Inversion using Machine Learning

Andrew Curtis

University of Edinburgh

Scotland

Neural Network Tomography (Dimensionality Reduction) Variational Inversion

Andrew Curtis

University of Edinburgh

Scotland

Neural Network Tomography

Andrew Curtis

Stephanie Earp, Xin Zhang, Mohammad Shahraeeni,
Ueli Meier, Rob Devilee, Jeannot Trampert

University of Edinburgh
ETH Zurich

JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 99, NO. B4, PAGES 6753–6768, APRIL 10, 1994

Neural networks and inversion of seismic data

Gunter Röth and Albert Tarantola

Institut de Physique du Globe de Paris, Paris, France

Neural networks can be viewed as applications that map one space, the input space, into some output space. In order to simulate the desired mapping the network has to go through a learning process consisting of an iterative change of the internal parameters, through the presentation of many input patterns and their corresponding output patterns. The training process is accomplished if the error between the computed output and the desired output pattern is minimal for all examples in the training set. The network will then simulate the desired mapping on the

JOURNAL OF GEOPHYSICAL RESEARCH, VOL. 104, NO. B12, PAGES 28,841–28,857, DECEMBER 10, 1999

An efficient, probabilistic neural network approach to solving inverse problems: Inverting surface wave velocities for Eurasian crustal thickness

R.J.R. Devilee A. Curtis,[1] and K. Roy-Chowdhury

Geodynamic Research Institute, Department of Geophysics, Utrecht University, the Netherlands

Abstract. Nonlinear inverse problems usually have no analytical solution and may be solved by Monte Carlo methods that create a set of samples, representative of the a posteriori distribution. We show how neural networks can be trained on these samples to give a continuous approximation to the inverse relation in a compact and computationally efficient form. We examine the strengths and weaknesses of

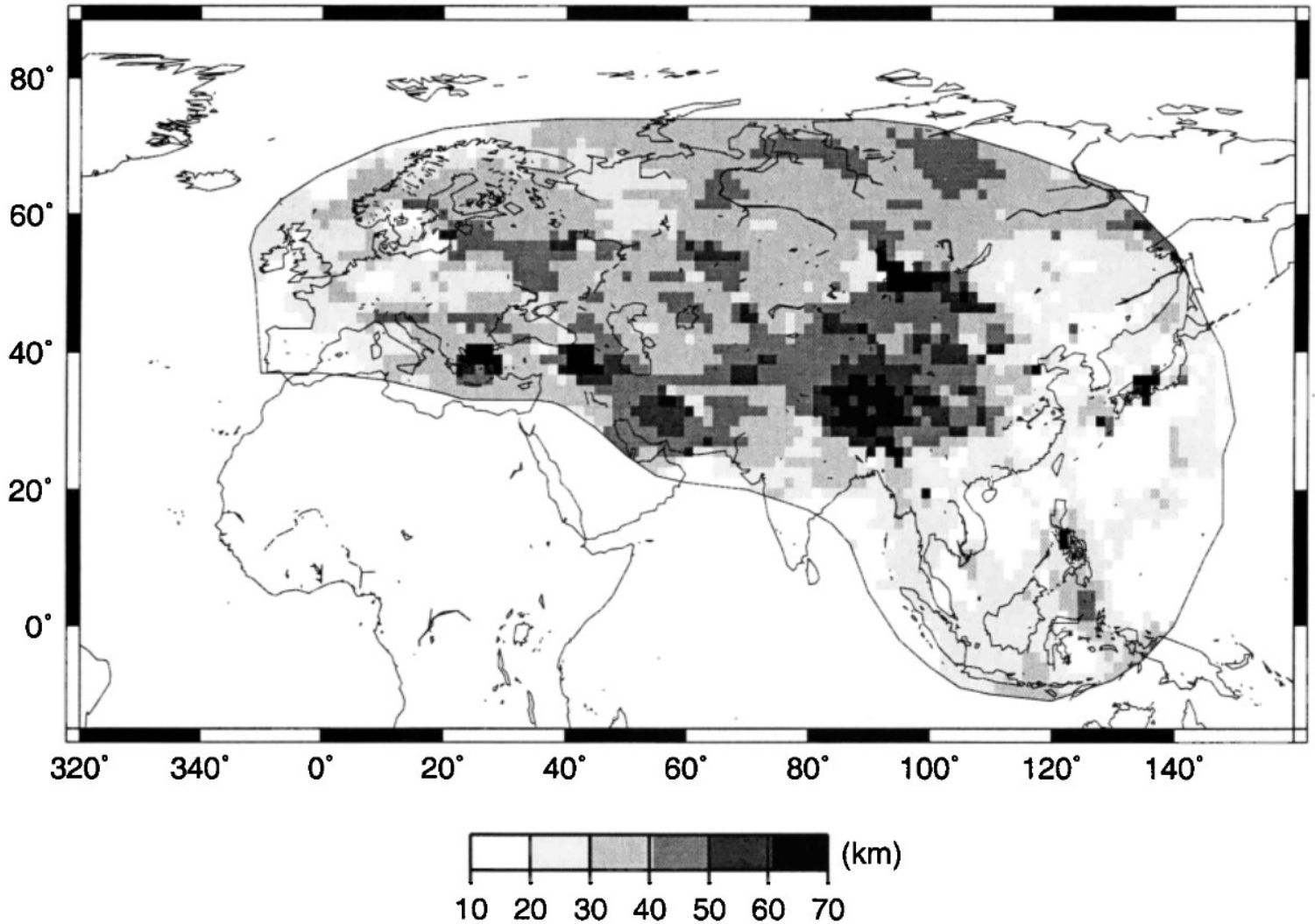


Figure 12. Maximum likelihood crustal thickness map across Eurasia obtained from the inversion of combined Love and Rayleigh wave data of data set II with a histogram network. The discretization interval of the solution is 10 km. Only areas in which phase velocities are resolved at length scales of < 1000 km are shown.

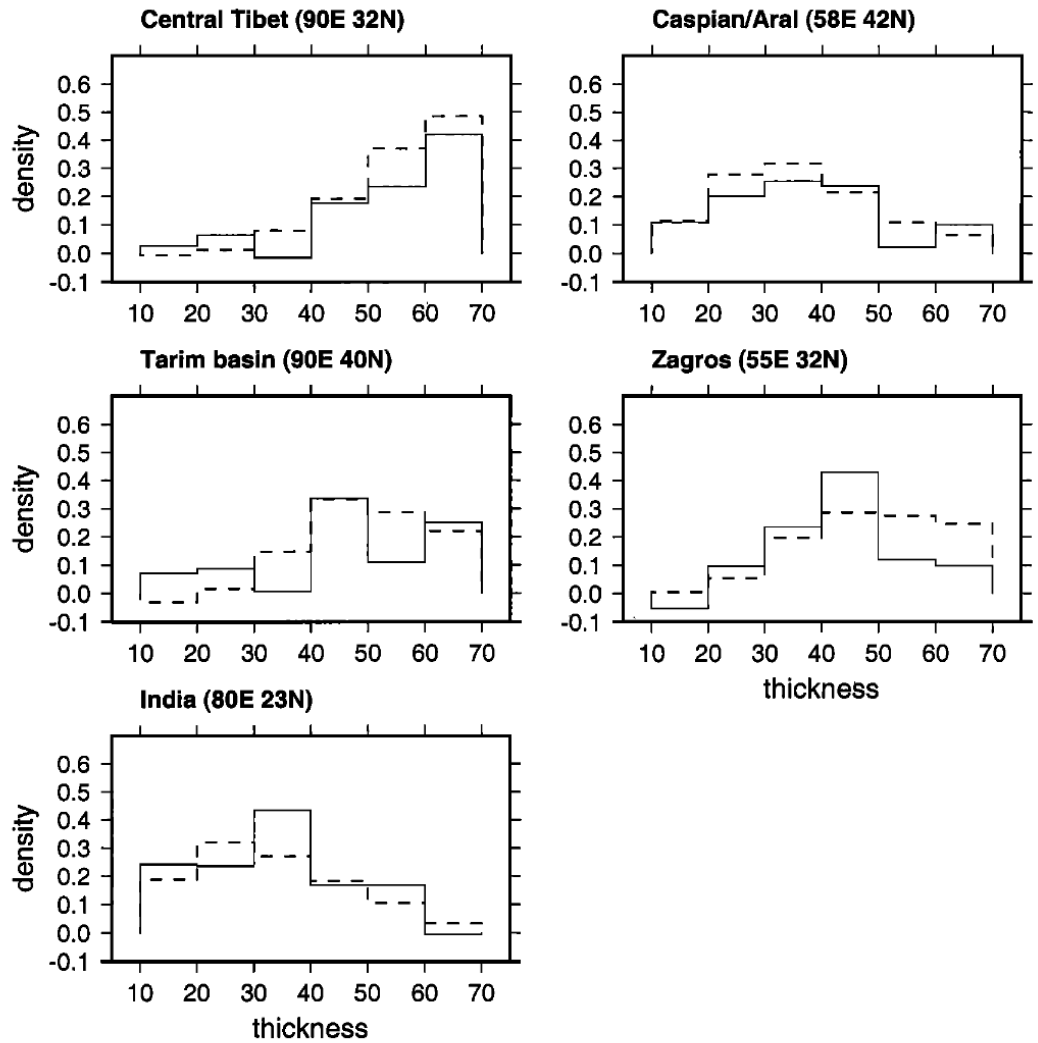


Figure 11. The full crustal thickness distributions for several coordinates in the Tibetan area,

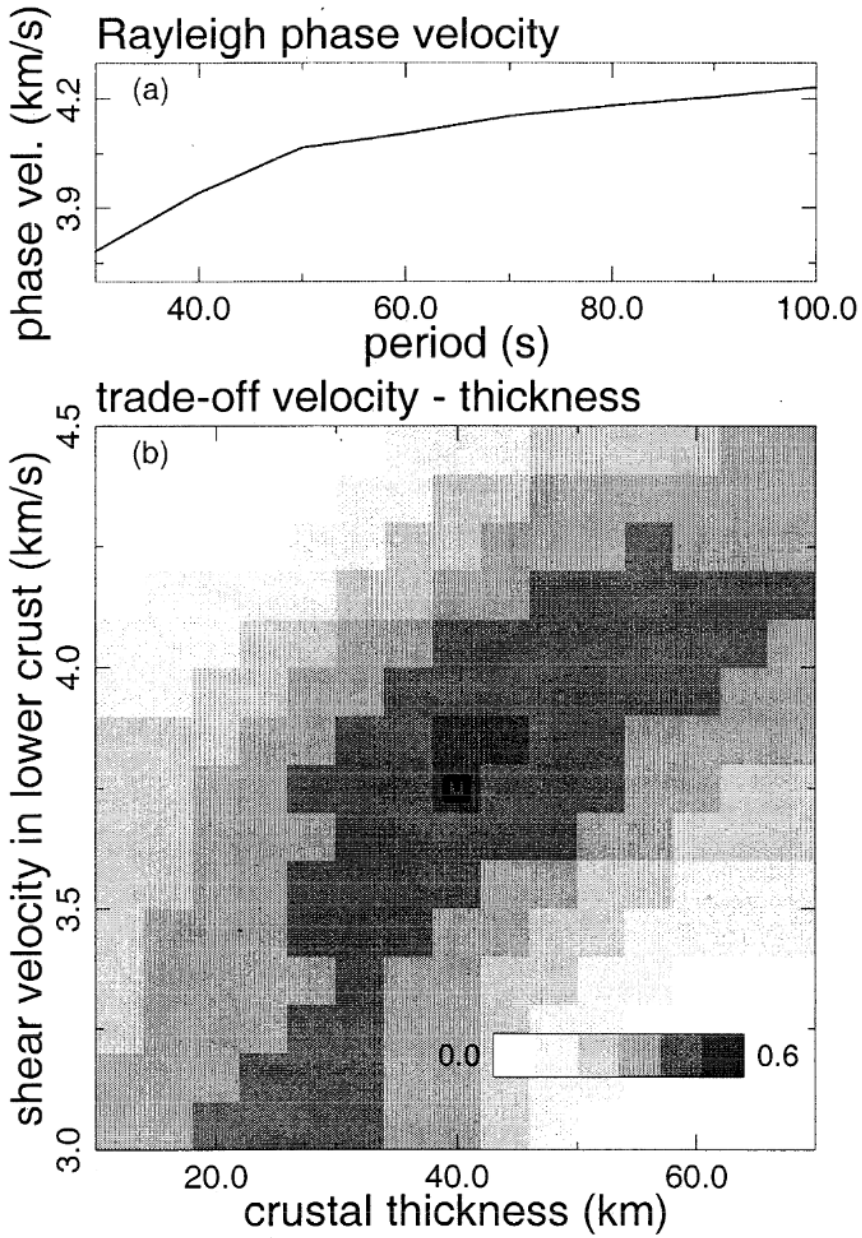


Figure 9. (a) Synthetic Rayleigh phase velocities calculated for the model of Table 1. We trained two

Geophys. J. Int. (2007) **169,** 706–722 doi: 10.1111/j.1365-246X.2007.03373.x

Global crustal thickness from neural network inversion of surface wave data

Ueli Meier,[1] Andrew Curtis[2,*] and Jeannot Trampert[1]

[1] *Utrecht University, Department of Earth Sciences, Budapestlaan* 4, 3584 *CD Utrecht, the Netherlands. E-mail: meierue@geo.uu.nl*
[2] *The University of Edinburgh, School of GeoSciences, Grant Institute, West Mains Road, Edinburgh,* EH9 3JW, *UK.*

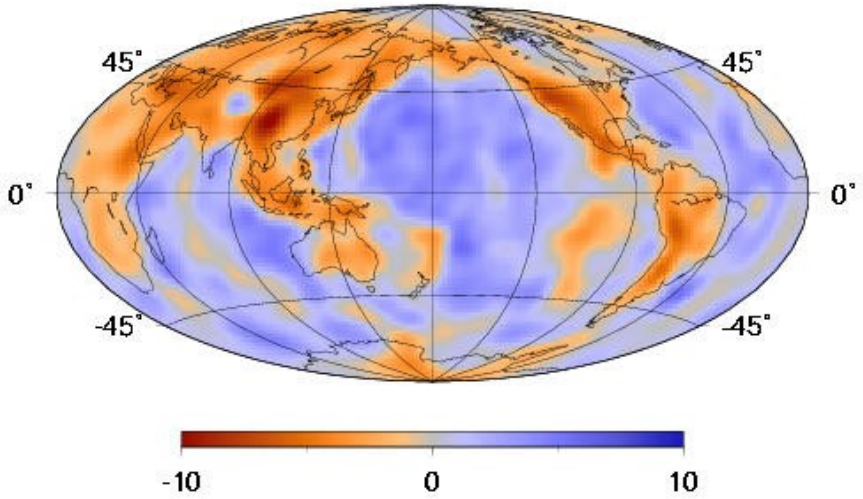
Accepted 2007 January 24. Received 2006 November 9; in original form 2006 August 11

Mixture Density Networks (MDN)

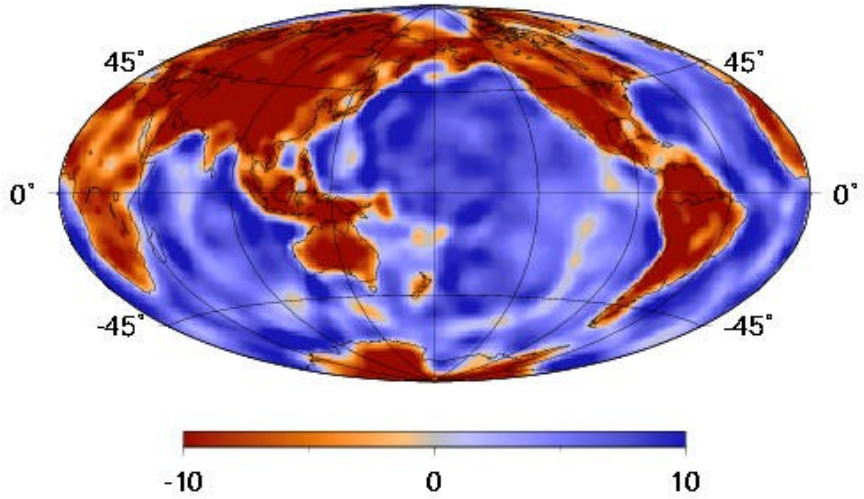
SUMMARY

We present a neural network approach to invert surface wave data for a global model of crustal thickness with corresponding uncertainties. We model the *a posteriori* probability distribution of Moho depth as a mixture of Gaussians and let the various parameters of the mixture model be given by the outputs of a conventional neural network. We show how such a network can be trained on a set of random samples to give a continuous approximation to the inverse relation in a compact and computationally efficient form. The trained networks are applied to real

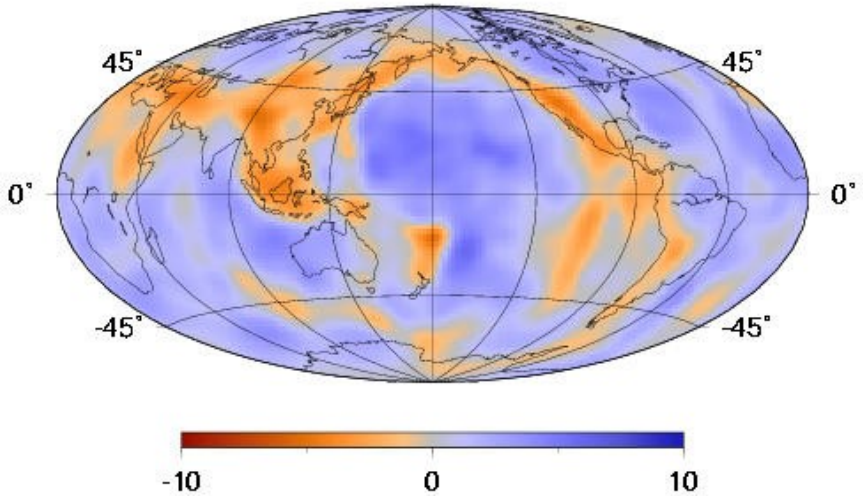
Love Phase 40 s



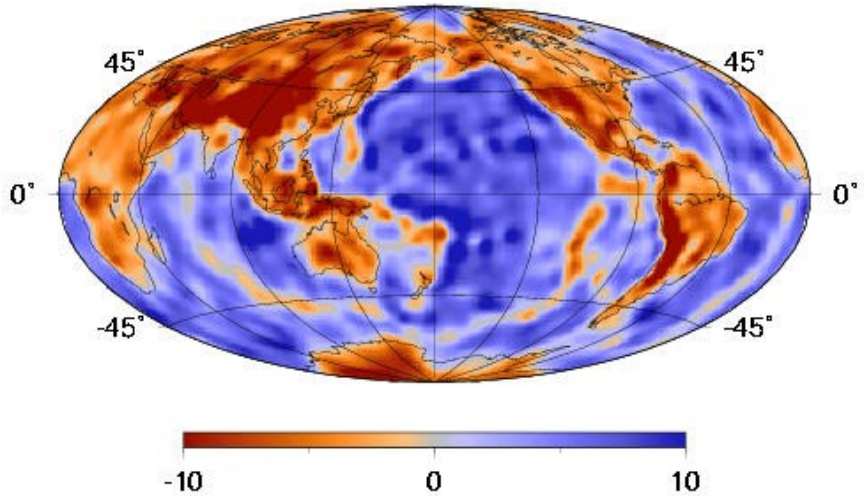
Love Group 40 s



% Perturbation Relative to PREM



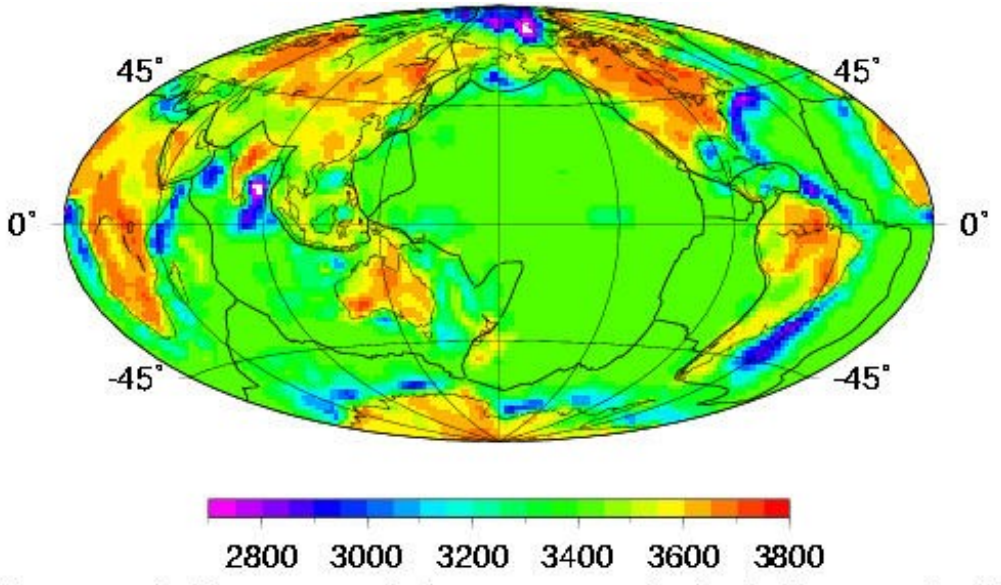
Rayleigh Phase 40 s



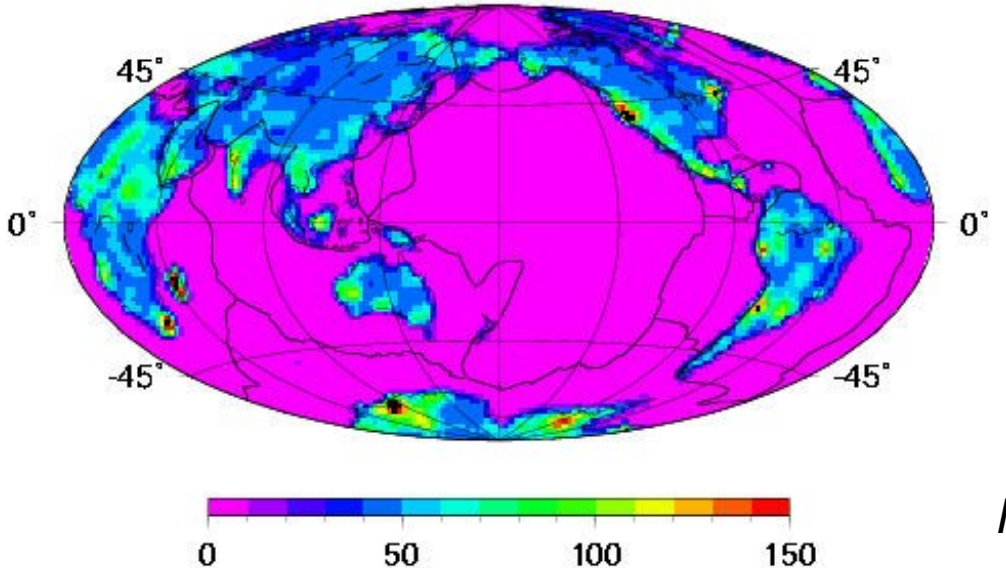
Rayleigh Group 40 s

Trampert and Woodhouse, 2003

Ritzwoller et al., 2002



Mean vertically averaged crustal Vs [m/s]

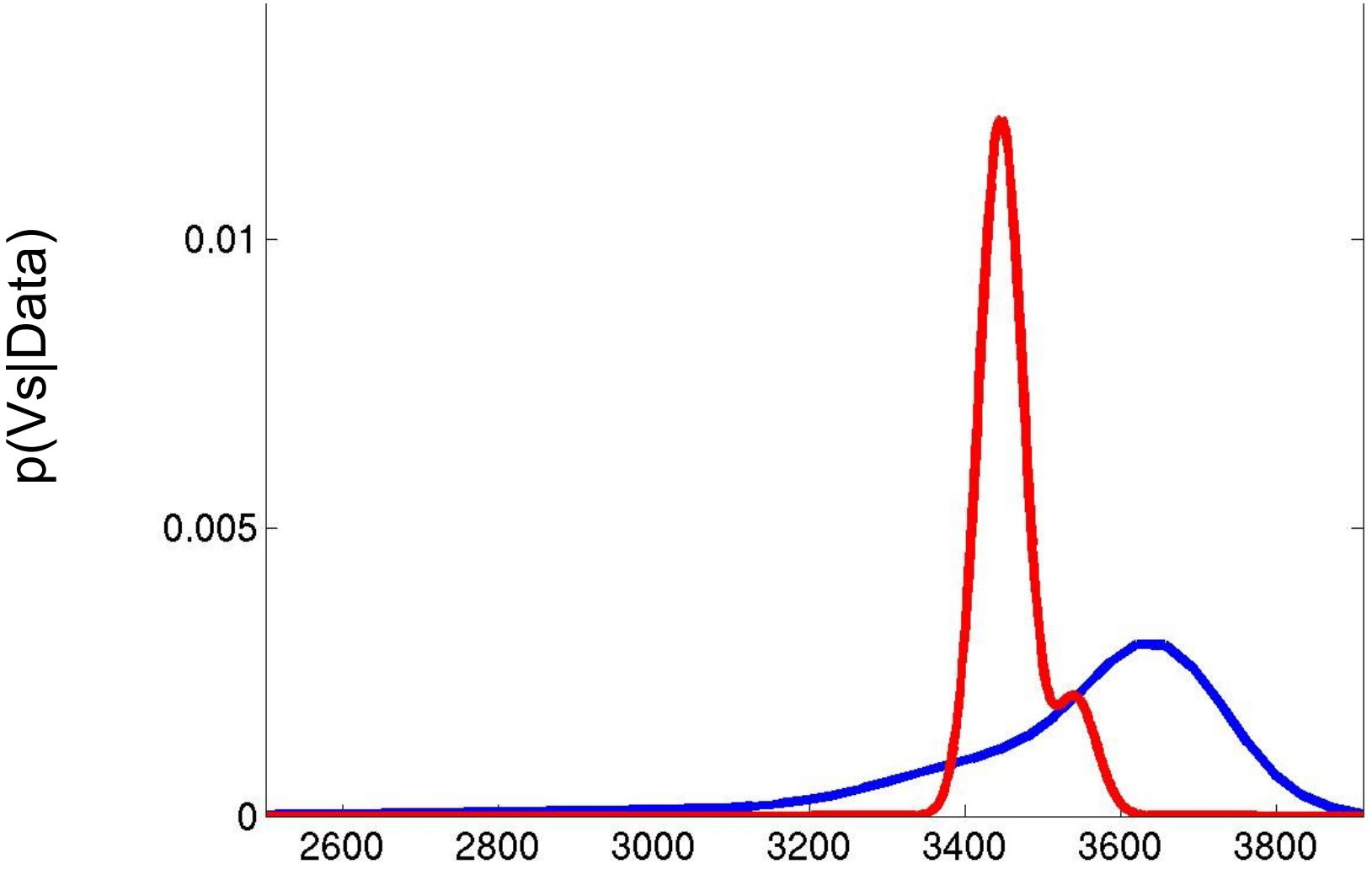


Meier et al., GRL 2007

Standard deviation σ [m/s]

Posterior **average crustal Vs** distribution in Central Tibet

(33N,87E)



GEOPHYSICS, VOL. 76, NO. 2 (MARCH-APRIL 2011); P. E45–E58, 12 FIGS., 3 TABLES.
10.1190/1.3540628

Fast probabilistic nonlinear petrophysical inversion

Mohammad S. Shahraeeni[1] and Andrew Curtis[1]

ABSTRACT

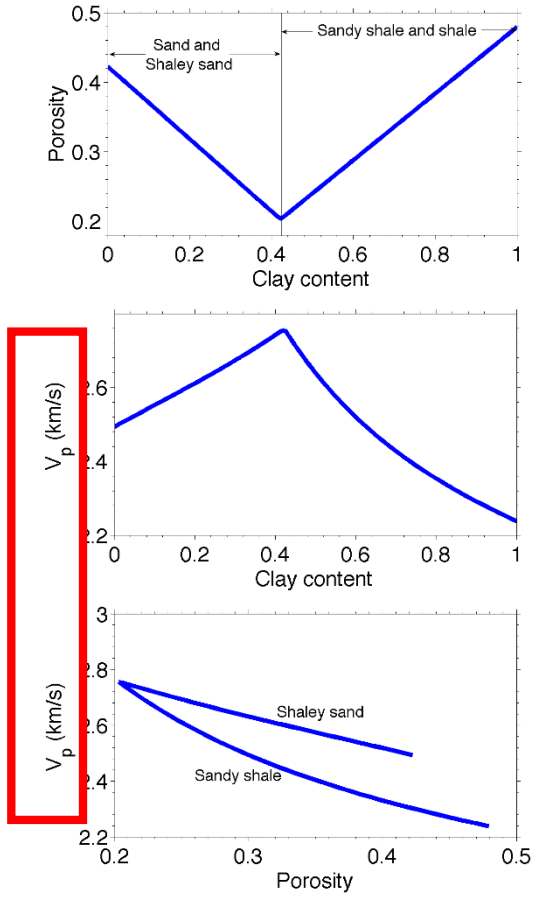
We have developed an extension of the mixture-density neural network as a computationally efficient probabilistic method to solve nonlinear inverse problems. In this method, any postinversion (a posteriori) joint probability density function (PDF) over the model parameters is represented by a weighted sum of multivariate Gaussian PDFs. A mixture-density neural network estimates the weights, mean vector, and covariance matrix of the Gaussians given any measured data set. In one study, we have jointly inverted compressional- and shear-wave velocity for the joint PDF of porosity, clay content, and water saturation in a synthetic, fluid-saturated, dispersed sand-shale system. Results show that if the method is applied appropriately, the joint PDF estimated by the neural network is comparable to the Monte Carlo sampled a posteriori solution of the inverse problem.

used to solve nonlinear geophysical inverse problems with 1D model spaces. Devilee et al. (1999) invert regional surface-wave dispersion velocities for crustal thickness across Eurasia, and Meier et al. (2007a) extend this to obtain a global crustal thickness map. Devilee et al. (1999) also show how the laws of probability can be used to combine the output of multiple neural networks, each solving a 1D inverse problem, to solve a multidimensional inverse problem.

Devilee et al.'s method is used by Meier et al. (2007b) to invert for a two-parameter (average velocity and Moho depth) global crustal model that could be used, among other applications, to make near-surface corrections for global deep-mantle tomography. Meier et al. (2009) extend the data and methodology to perform petrophysical inversion for global water content and temperature in the earth's mantle transition zone (approximately 440–660 km deep) in an inversion that also constrains parameters in the petrophysical forward relations between temperature, water content, and seismic velocities.

A rock physical inverse problem

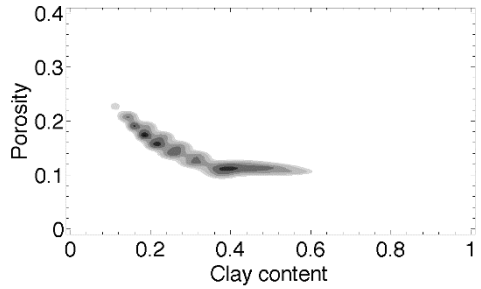
- Data vector: **d**=(V_p,V_s)
  - σ_{Vp}=0.05 V_p.
  - σ_{Vs}=0.07 V_s.
  - Model vector: **m**=(φ,V_{cl},S_w)
- Forward function: Dvorkin-Gutierrez (2001) bi-value petrophysical model.
- Confounding parameters of the forward functions:
  - K_s, G_s & ρ_s: elastic moduli and density of sand grains.
  - K_c, G_c & ρ_c: elastic moduli and density of clay grains.
  - z: depth



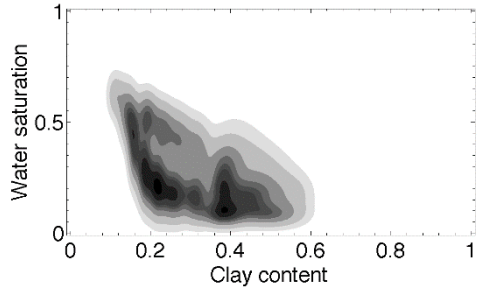
Comparison between MC sampling and MDN solution

MDN solution

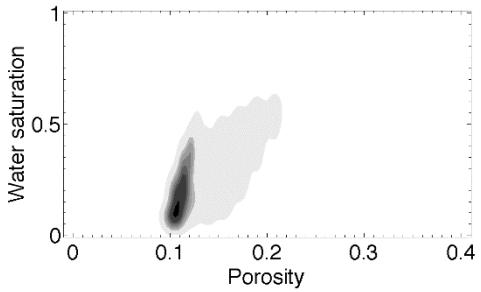
$d=(V_p,V_s)$ =(2818 m/s, 1675 m/s)



But is this correct…?

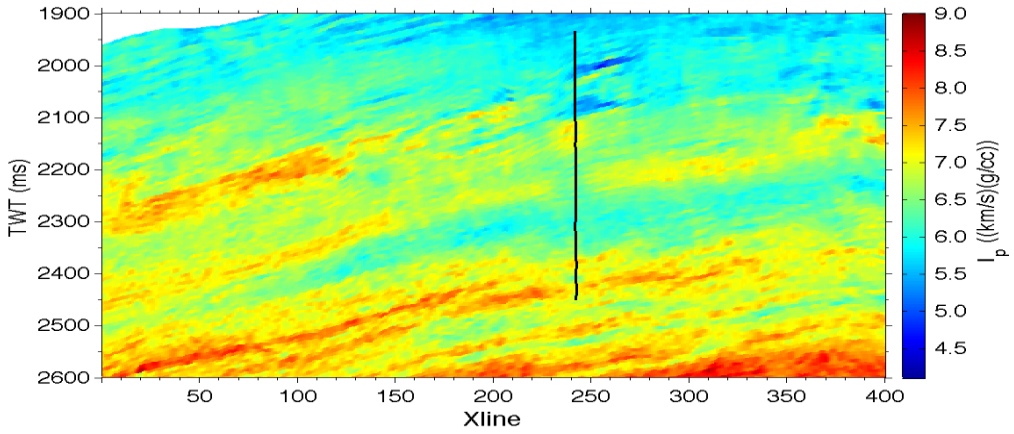


“Bayesian’s Uncertainty”

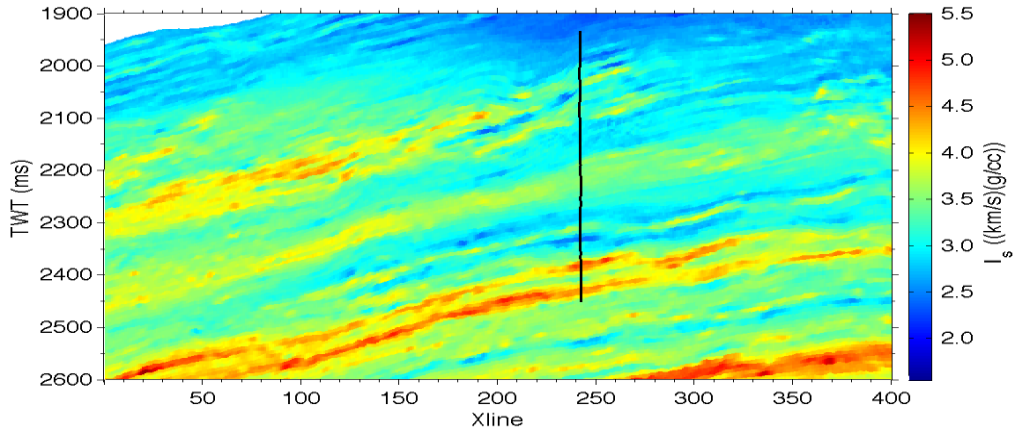


AVO seismic data

P-wave impedance

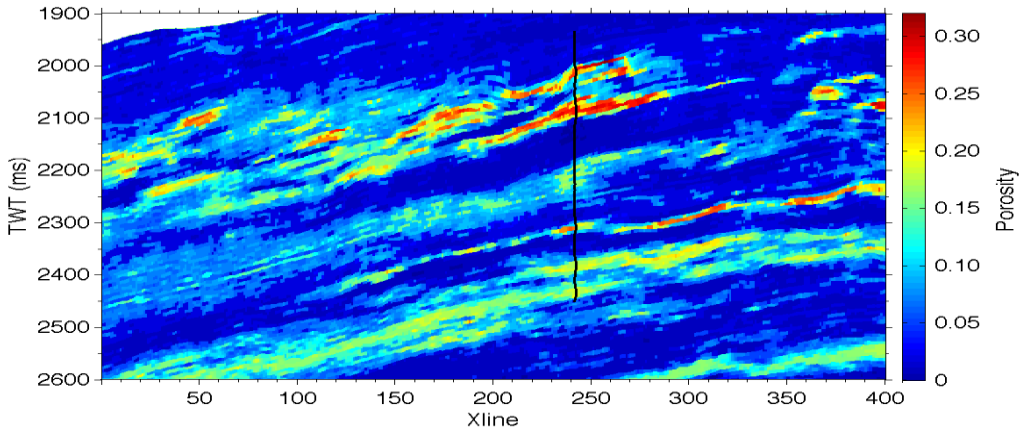


S-wave impedance

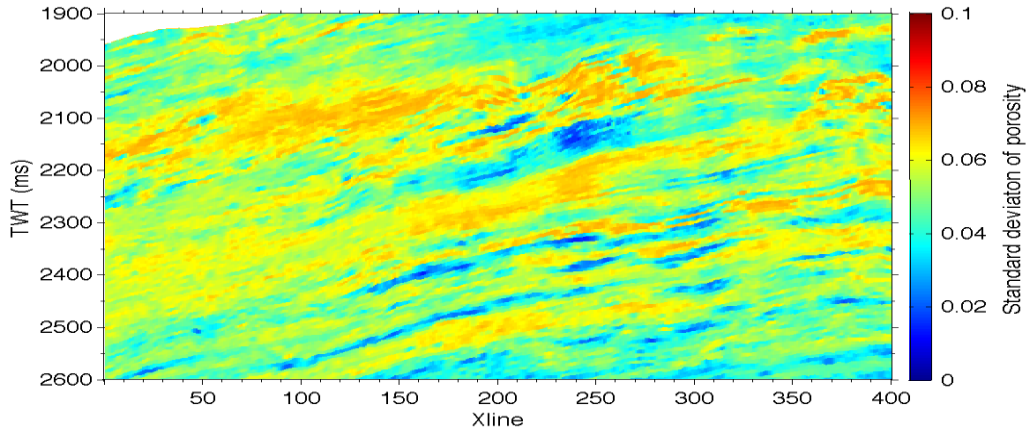


Inversion result:
Porosity

MAP

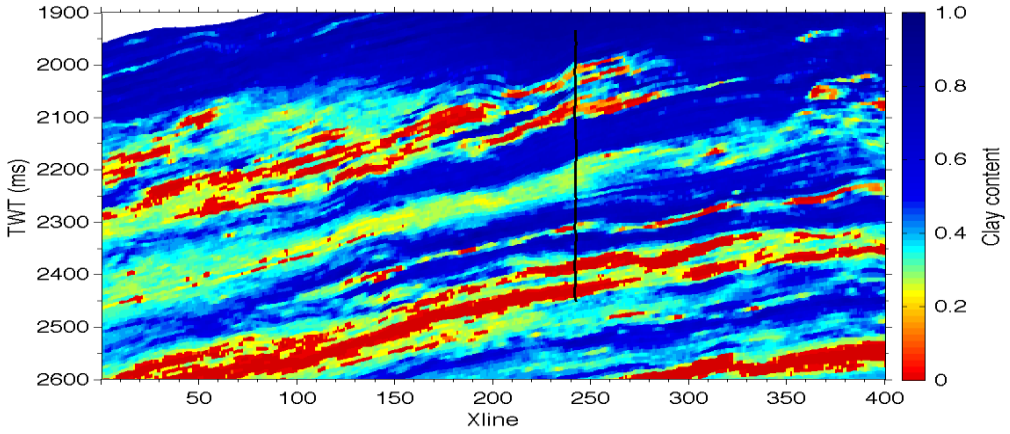


Standard deviation

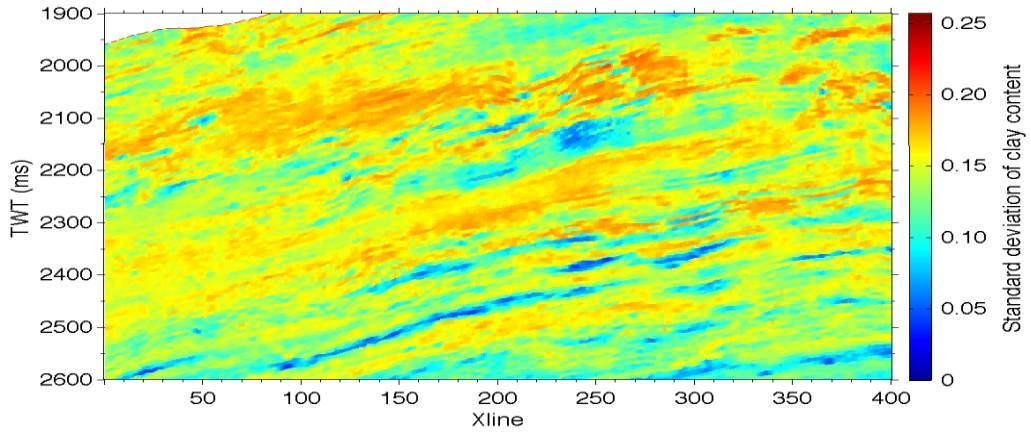


Inversion result:
Clay Content

MAP



Standard deviation



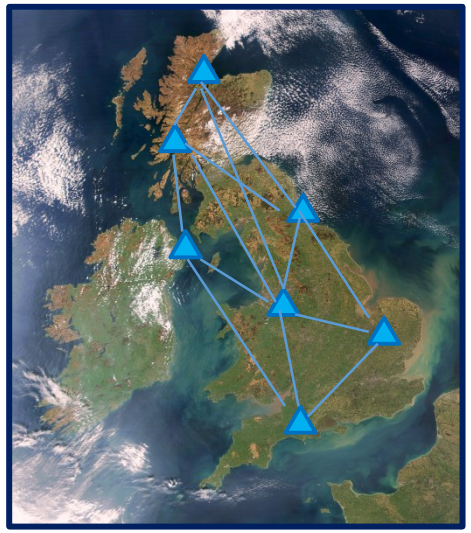
THE UNIVERSITY of EDINBURGH

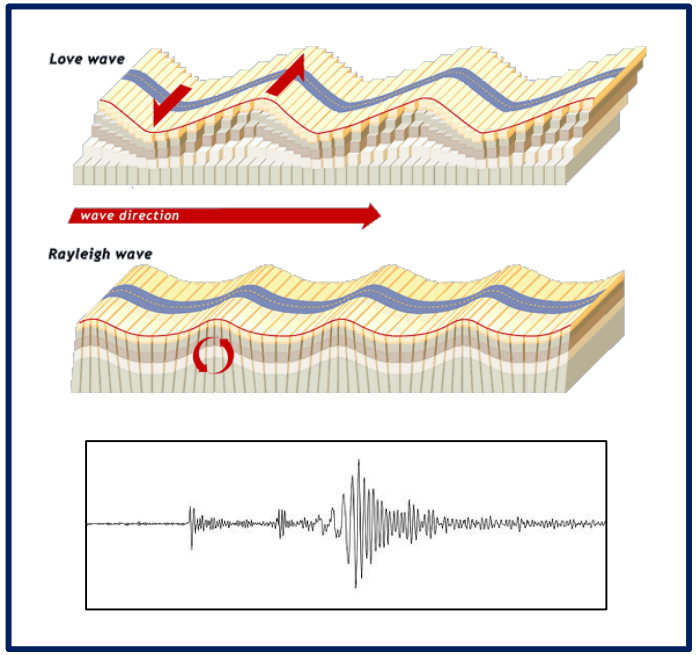
edinburgh interferometry project

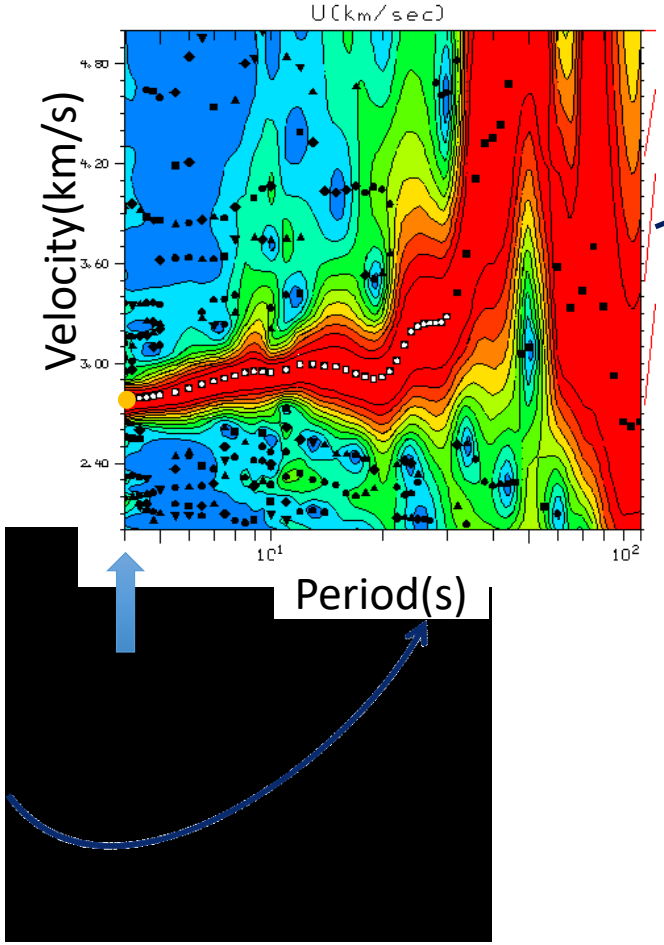
3D Surface Wave Tomography

Xin Zhang, Andrew Curtis
Erica Galetti, Sjoerd de Ridder

Seismic Surface Wave Tomography: typical workflow

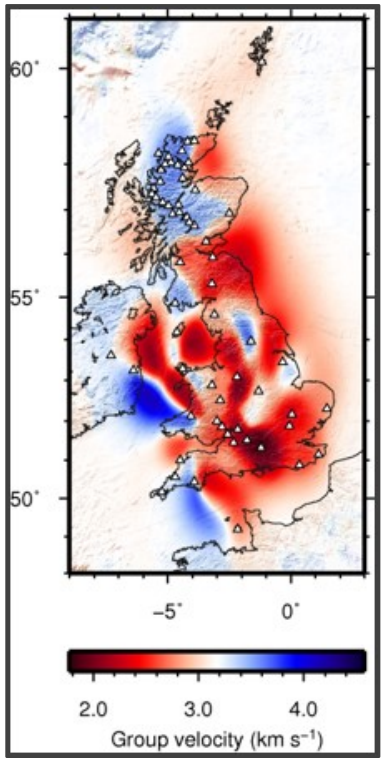




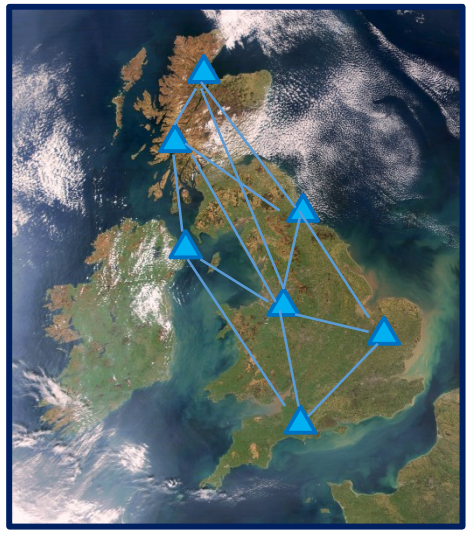


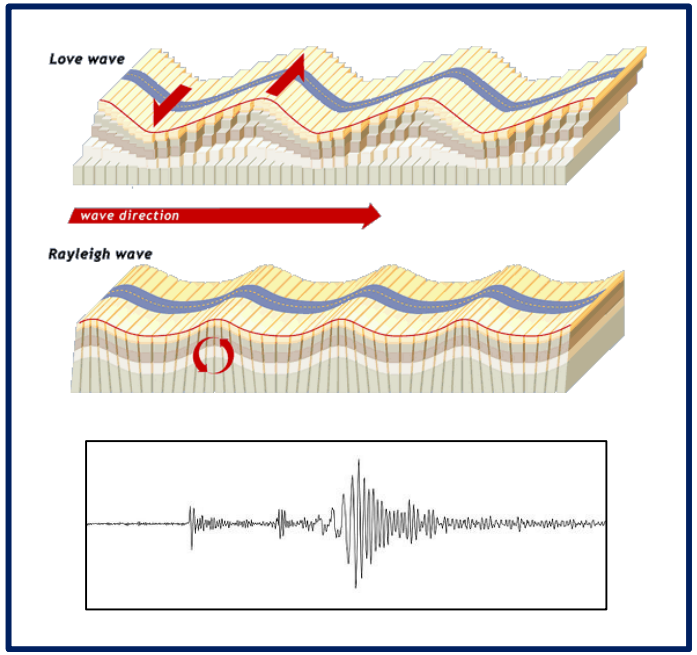
Travel-Time Tomography

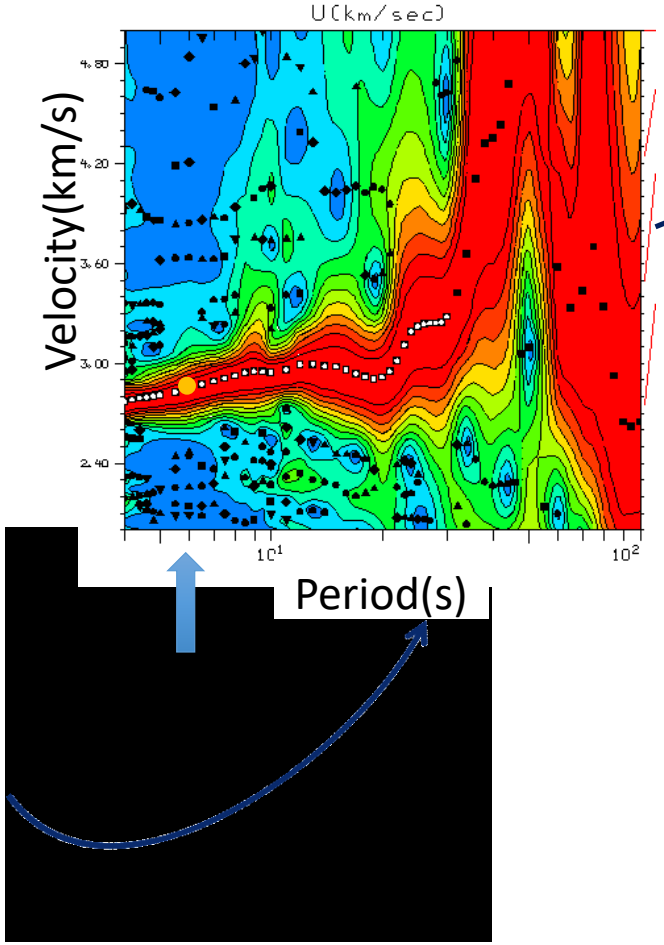
4s Group Velocity



Seismic Surface Wave Tomography : typical workflow

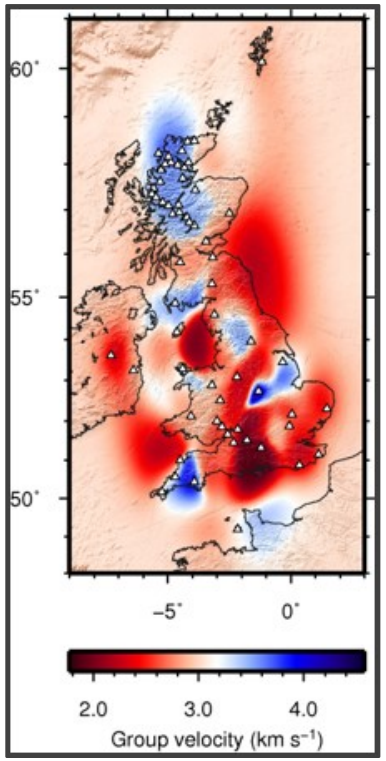




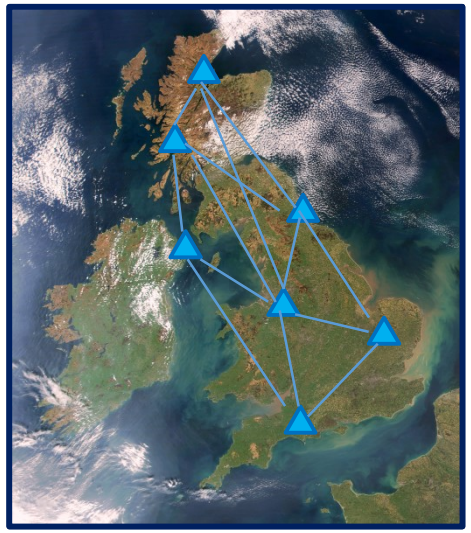


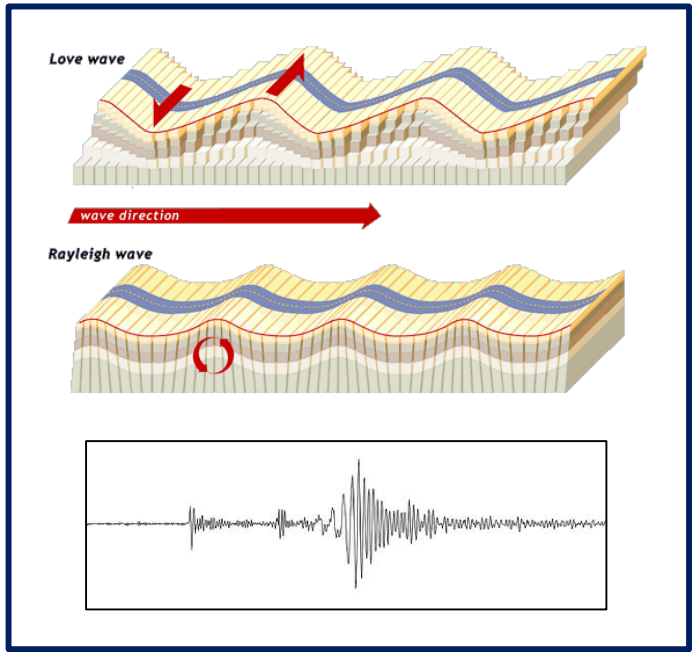
Travel-Time Tomography

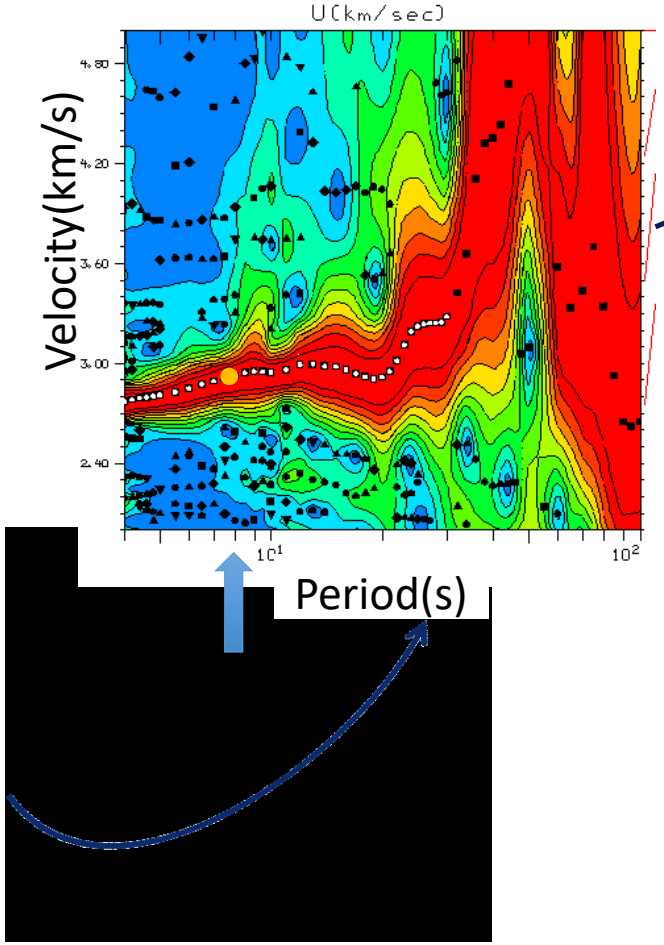
6s Group Velocity



Seismic Surface Wave Tomography : typical workflow

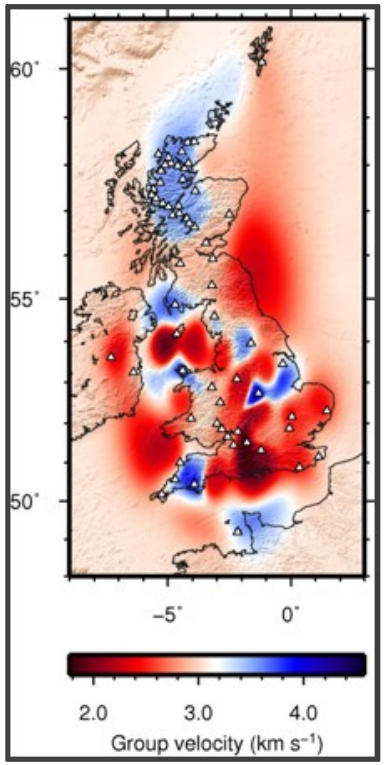




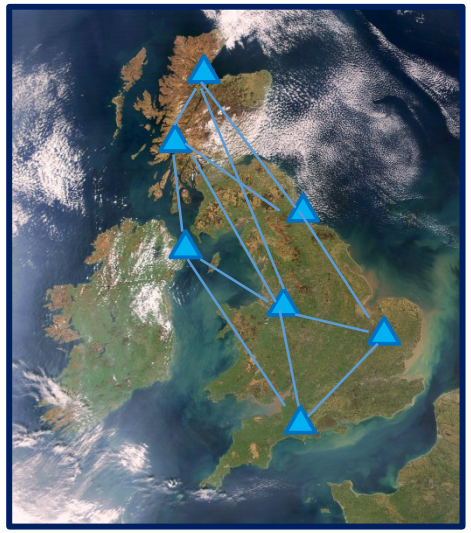


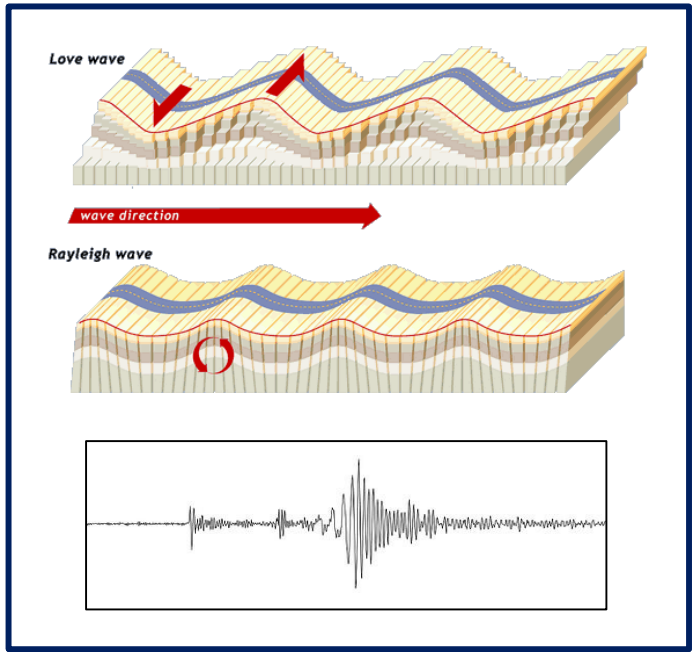
Travel-Time Tomography

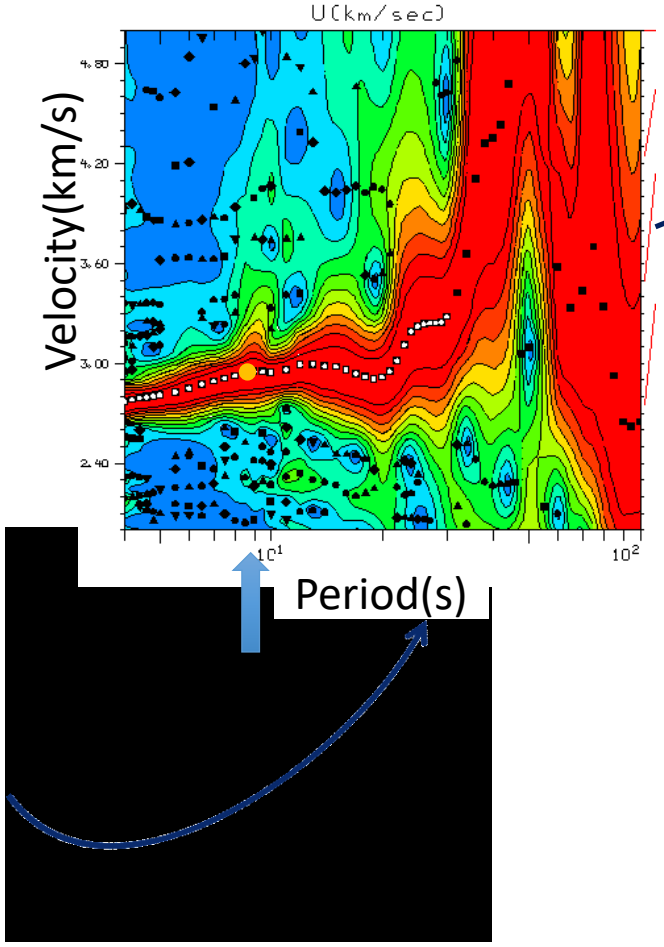
8s Group Velocity



Seismic Surface Wave Tomography : typical workflow

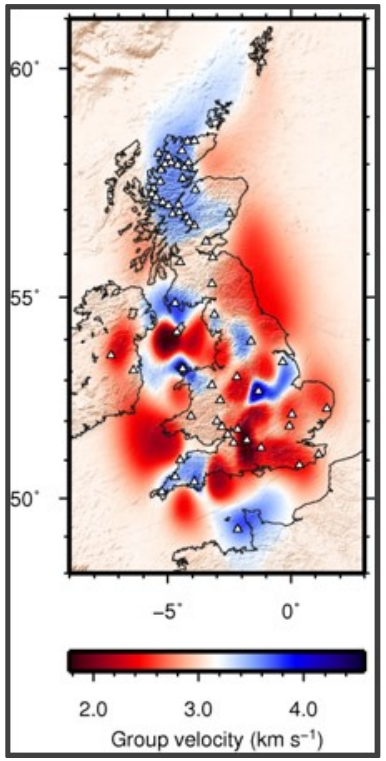




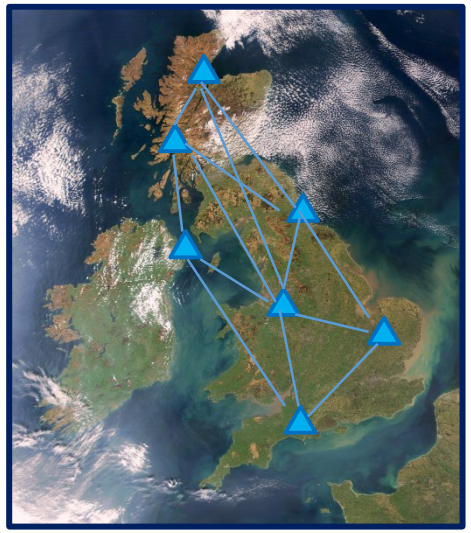


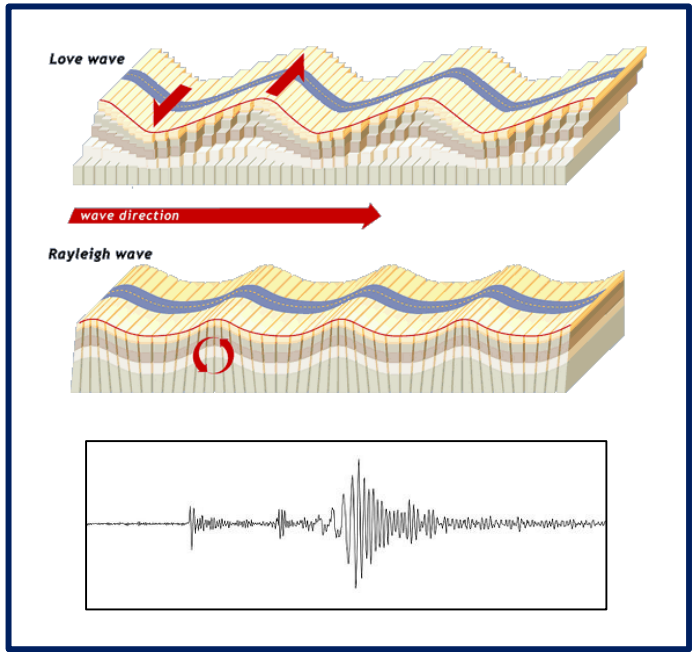
Travel-Time Tomography

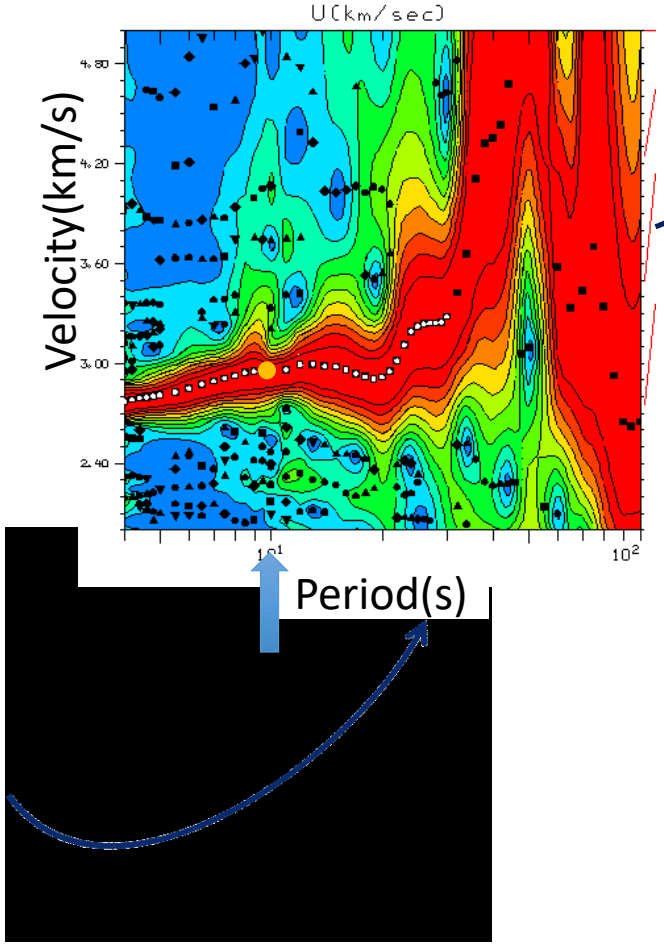
9s Group Velocity



Seismic Surface Wave Tomography : typical workflow

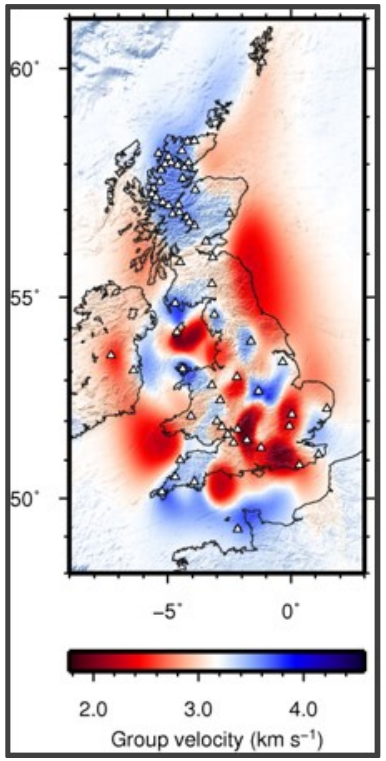




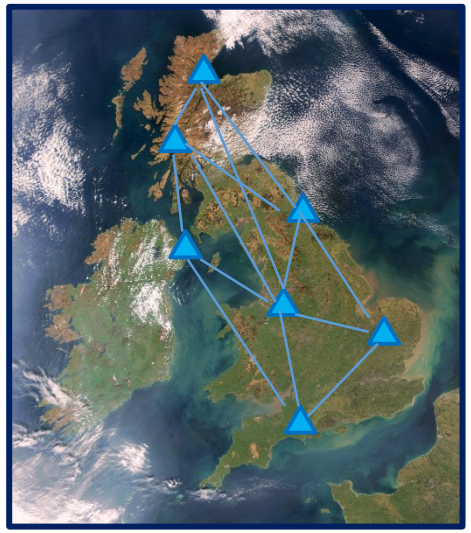


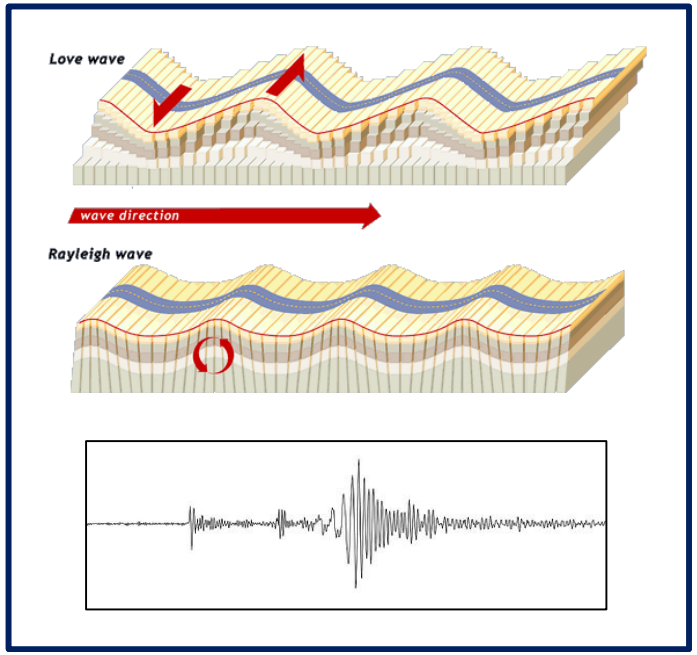
Travel-Time Tomography

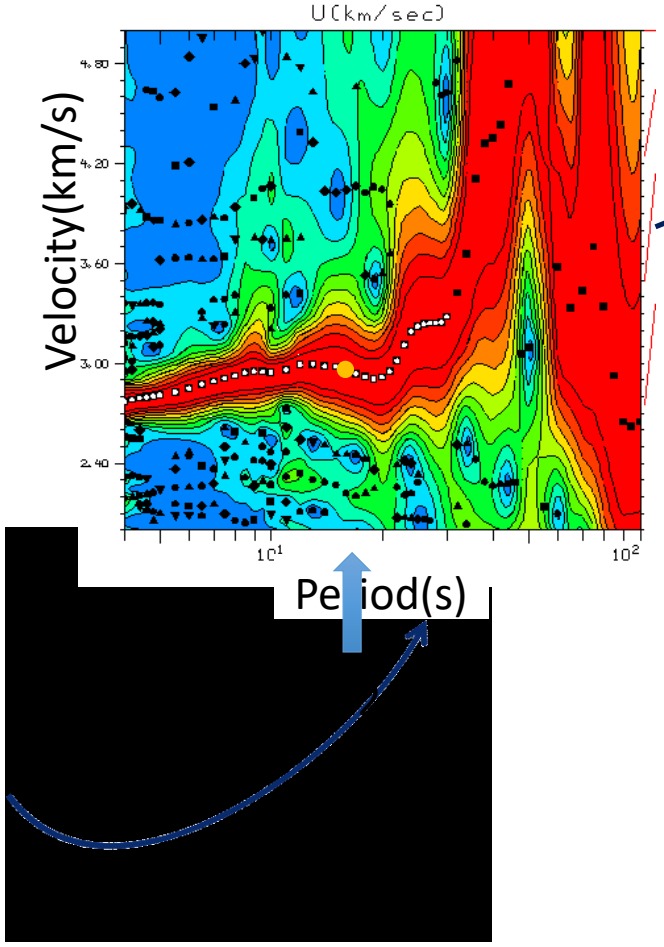
10s Group Velocity



Seismic Surface Wave Tomography : typical workflow

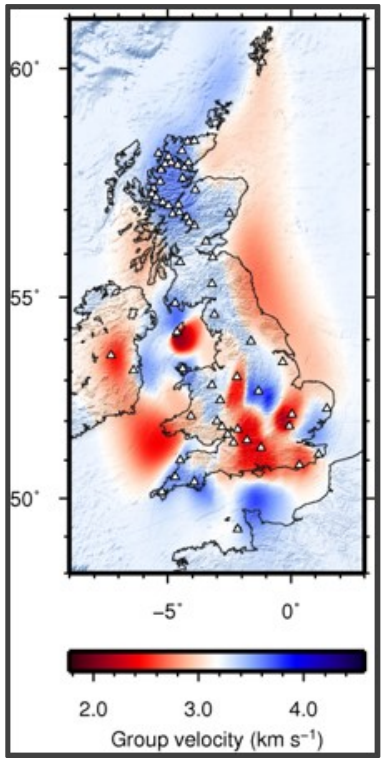




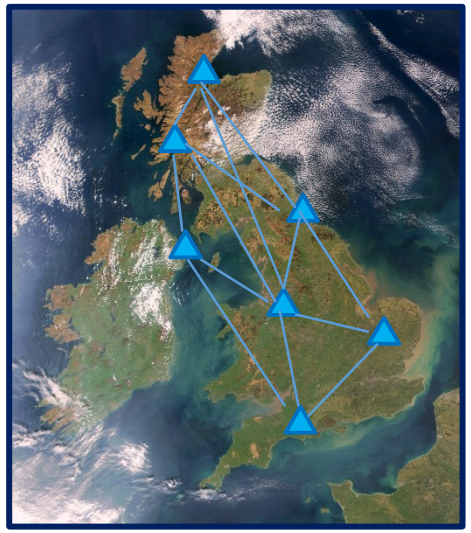


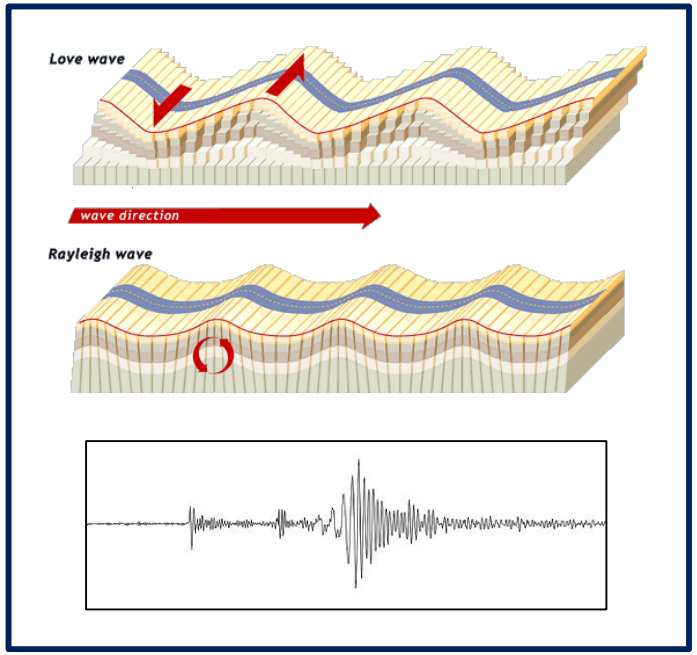
Travel-Time Tomography

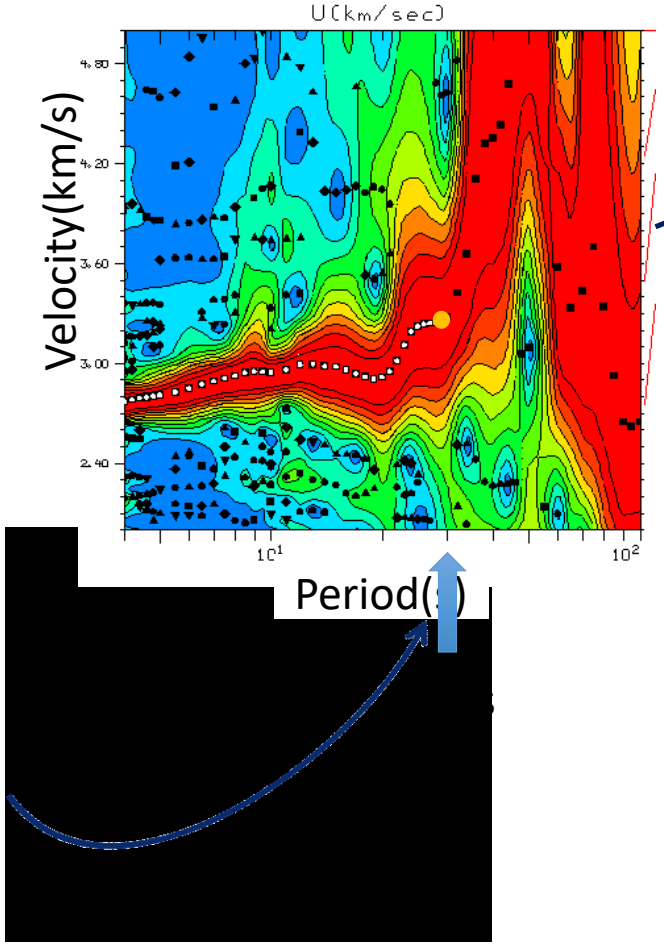
12s Group Velocity



Seismic Surface Wave Tomography : typical workflow

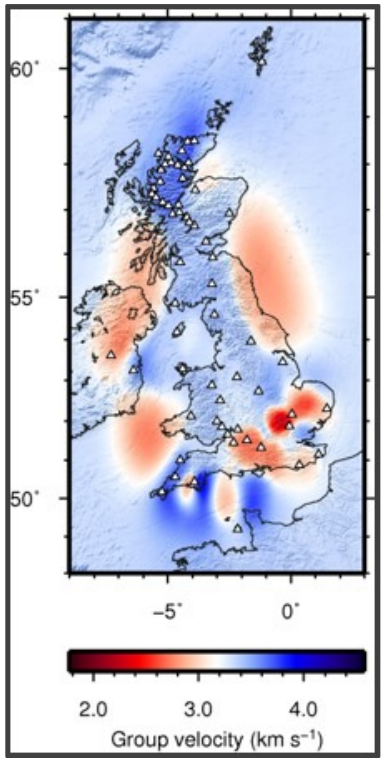




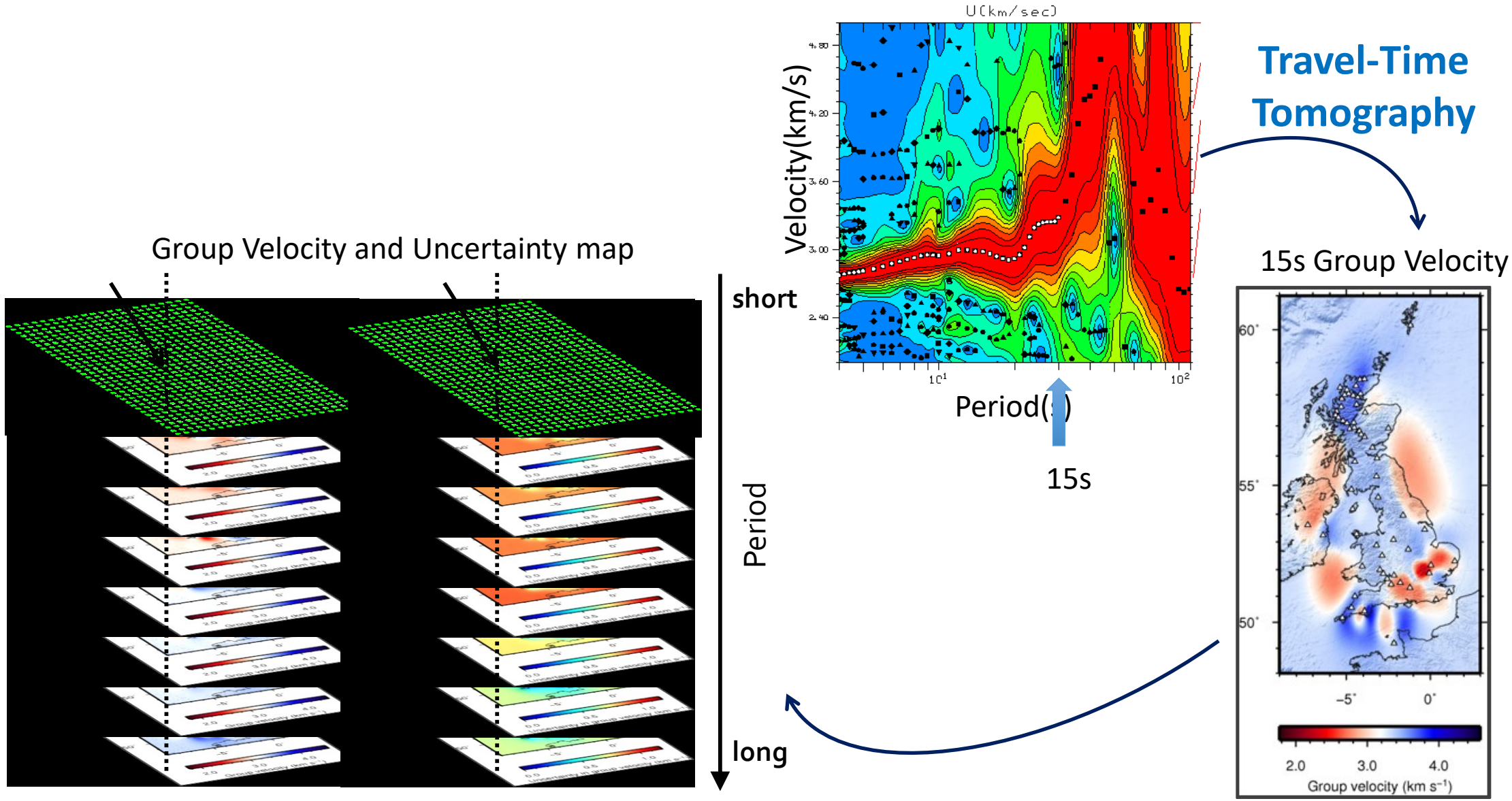


Travel-Time Tomography

15s Group Velocity

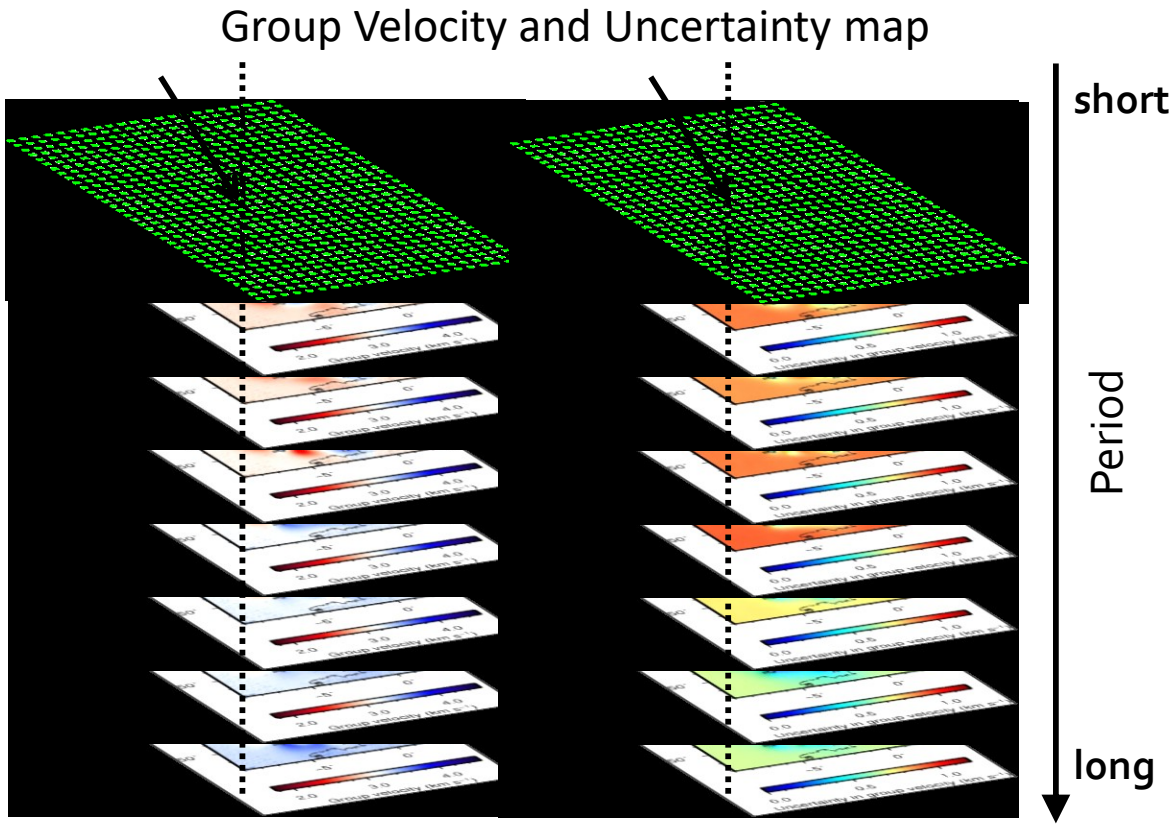


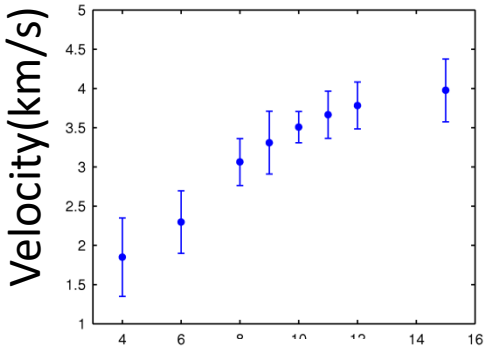
Seismic Surface Wave Tomography : typical workflow



Seismic Surface Wave Tomography : typical workflow

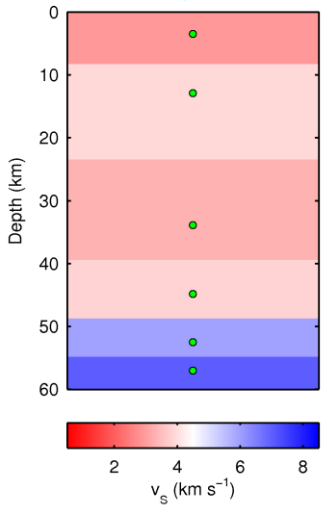
- Stage 1: construct 2D phase/group velocity map
- Stage 2: 1D depth inversion at each grid point
  Many points ➔ 3D model

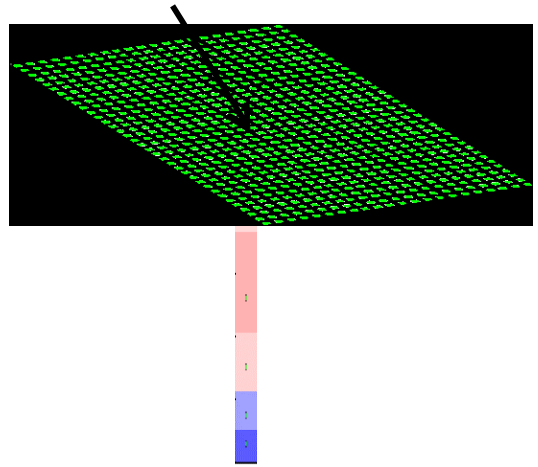




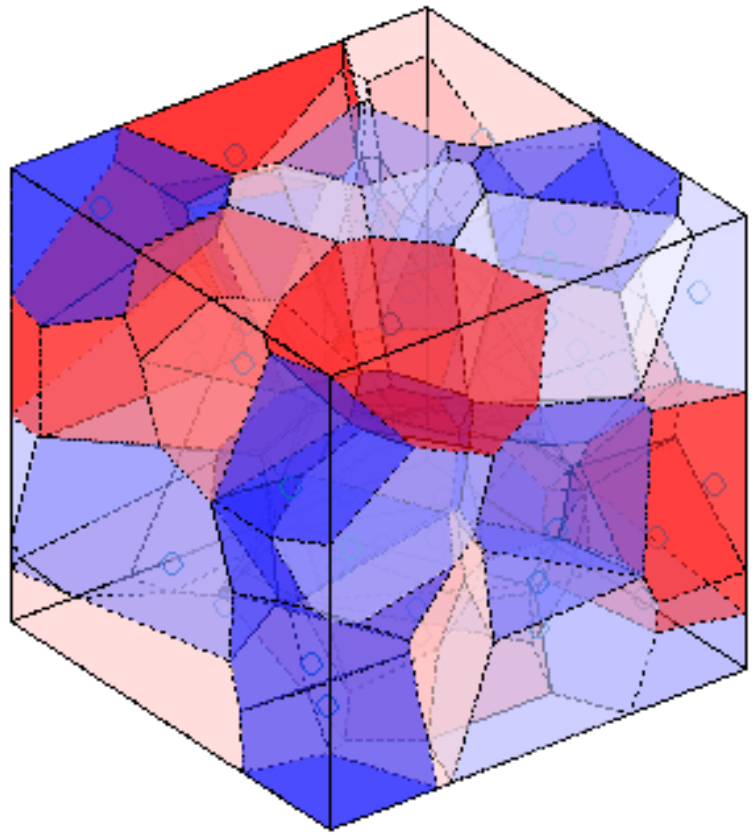
1D depth inversion

1D Depth Tomography

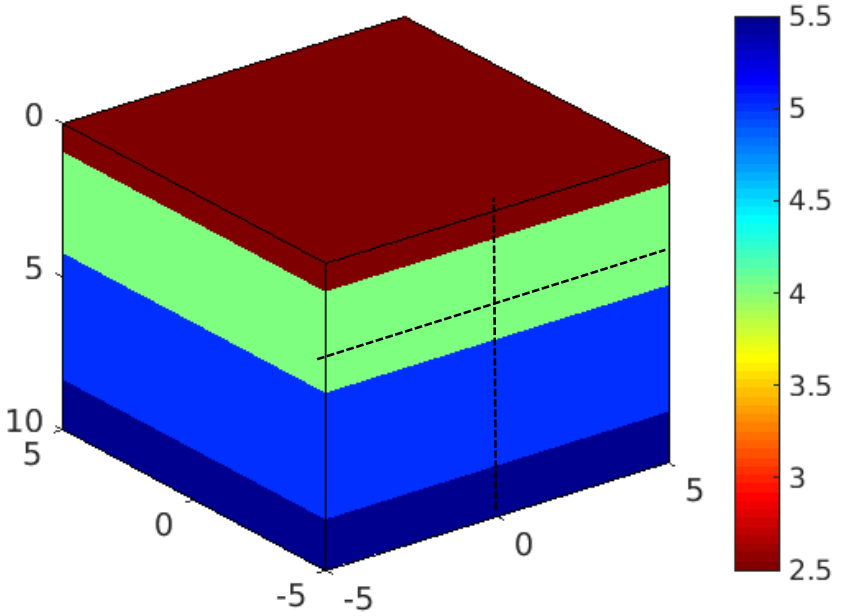


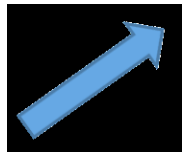


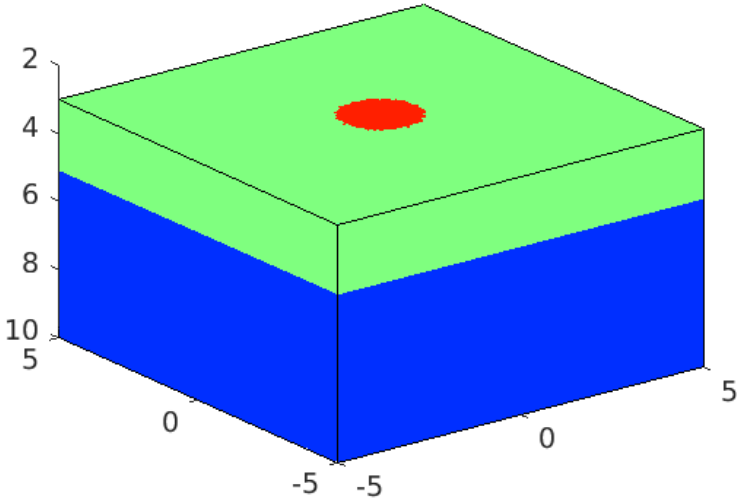
3D Voronoi

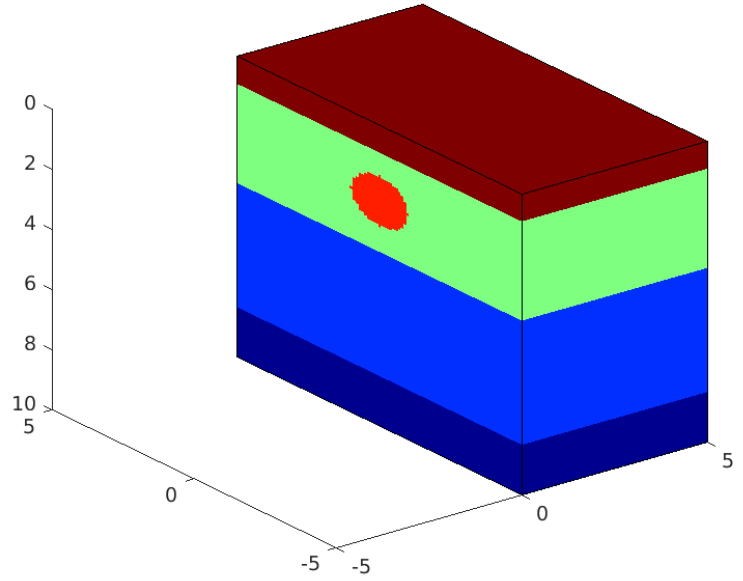


Synthetic test

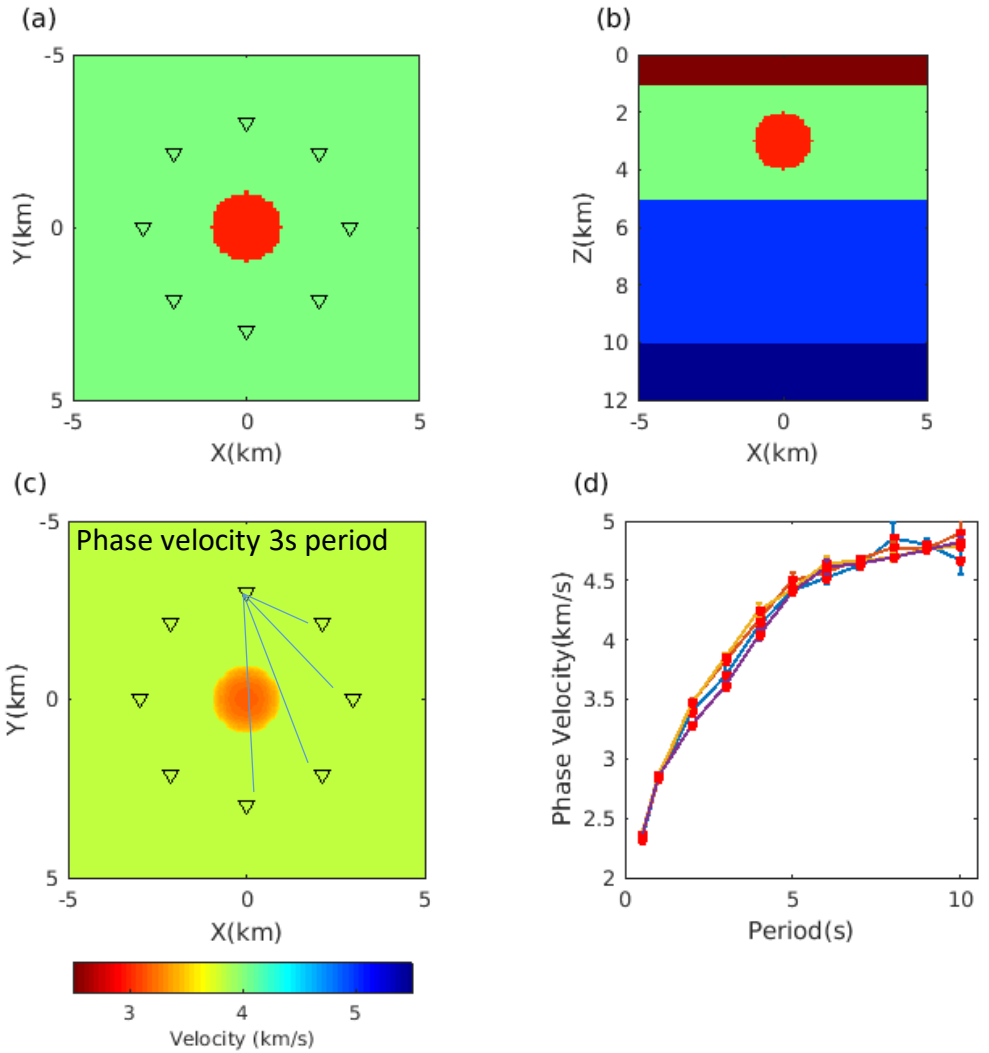




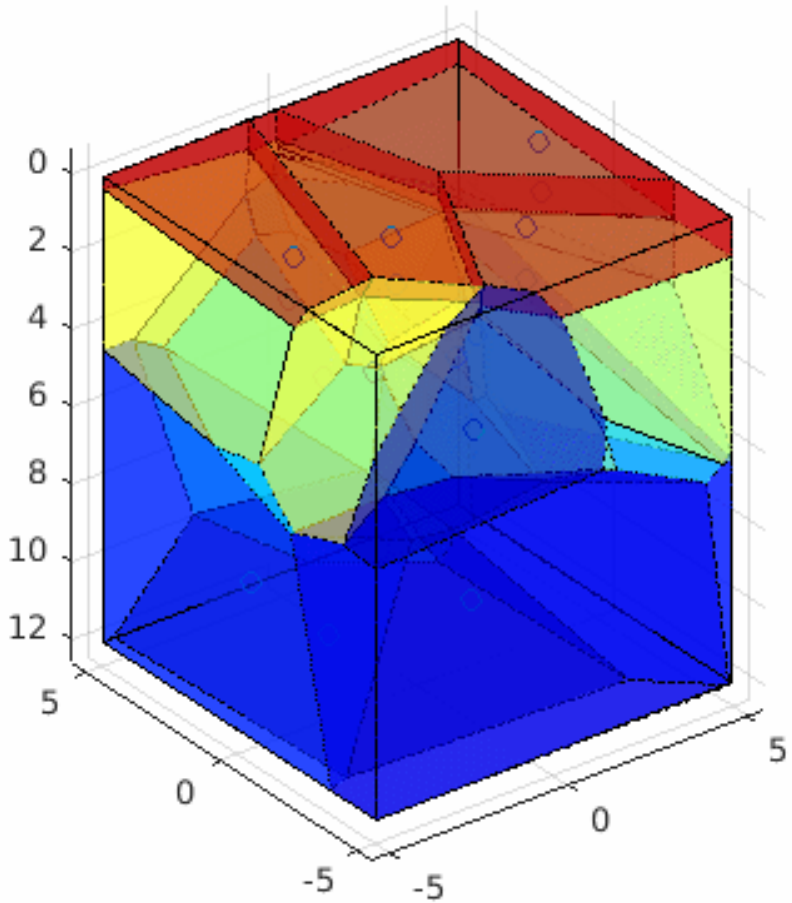




Synthetic test

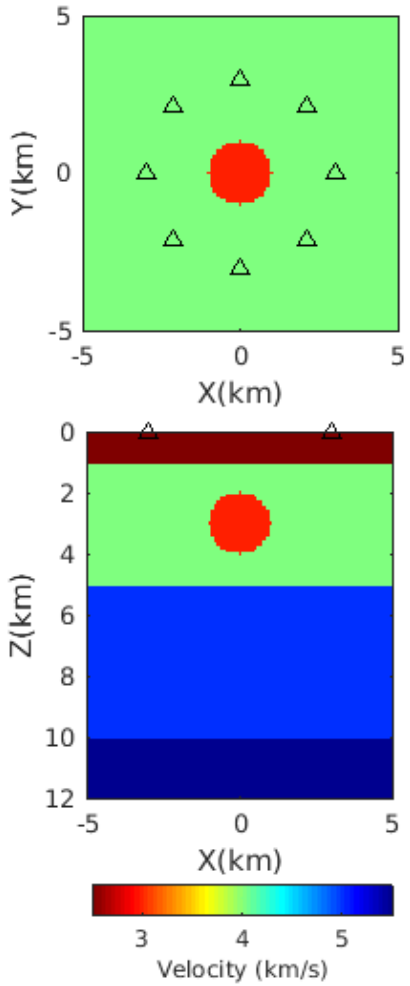


Markov chain



3D McMC

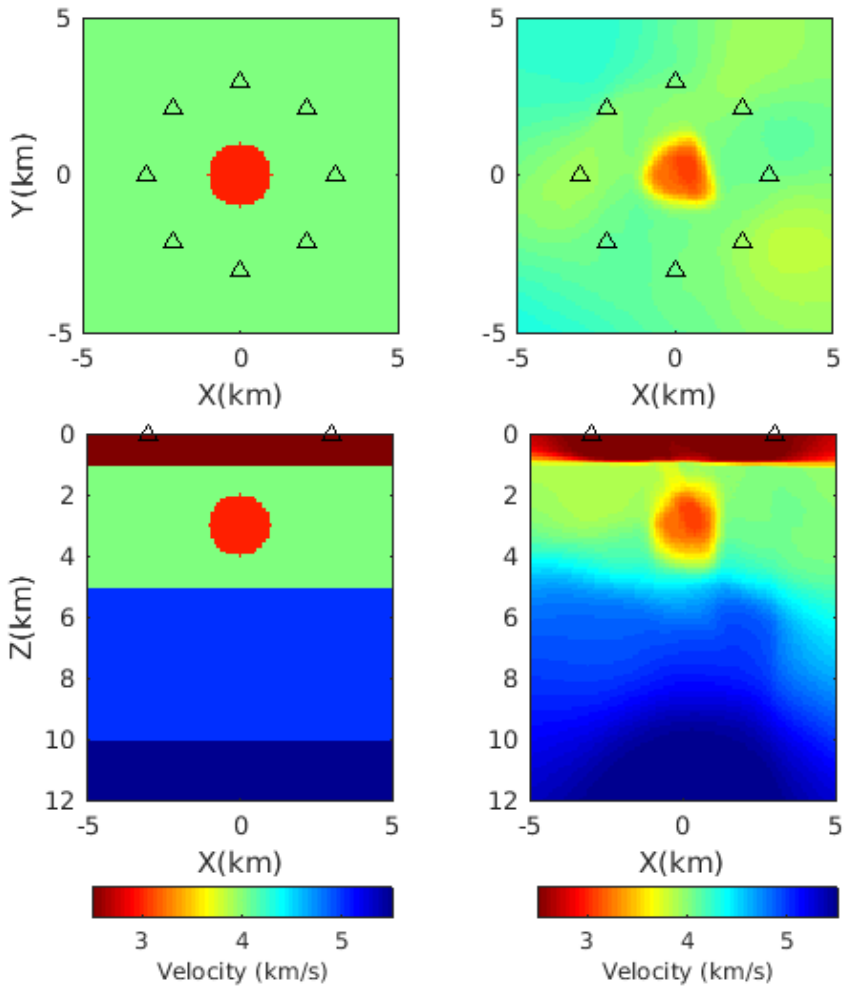
True



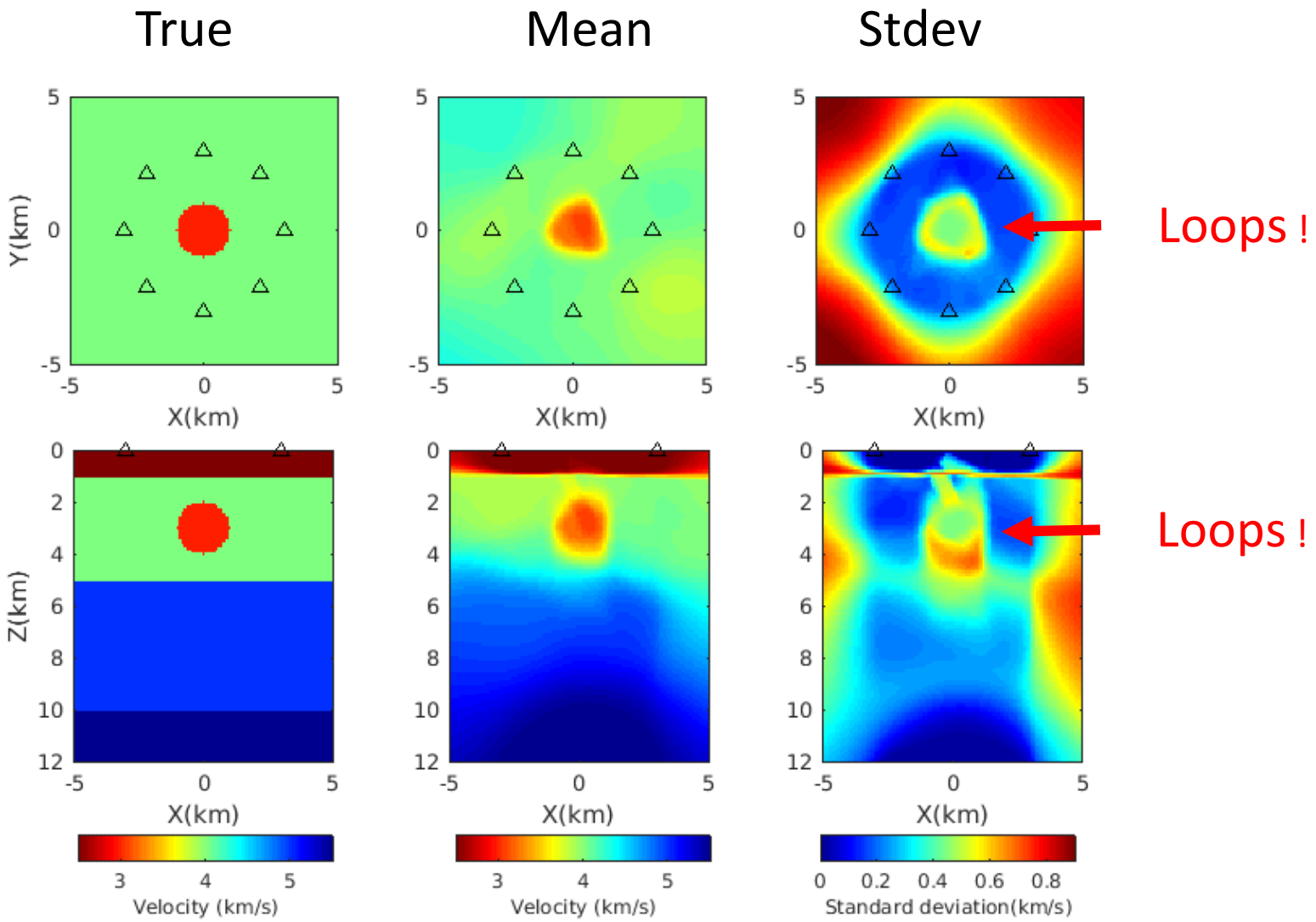
3D McMC

True

Mean



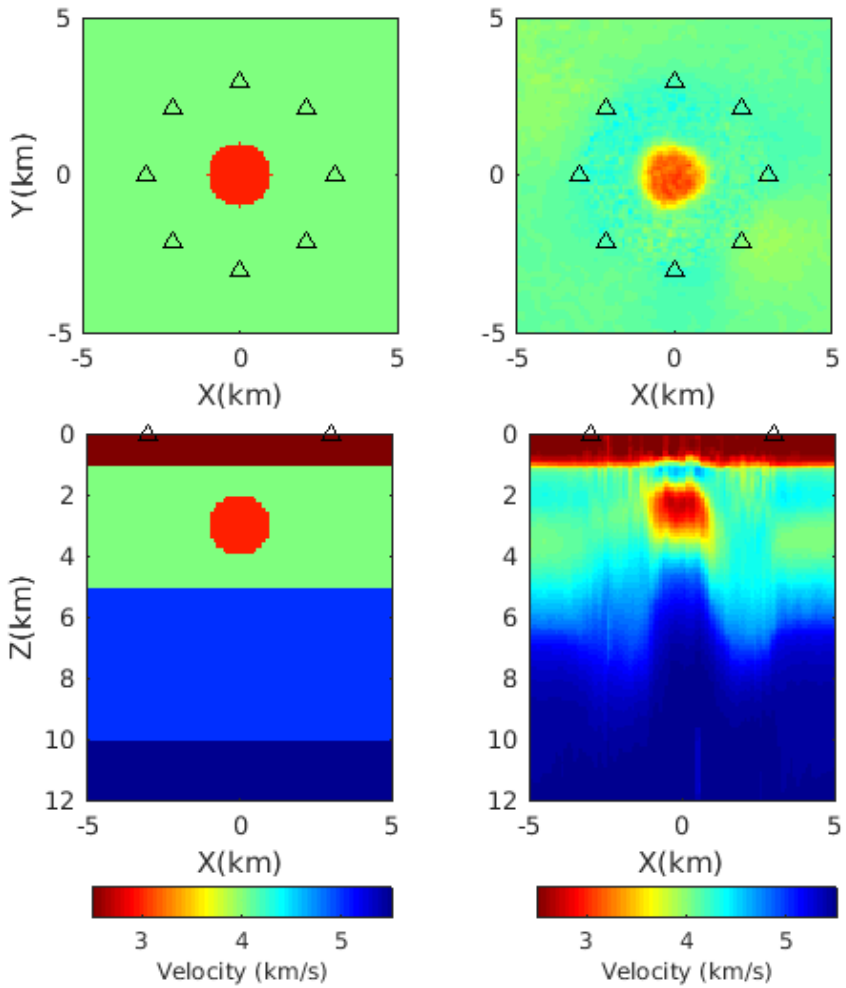
3D McMC


True
Mean
Stdev
Y(km)
X(km)
Z(km)
Velocity (km/s)
Standard deviation(km/s)
Loops !
Loops !

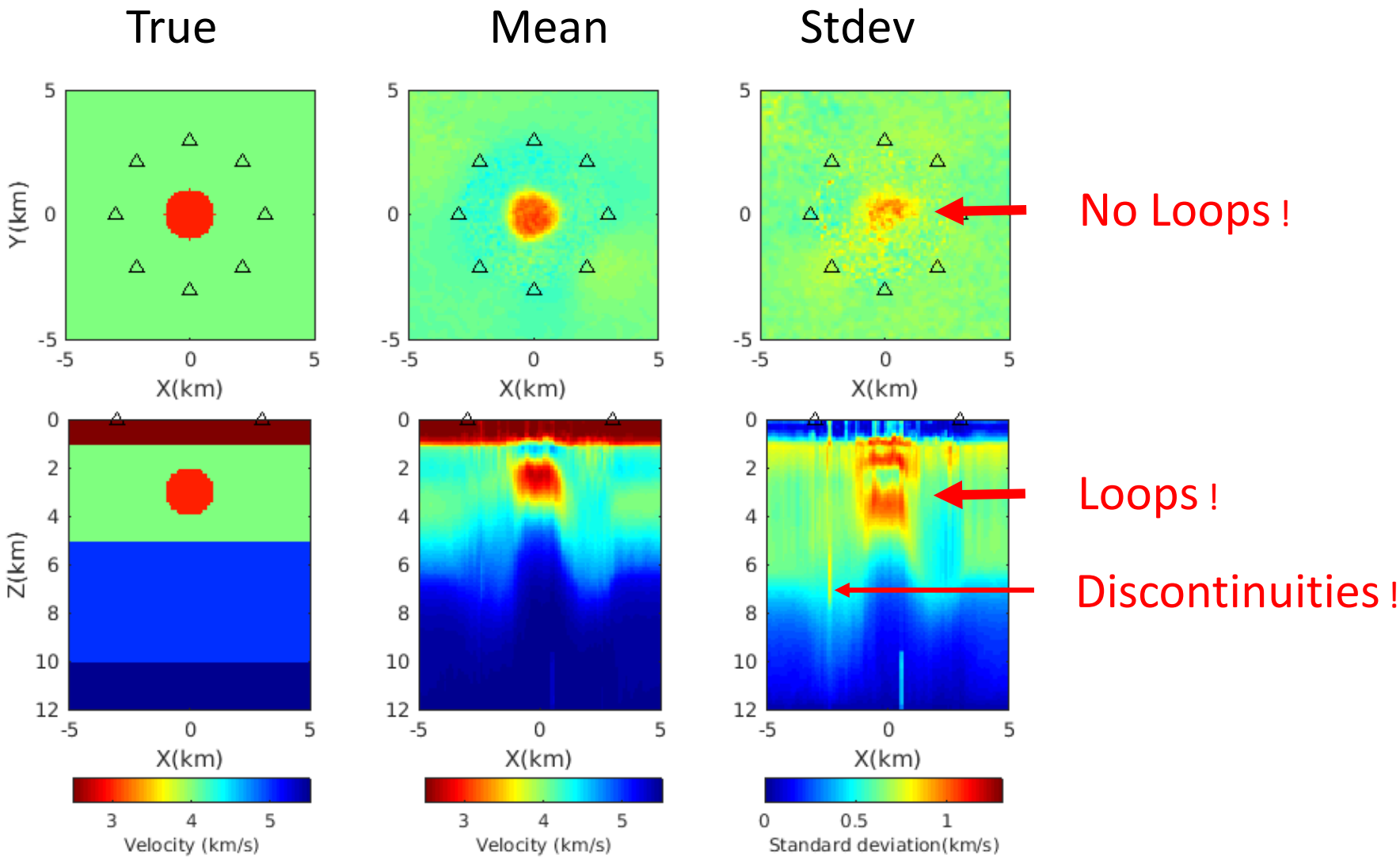
2-step McMC

True

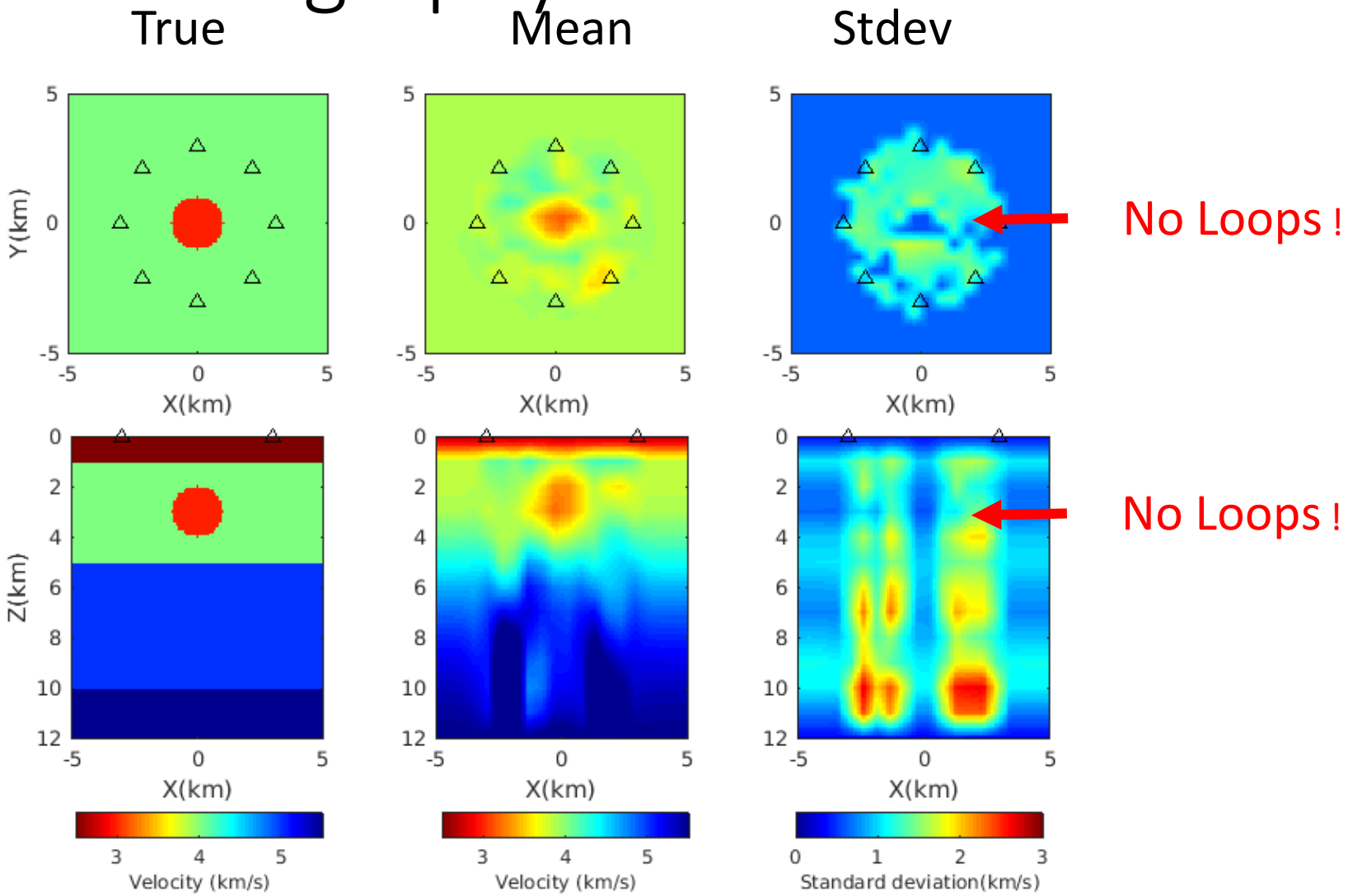
Mean



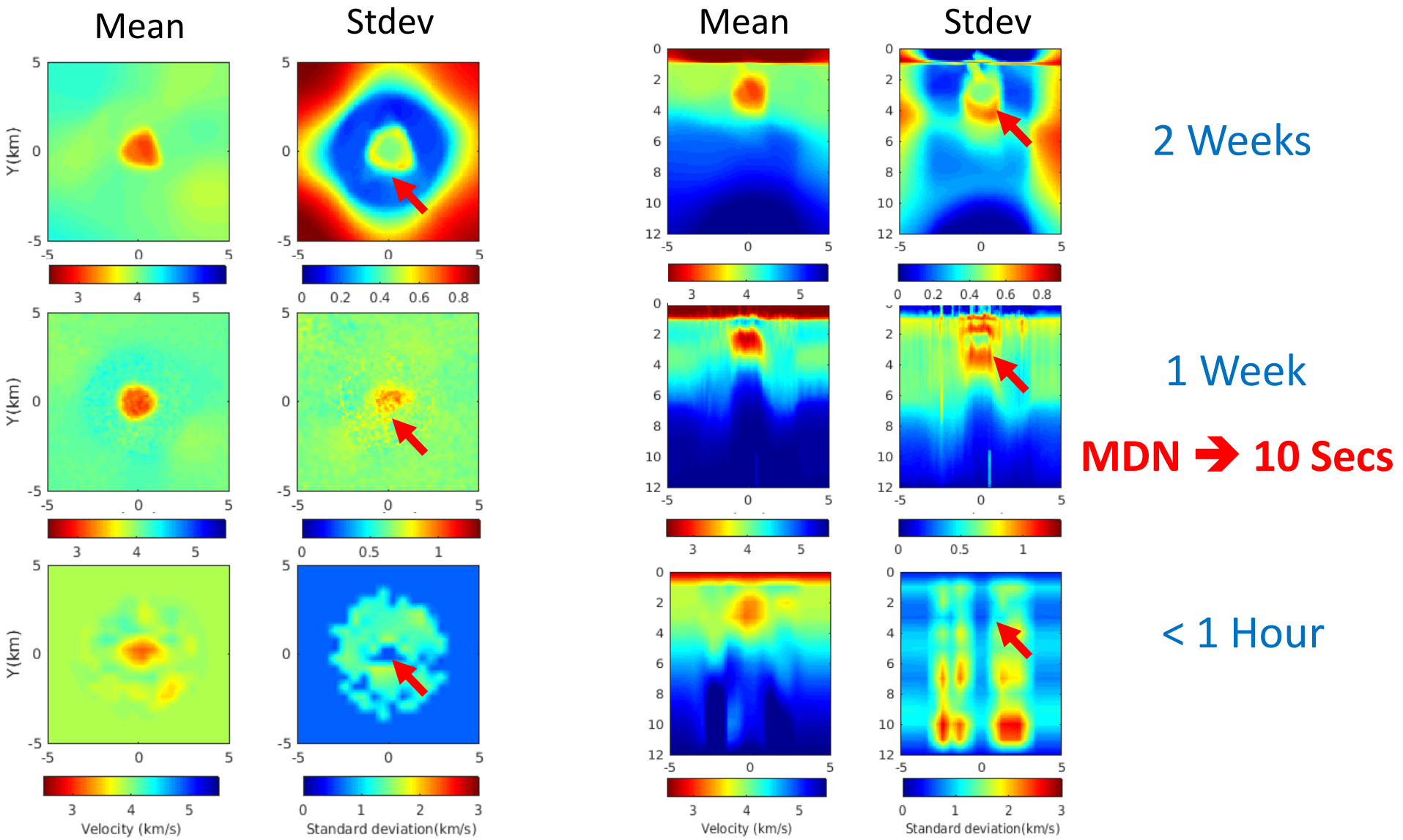
2-step McMC


True
Mean
Stdev
Y(km)
X(km)
Z(km)
Velocity (km/s)
Standard deviation(km/s)
No Loops !
Loops !
Discontinuities !

Linearised tomography



Comparison


Mean
Stdev
Mean
Stdev
3D McMC
2-step McMC
Linear inversion
Y(km)
Velocity (km/s)
Standard deviation(km/s)
2 Weeks
1 Week
MDN → 10 Secs
< 1 Hour

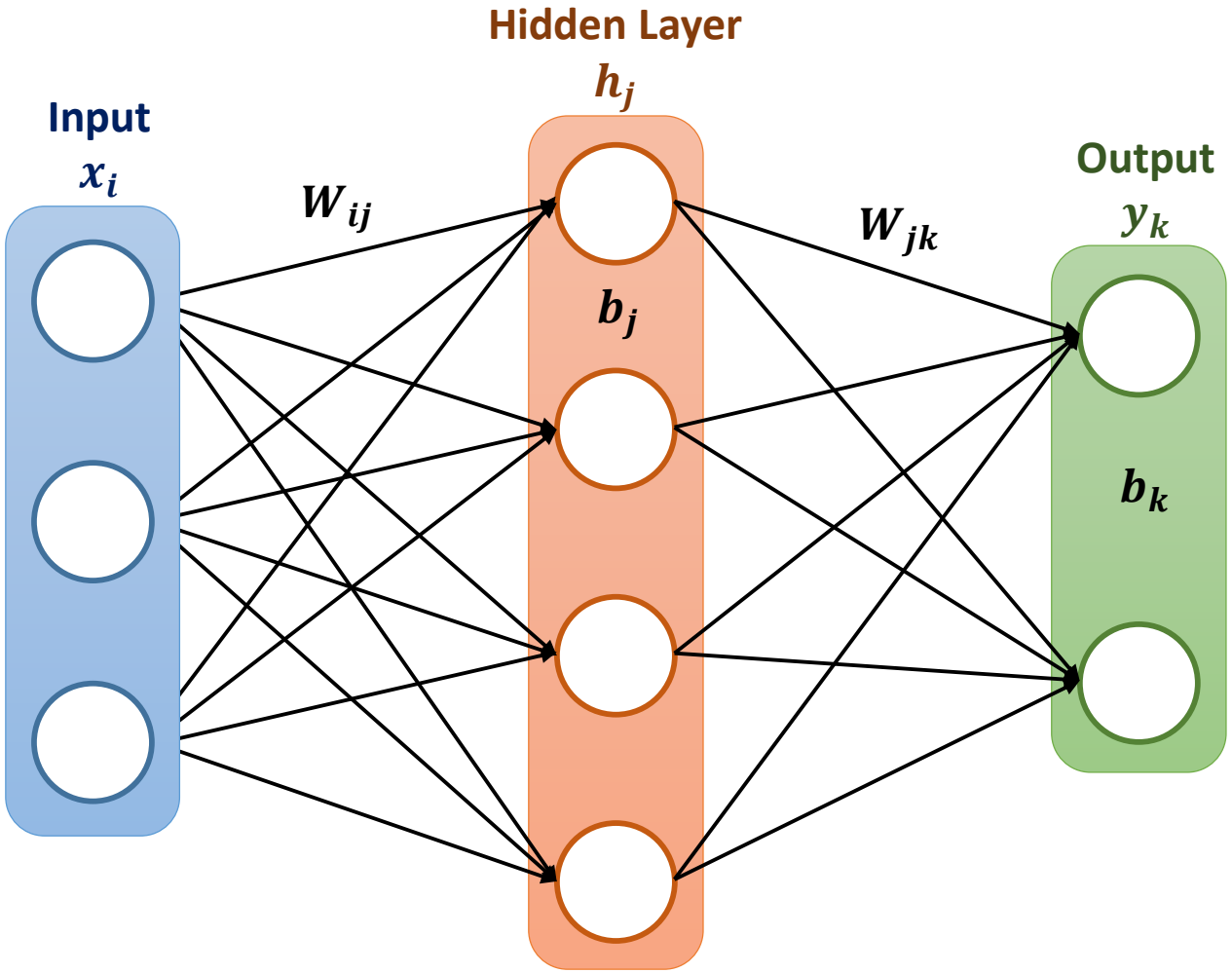
THE UNIVERSITY of EDINBURGH

edinburgh interferometry project

Neural Network Surface Wave Tomography

Stephanie Earp & Andrew Curtis

Neural Network



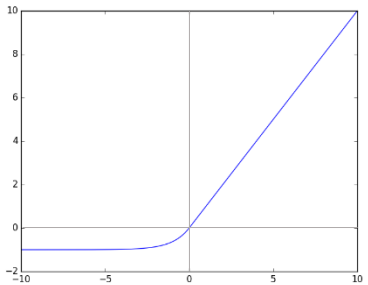
$$h_j = \sum_i W_{ij} x_i + b_j$$

$$h'_j = elu(h_j)$$

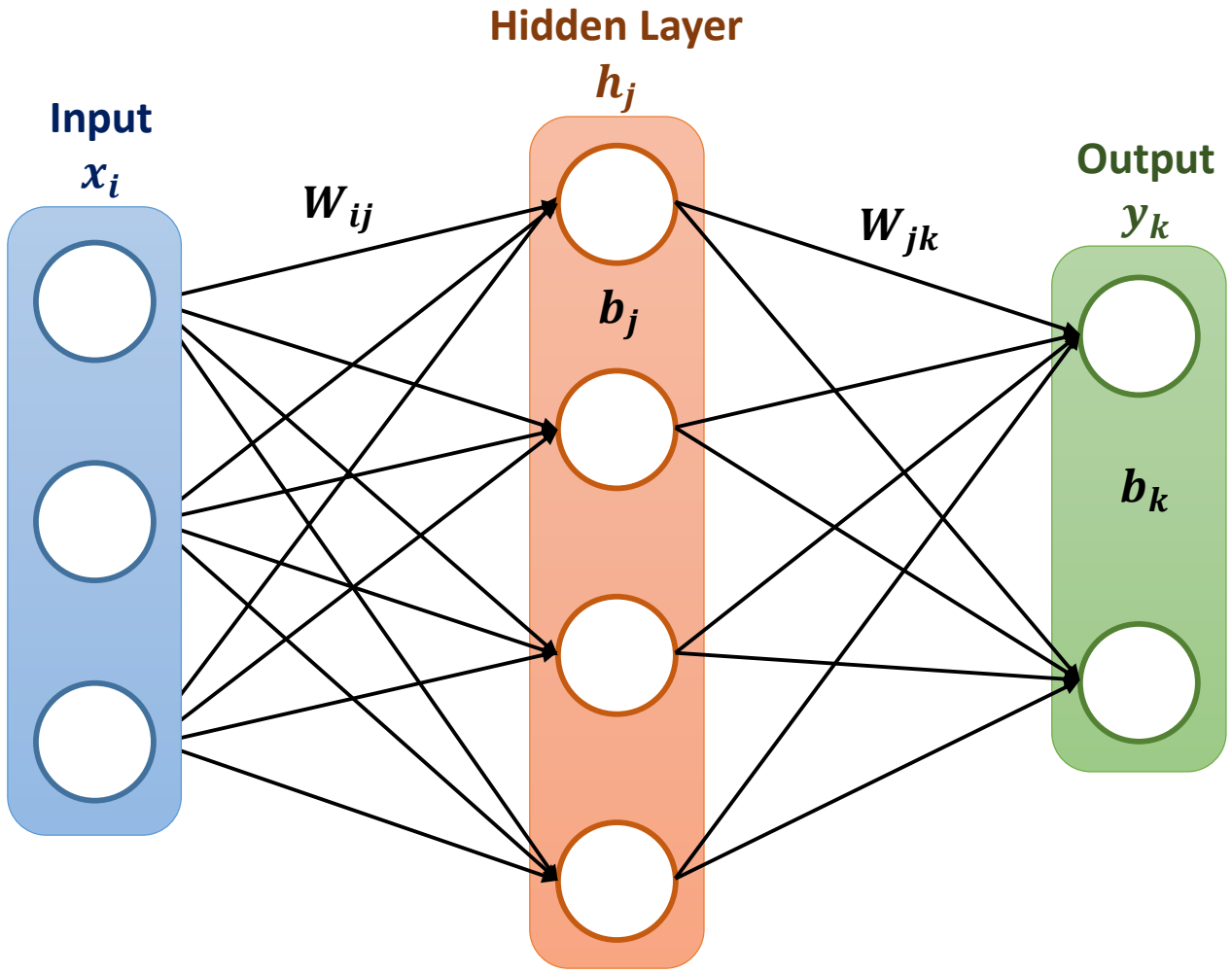
Activation function

$$y_k = \sum_j W_{jk} h'_j + b_k$$

$elu(x)$



Neural Network



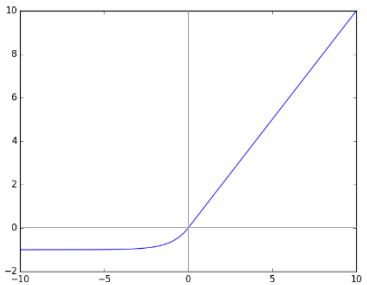
$$h_j = \sum_i W_{ij} x_i + b_j$$

$$h'_j = elu(h_j)$$

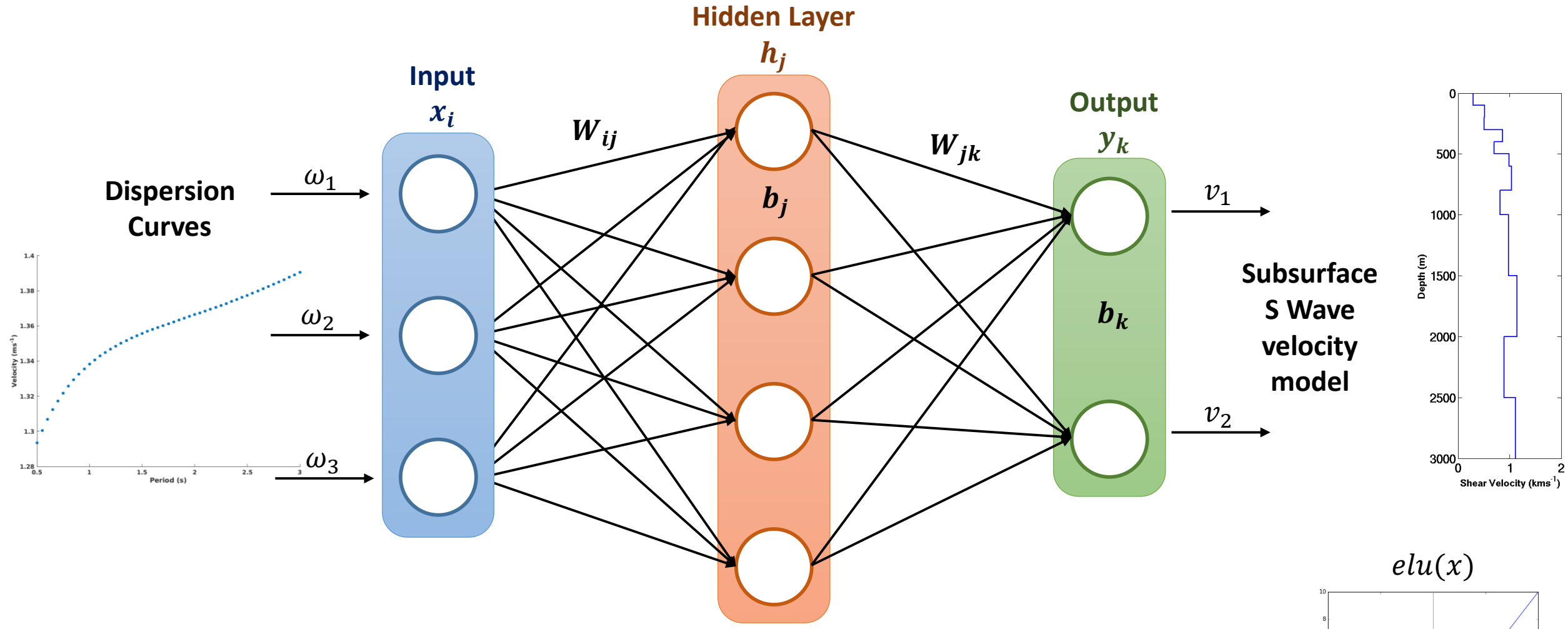
Activation function

$$y_k = \sum_j W_{jk} h'_j + b_k$$

$elu(x)$



Neural Network



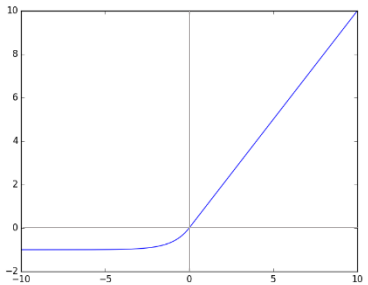
$$h_j = \sum_i W_{ij} x_i + b_j$$

$$h'_j = elu(h_j)$$

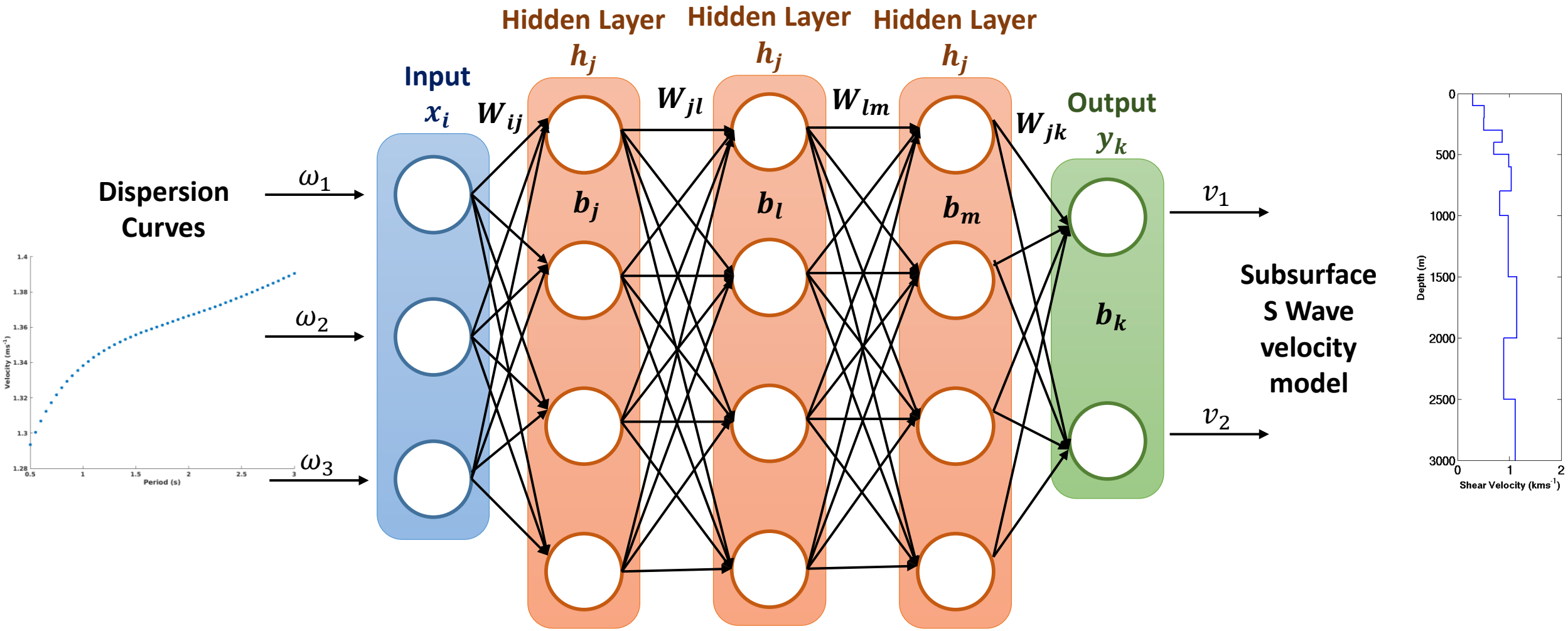
Activation function

$$y_k = \sum_j W_{jk} h'_j + b_k$$

$elu(x)$



Neural Network

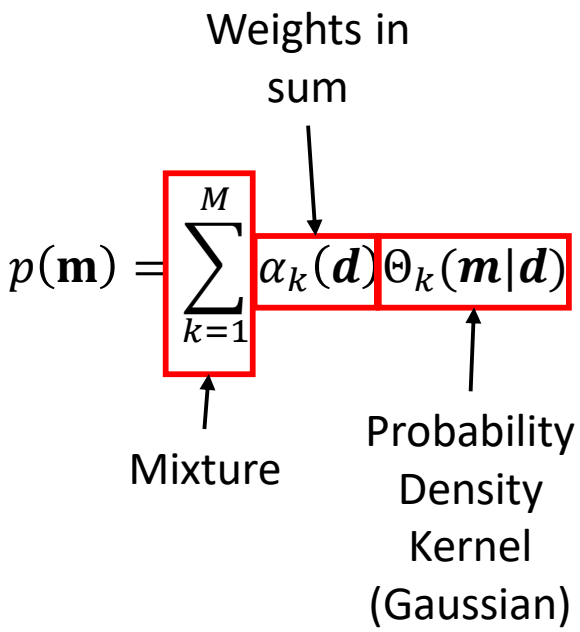


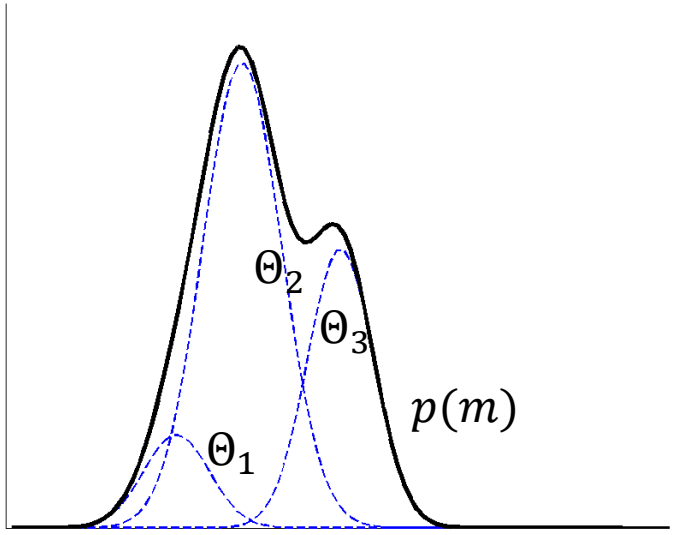
Recursive, convolutional, adversarial, etc.

Mixture Density Network

- Standard Neural Network (NN) gives no uncertainty information

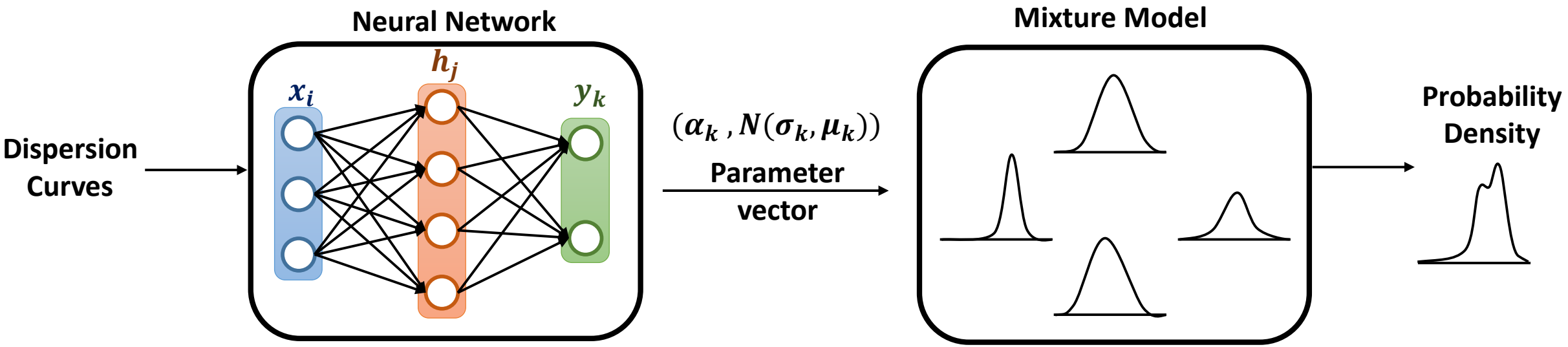
- Parameterise uncertainty using mixture of densities (→ MDN)



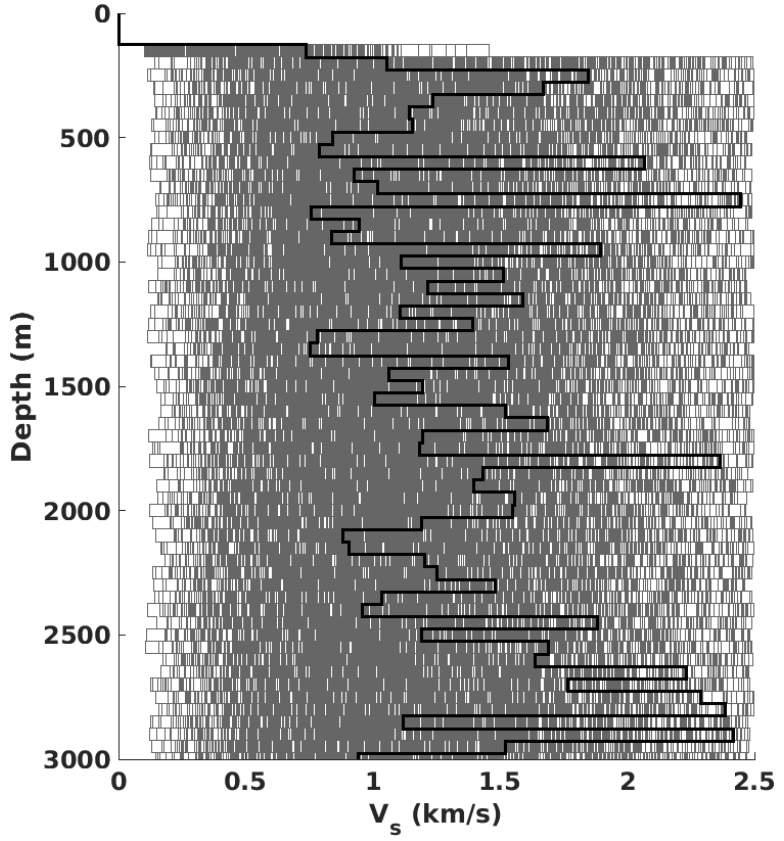


Mixture Density Network

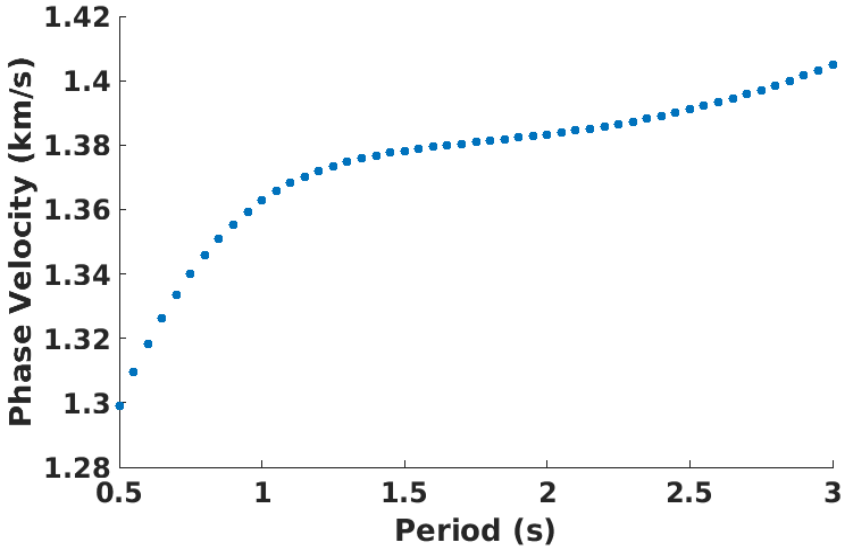
$$p(\mathbf{m}) = \sum_{k=1}^{M} \alpha_k(\boldsymbol{d})\Theta_k(\boldsymbol{m}|\boldsymbol{d})$$

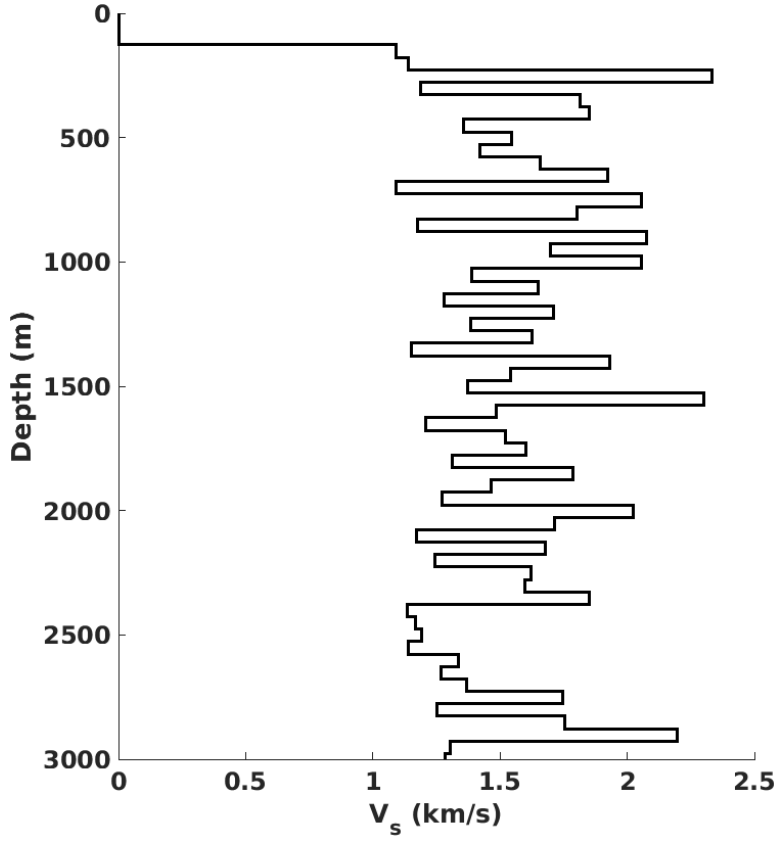


Method



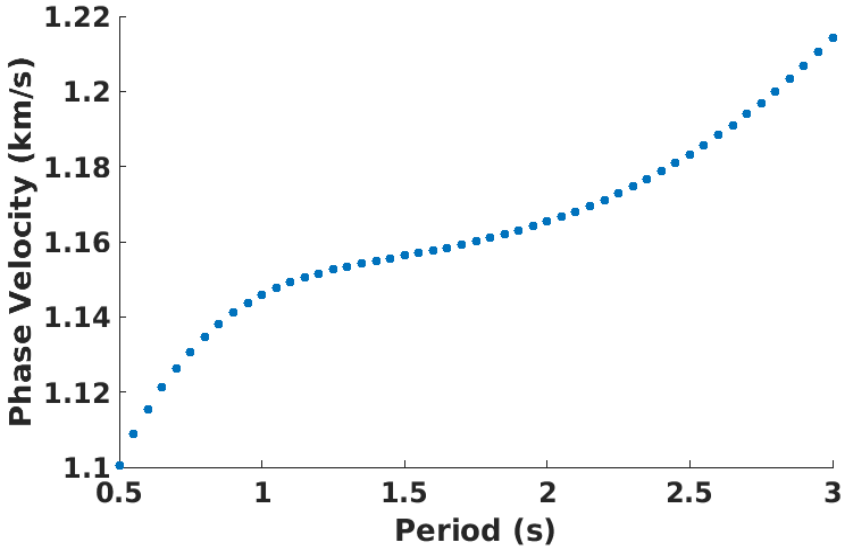
Method

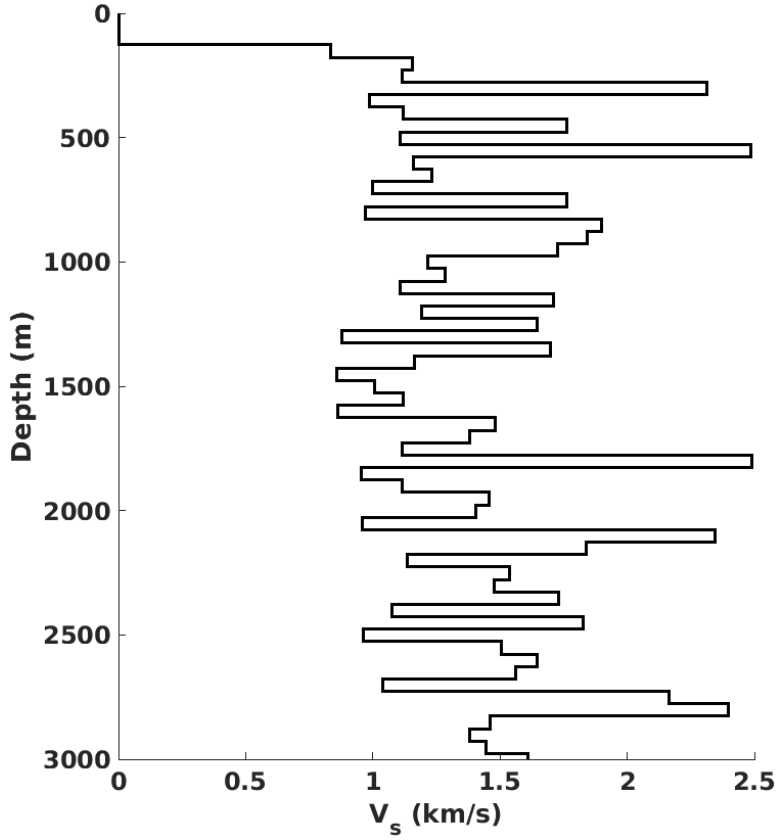




Forward Model

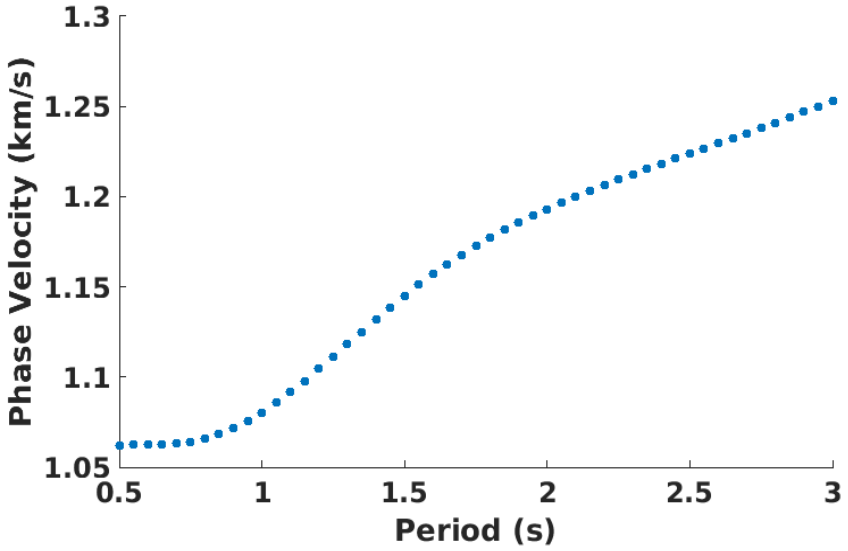
Method

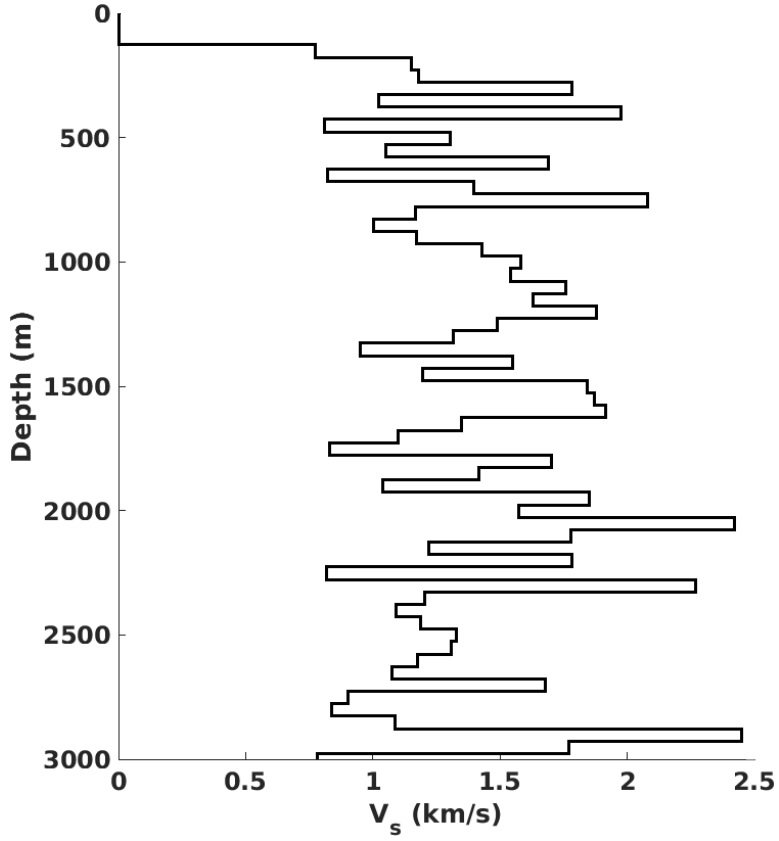




Forward Model

Method

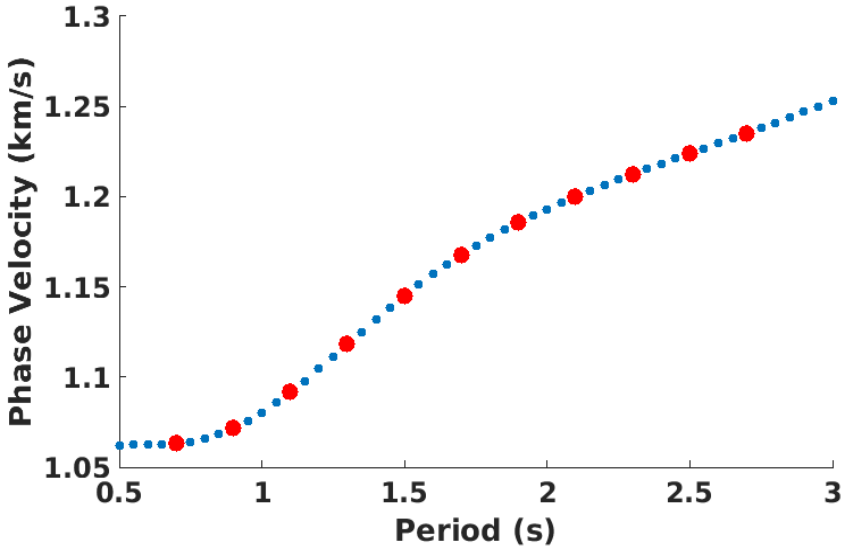


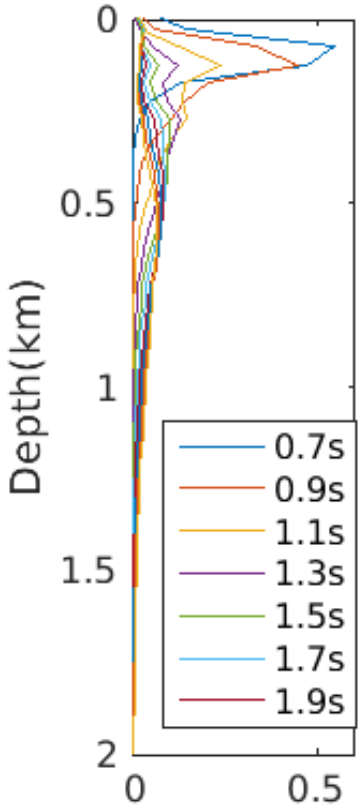


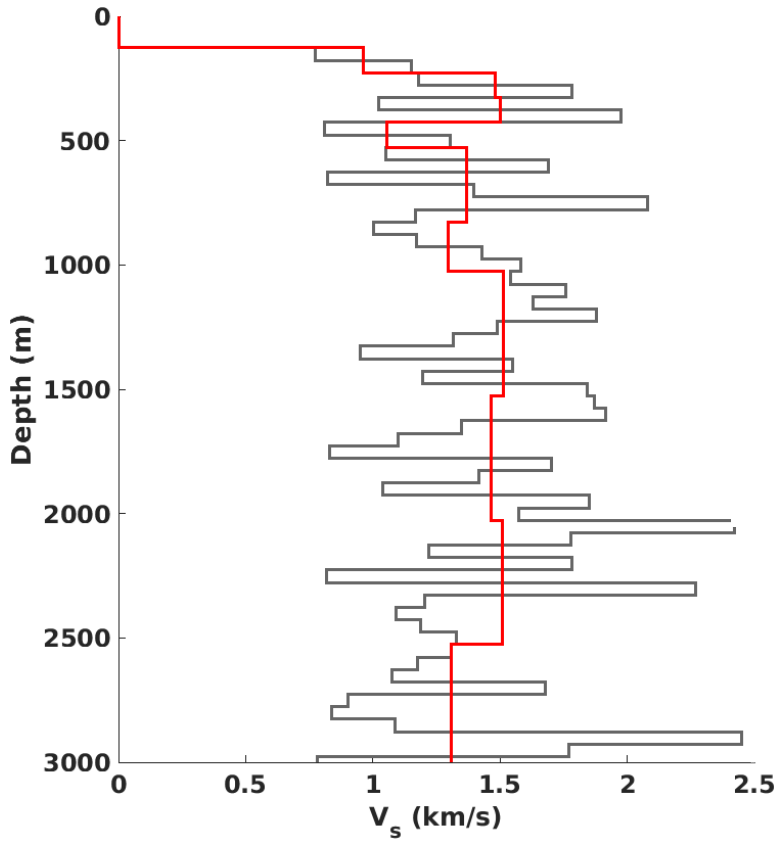
Forward Model

Method

Inversion

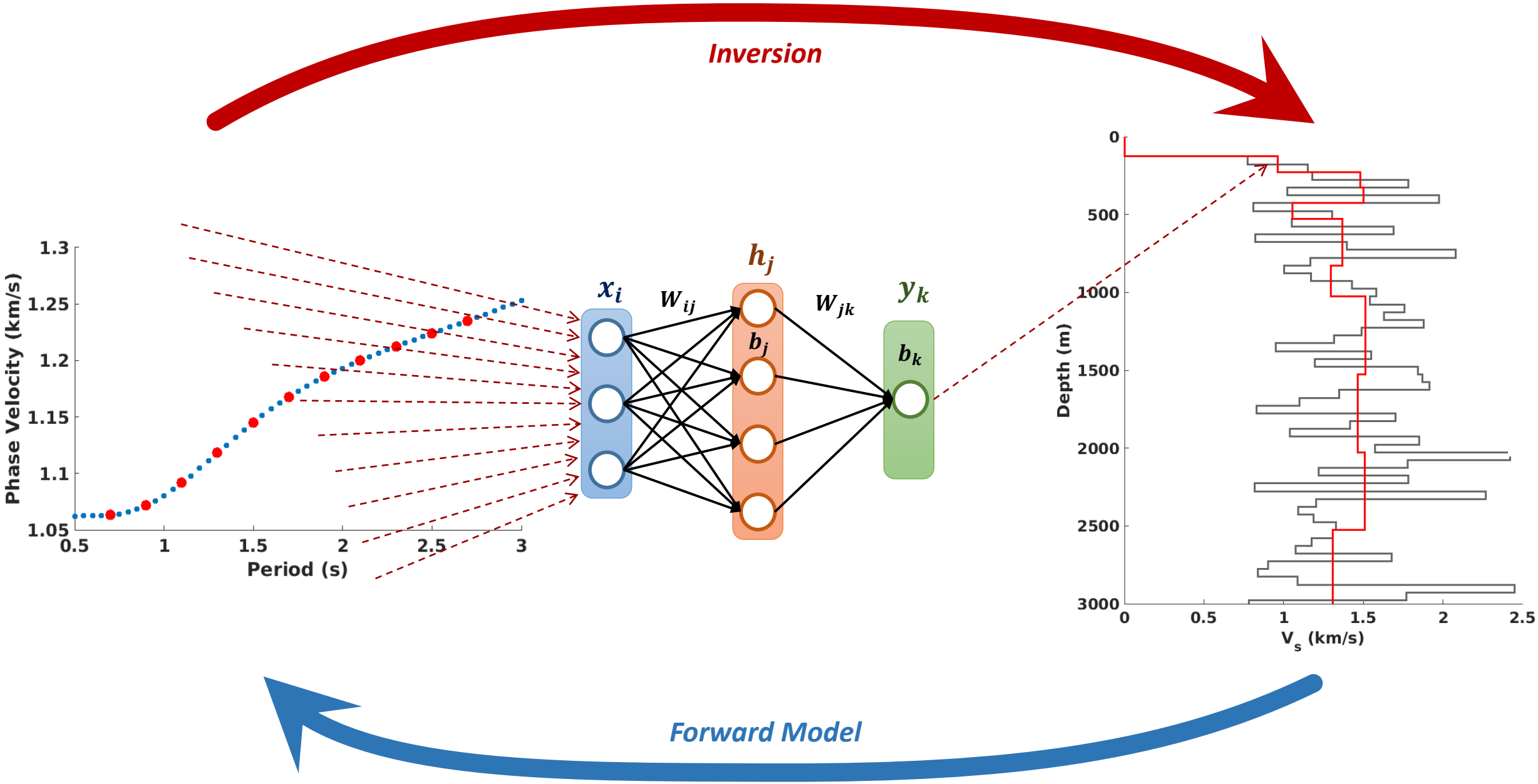




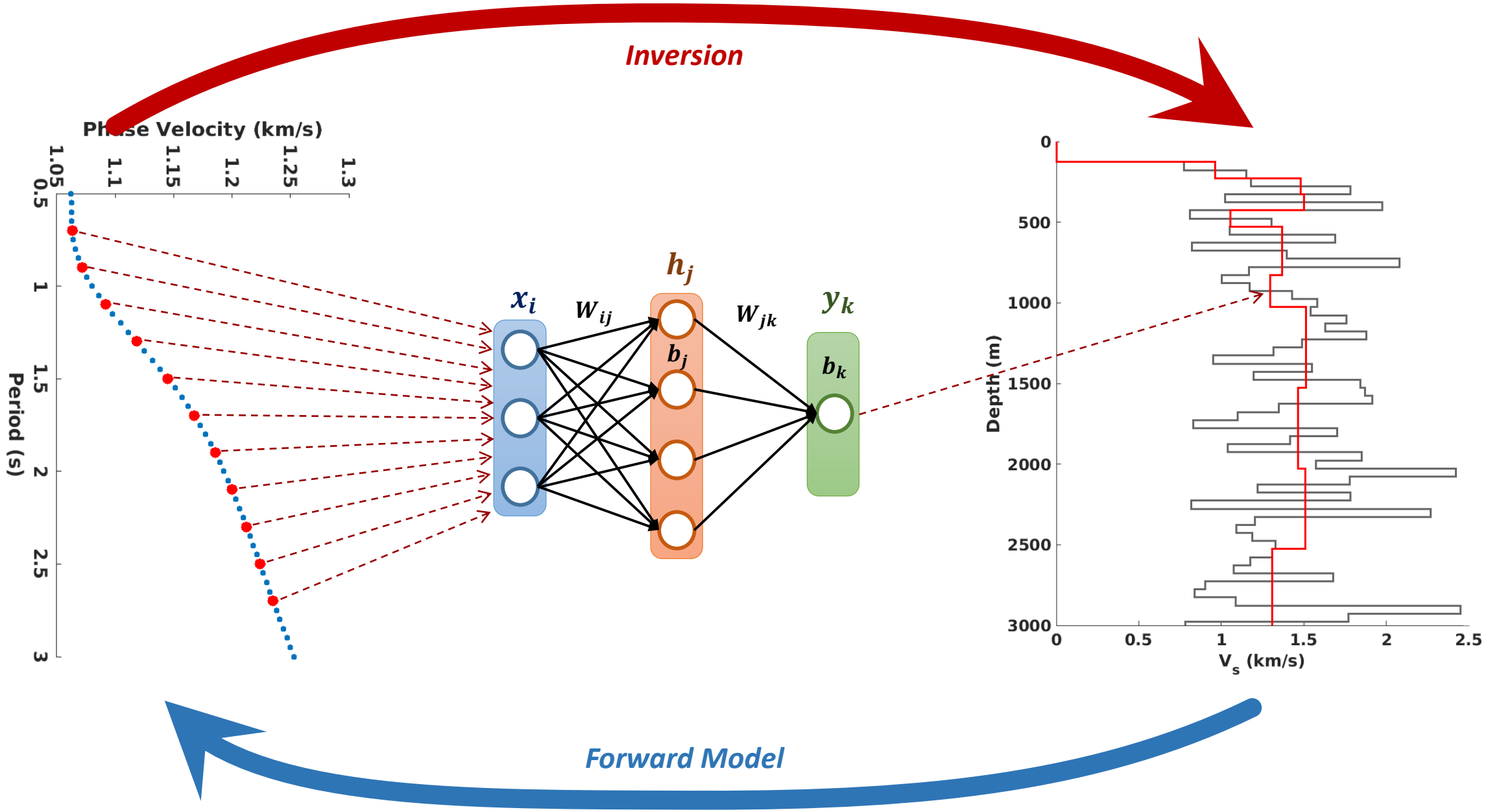


Forward Model

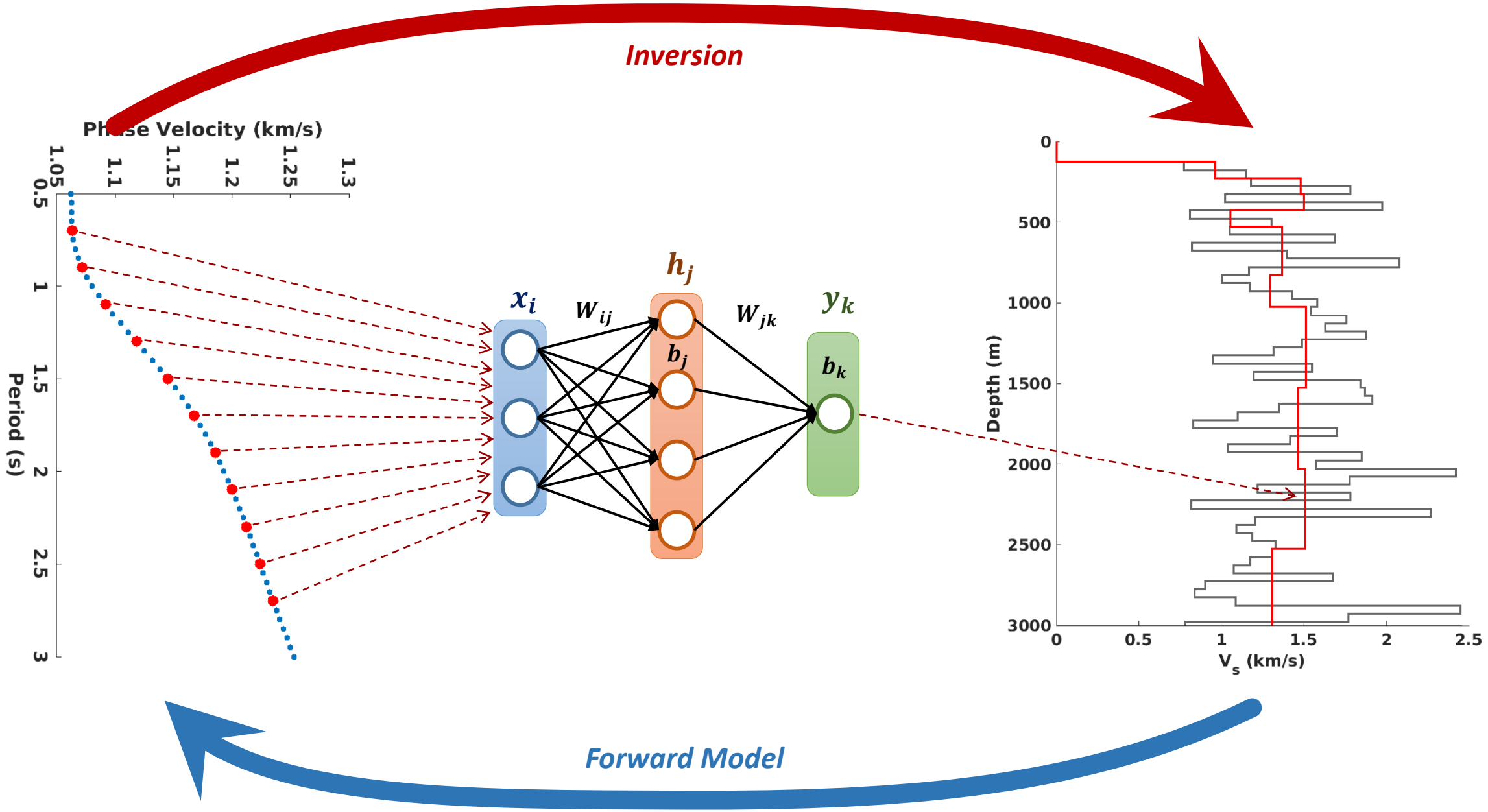
Method


Inversion
Phase Velocity (km/s)
Period (s)
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
V_s (km/s)
Forward Model

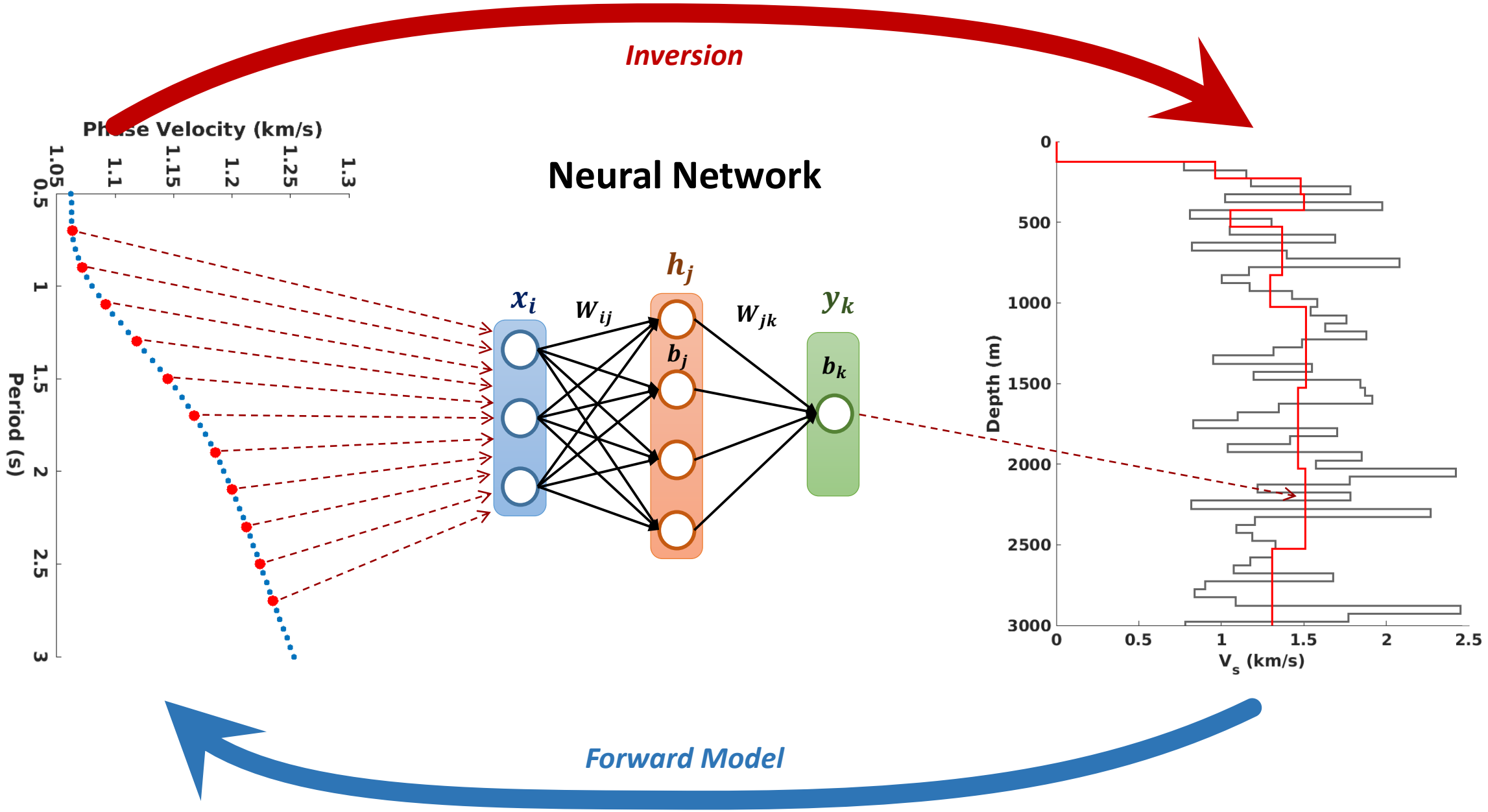
Method


Inversion
Phase Velocity (km/s)
1.05
1.1
1.15
1.2
1.25
1.3
Period (s)
0.5
1
1.5
2
2.5
3
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
0
500
1000
1500
2000
2500
3000
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

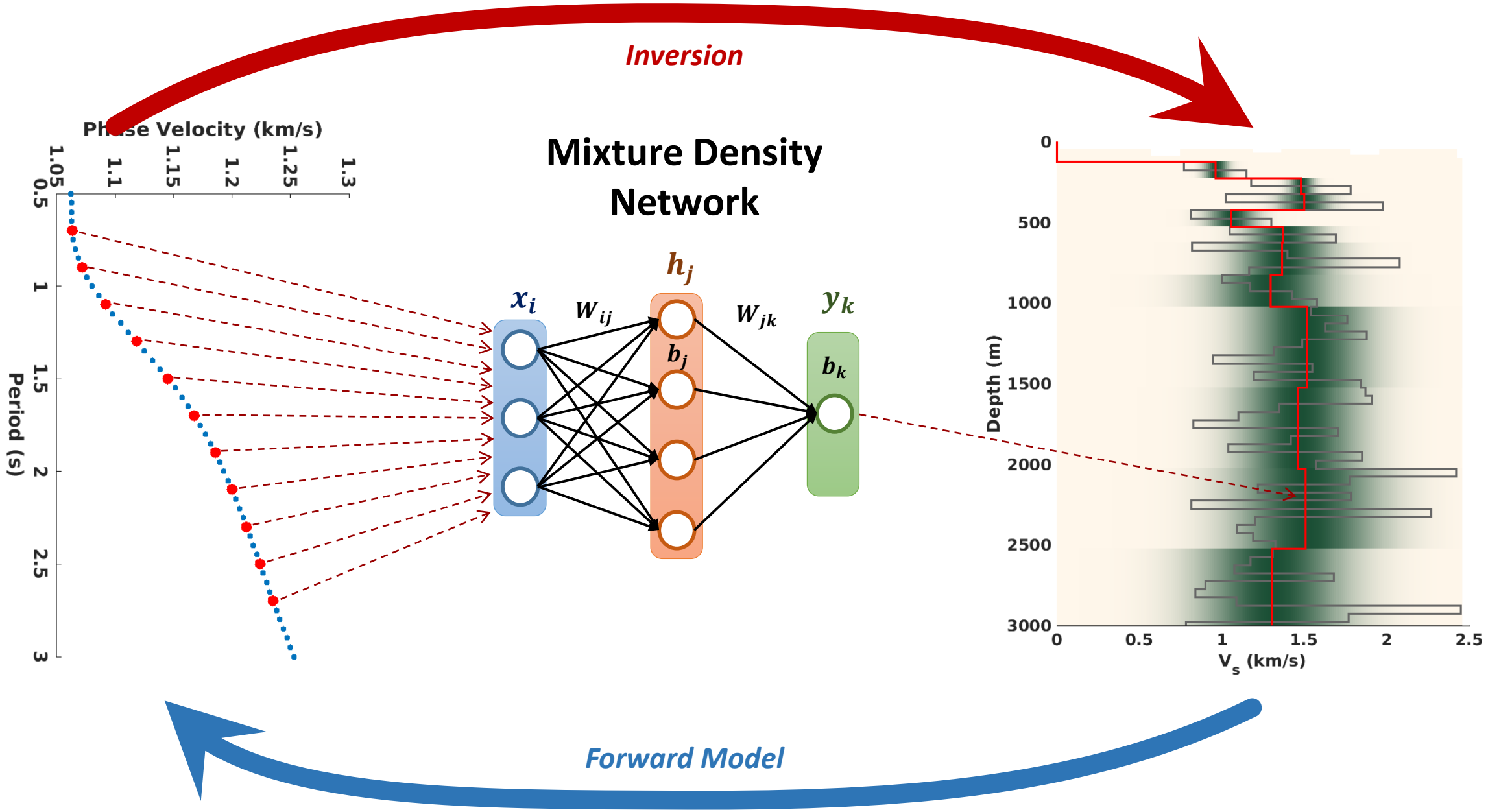
Method


Inversion
Phase Velocity (km/s)
1.05
1.1
1.15
1.2
1.25
1.3
Period (s)
0.5
1
1.5
2
2.5
3
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
0
500
1000
1500
2000
2500
3000
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

Method


Inversion
Phase Velocity (km/s)
1.05
1.1
1.15
1.2
1.25
1.3
Period (s)
0.5
1
1.5
2
2.5
3
Neural Network
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
0
500
1000
1500
2000
2500
3000
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

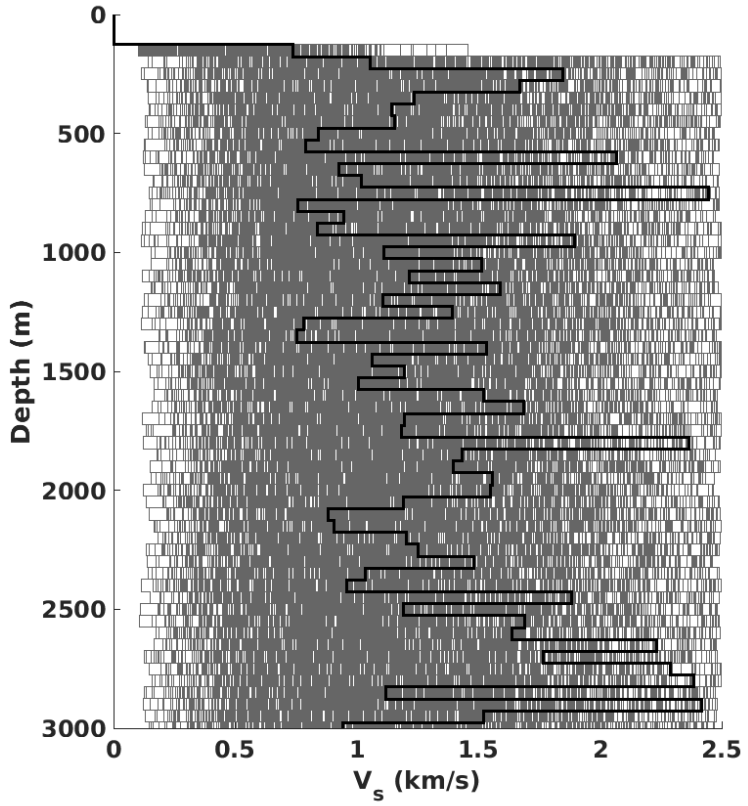
Method


Inversion
Phase Velocity (km/s)
1.05
1.1
1.15
1.2
1.25
1.3
Period (s)
0.5
1
1.5
2
2.5
3
Mixture Density Network
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
0
500
1000
1500
2000
2500
3000
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

Method

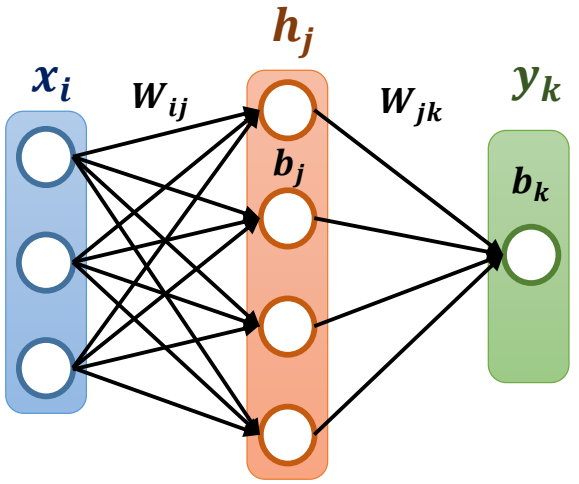
- Generate 1,000,000 models ➔ phase velocities

- Training set – 850,000 random models
- Validation set – 100,000 random models
  - Used to check network
  - ‘Sees’ model during training
- Test set – 50,000 random models
  - Checks model on ‘unseen’ data



Method

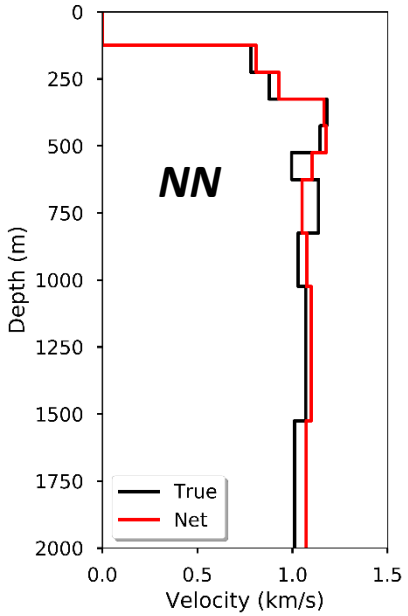
- Optimise hyperparameters for feed-forward network
  - Number of hidden layers
  - Nodes in hidden layer
  - Learning Rate
  - Activation function
  - Weights and bias initialisation

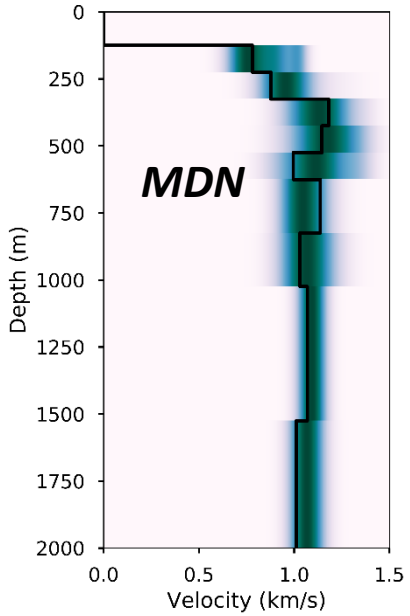


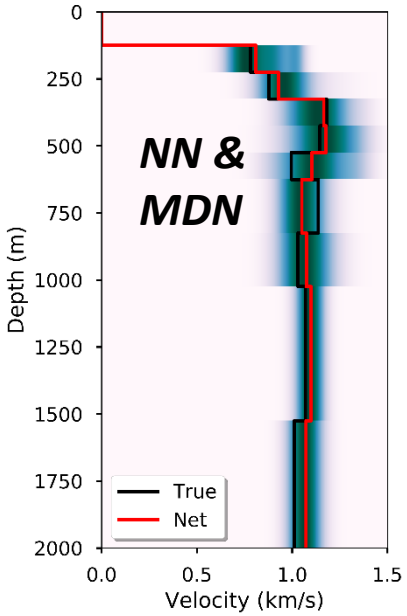
- Selected network parameters with lowest cost value for test data

Synthetic Results

- Trained
  - 118 Neural Networks for every depth layer
  - 135 Mixture Density Networks for each layer
  - MDN contains a mixture of 6 Gaussians
- Selected networks with lowest cost value for test data

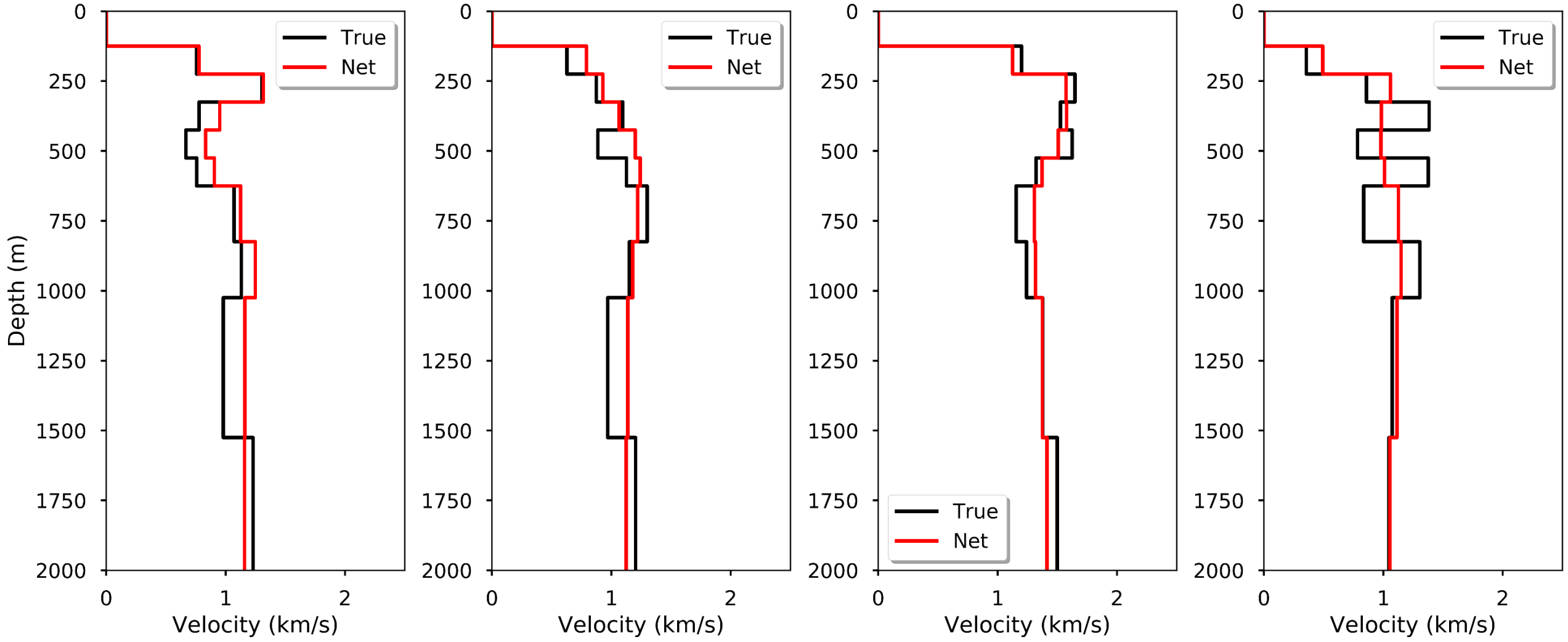






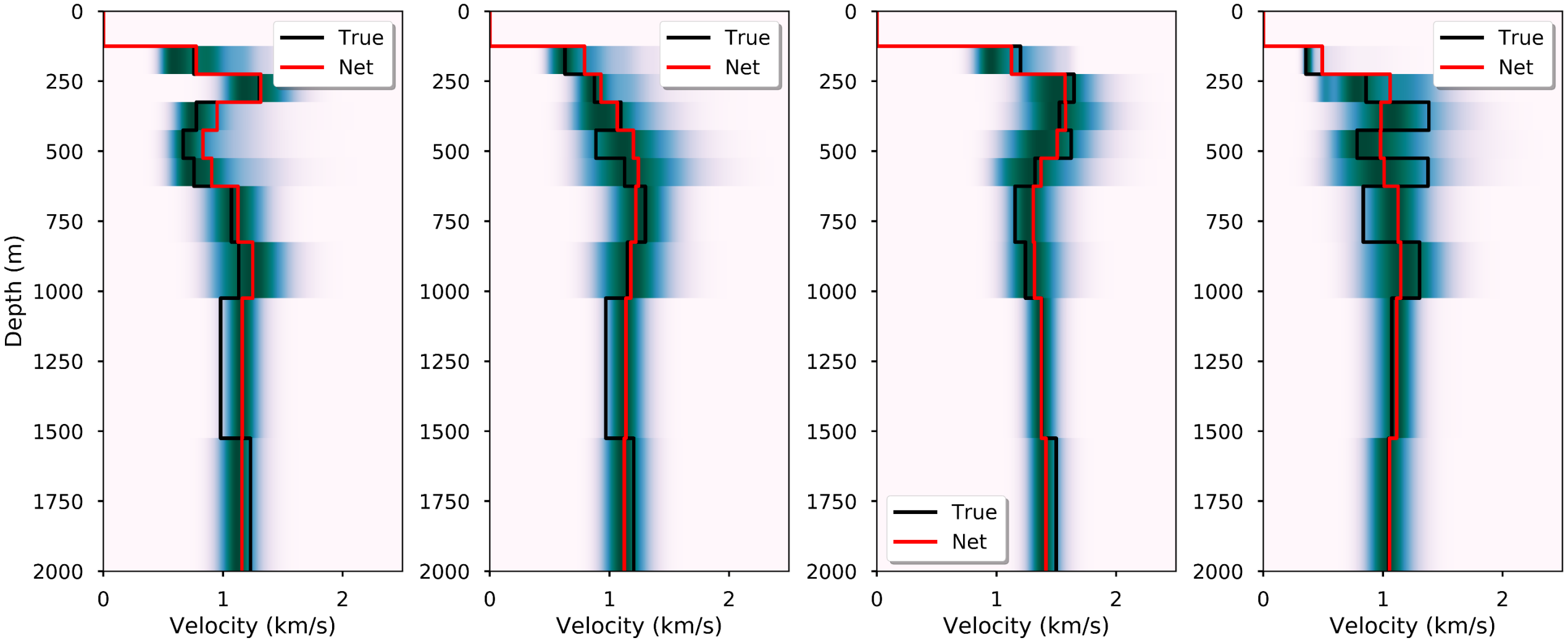
Synthetic Results

Neural Network



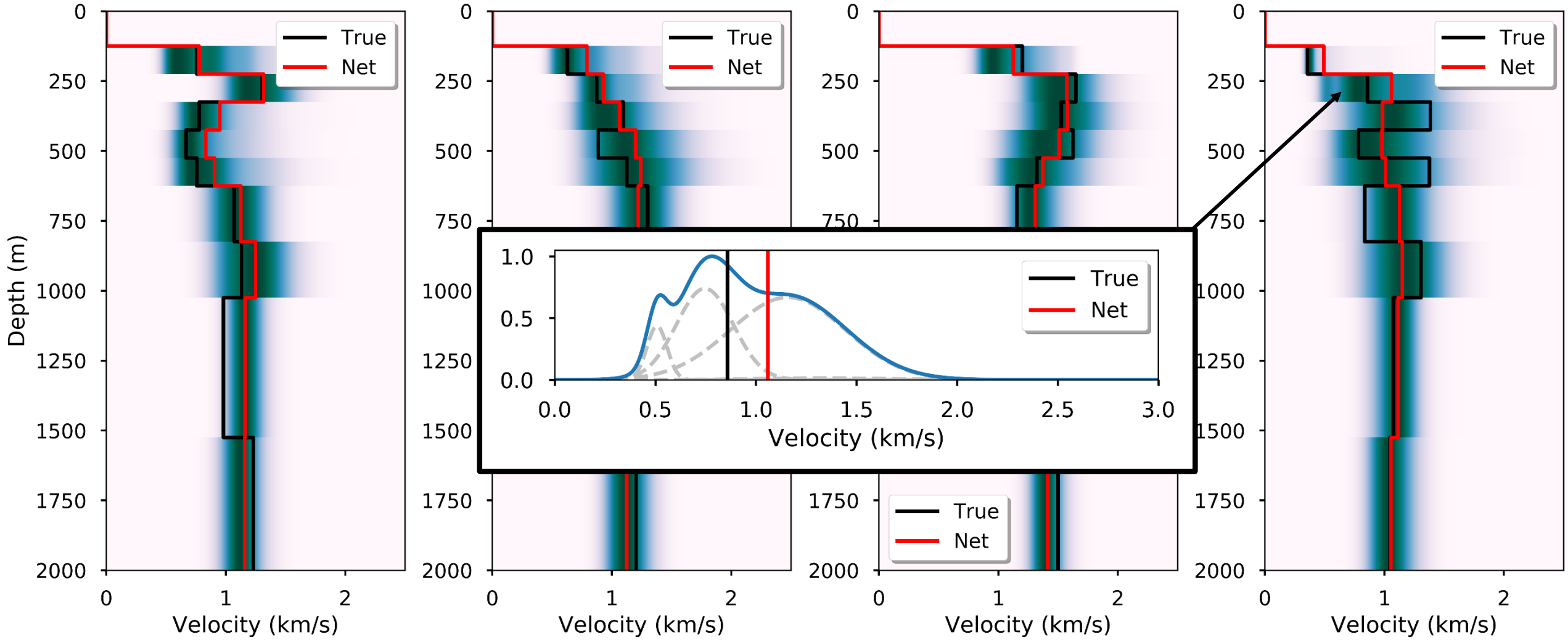
Synthetic Results

Mixture Density and Neural Network



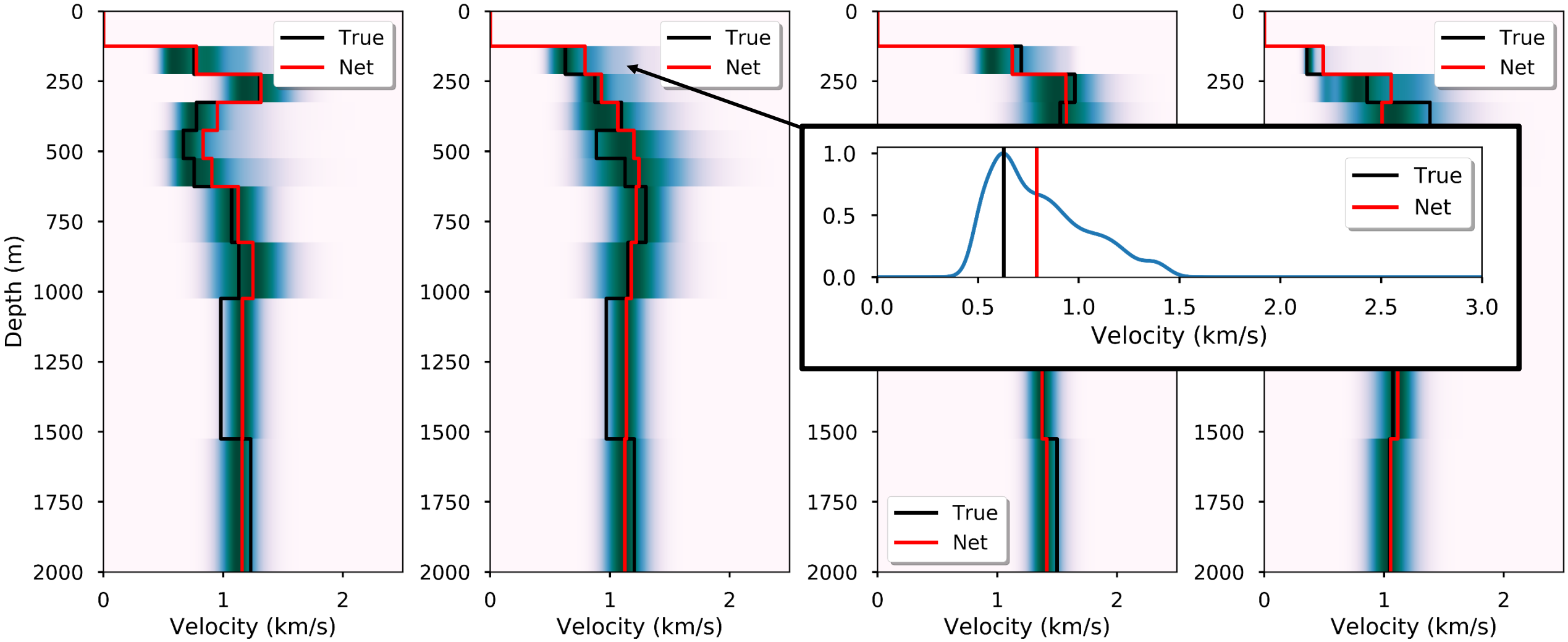
Synthetic Results

Mixture Density and Neural Network



Synthetic Results

Mixture Density and Neural Network

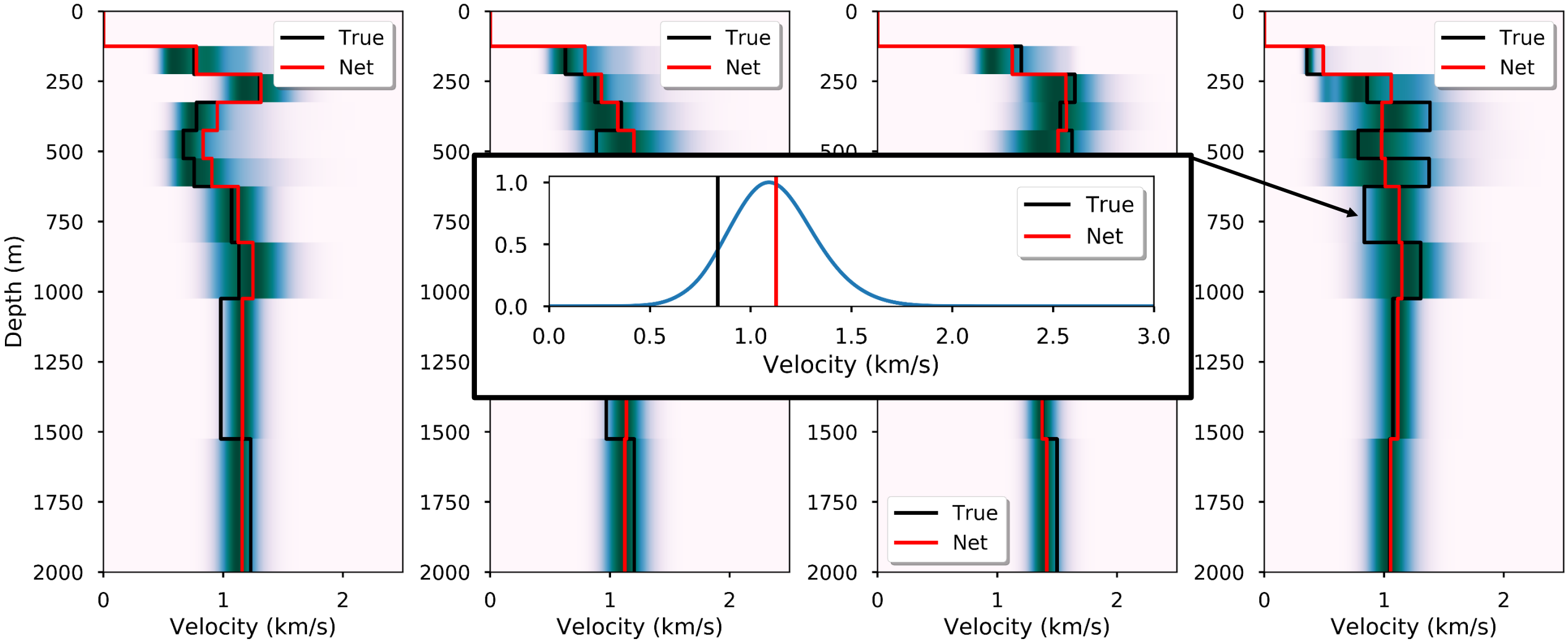


Explains Devilee & Meier et al.

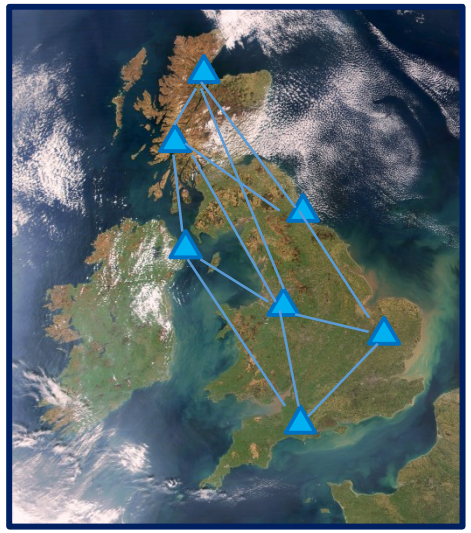
Synthetic Results

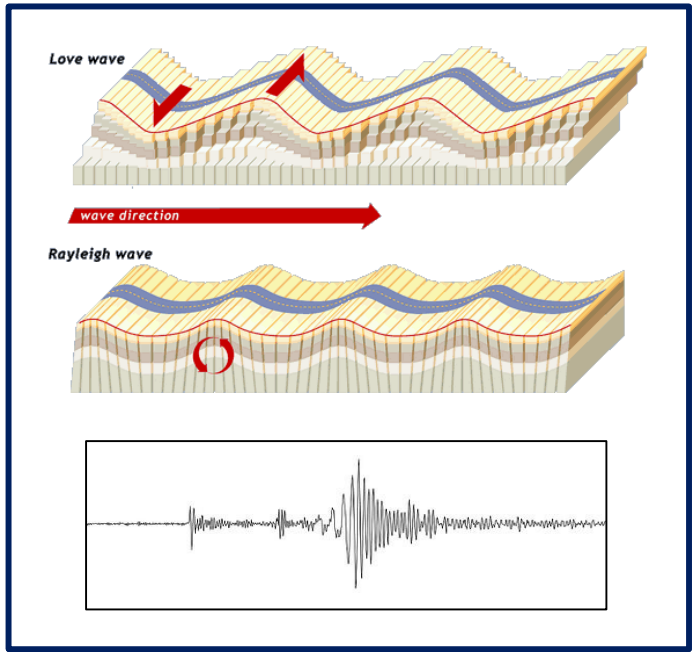
Mixture Density and Neural Network

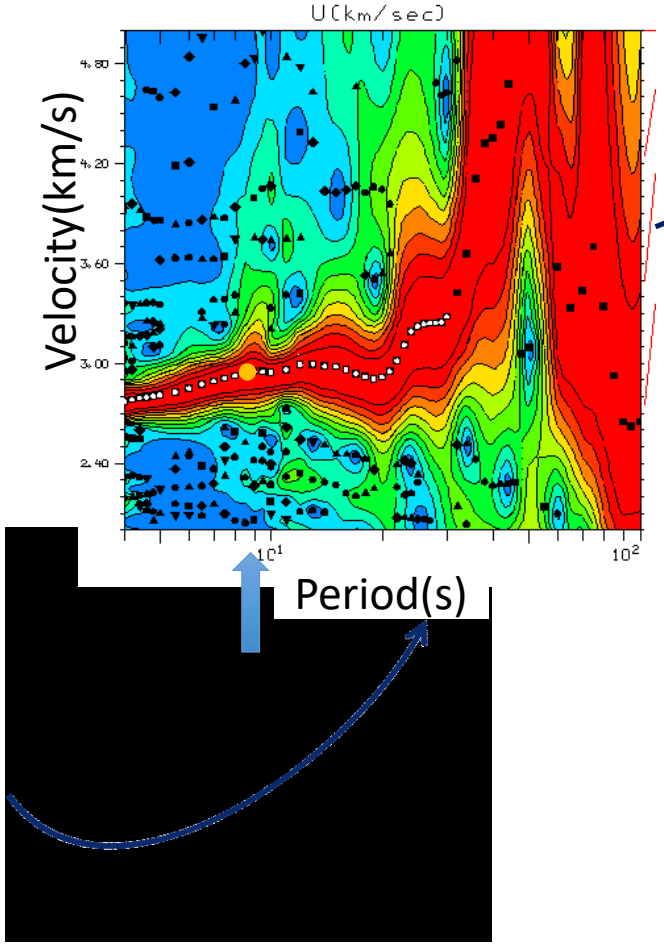
Fully nonlinear probabilistic Inversion in ~1 second



Seismic Surface Wave Tomography : typical workflow

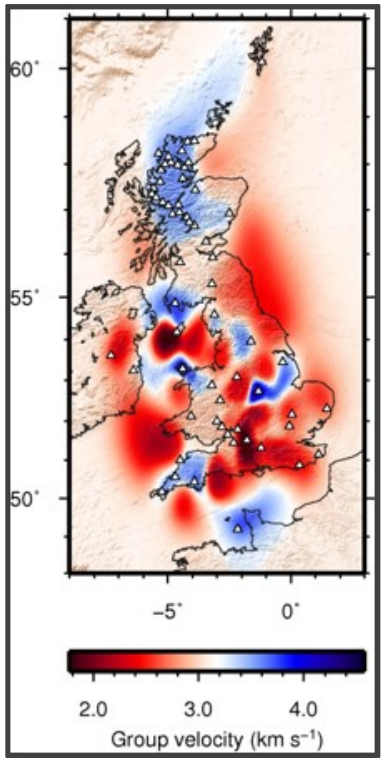






Travel-Time Tomography

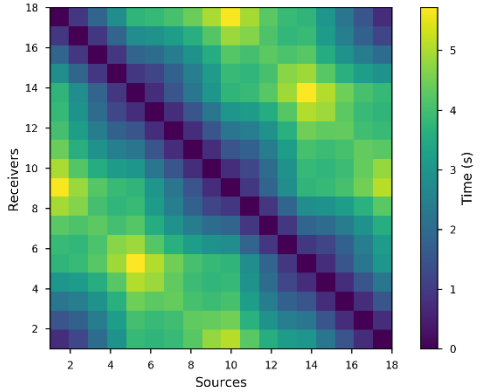
9s Group Velocity



Neural Network Travel-time Tomography

Travel-time Tomography

Source-Receiver travel-times

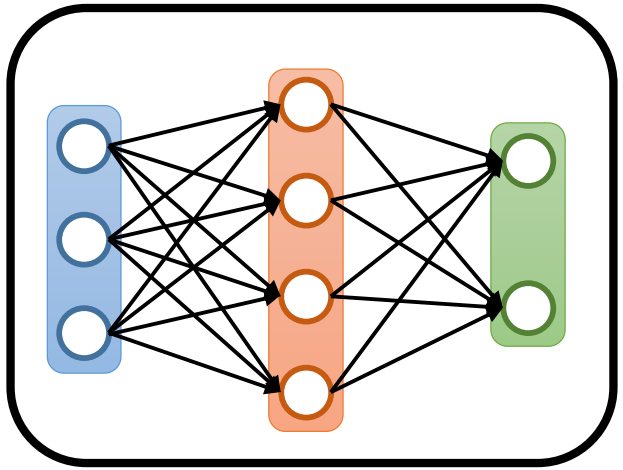


t_1

t_2

t_3

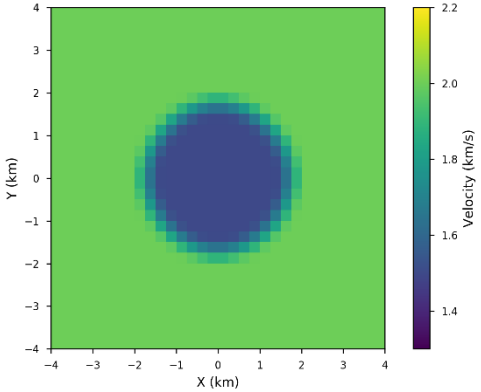
Neural Network



v_1

v_2

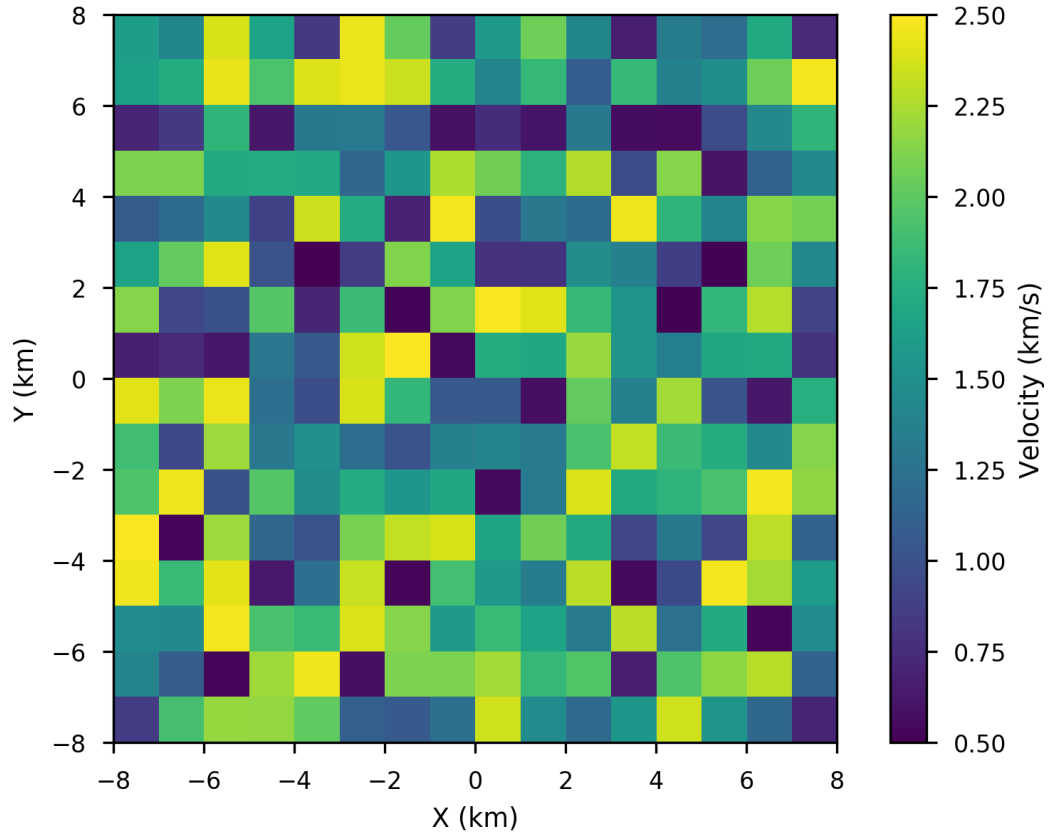
Velocity model



Forward Model

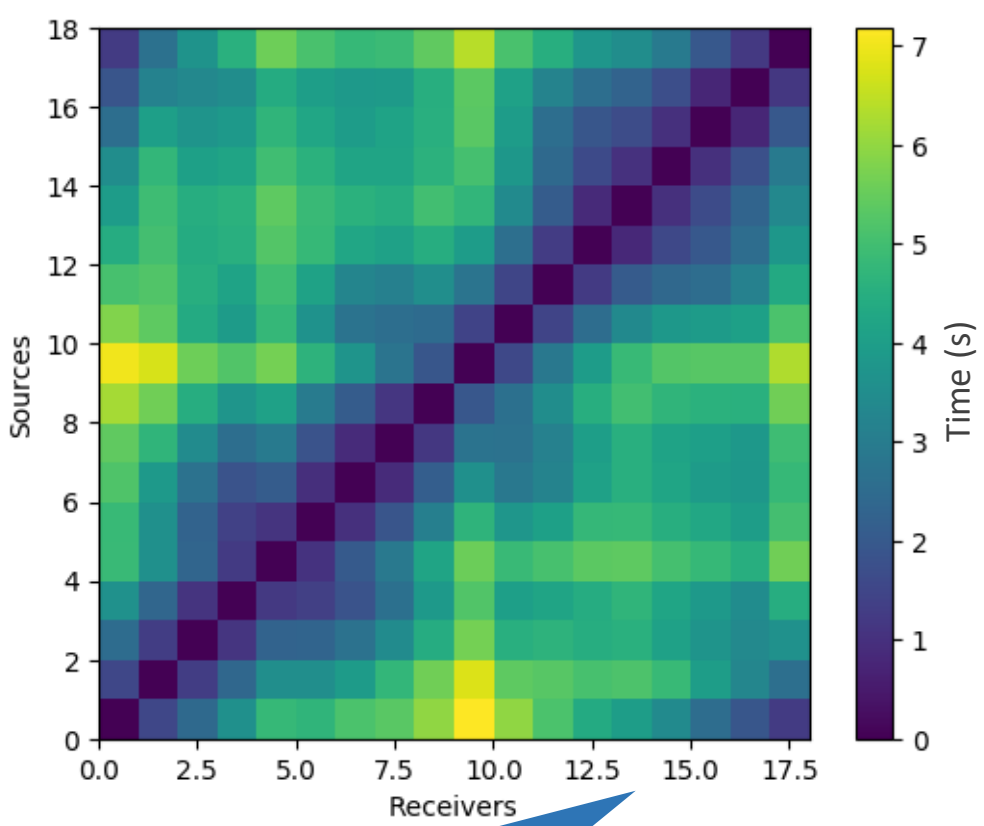
Travel-time Tomography

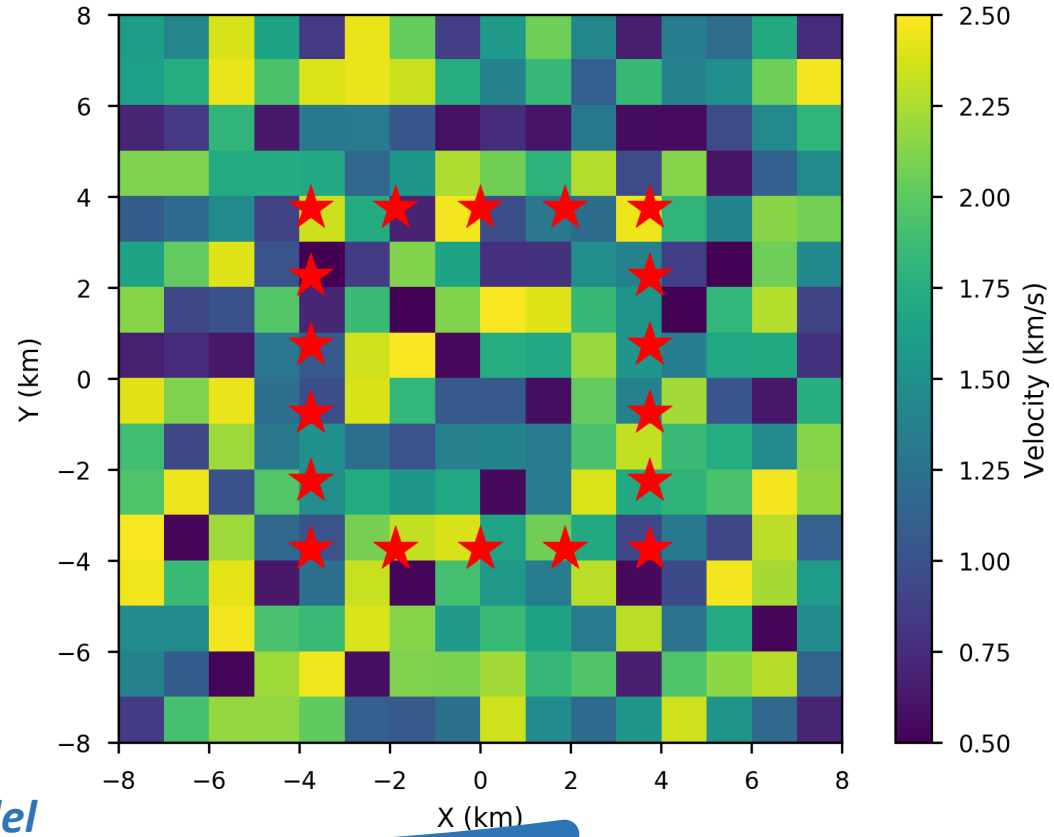
- **500,000 random models**
- Selected from $\mathcal{U}(0.5,2.5)$km/s



Travel-time Tomography

- **500,000 random models**
- Selected from $\mathcal{U}(0.5,2.5)$km/s

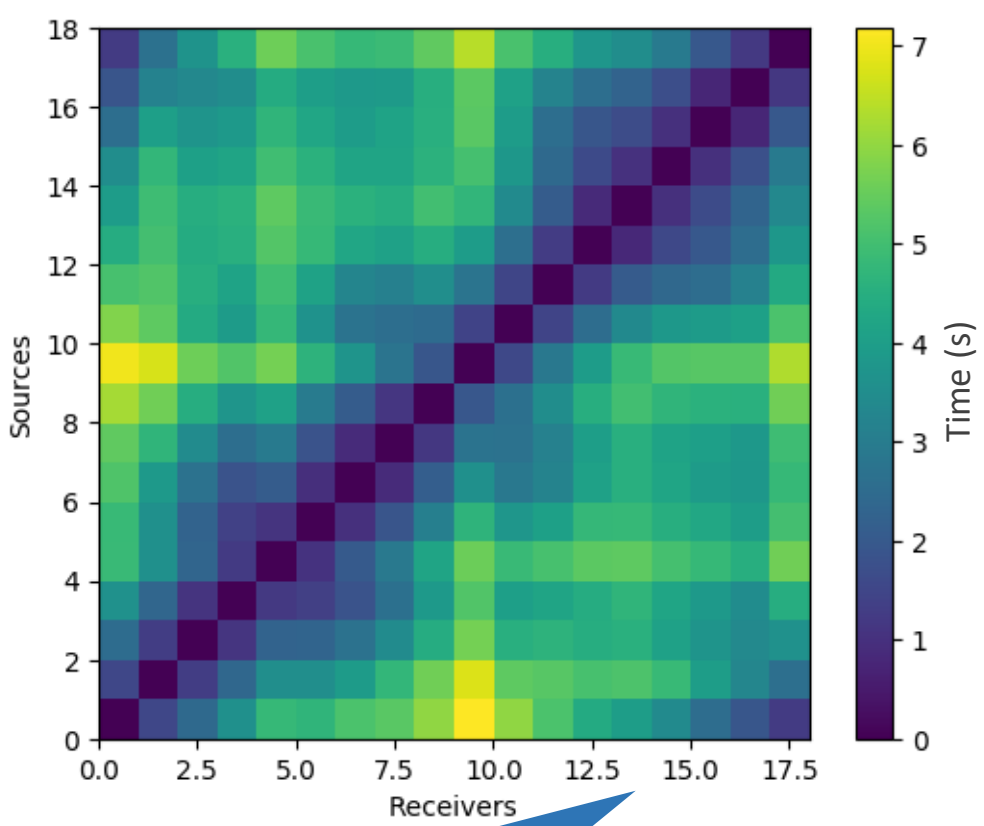


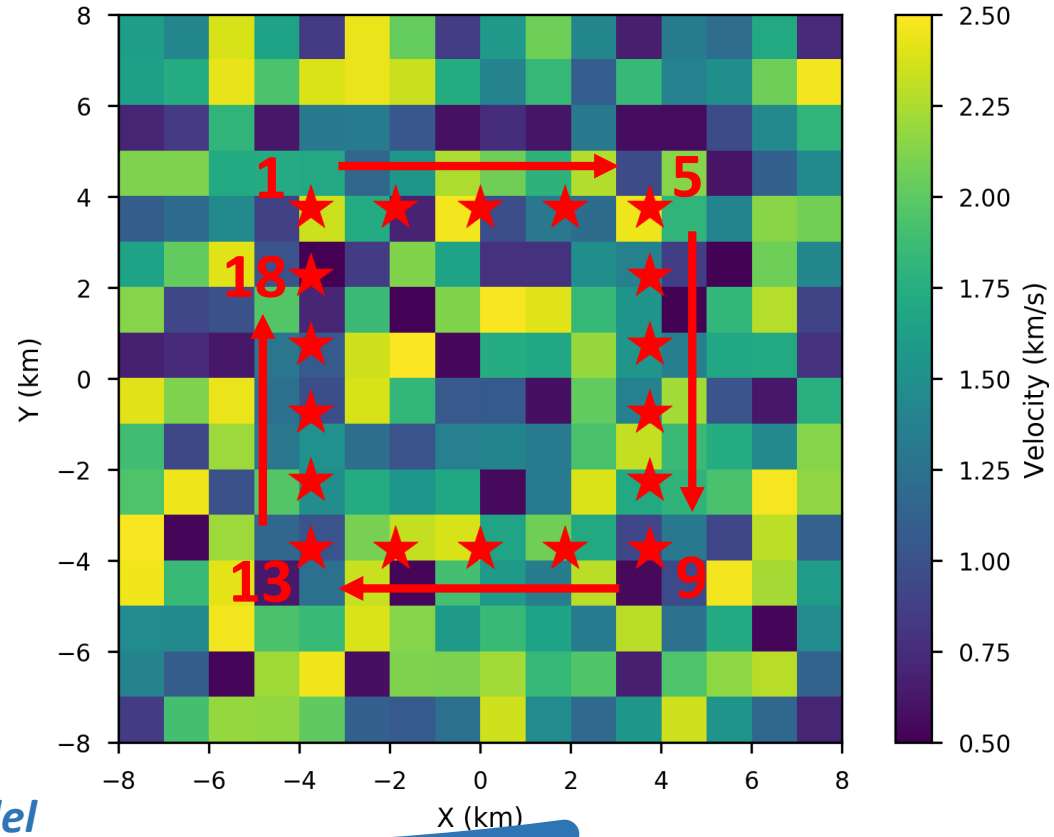


Forward Model

Travel-time Tomography

- **500,000 random models**
- Selected from $\mathcal{U}(0.5,2.5)$km/s

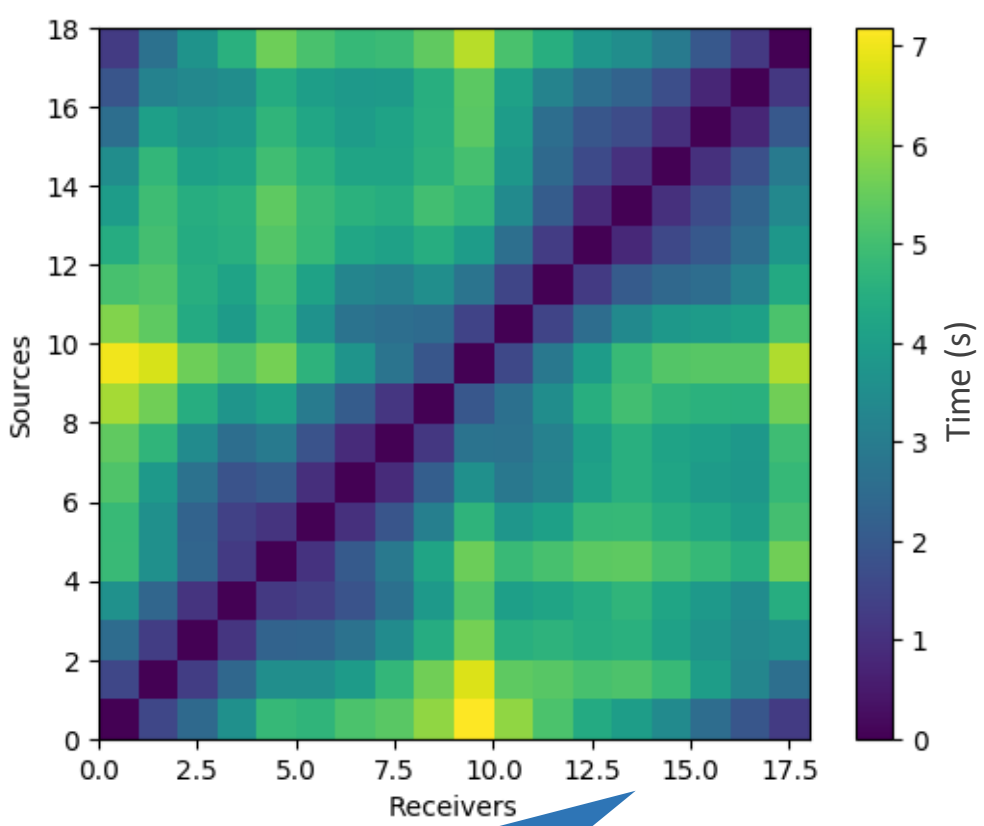


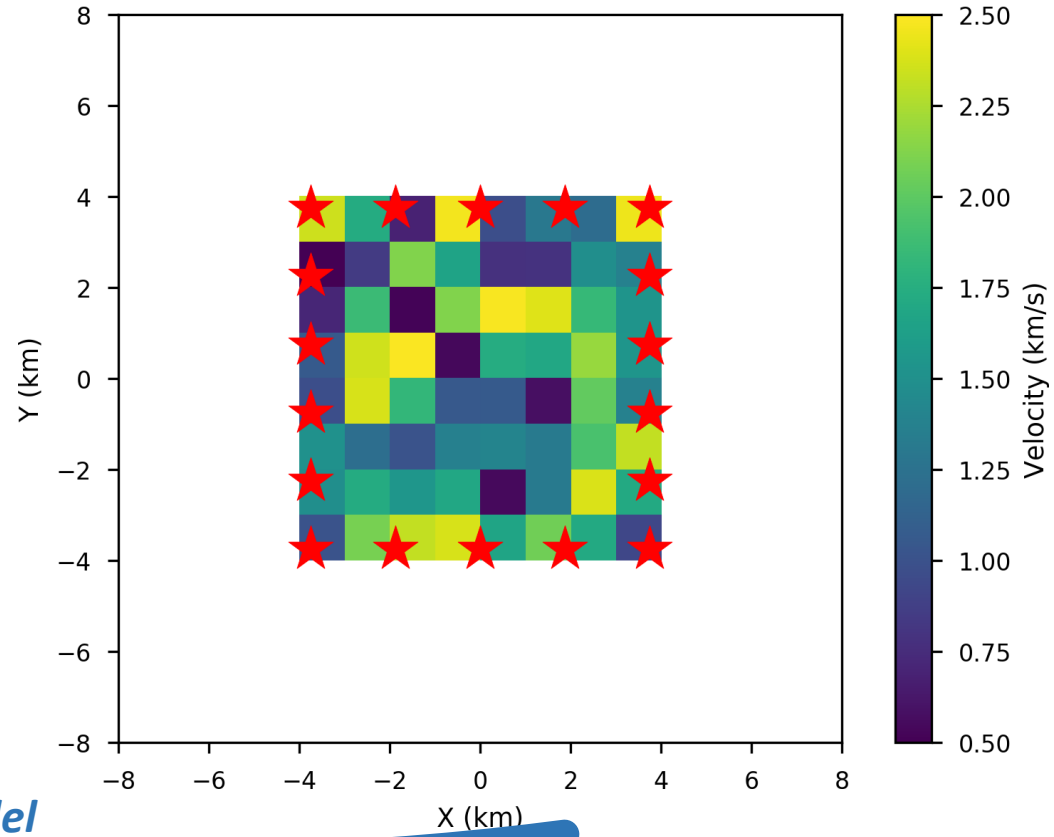


Forward Model

Travel-time Tomography

- **500,000 random models**
- Selected from $\mathcal{U}(0.5,2.5)$km/s

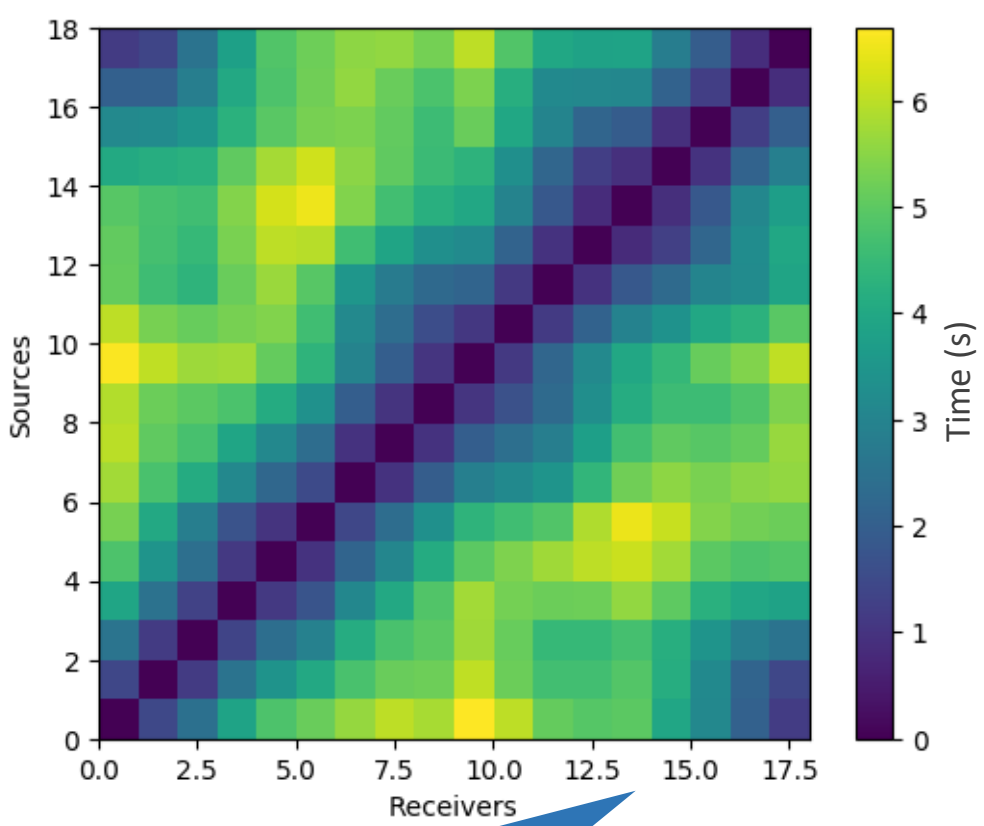


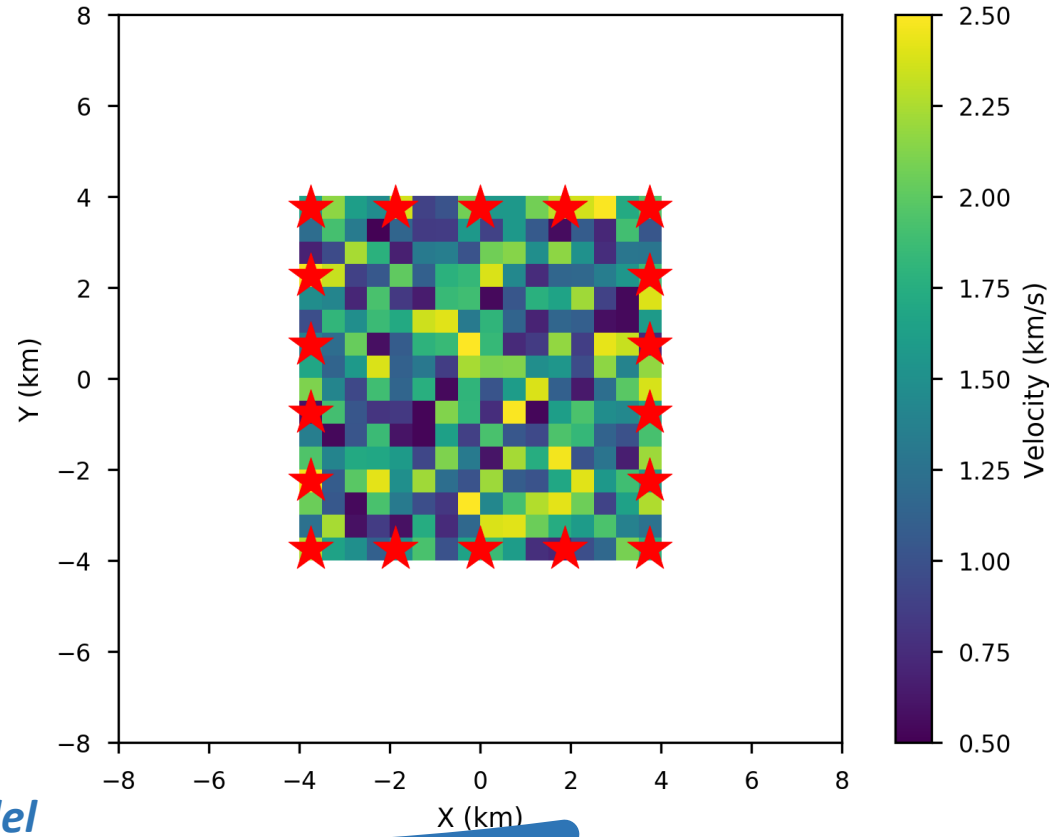


Forward Model

Travel-time Tomography

- **500,000 random models**
- Selected from $\mathcal{U}(0.5,2.5)$km/s



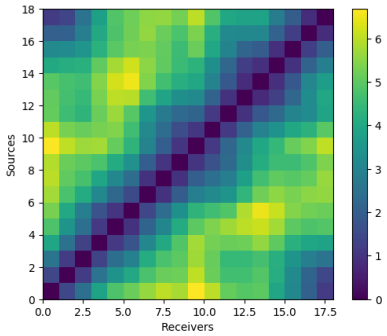


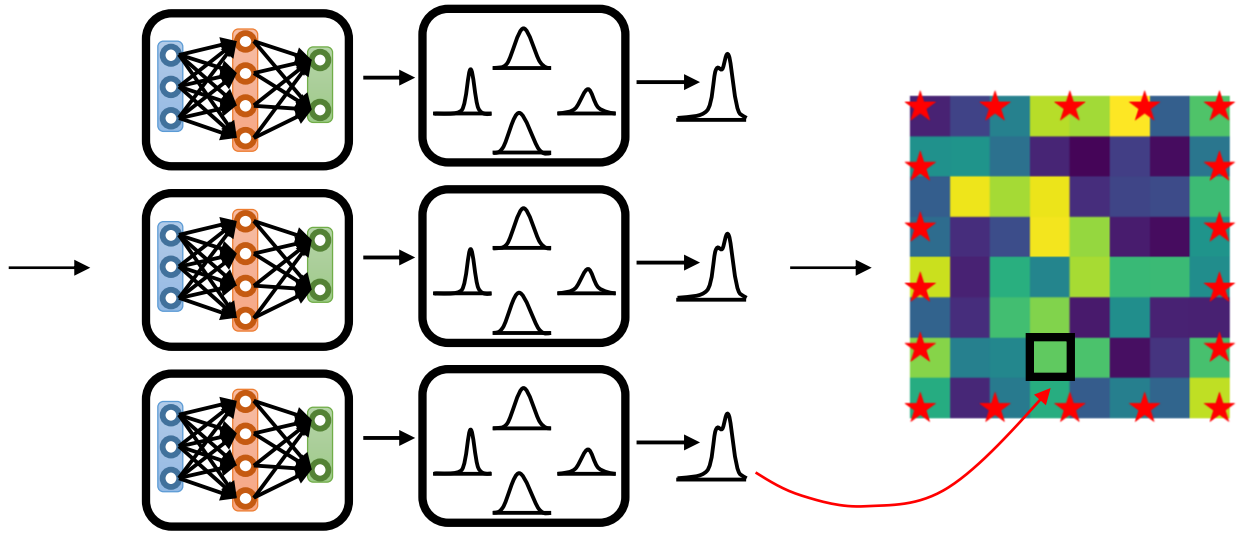
Forward Model

Travel-time Tomography

Mixture Density Network

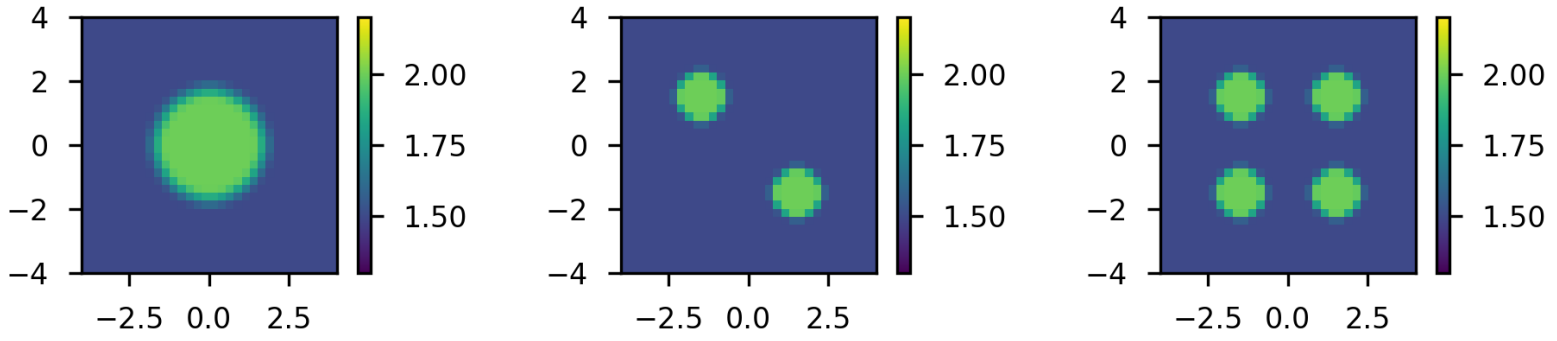
Each pixel separately



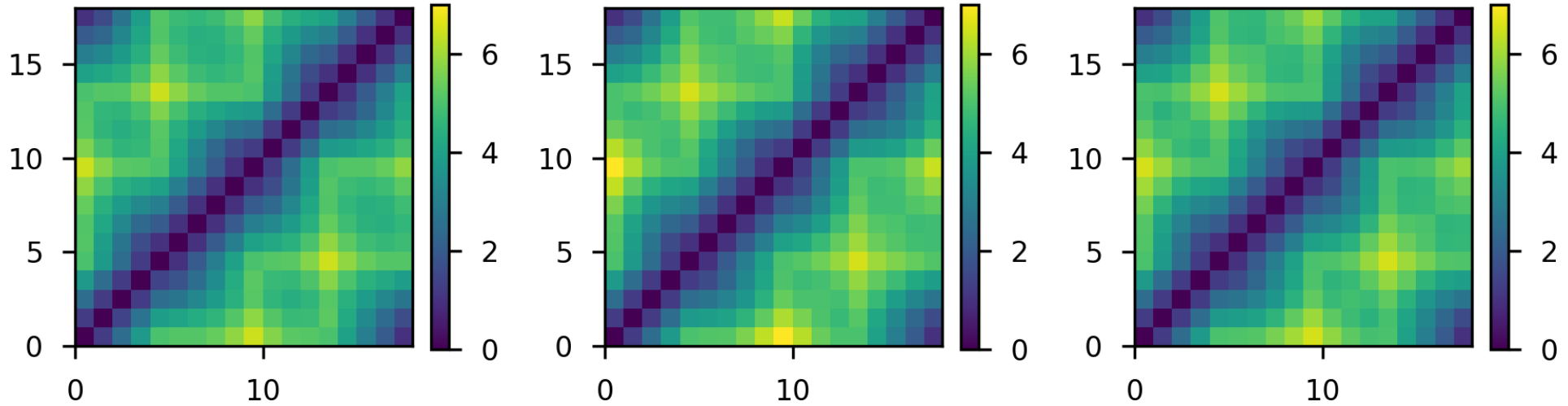


Travel-time Tomography

True Model

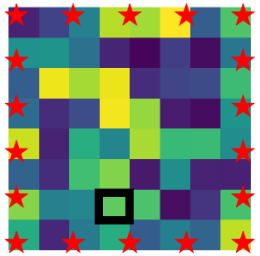


Data

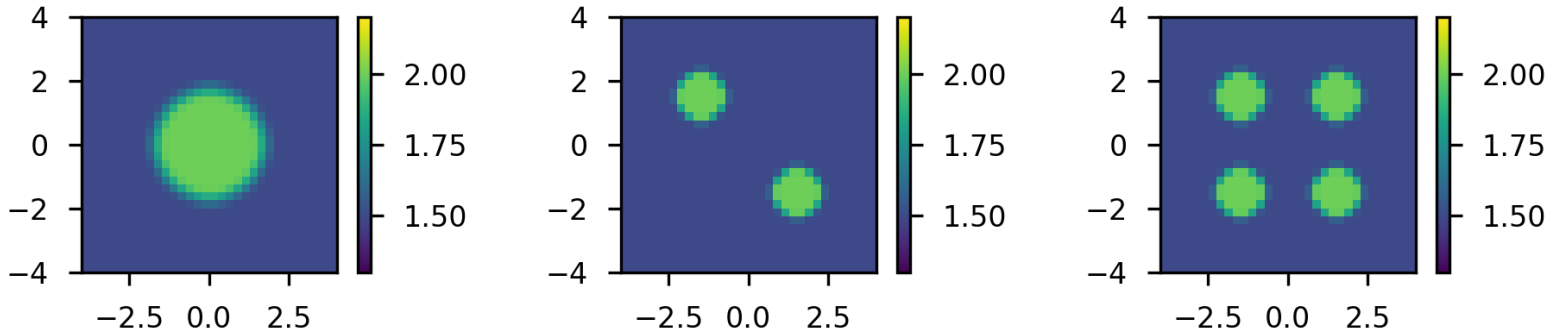


Mixture Density Network

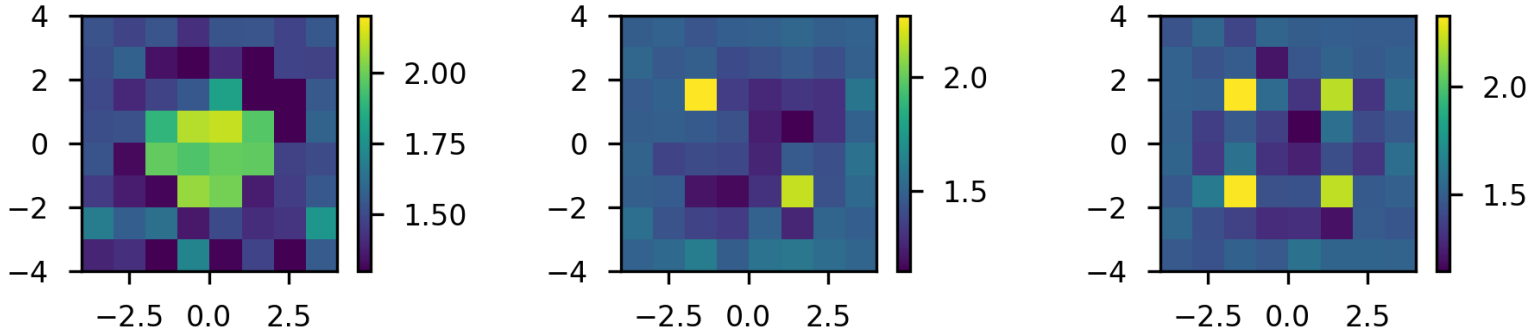
Separate pixels



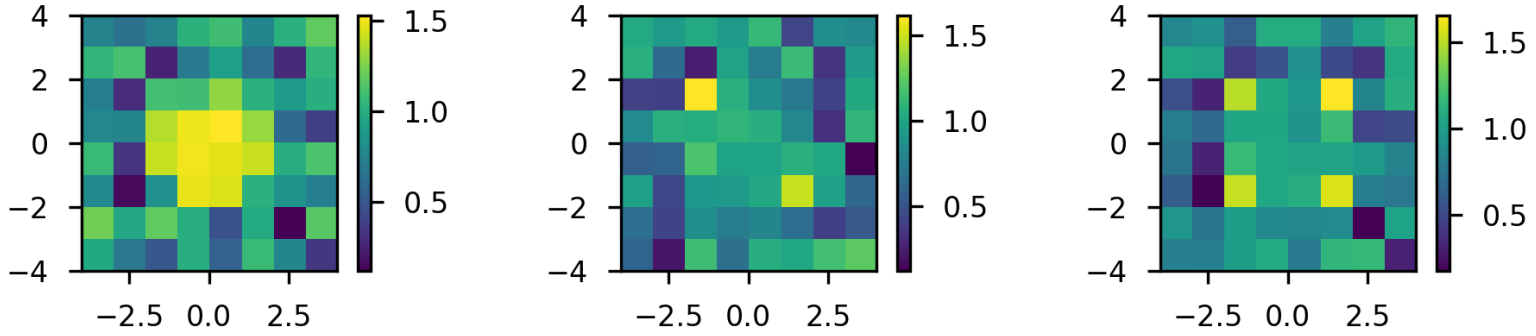
True Model



Mean

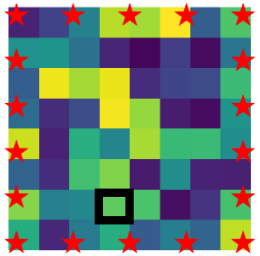


Standard Deviation

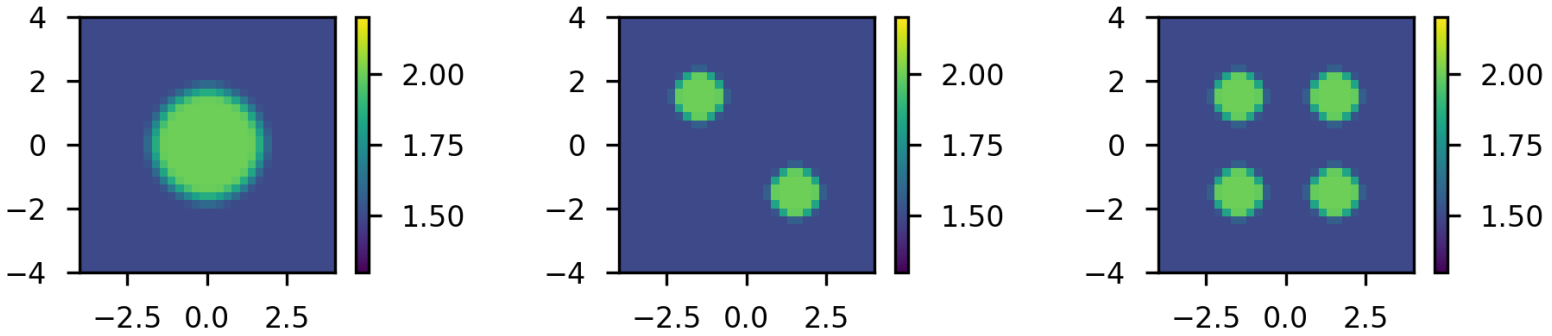


Mixture Density Network

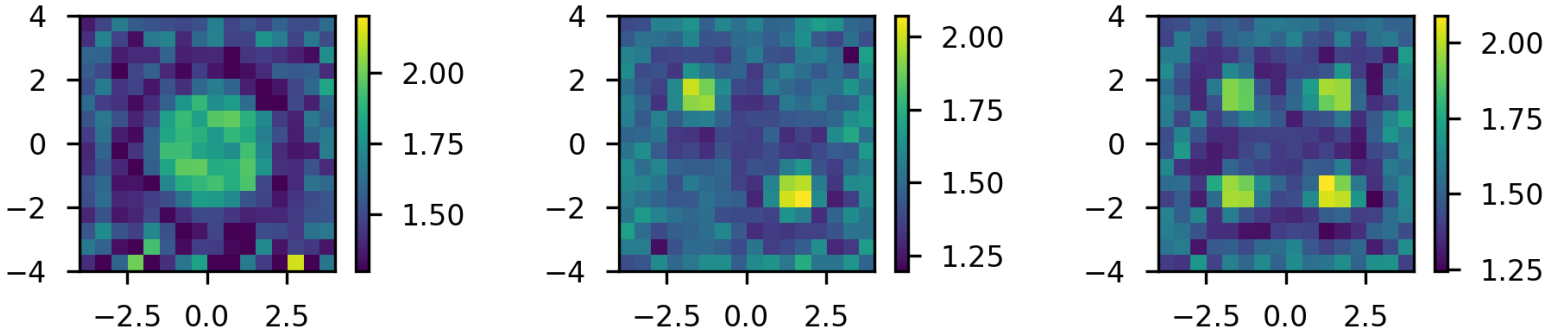
Separate pixels



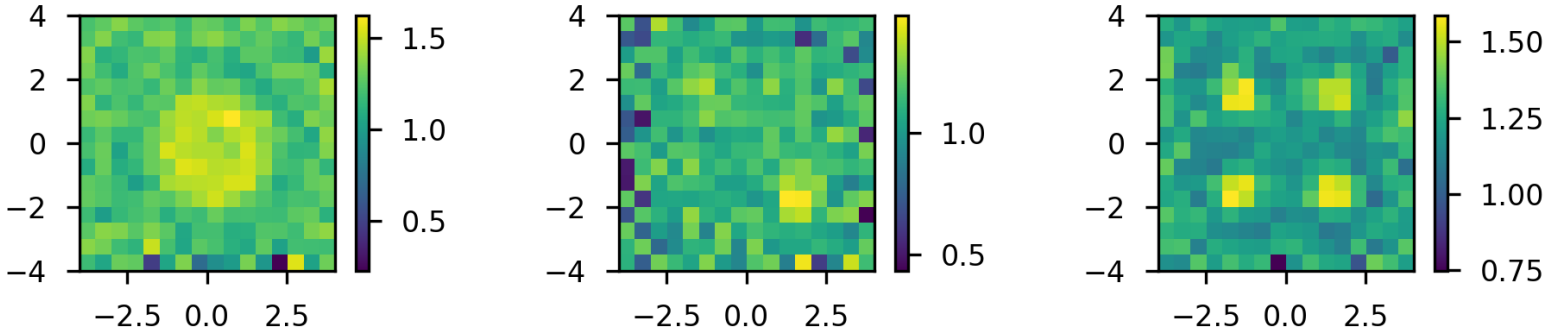
True Model



Mean

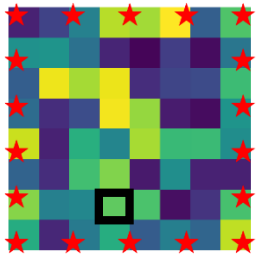


Standard Deviation

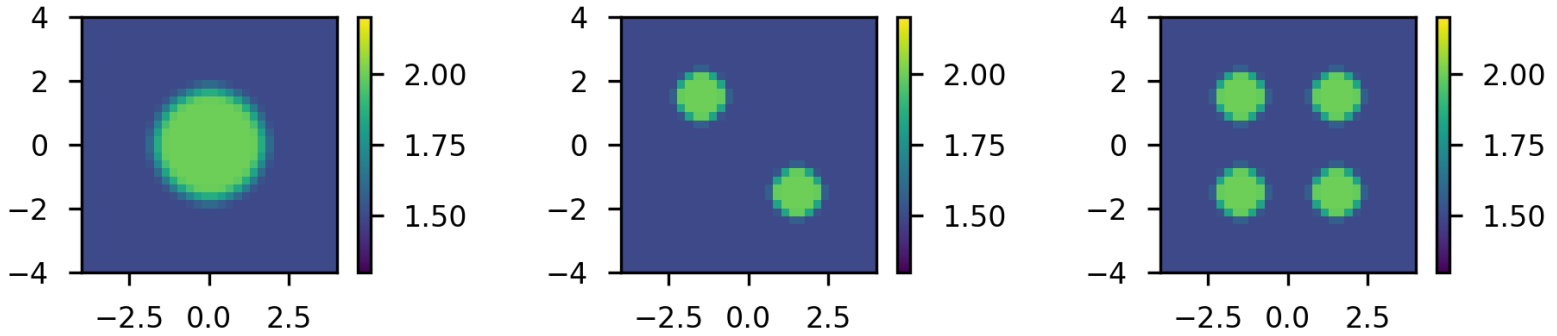


Mixture Density Network

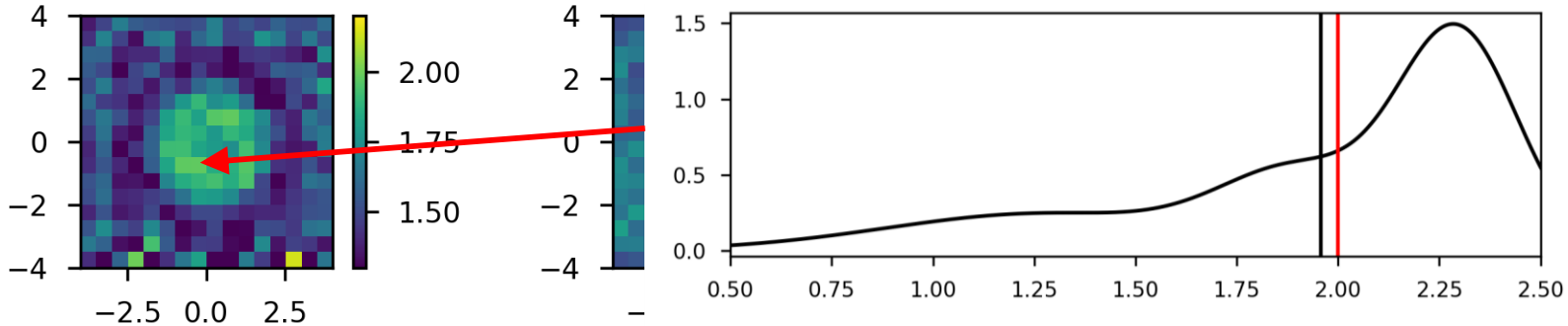
Separate pixels



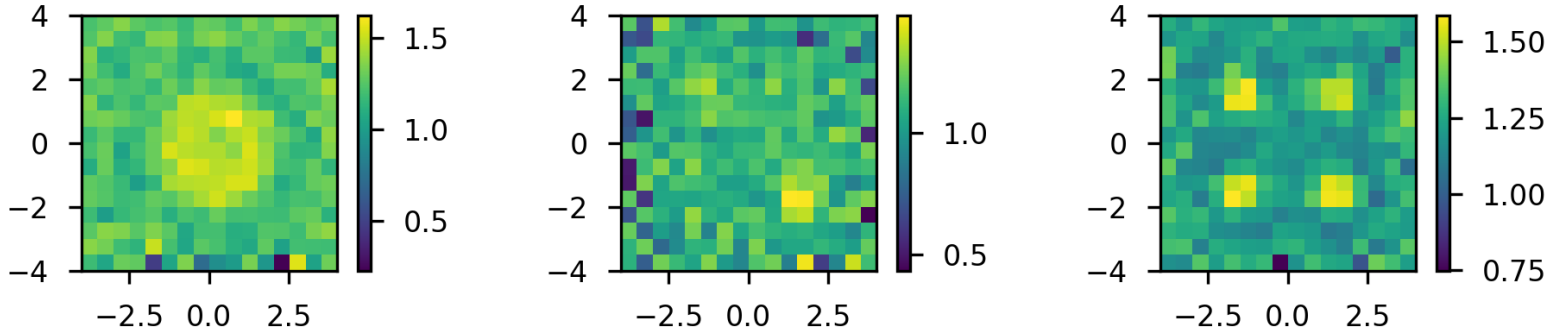
True Model



Mean

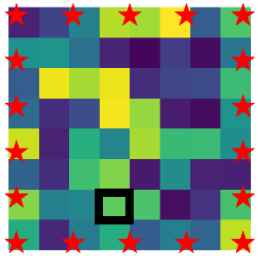


Standard Deviation

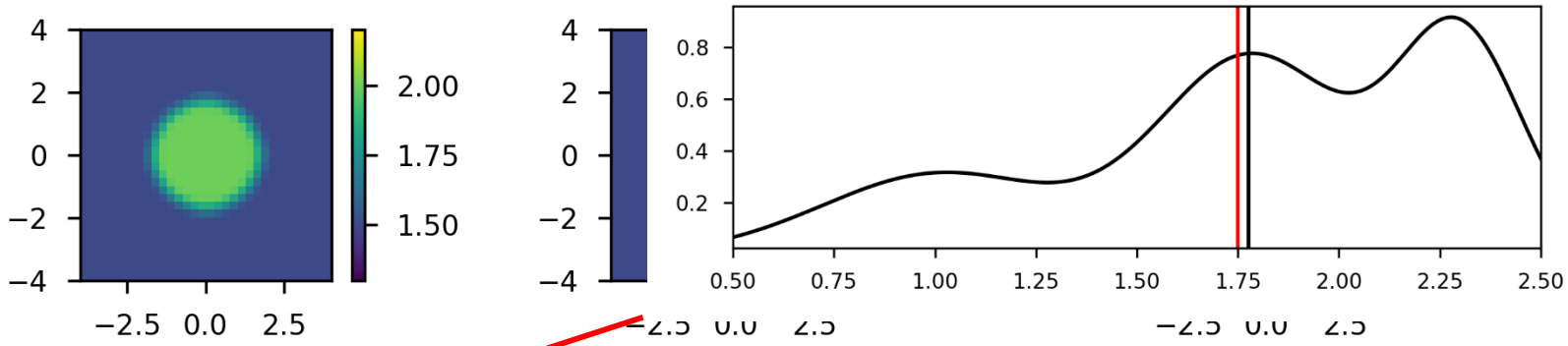


Mixture Density Network

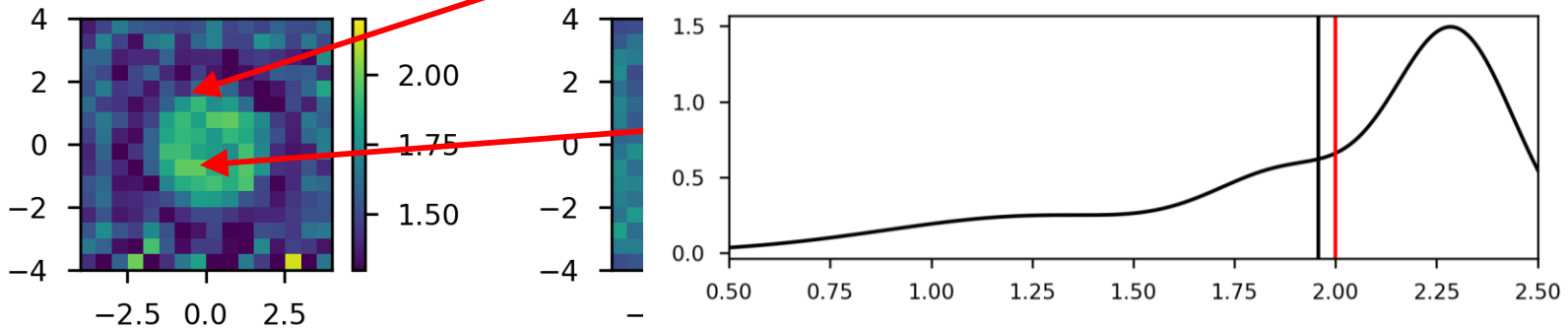
Separate pixels



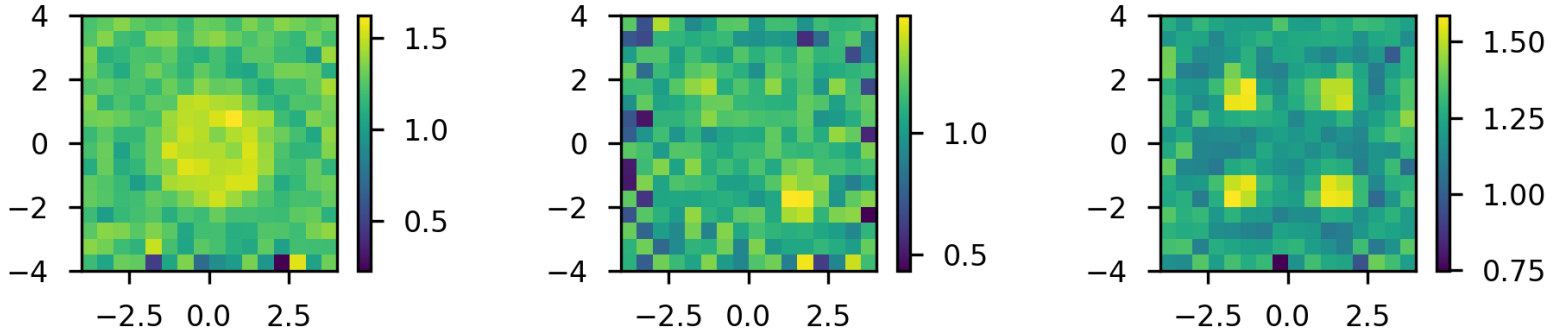
True Model



Mean

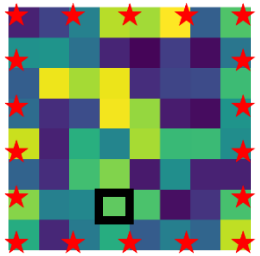


Standard Deviation

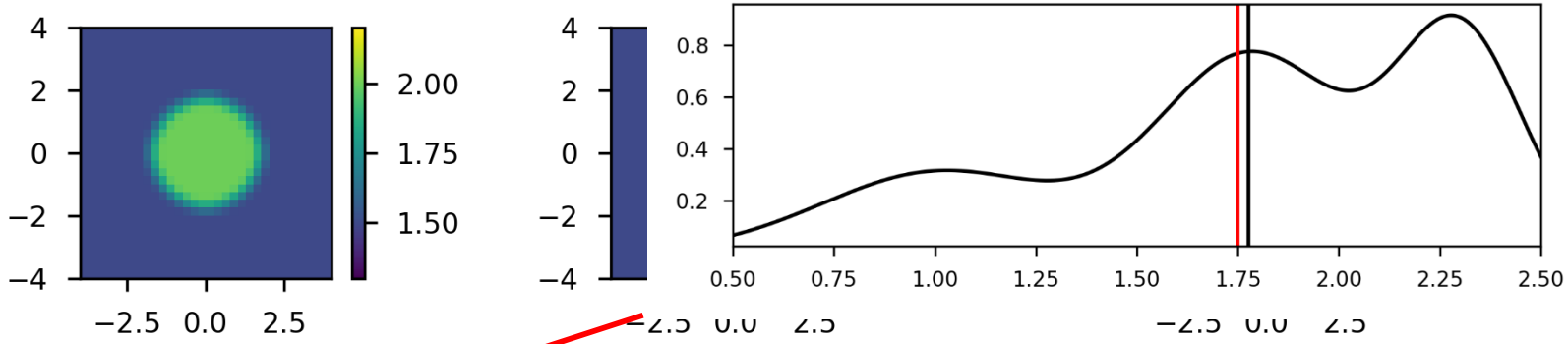


Mixture Density Network

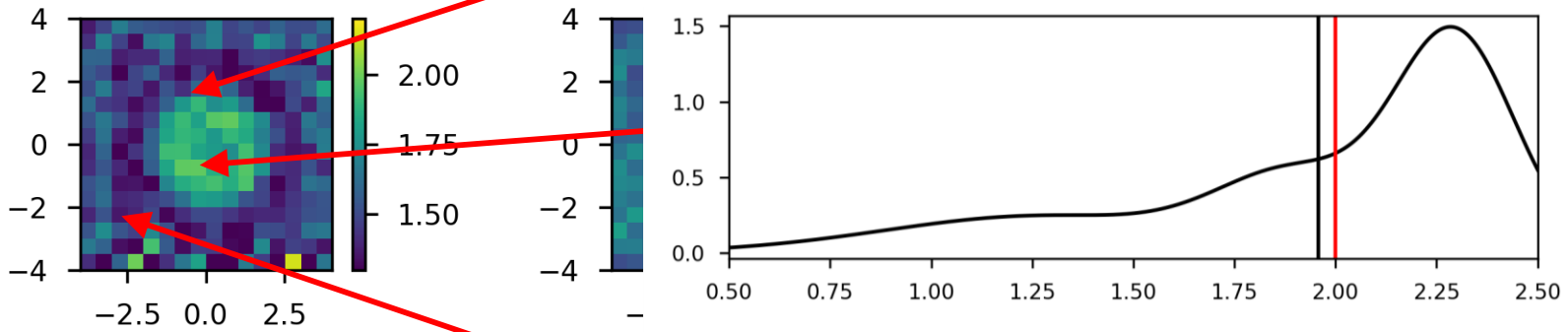
Separate pixels



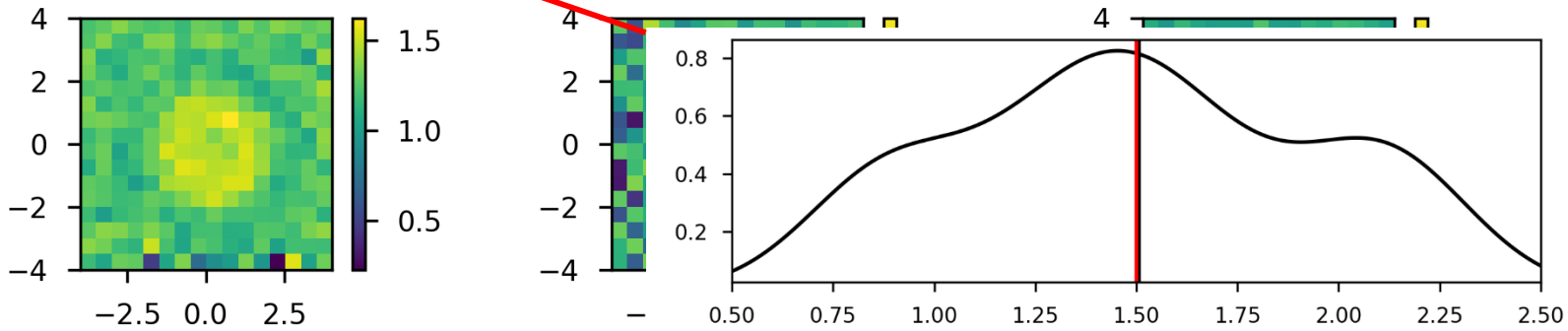
True Model



Mean



Standard Deviation



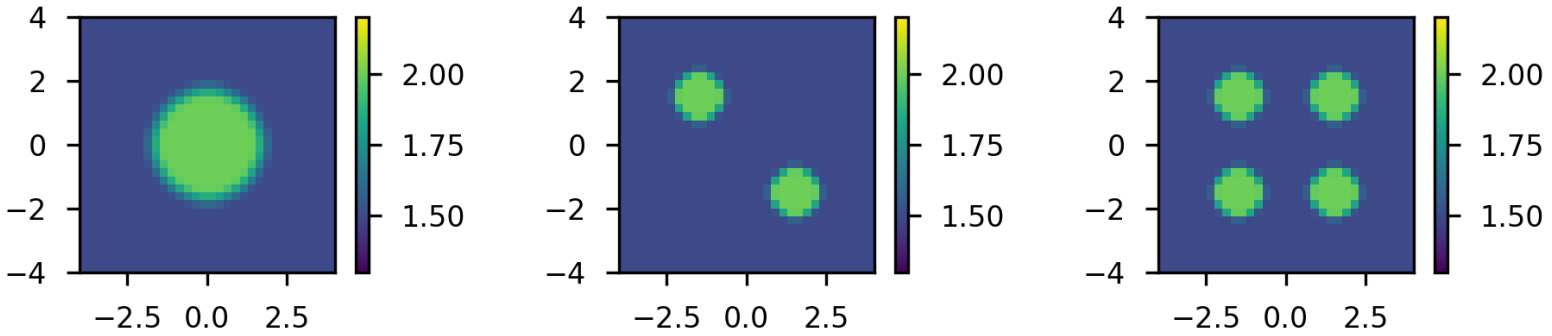
Fully nonlinear probabilistic Tomography in ~1 second

Earp & Curtis 2019 (ArXiv)

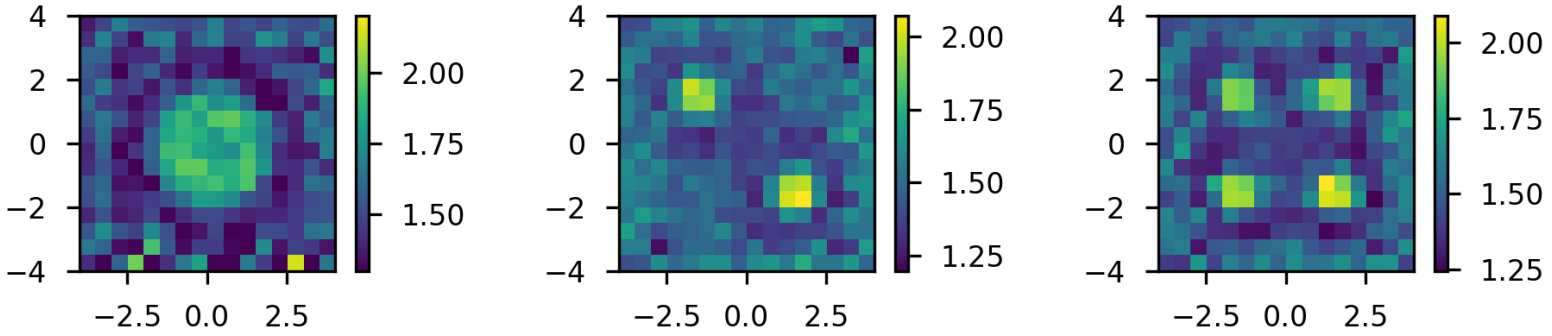
Mixture Density Network

Separate pixels

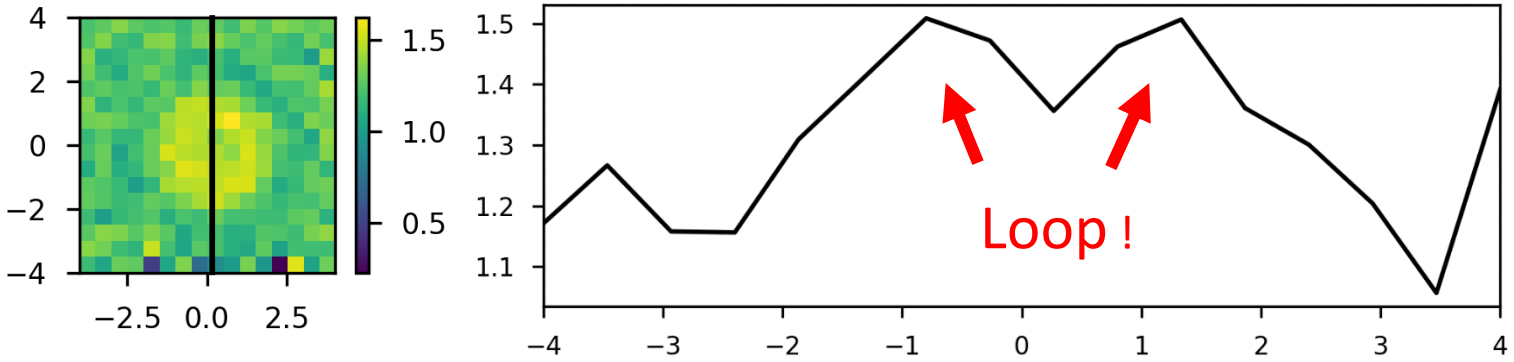
True Model



Mean



Standard Deviation

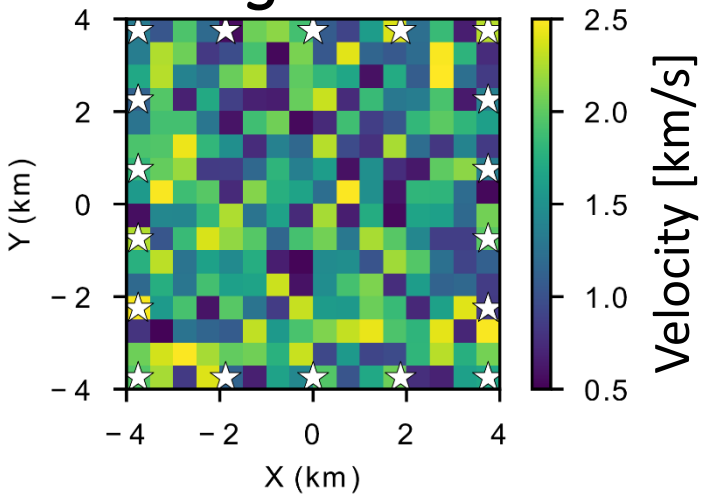


Training Set

- Sample prior model distribution
- 2.5 million models
  - 8x8 model
  - 16x16 model
- Convolutional networks

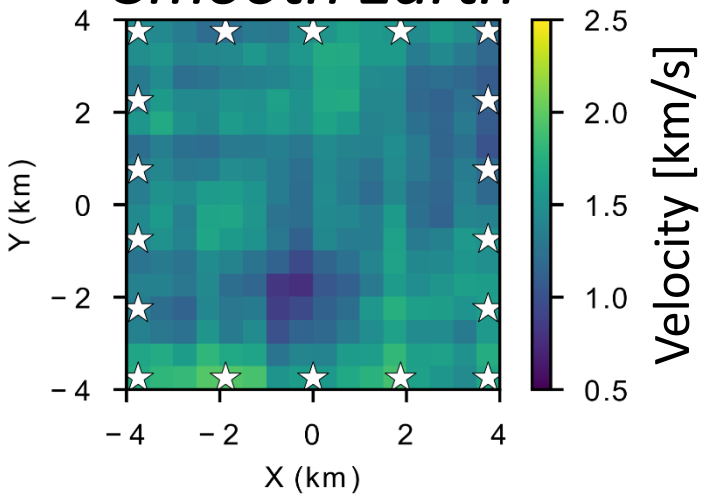
Uniformly Random

Rough Earth



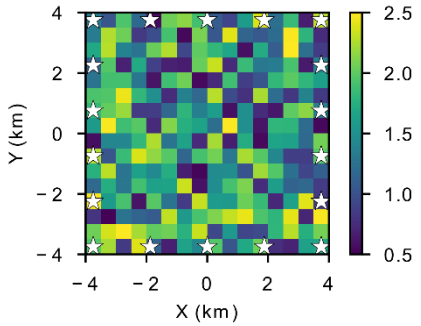
More prior info:

Smooth Earth

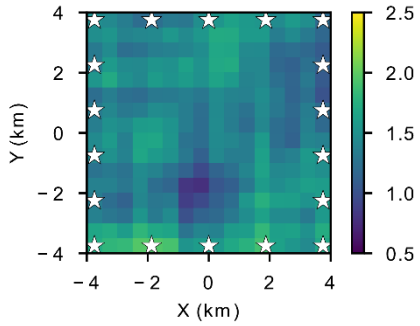


EAGE Abstract + arXiv: Earp & Curtis, 2019

Rough Prior

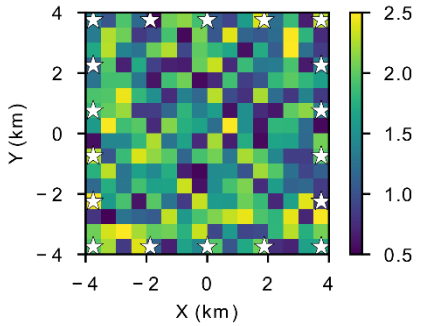


Smooth Prior

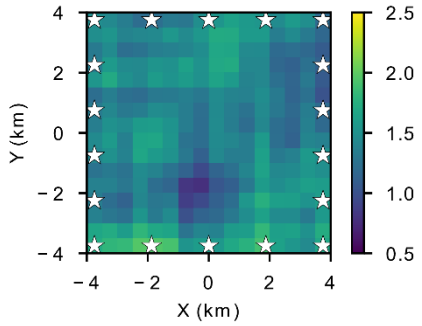


EAGE Abstract + arXiv: Earp & Curtis, 2019

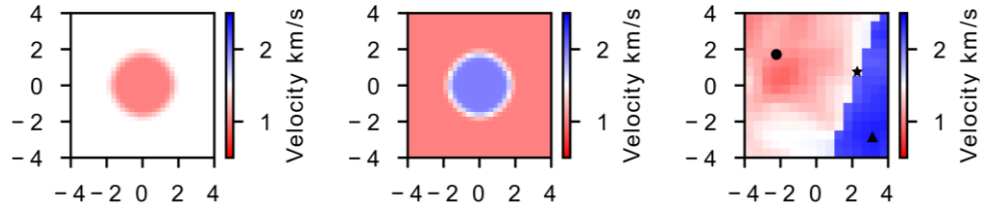
Rough Prior



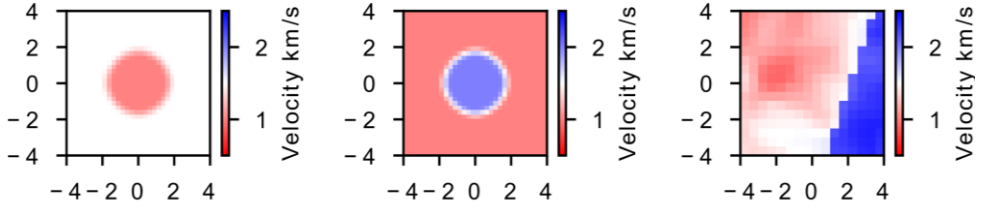
Smooth Prior



True Model

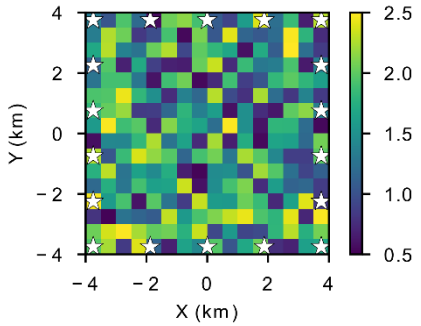


True Model

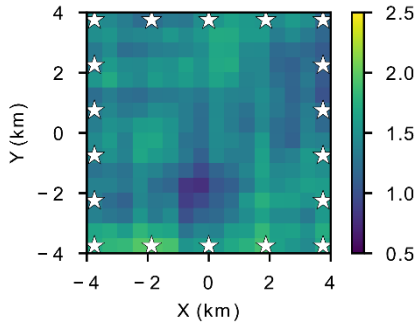


EAGE Abstract + arXiv: Earp & Curtis, 2019

Rough Prior

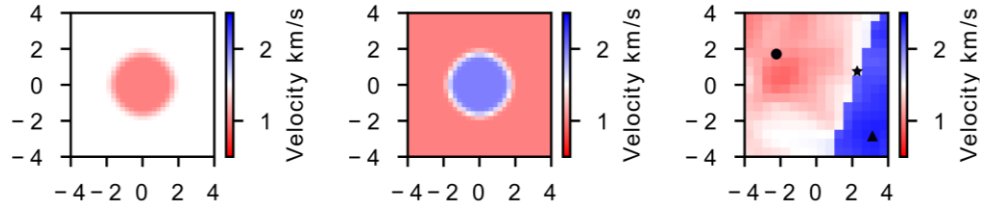


Smooth Prior

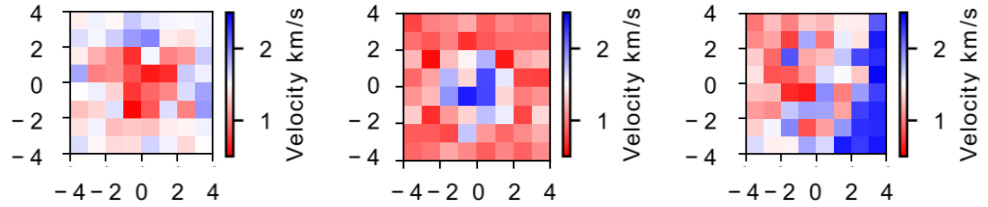


Samples from posterior marginal pdf's

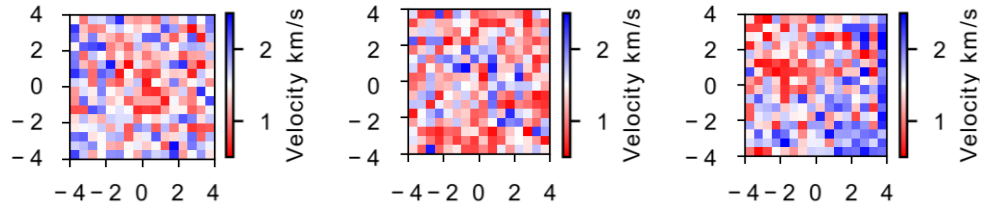
True Model



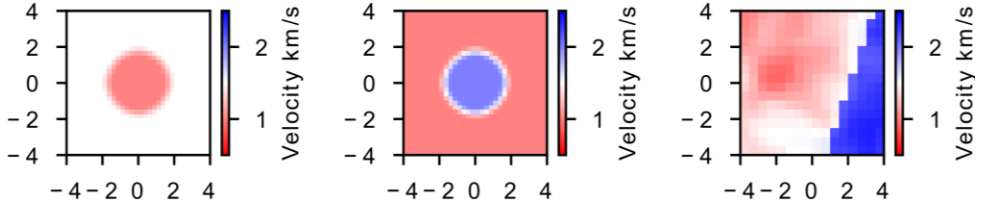
8 x 8



16 x 16

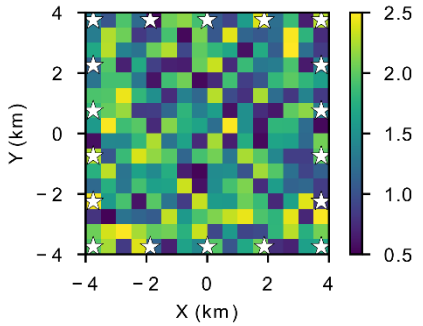


True Model

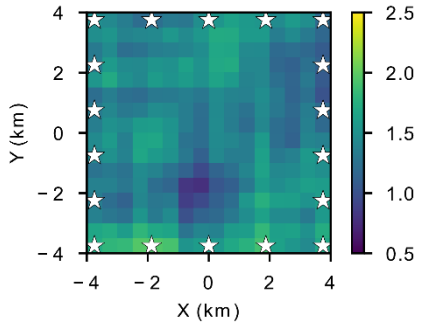


EAGE Abstract + arXiv: Earp & Curtis, 2019

Rough Prior



Smooth Prior

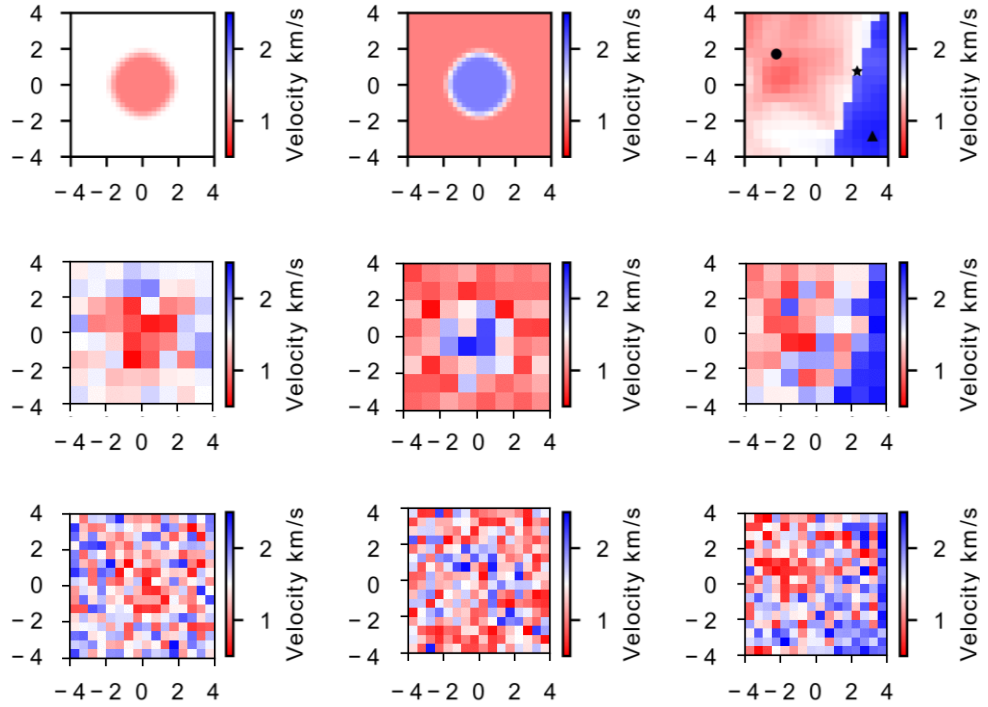


Samples from posterior marginal pdf's

True Model

8 x 8

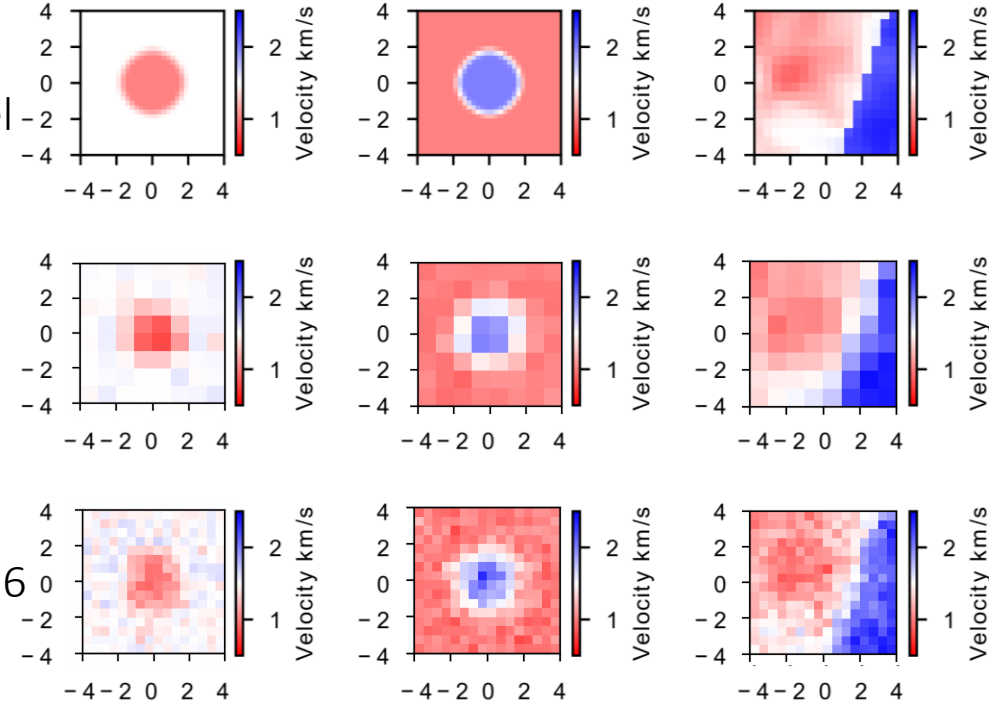
16 x 16



True Model

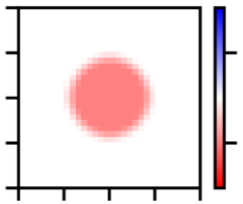
8 x 8

16 x 16



EAGE Abstract + arXiv: Earp & Curtis, 2019

Rough Prior



Smooth Prior

Summary

- MDNs can solve **travel time tomography** for local velocity uncertainty
- MDNs can **invert Phase velocities for Shear velocity** structure with depth
- Both take ~1 second per inversion **post-training**: but **need to 'sample' the prior pdf…**
- **Adds ~3 lines to standard neural network training scripts** (PyTorch, TensorFlow,…)

➔ **3D nonlinear tomography with uncertainties within ~ 1 hour of recording**

Future…

Test and apply to **your** problems! (I can help)