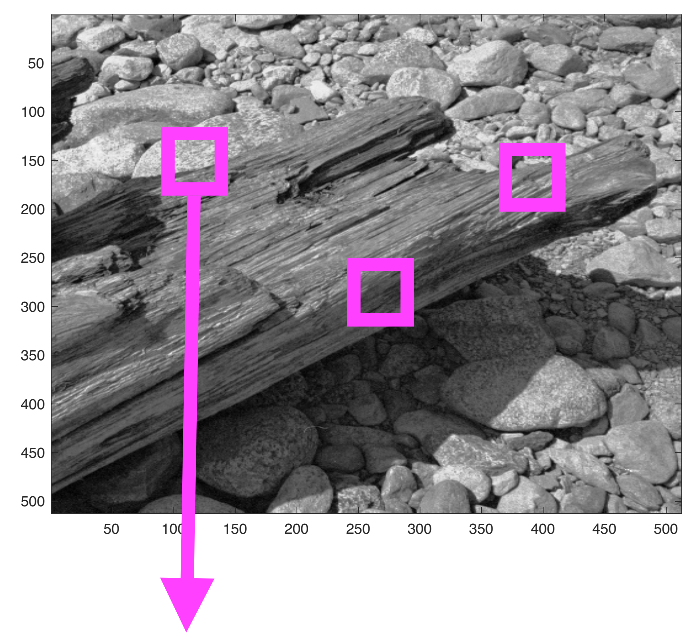
$$\mathbf{D} \in \mathbb{R}^{n\times Q}$$

$$\mathbf{R}_i \in \{0,1\}^{n\times N}$$

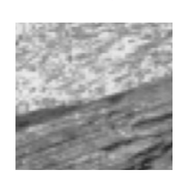
Natural image



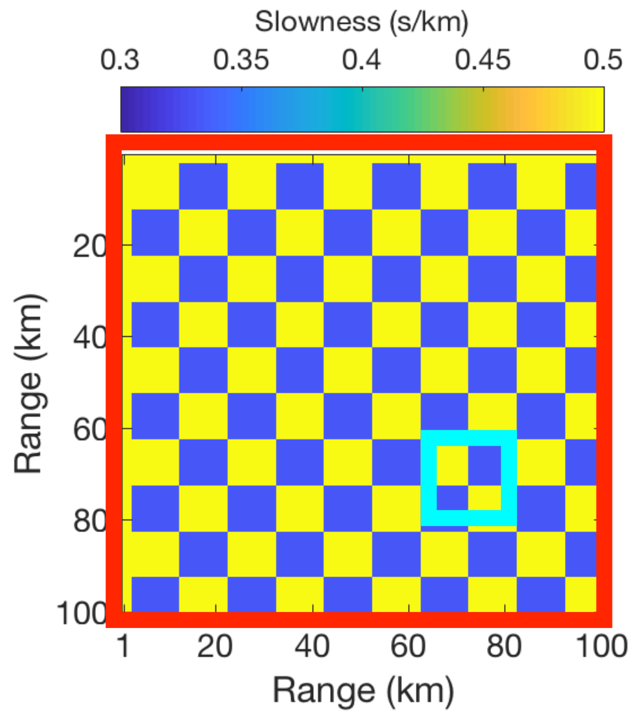
Olshausen 2009

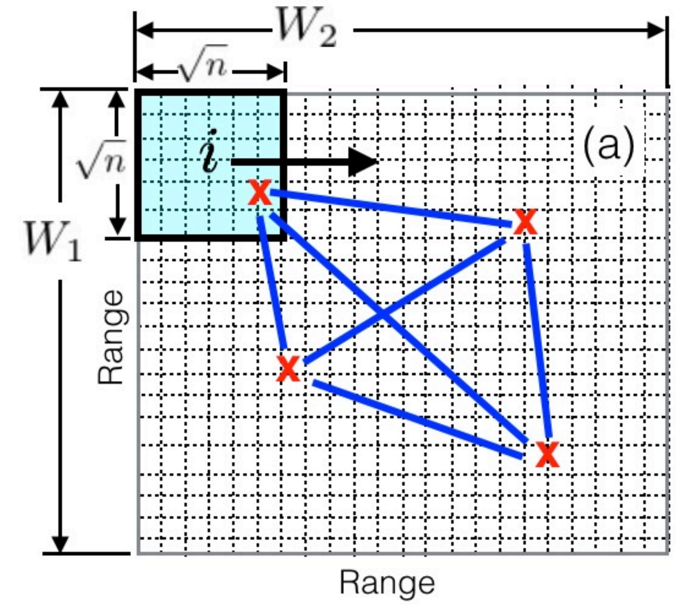


$$\widehat{\mathbf{x}}_i = \arg\min_{\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T$$

$\mathbf{y} =$  $=$  $x_1 +$  $x_2 + \ldots$

LST slowness image and sampling





Pixels and “patches”

x’s are stations

Slowness map and sampling:

- Discrete slowness map $N = W_1 \times W_2$ pixels
- I overlapping $\sqrt{n} \times \sqrt{n}$ pixel patches
- M straight-ray paths

Tomography matrix (straight ray) $\mathbf{A} \in \mathbb{R}^{M \times N}$

Slowness dictionary $\mathbf{D} \in \mathbb{R}^{n \times Q}$

$Q \ll I$

“Local” model $\widehat{\mathbf{x}}_i = \underset{\mathbf{x}_i}{\arg\min} \|\mathbf{R}_i \mathbf{s}_\mathrm{s} - \mathbf{D}\mathbf{x}_i\|_2^2 \text{ subject to } \|\mathbf{x}_i\|_0 = T$

“Global” model $\mathbf{t} = \mathbf{A}\mathbf{s}_\mathrm{g} + \epsilon$ $\widehat{\mathbf{s}}_\mathrm{g} = \underset{\mathbf{s}_\mathrm{g}}{\arg\min} \|\mathbf{t} - \mathbf{A}\mathbf{s}_\mathrm{g}\|_2^2 + \lambda_1 \|\mathbf{s}_\mathrm{g} - \mathbf{s}_\mathrm{s}\|_2^2,$

Formulation of LST and algorithm

Bayesian MAP objective:

$$\{\widehat{\mathbf{s}}_{\mathrm{g}}, \widehat{\mathbf{s}}_{\mathrm{s}}, \widehat{\mathbf{X}}\} = \underset{\mathbf{s}_{\mathrm{g}}, \mathbf{s}_{\mathrm{s}}, \mathbf{X}}{\arg\min} \left\{ \frac{1}{\sigma_\epsilon^2} \|\mathbf{t} - \mathbf{A}\mathbf{s}_{\mathrm{g}}\|_2^2 + \frac{1}{\sigma_{\mathrm{g}}^2} \|\mathbf{s}_{\mathrm{g}} - \mathbf{s}_{\mathrm{s}}\|_2^2 + \frac{1}{\sigma_{p,i}^2} \sum_i \|\mathbf{D}\mathbf{x}_i - \mathbf{R}_i \mathbf{s}_{\mathrm{s}}\|_2^2 \right\}$$

$$\text{subject to } \|\mathbf{x}_i\|_0 = T \ \forall\ i.$$

Solution via block-coordinate descent

- **Global model:** the global slowness is solved as

$$\widehat{\mathbf{s}}_{\mathrm{g}} = \underset{\mathbf{s}_{\mathrm{g}}}{\arg\min}\ \|\mathbf{t} - \mathbf{A}\mathbf{s}_{\mathrm{g}}\|_2^2 + \lambda_1 \|\mathbf{s}_{\mathrm{g}} - \mathbf{s}_{\mathrm{s}}\|_2^2, \quad \lambda_1 = (\sigma_\epsilon / \sigma_{\mathrm{g}})^2$$

- **Local model:** sparse coding and dictionary learning, decoupled from MAP objective

$$\widehat{\mathbf{x}}_i = \underset{\mathbf{x}_i}{\arg\min}\ \|\mathbf{D}\mathbf{x}_i - \mathbf{R}_i \widehat{\mathbf{s}}_{\mathrm{g}}\|_2^2 \ \text{ subject to } \|\mathbf{x}_i\|_0 = T \qquad (\mathbf{s}_{\mathrm{s}} = \widehat{\mathbf{s}}_{\mathrm{g}})$$

Dictionary learning by iterative thresholding and signed K-means (ITKM) algorithm, Schnass 2015

$$\max_{\mathbf{D}} \sum_i \max_{|K|=T} \|\mathbf{D}_K^{\mathrm{T}} \mathbf{y}_i\|_1,$$

- The sparse slowness is then solved from

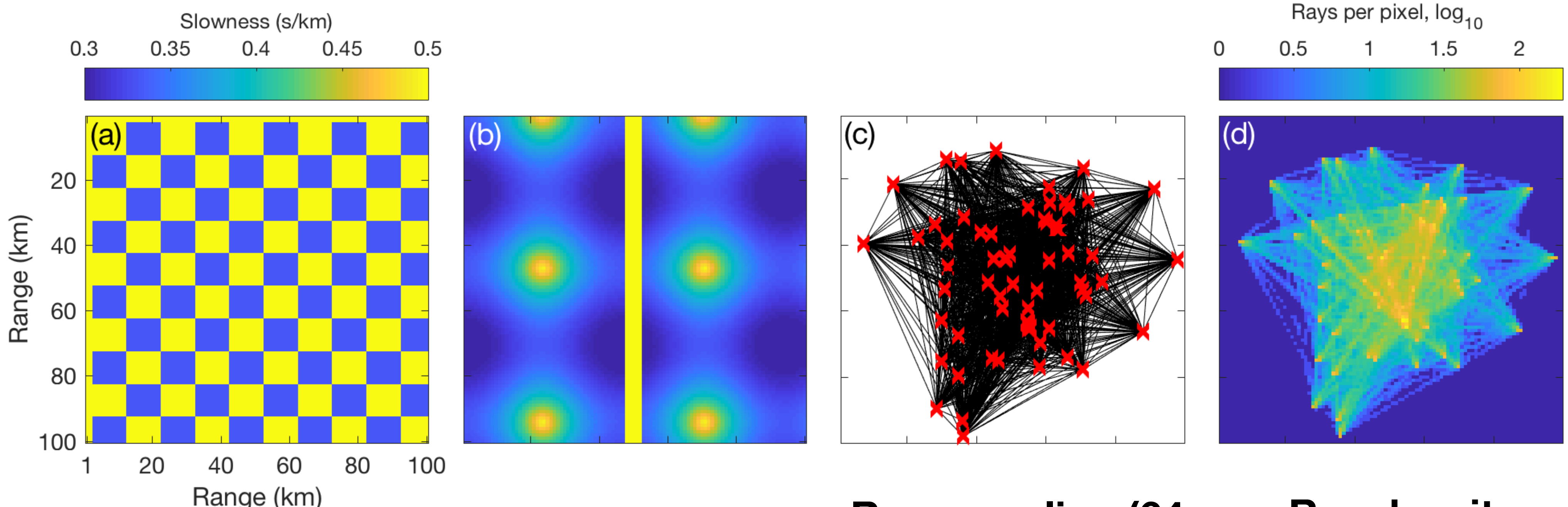
$$\widehat{\mathbf{s}}_{\mathrm{s}} = \underset{\mathbf{s}_s}{\arg\min}\ \lambda_2 \|\widehat{\mathbf{s}}_{\mathrm{g}} - \mathbf{s}_{\mathrm{s}}\|_2^2 + \sum_i \|\mathbf{D}\widehat{\mathbf{x}}_i - \mathbf{R}_i \mathbf{s}_{\mathrm{s}}\|_2^2, \quad \lambda_2 = (\sigma_p / \sigma_{\mathrm{g}})^2$$

"Slowness at pixel n"

$$\widehat{\mathbf{s}}_{\mathrm{s}} = \left(\lambda_2 \mathbf{I} + \sum_i \mathbf{R}_i^{\mathrm{T}} \mathbf{R}_i\right)^{-1} \left(\lambda_2 \widehat{\mathbf{s}}_{\mathrm{g}} + \sum_i \mathbf{R}_i^{\mathrm{T}} \mathbf{D}\widehat{\mathbf{x}}_i\right) \longrightarrow \widehat{s}_{\mathrm{s,n}} = \frac{\lambda_2 \widehat{s}_{\mathrm{g,n}} + b_{\mathrm{n}} s_{p,\mathrm{n}}}{\lambda_2 + b_{\mathrm{n}}}$$

$$\mathbf{b} = \mathrm{diag}\left(\textstyle\sum_i \mathbf{R}_i^{\mathrm{T}} \mathbf{R}_i\right) \in \mathbb{Z}^N \qquad s_{p,\mathrm{n}} = p_{\mathrm{n}} / b_{\mathrm{n}}$$

Synthetic slownesses and dictionaries



Checkerboard

"Fault" profile

Ray sampling (64 stations, 2016 rays)

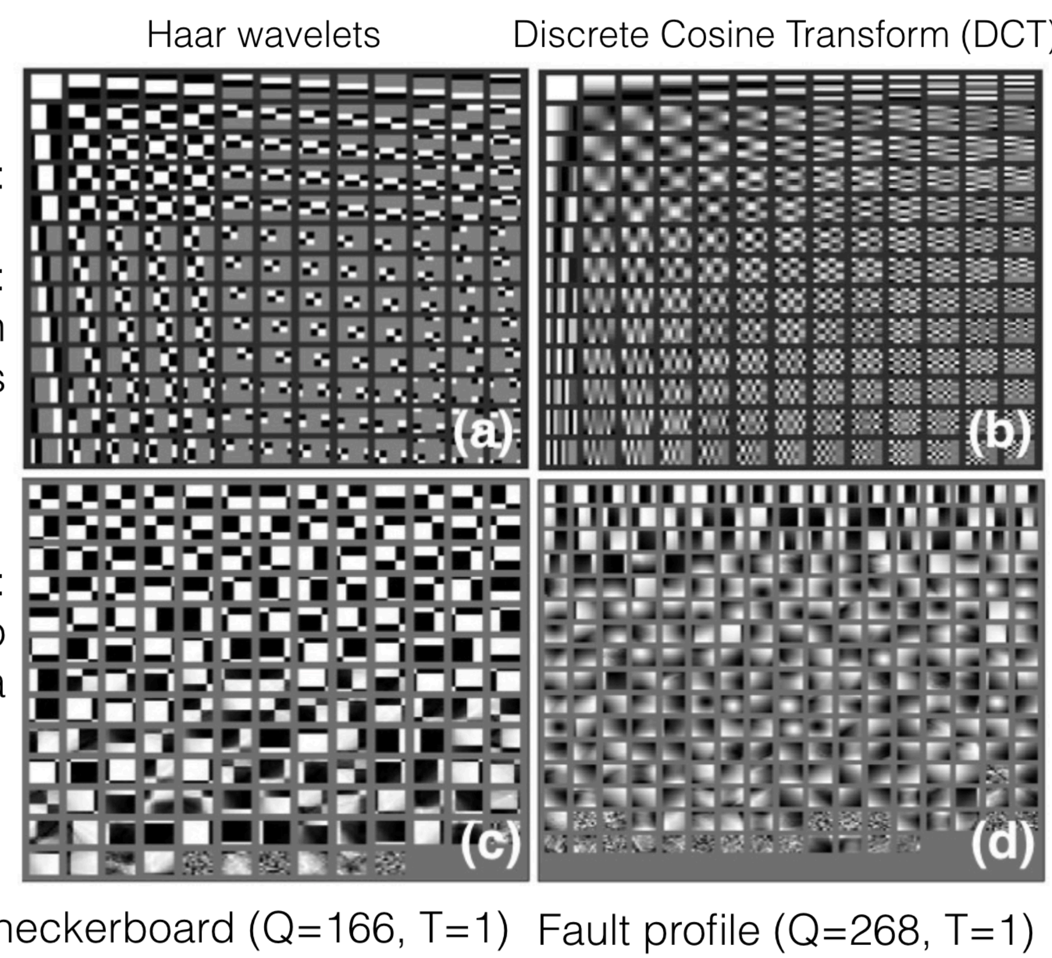
Ray density

Dictionaries: Prescribed and Learned

Dictionaries:

'Generic': Perform well in many tasks

'Learned': Adapted to specific data



LST vs. conventional method: synthetic inversions without noise

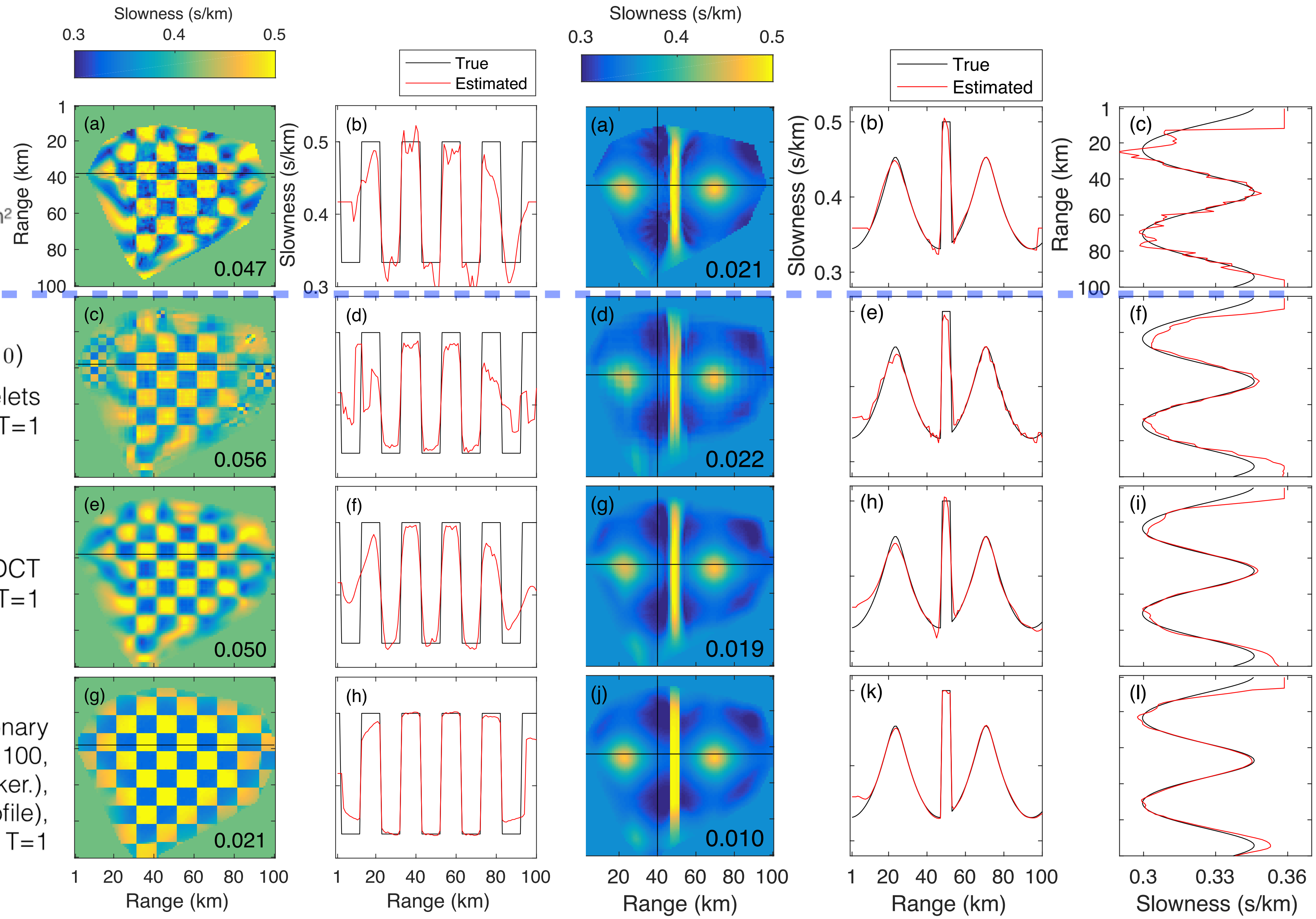
Each example took ~5 min on MacBook Pro

Conventional
L=10 km, η = 0.1 km²

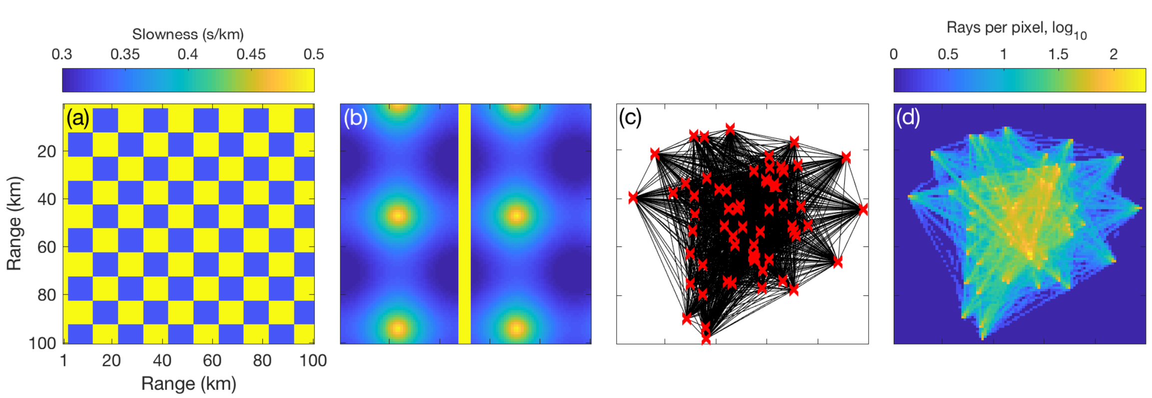
LST ($\lambda_1 = \lambda_2 = 0$)
Haar wavelets
n=64, Q=169, T=1

DCT
n=64, Q=169, T=1

Learned dictionary
n=100,
Q=166 (checker.),
Q=268 (fault profile),
T=1



Slowness RMSE (s/km) written on 2D estimates



Conventional tomography method
(Rodgers 2000)

$$\hat{\mathbf{s}}_{\mathrm{g}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \eta\boldsymbol{\Sigma}_L^{-1}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{t}, \quad \boldsymbol{\Sigma}_L(i,j) = \exp\left(-D_{i,j}/L\right)$$

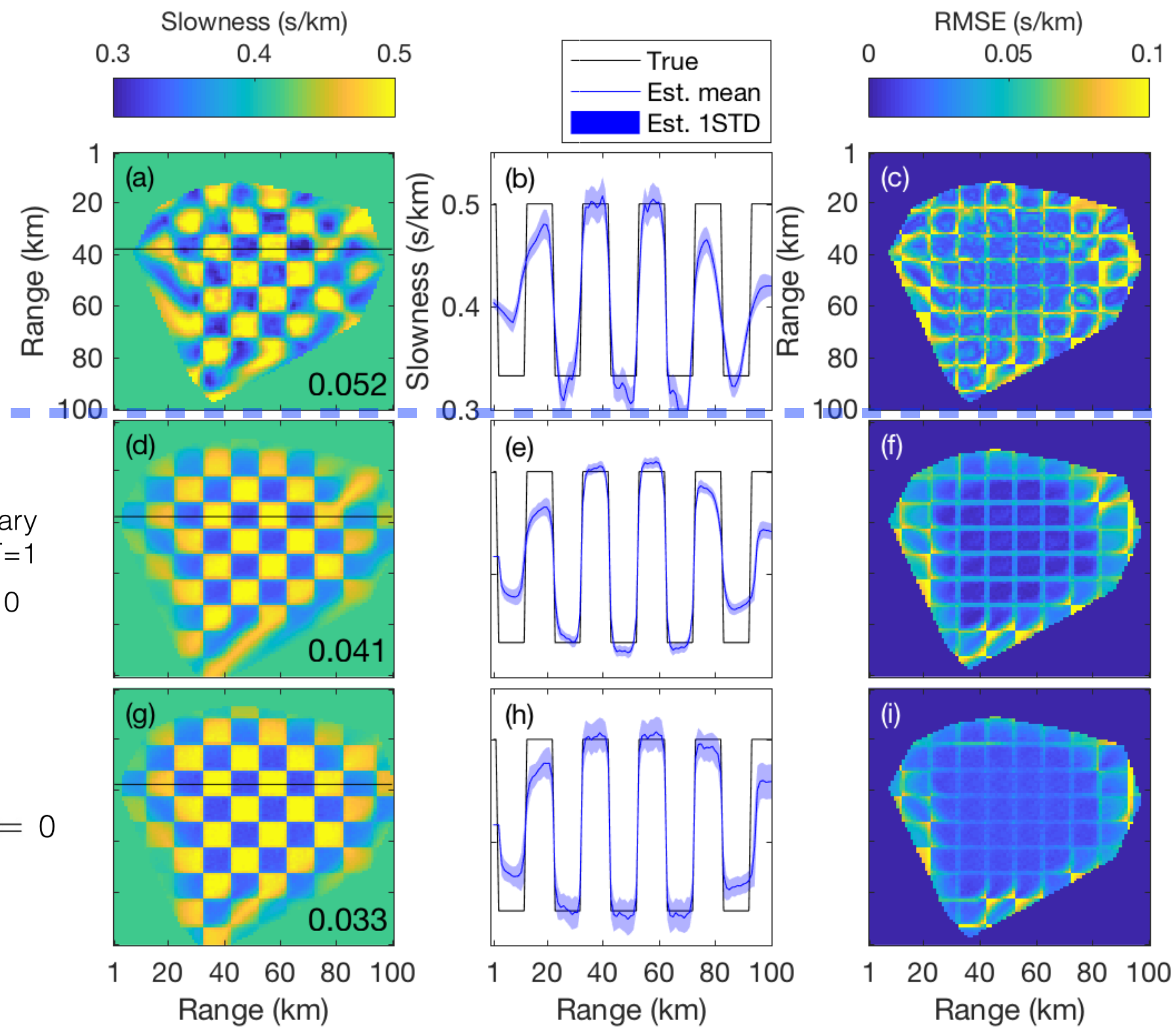
with $\eta = (\sigma_\epsilon/\sigma_{\mathrm{g}})^2$

LST vs. conventional method: synthetic inversions with travel time noise
Checkerboard

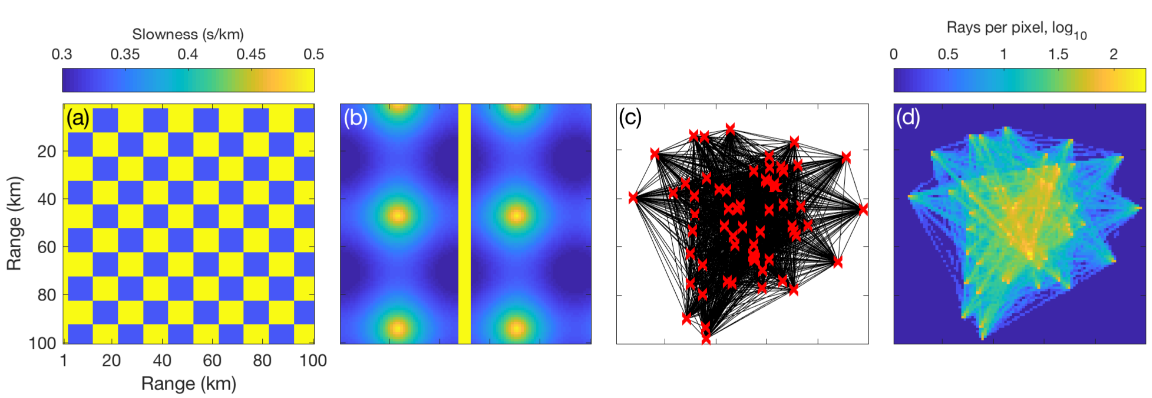
Conventional
L=6 km, $\eta = 10$ km²

LST
Learned dictionary
n=100, Q=144, T=1
$\lambda_1 = 5$ km², $\lambda_2 = 0$

$\lambda_1 = 10$ km², $\lambda_2 = 0$



- Slowness RMSE (s/km) written on 2D estimates
- Noise is Gaussian with STD 2% of mean travel time



Conventional tomography method
(Rodgers 2000)

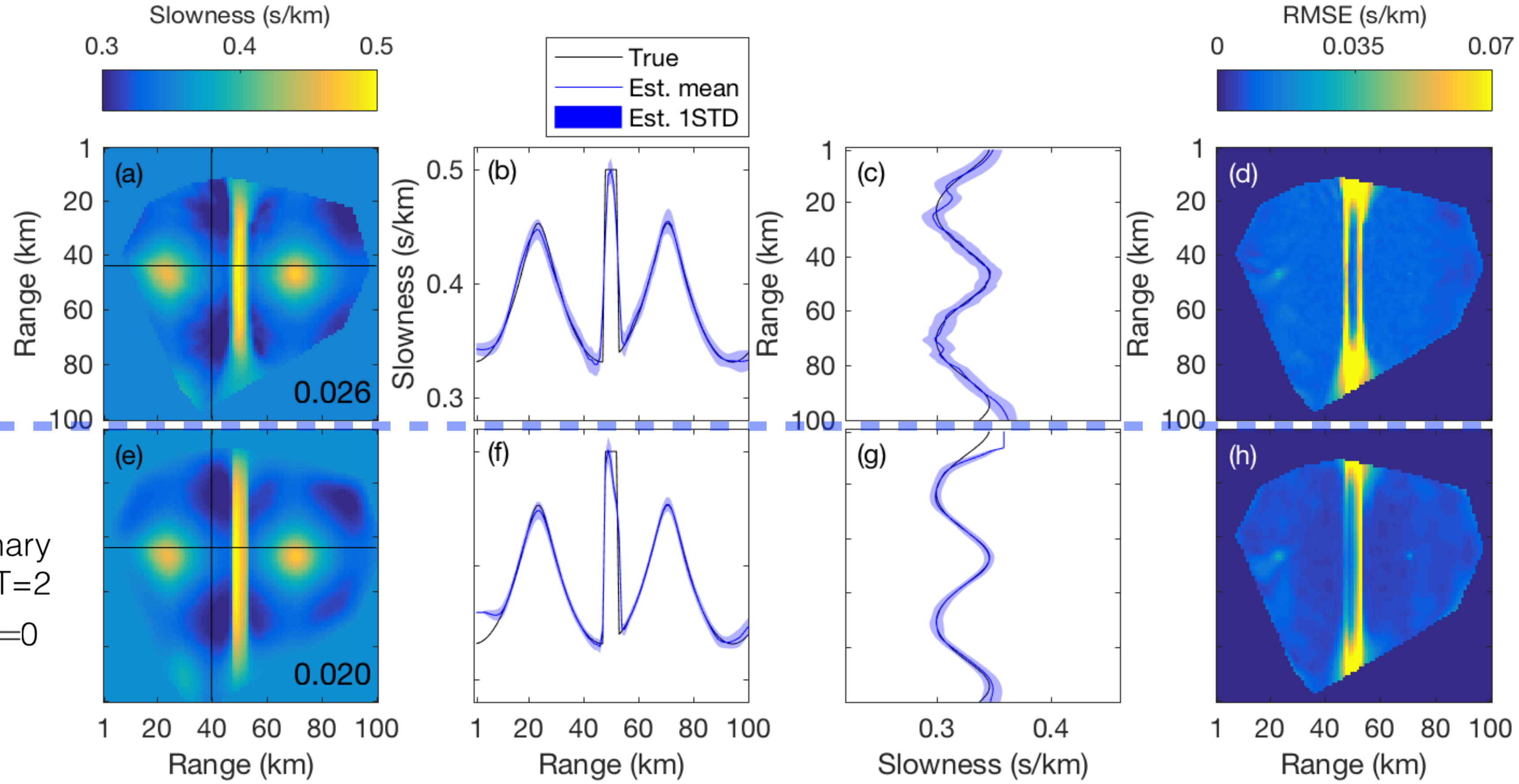
$$\widehat{\mathbf{s}}_{\mathrm{g}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \eta \boldsymbol{\Sigma}_L^{-1}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{t}, \quad \boldsymbol{\Sigma}_L(i,j) = \exp\left(-D_{i,j}/L\right)$$

with $\quad \eta = (\sigma_\epsilon/\sigma_{\mathrm{g}})^2$

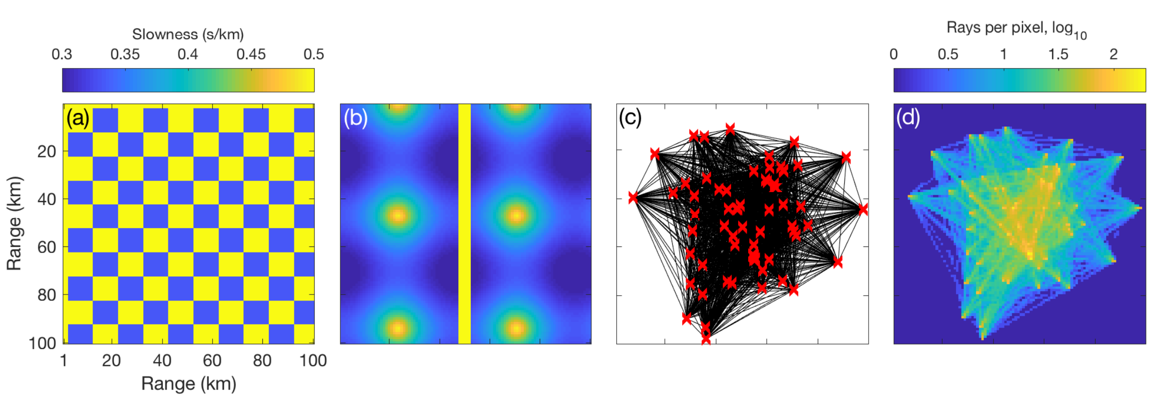
LST vs. conventional method: synthetic inversions with travel time noise
Fault profile

Conventional
L=12 km, η =10 km²

LST
Learned dictionary
n=100, Q=100, T=2
λ_1 = 5 km², λ_2 =0



- Slowness RMSE (s/km) written on 2D estimates
- Noise is Gaussian with STD 2% of mean travel time



Conventional tomography method
(Rodgers 2000)

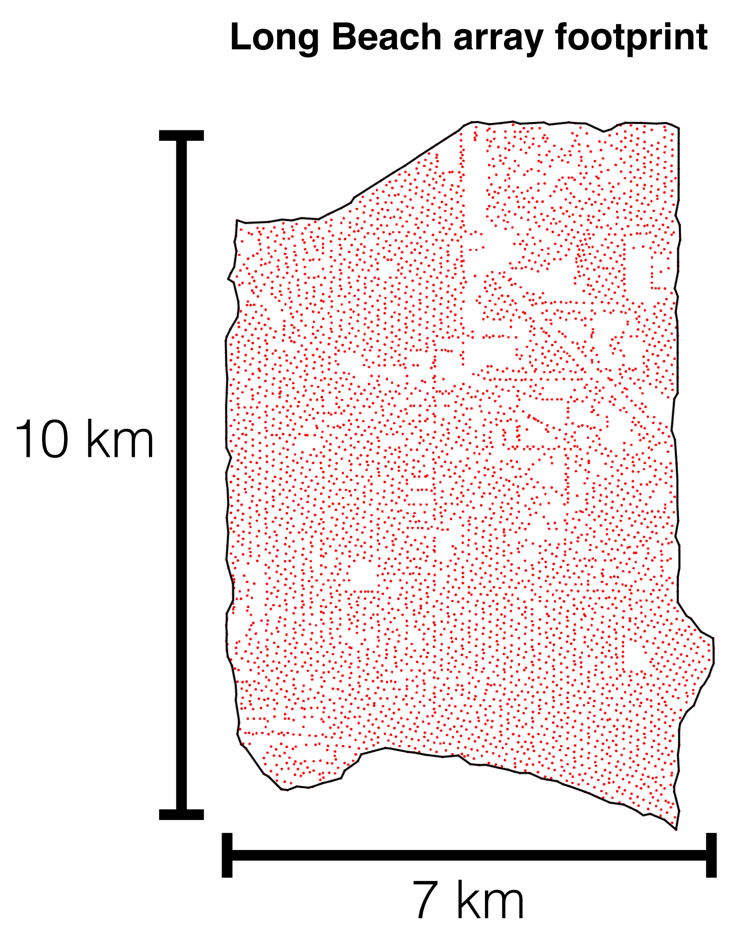
$$\widehat{\mathbf{s}}_{\mathrm{g}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \eta \mathbf{\Sigma}_L^{-1}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{t}, \quad \mathbf{\Sigma}_L(i,j) = \exp\left(-D_{i,j}/L\right)$$

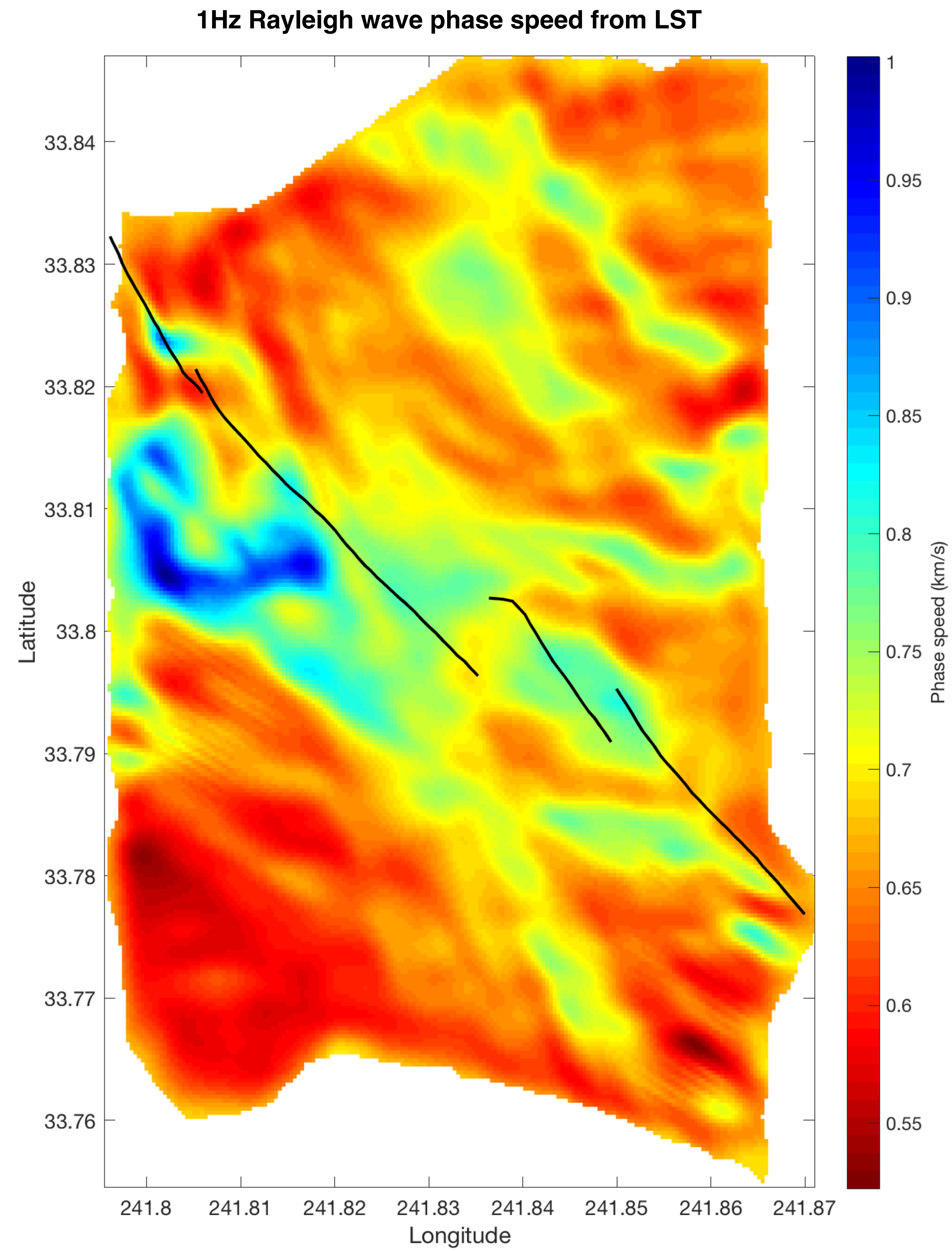
with $\quad \eta = (\sigma_\epsilon/\sigma_{\mathrm{g}})^2$

Imaging Long Beach, CA using LST: Big Data task

- In March 2011, 5200 seismic stations were deployed in Long Beach, California over 70km^2 area
- Ambient seismic noise cross-correlations were obtained for all unique virtual source-receiver pairs (~14 million ray paths) using 3 weeks of data
- We consider only the 1Hz vertical component data, corresponding to Rayleigh surface waves (from Lin et al. 2013)
- After quality control there were ~8 million ray paths

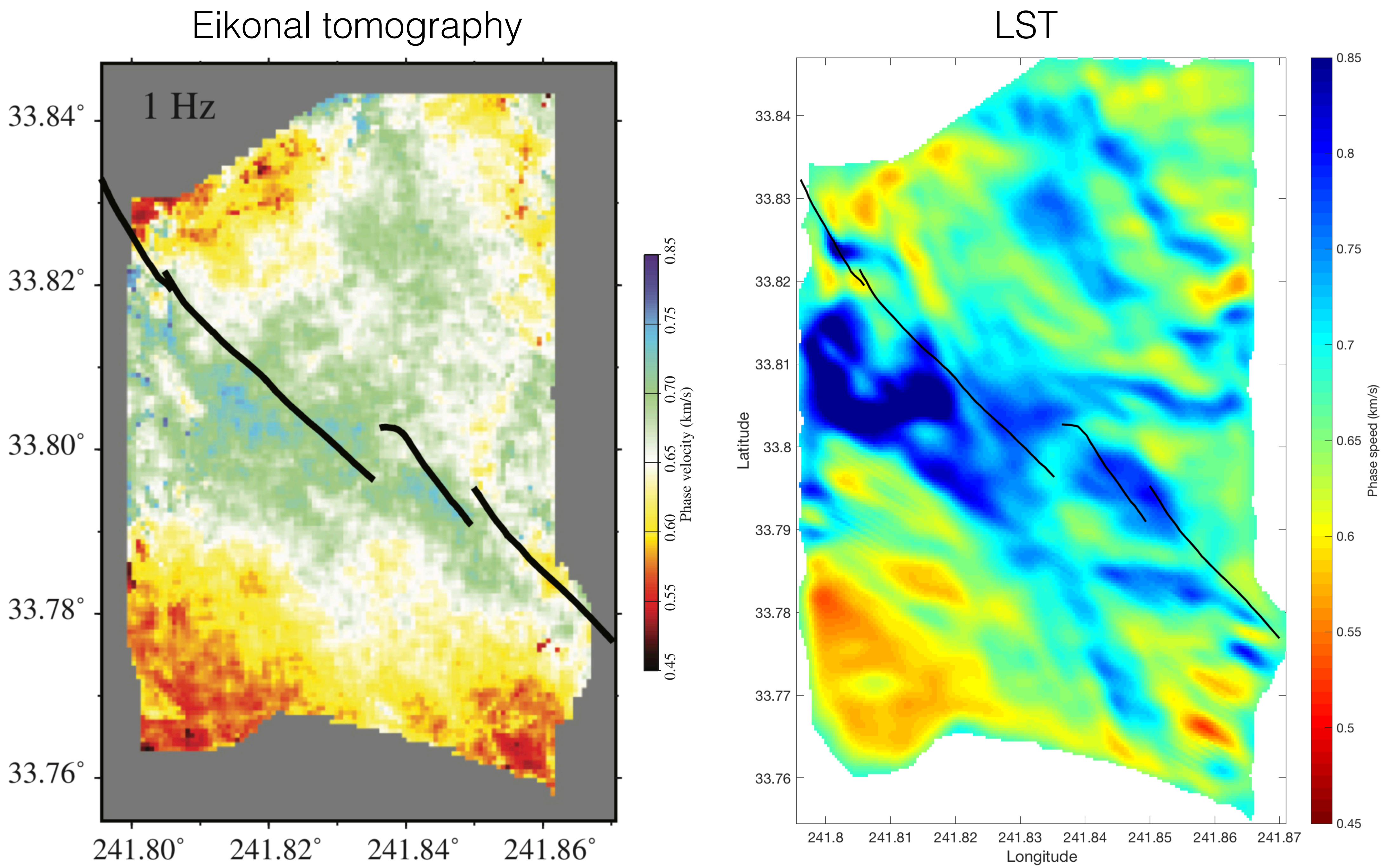
High-resolution LST phase speed map from 8 million cross-correlations





- For LST we generate a 300x200 pixel slowness map with 8 million rays (tomography matrix **A** has dimensions M=8 million, N=60000)
- 10 iterations, used ~2 cpu-hours
- Since we are not imposing global correlations on pixels, can treat **A** as sparse matrix, get fast inversion for global model (which is bottleneck)
- Newport-Inglewood fault network shown as black line

LST comparison with eikonal tomography (Lin et al. 2013)



- We observe the same general trends between eikonal and LST
- From LST have improved contrast along fault lines, for example near Signal Hill
- The LST results are preliminary and they can likely be improved with more careful preprocessing (future work)