Novel Methods for Ambient Noise Tomography

Andrew Curtis

Xin Zhang, Stephanie Earp

Sjoerd de Ridder*, Ruicun Cao, Erica Galetti

University of Edinburgh

*now at University of Leeds

Introduction

➔ *Bayesian Nonlinear Ambient Noise Tomography in Near-Real Time*

Introduction

➔ *Bayesian Nonlinear Ambient Noise Tomography in Near-Real Time*

Introduction

Family of plausible models

➔ *Bayesian Nonlinear Ambient Noise Tomography in Near-Real Time*

Introduction

Family of plausible models

→ *Bayesian Nonlinear Ambient Noise Tomography in Near-Real Time*

"Assume the Physics in linear…"!

Nonlinearity is your friend **: Coding is easier** – no derivatives!

Family of plausible models → **smaller**

Introduction

Family of plausible models

➔ *Bayesian Nonlinear Ambient Noise Tomography in Near-Real Time*

"Assume the Physics in linear…"!

Nonlinearity is your friend **: Coding is easier** – no derivatives!

Family of plausible models ➔ **smaller**

Real-time applications

- Monitoring induced seismicity/deformation
- Earthquake/volcano early warning
- Subsurface leak/pollution detection
- Rolling arrays

Academic Challenge ➔ **new methods** for other science & technology

Introduction

➔ *Bayesian Nonlinear Ambient Noise Tomography in Near-Real Time*

➔ This is possible! (or at least is a reasonable goal)

➔ Generalised Novel Methods for Geophysical Inversion

Bayesian Inference/Inversion

Bayes' theorem

likelihood

prior

$$p(\mathbf{m}|\mathbf{d}_{obs}) = \frac{p(\mathbf{d}_{obs}|\mathbf{m})\,p(\mathbf{m})}{p(\mathbf{d}_{obs})}$$

posterior

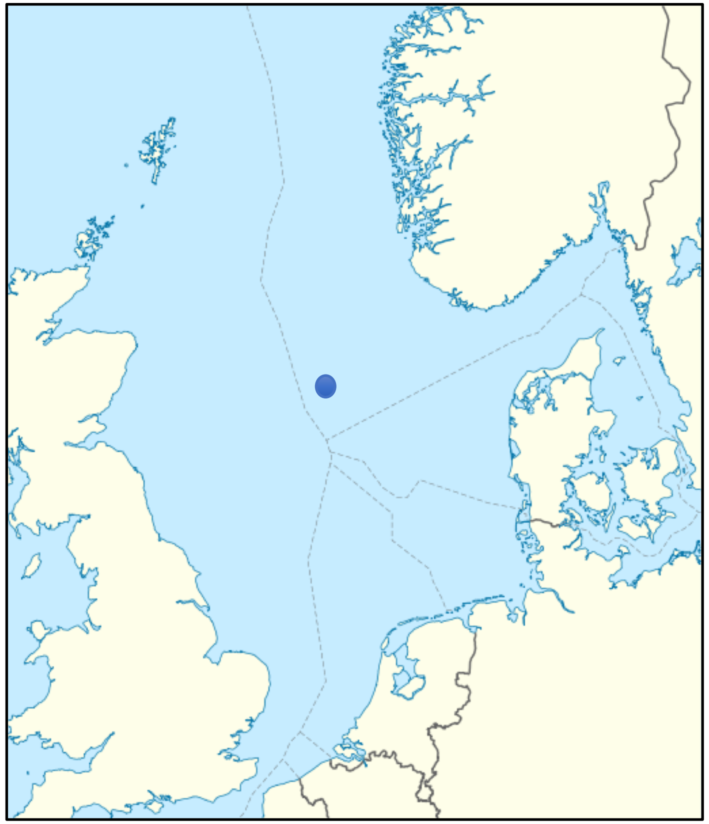
evidence

Talk Plan

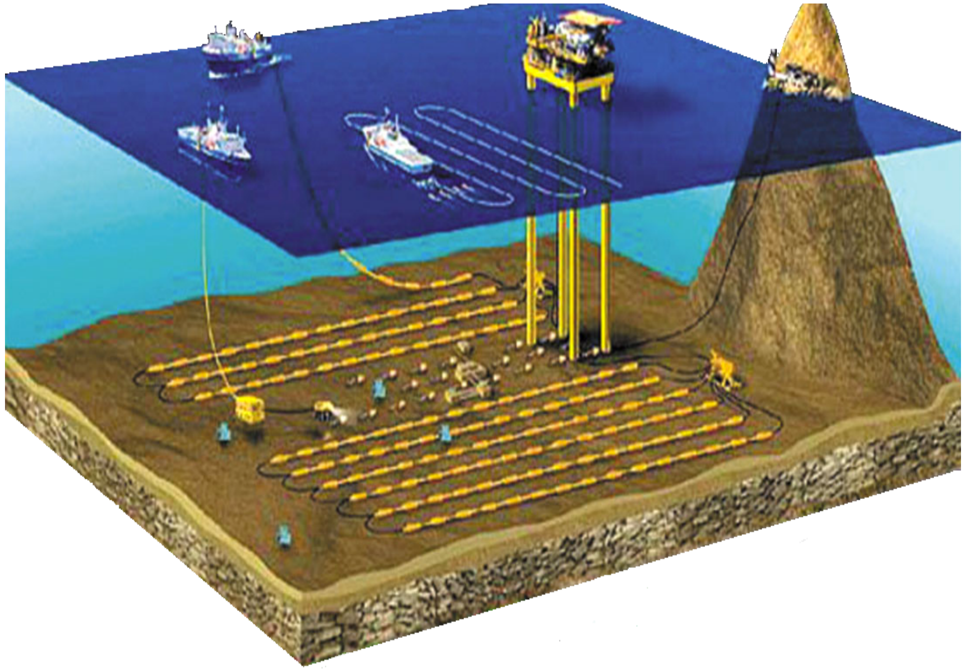
- ➔ Dense Array Ambient Noise Tomography
- ➔ Sparse Array Ambient Noise Tomography
- ➔ *Bayesian Nonlinear tomography in near-real time* is possible!
- ➔ Generalised methods to estimate probabilities

Ocean Bottom Cables

Ekofisk



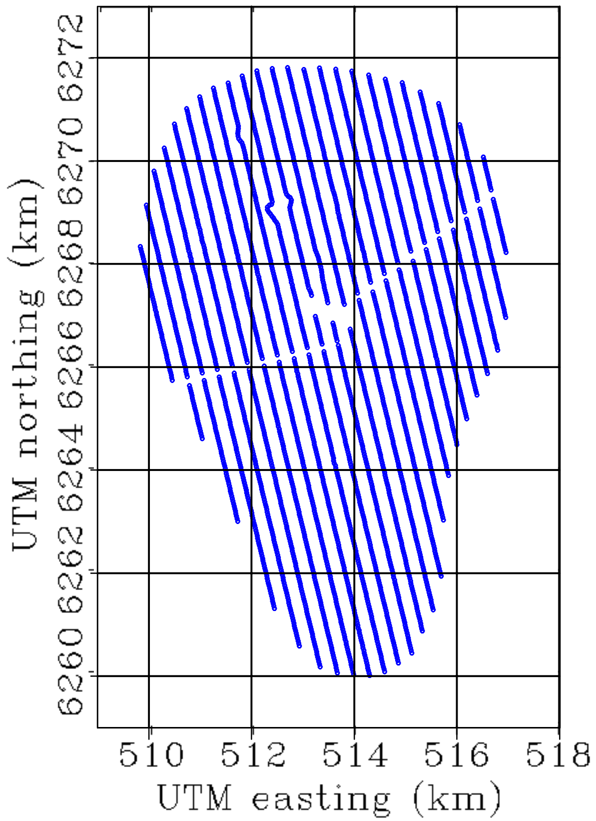
Thank you to Ekofisk Field partners:
equinor, Petoro, Eni, ConocoPhillips, Total

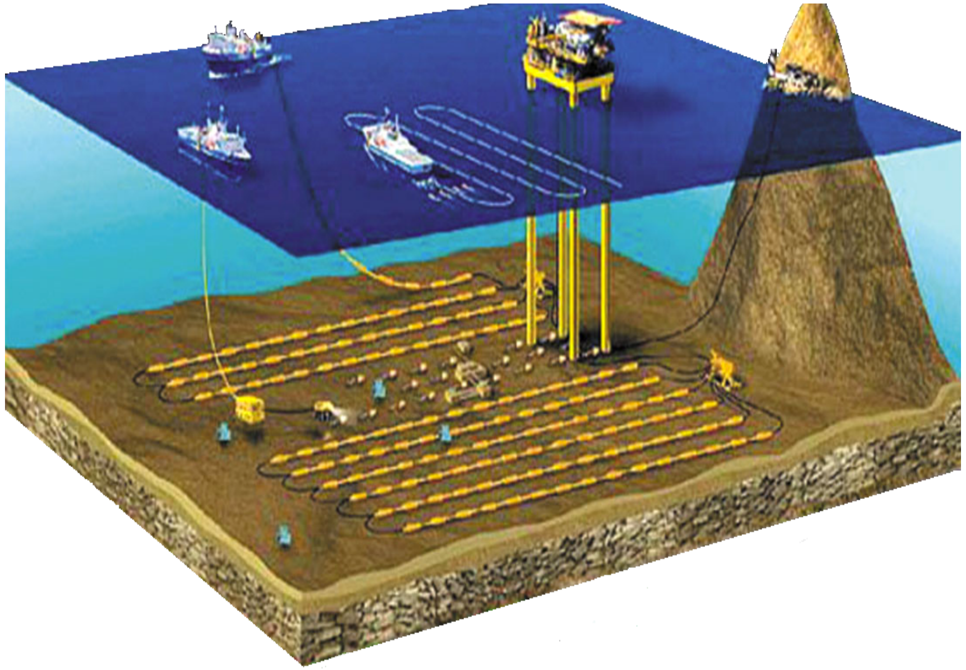


de Ridder & Biondi (2015)

Ocean Bottom Cables

Ekofisk OBC Array



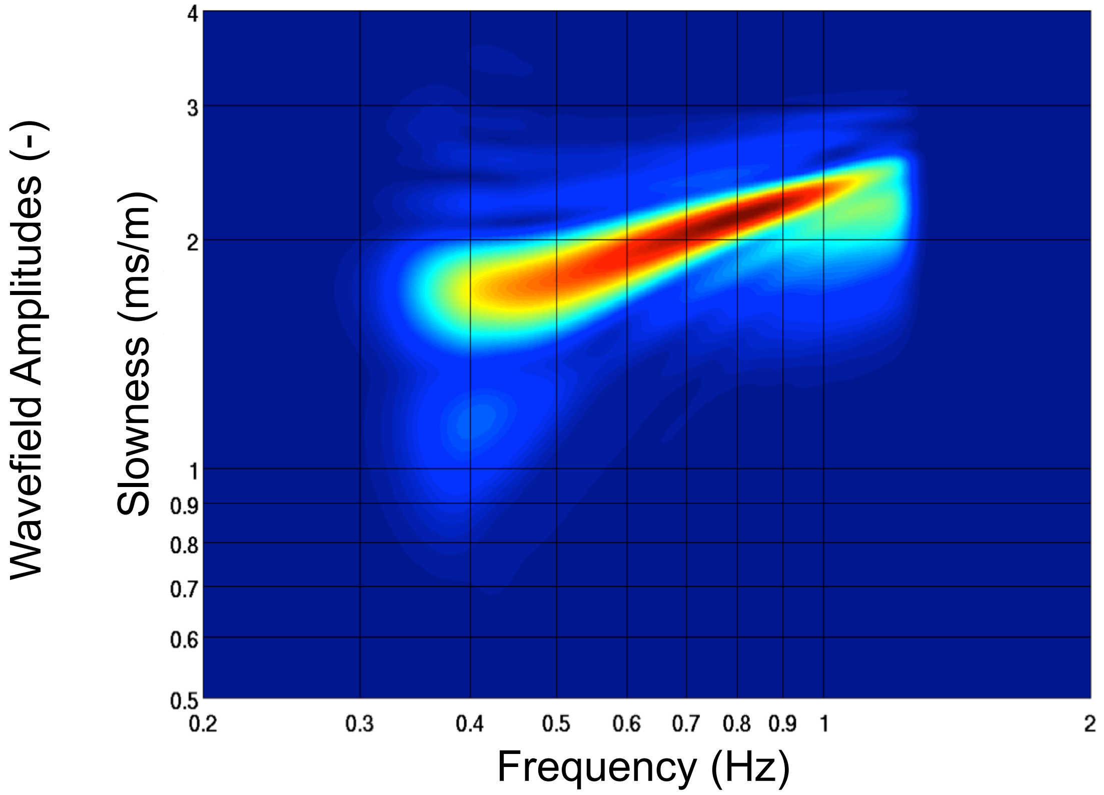


de Ridder & Biondi (2015)

Ambient Seismic recorded by OBC

Seismic Noise

Dispersion

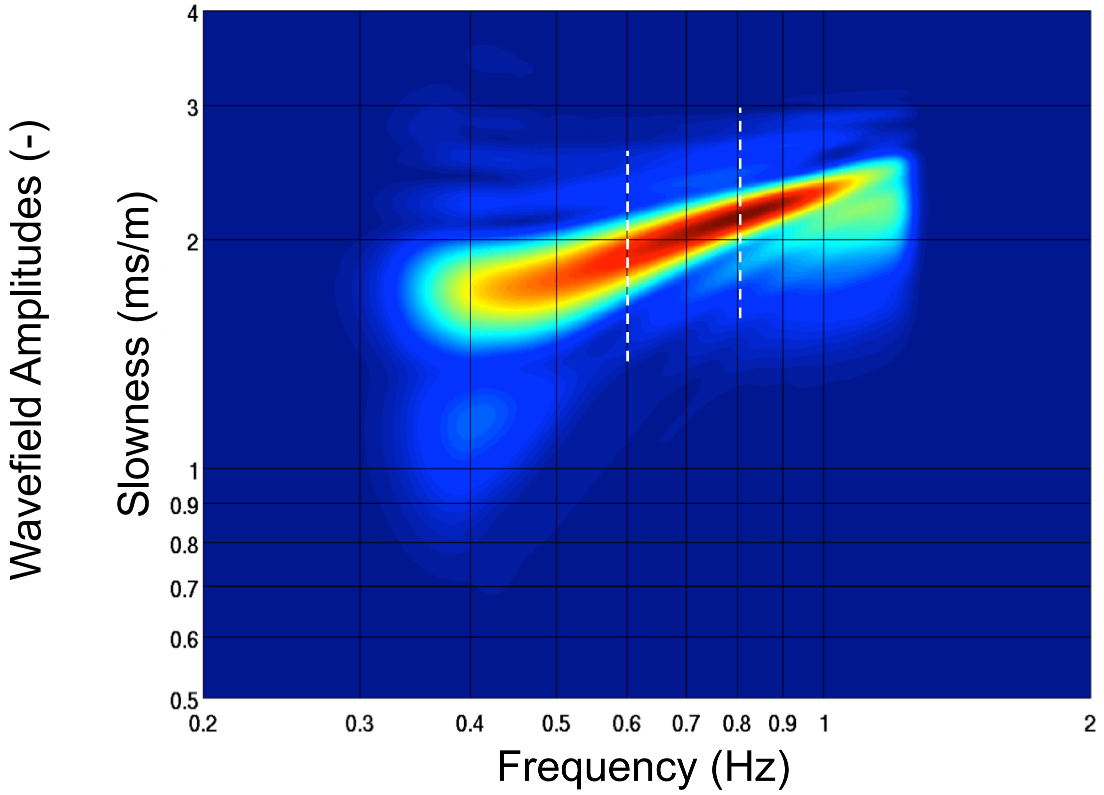


de Ridder & Biondi (2015)

Ambient Seismic recorded by OBC

Seismic Noise

Dispersion



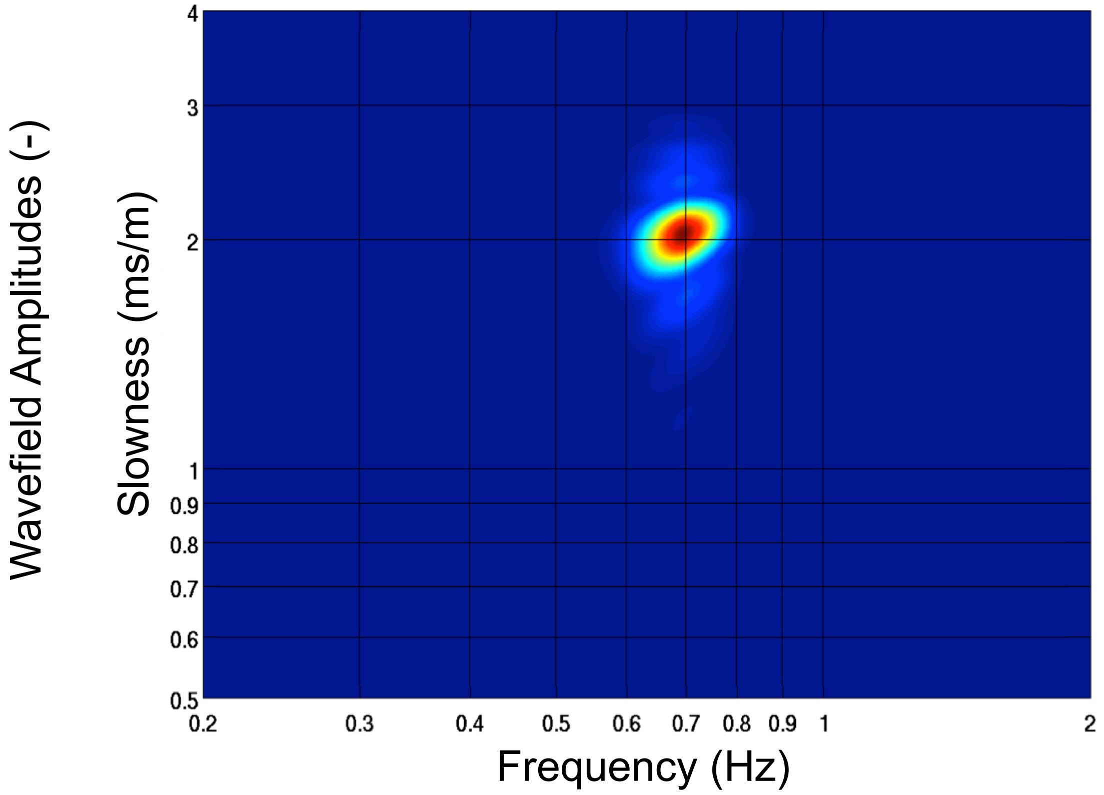
de Ridder & Biondi (2015)

$$\partial^2 U(t,\boldsymbol{x}) = c^2(x)\,\nabla^2 U(t,\boldsymbol{x})$$

***Seismic Gradiometry** – Curtis & Robertsson (2002)*

Seismic Noise

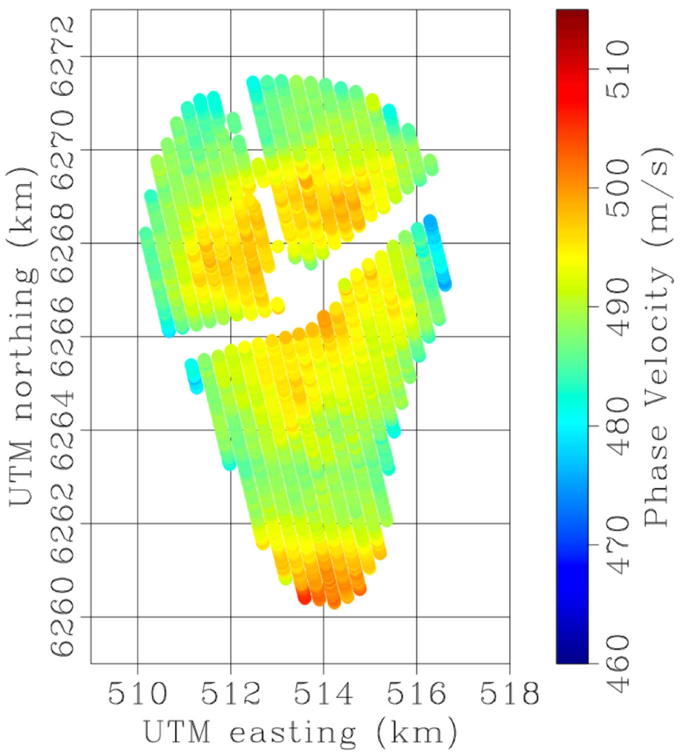
Dispersion



de Ridder & Biondi (2015)

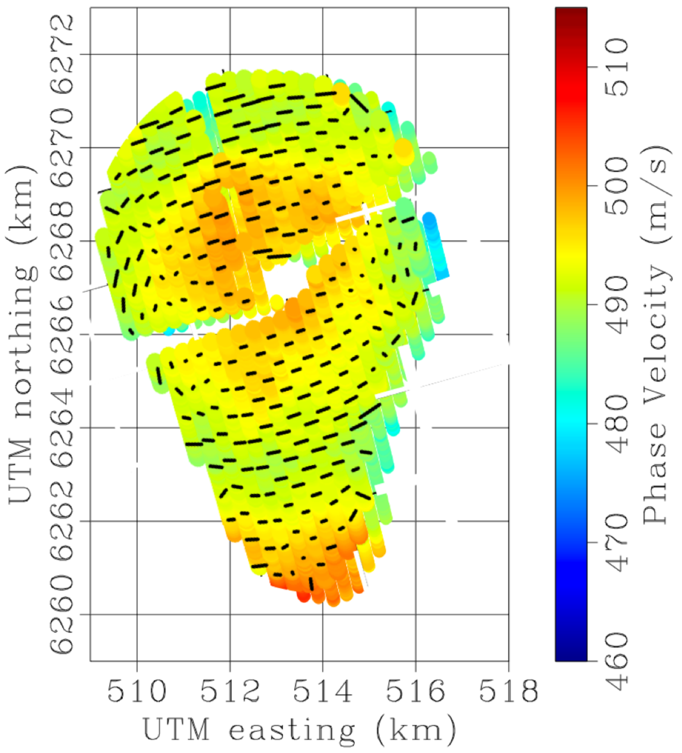
$$\partial^2 U(t,\boldsymbol{x}) = c^2(x)\,\nabla^2 U(t,\boldsymbol{x})$$

Isotropic



(De Ridder and Biondo, 2015)

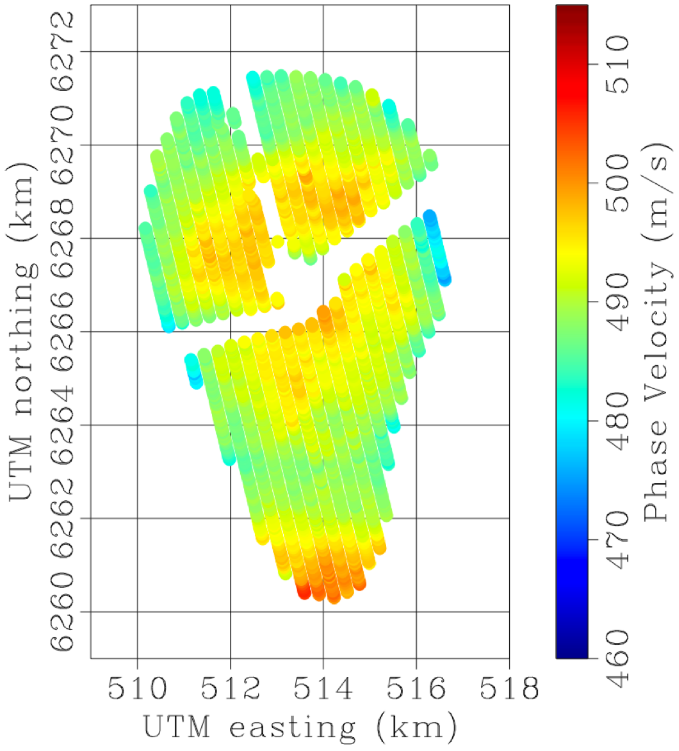
Anisotropic



(De Ridder and Curtis, 2017)

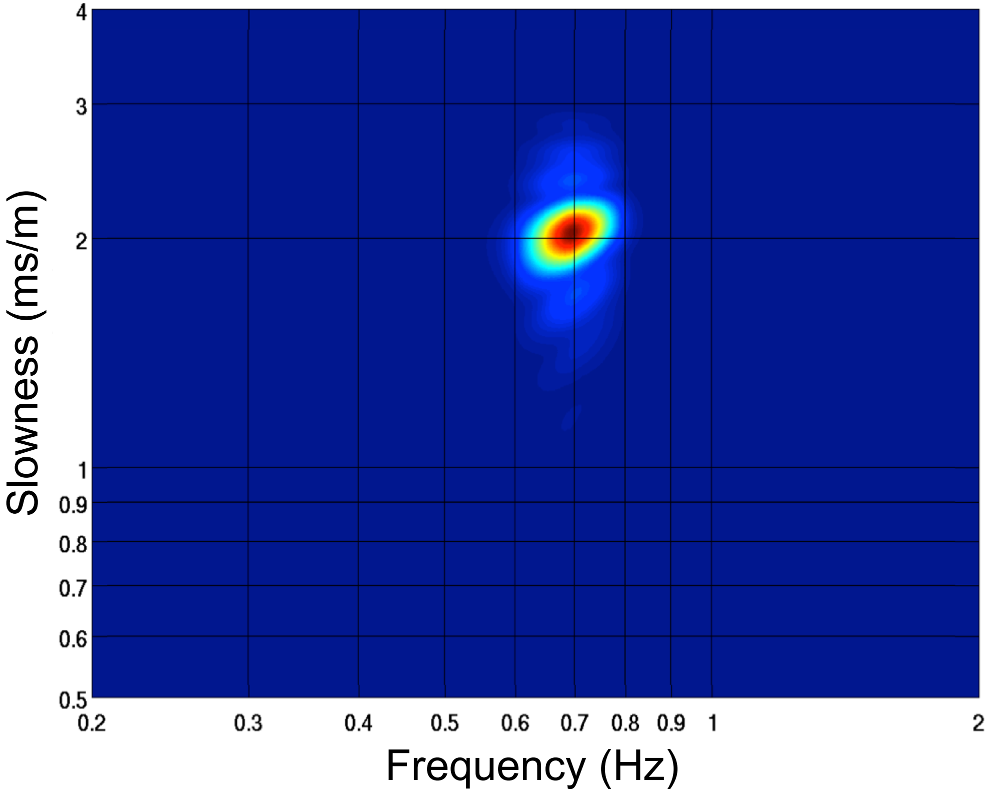
$$\partial^2 U(t, \boldsymbol{x}) = c^2(x)\, \nabla^2 U(t, \boldsymbol{x})$$

Isotropic



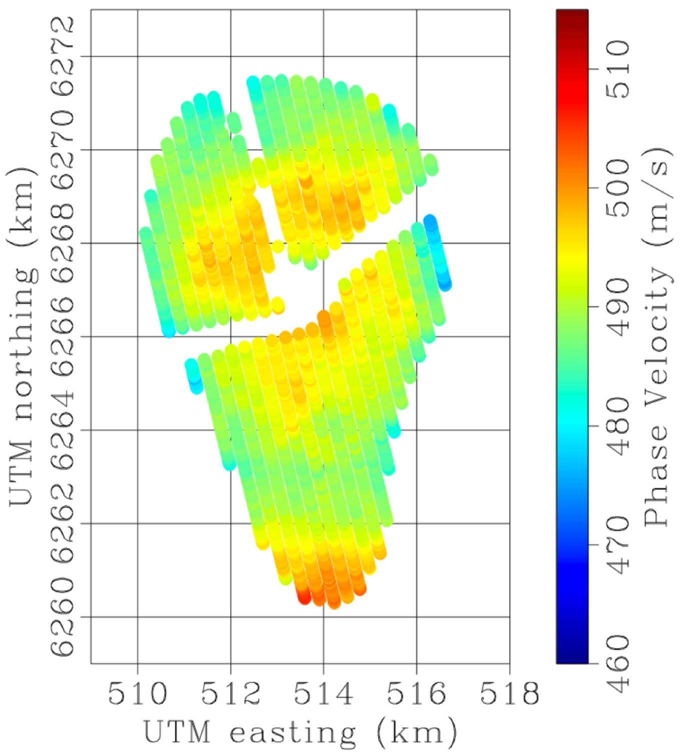
(De Ridder and Biondo, 2015)

Repeat at each frequency



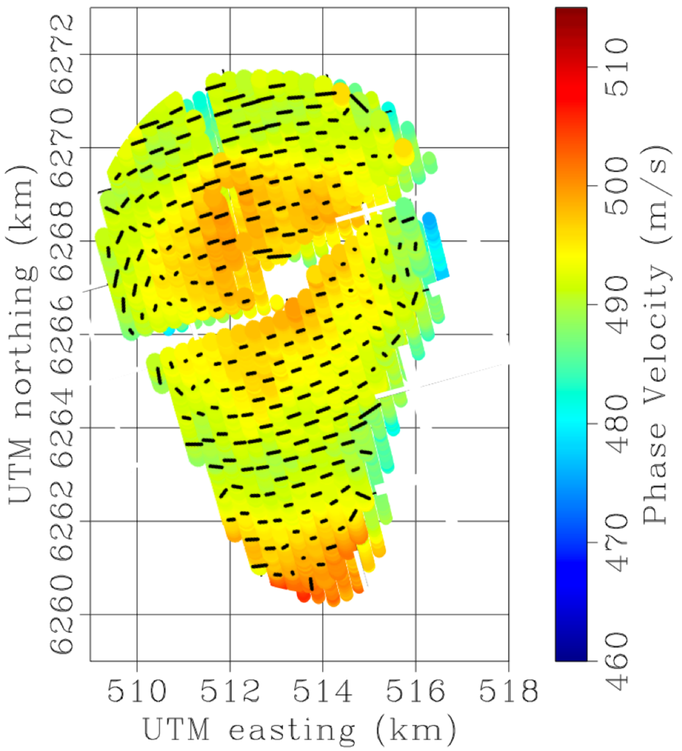
Seismic Gradiometry

Isotropic



(De Ridder and Biondo, 2015)

Anisotropic



(De Ridder and Curtis, 2017)

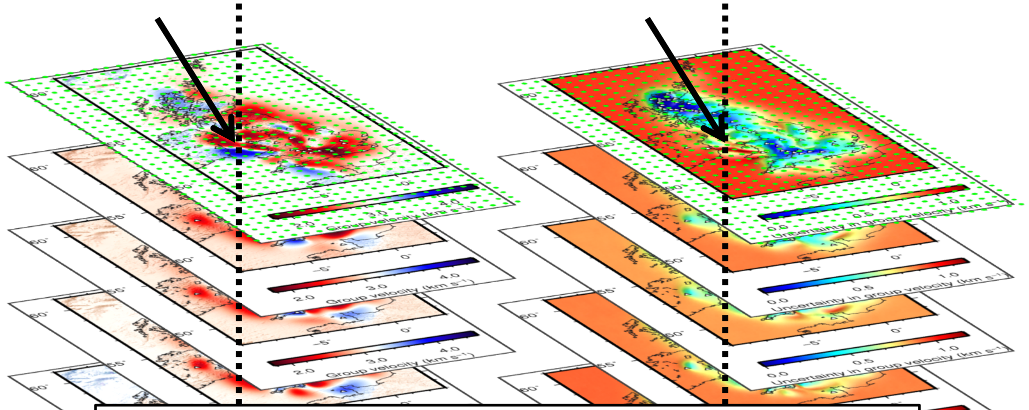
Mordret et al. (2012 – 2015)

Seismic Surface Wave Tomography : typical workflow

- **Step 1**: construct *m* x 2D phase/group velocity maps
- **Step 2**: 1D depth inversion at each grid point
  Repeat for *n* grid points ➔ 3D model

➔ **Decomposition**: 3D tomog = *m* x 2D + *n* x 1D inversions
(*m* maps, *n* locations)

Phase Velocity and Uncertainty map

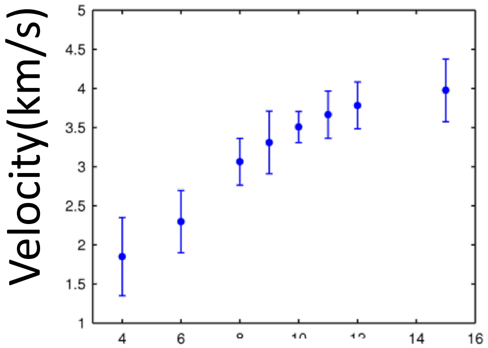


short

Period

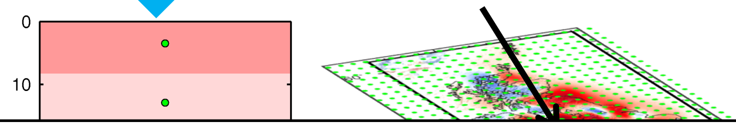
long

Step 1



1D depth inversion

Step 2



Gradiometry

Recording ~ 1 - 10 mins

Processing ~ ¼ - 10 mins

➔ <u>Near-Real Time</u>

Depth Tomography

Need uncertainty: Monte Carlo

Each grid point ~ 1 hour

All points parallelized ~ 1 - 100 hours

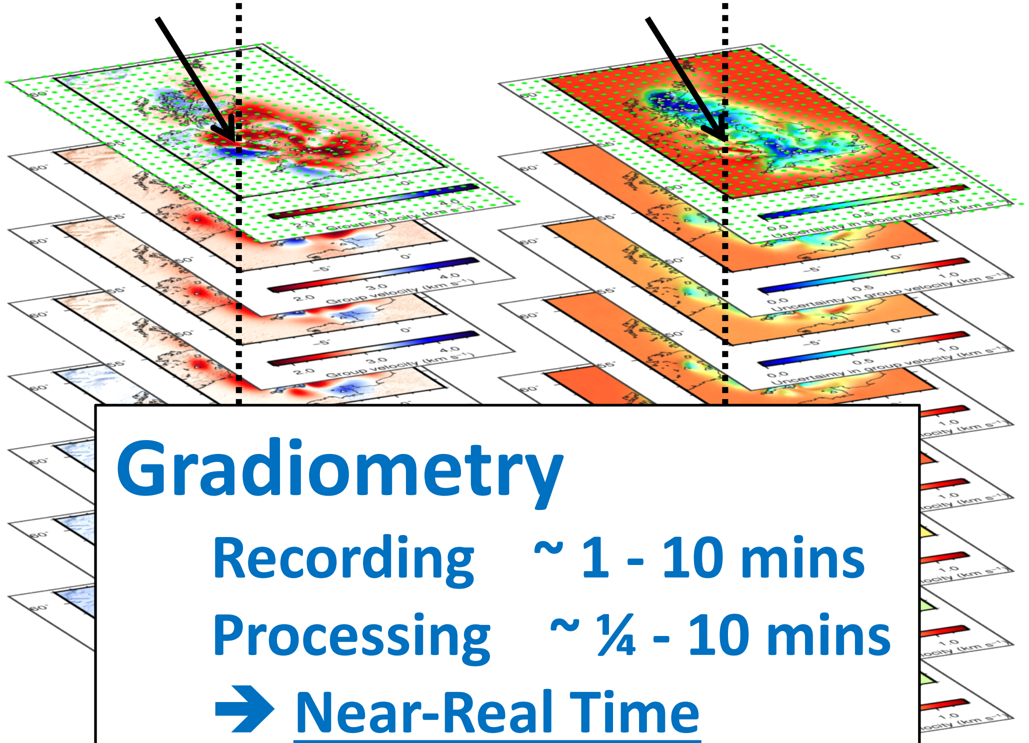
v_S (km s^{-1})

Seismic Surface Wave Tomography : typical workflow

- **Step 1**: construct *m* x 2D phase/group velocity maps
- **Step 2**: 1D depth inversion at each grid point
  Repeat for *n* grid points ➔ 3D model

➔ **Decomposition**: 3D tomog = *m* x 2D + *n* x 1D inversions
(*m* maps, *n* locations)

Phase Velocity and Uncertainty map



short

Period

long

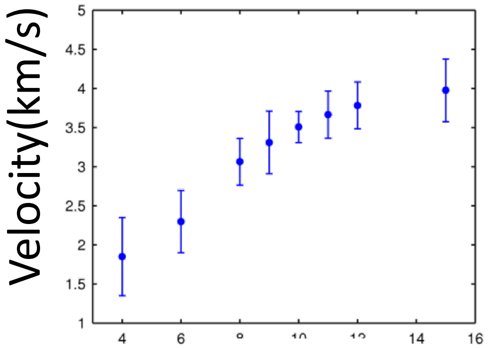
Step 1

Gradiometry

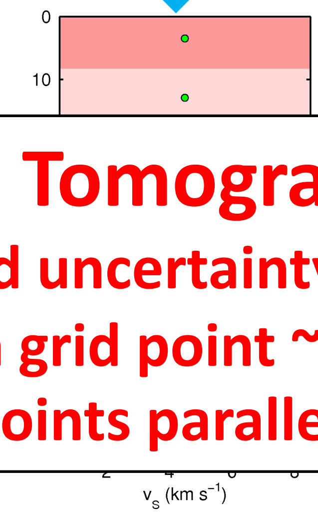
Recording ~ 1 - 10 mins

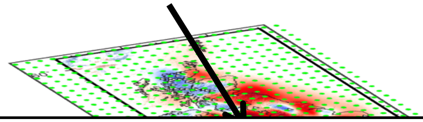
Processing ~ ¼ - 10 mins

➔ Near-Real Time



1D depth inversion





Step 2

Depth Tomography

Need uncertainty: Monte Carlo

Each grid point ~ 1 hour

All points parallelized ~ 1 - 100 hours

Neural Networks

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

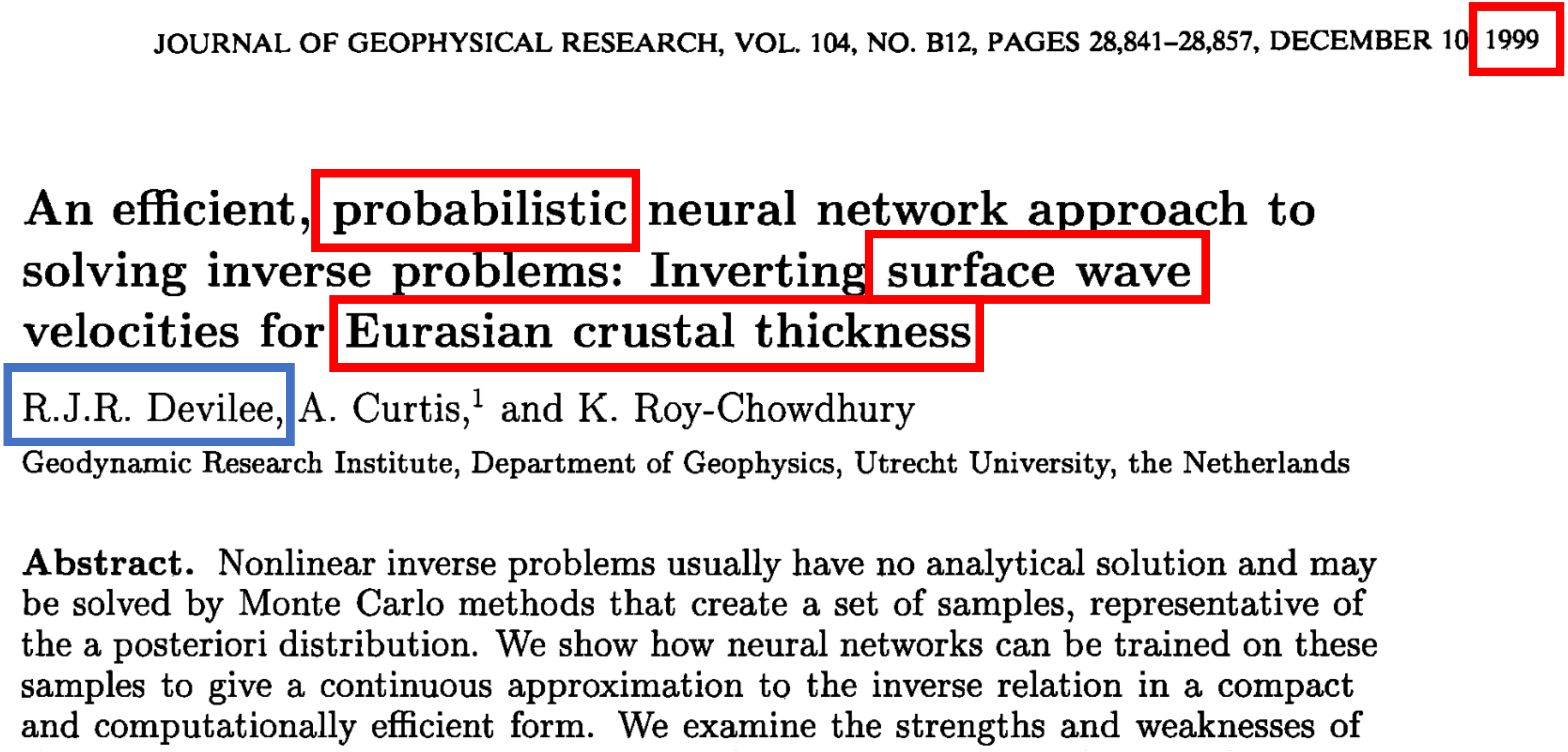
Neural networks and inversion of seismic data

Gunter Röth and Albert Tarantola

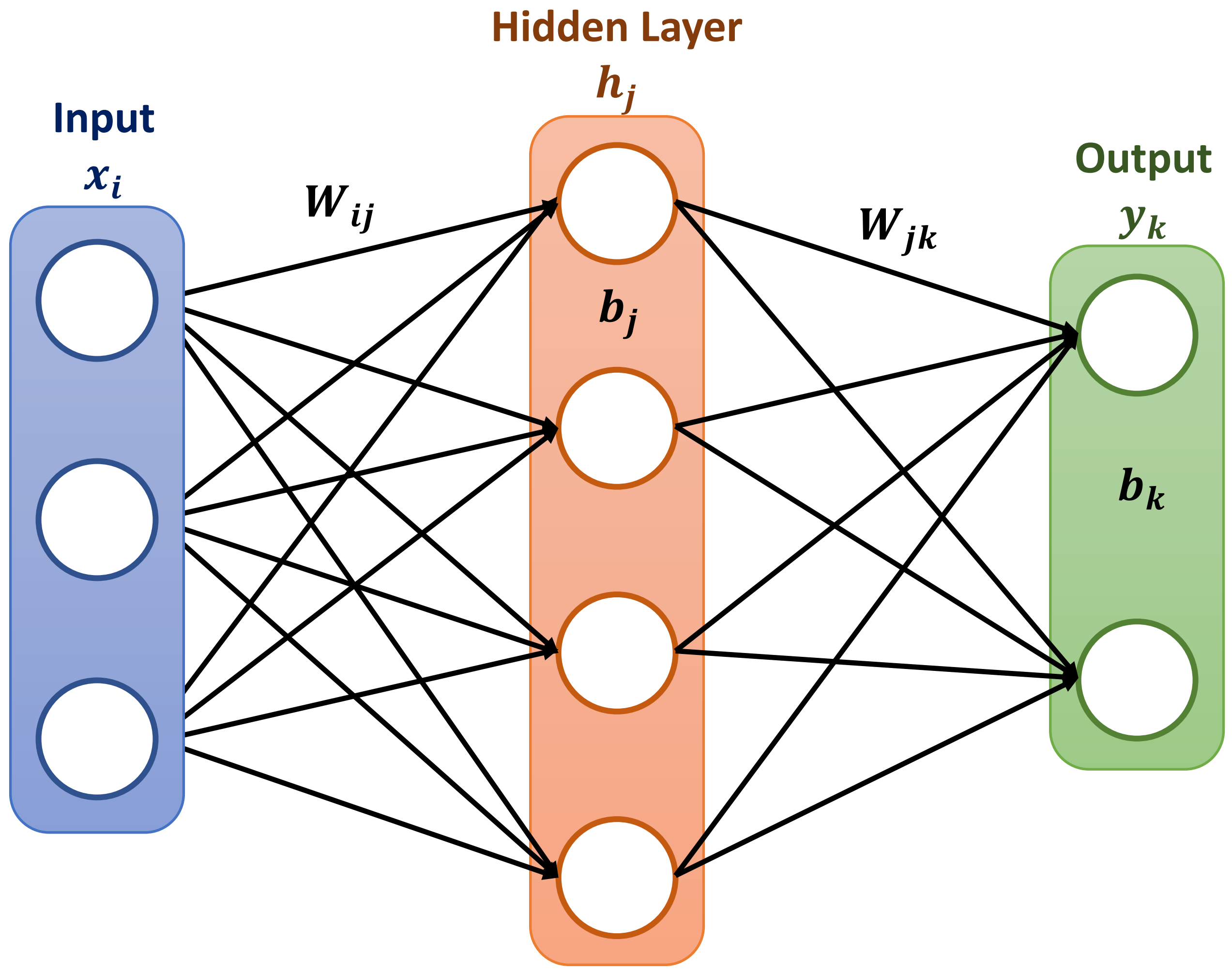
Institut de Physique du Globe de Paris, Paris, France

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

Neural Networks



Neural Network

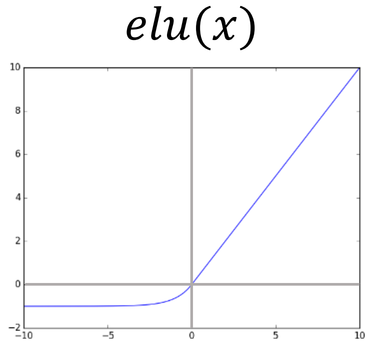


$$h_j = \sum_i W_{ij} x_i + b_j$$

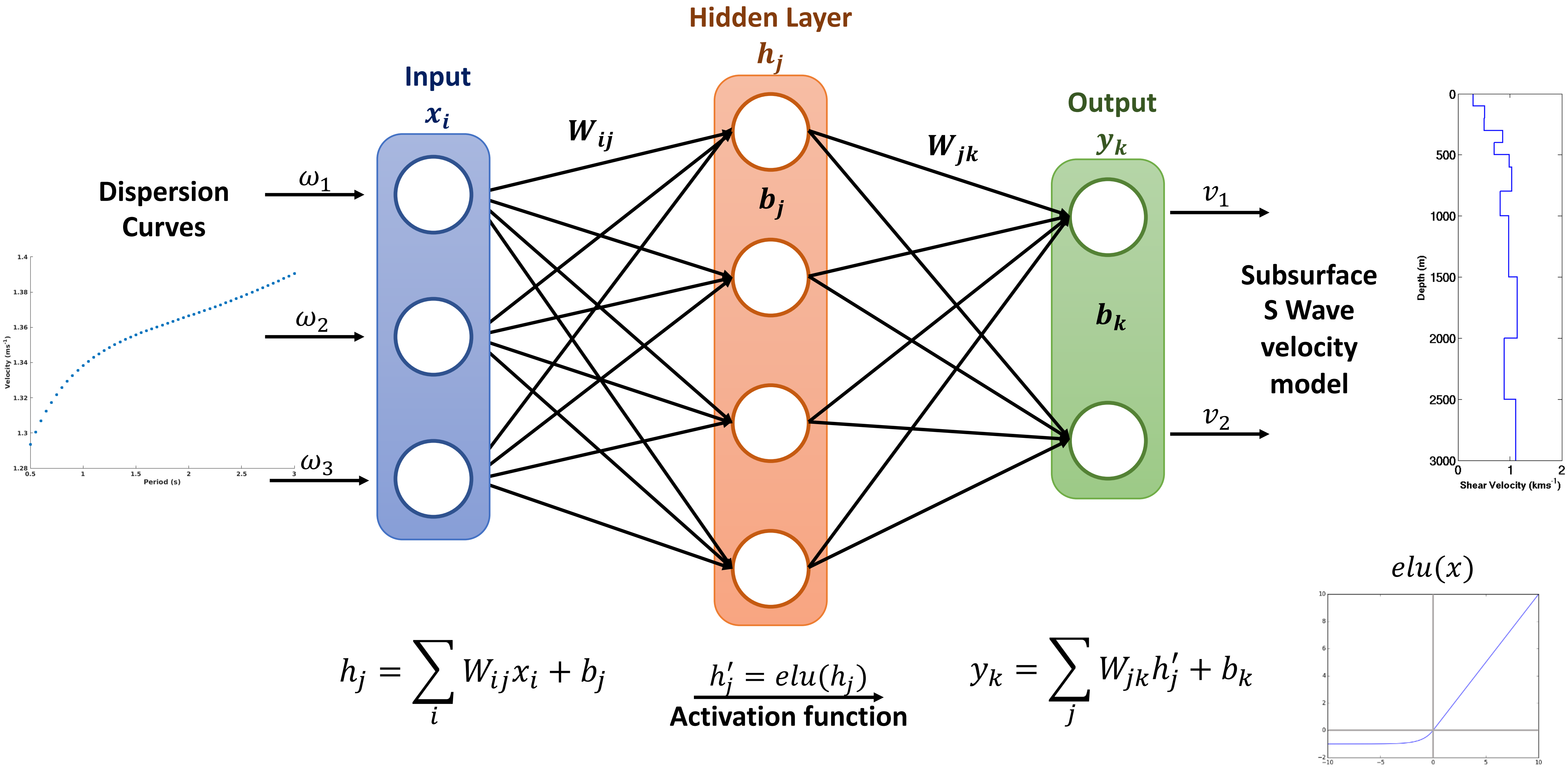
$$h'_j = elu(h_j)$$

Activation function

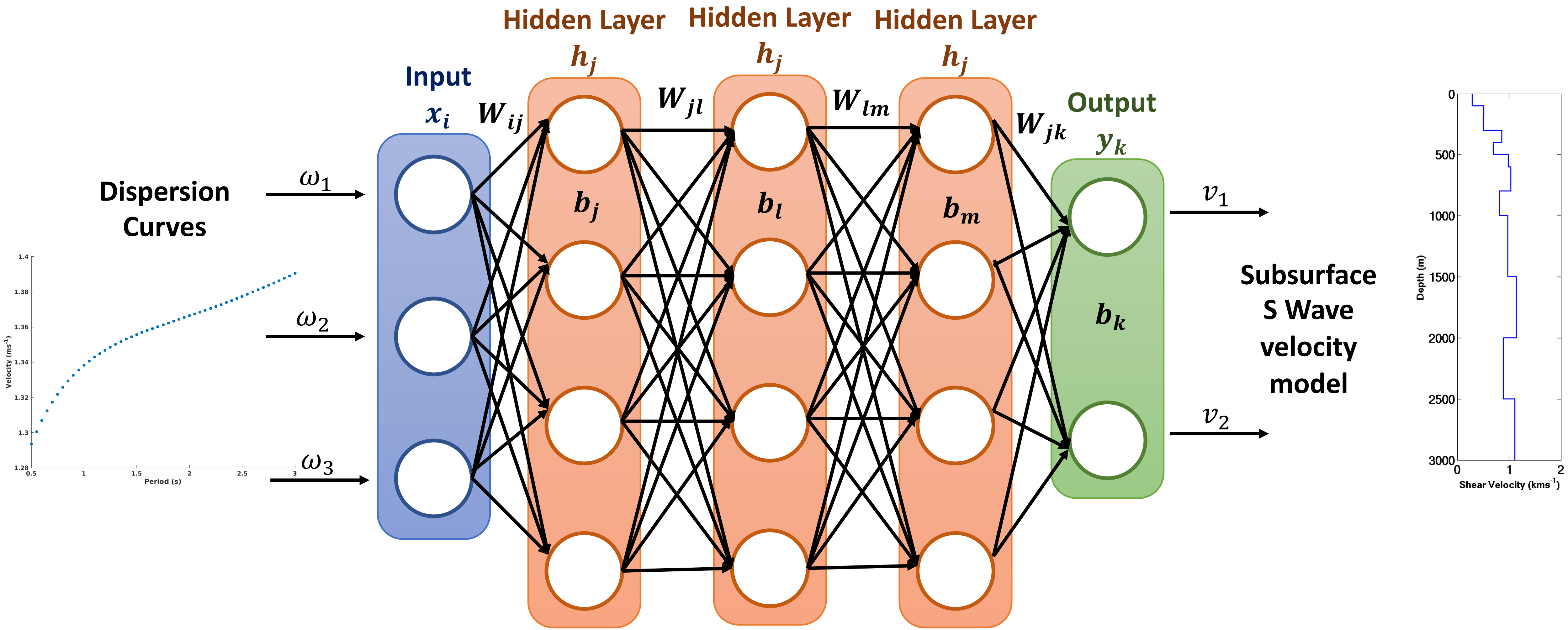
$$y_k = \sum_j W_{jk} h'_j + b_k$$



Neural Network



Neural Network



or any other network structure (GAN's, recursive, convolutional, etc.)

Mixture Density Network

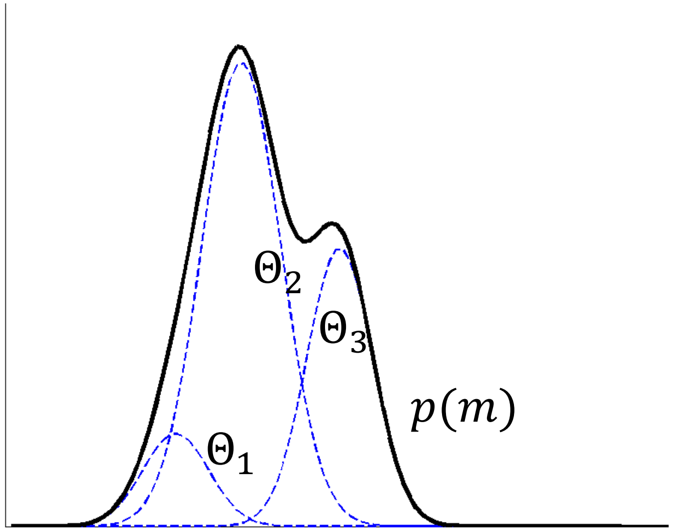
- Standard Neural Network (NN) gives no uncertainty information
- Parameterise uncertainty using mixture densities (MDN)

Weights in sum

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

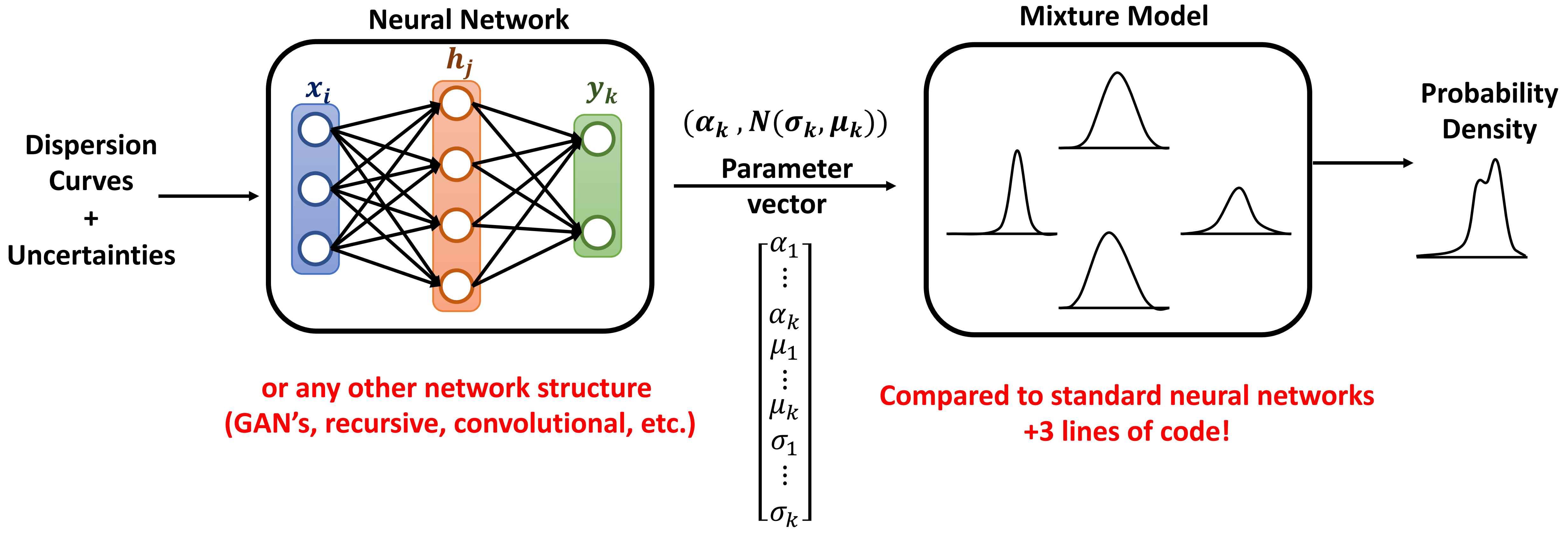
Mixture

Probability Density Kernel (Gaussian)

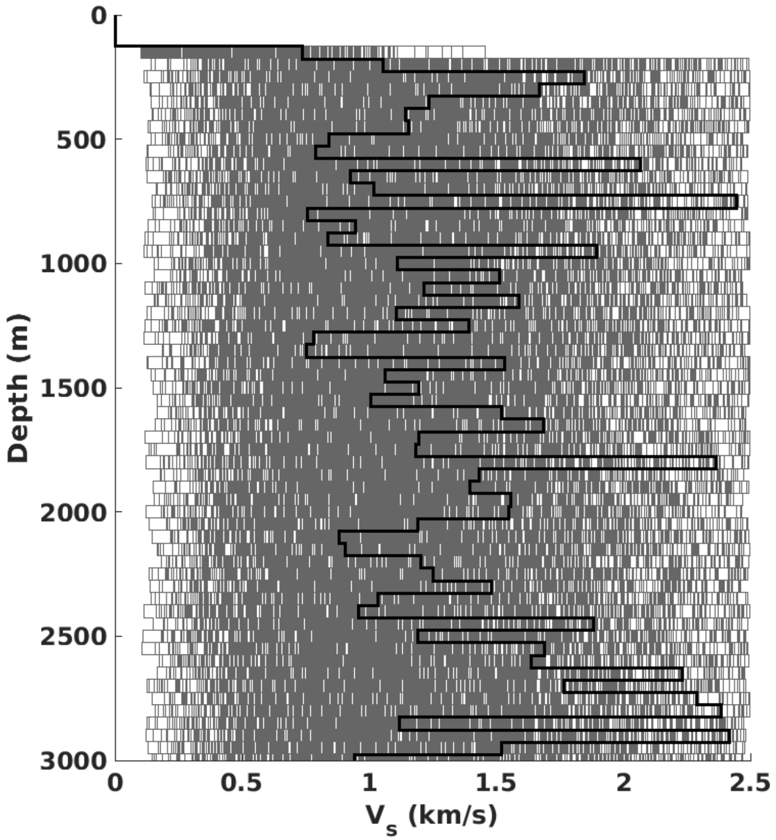


Mixture Density Network

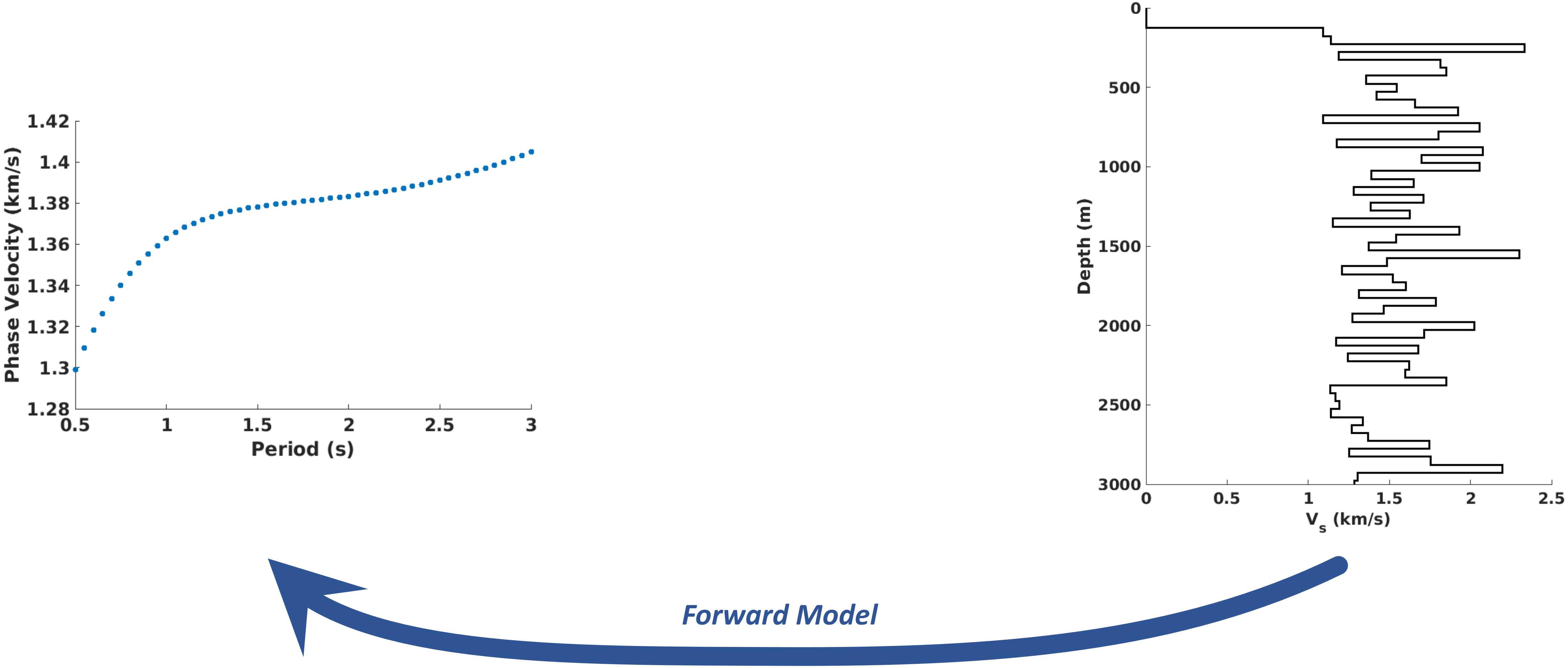
$$p(\mathbf{m}) = \sum_{k=1}^{M} \alpha_k(\boldsymbol{d})\Theta_k(\boldsymbol{m}|\boldsymbol{d})$$



Method

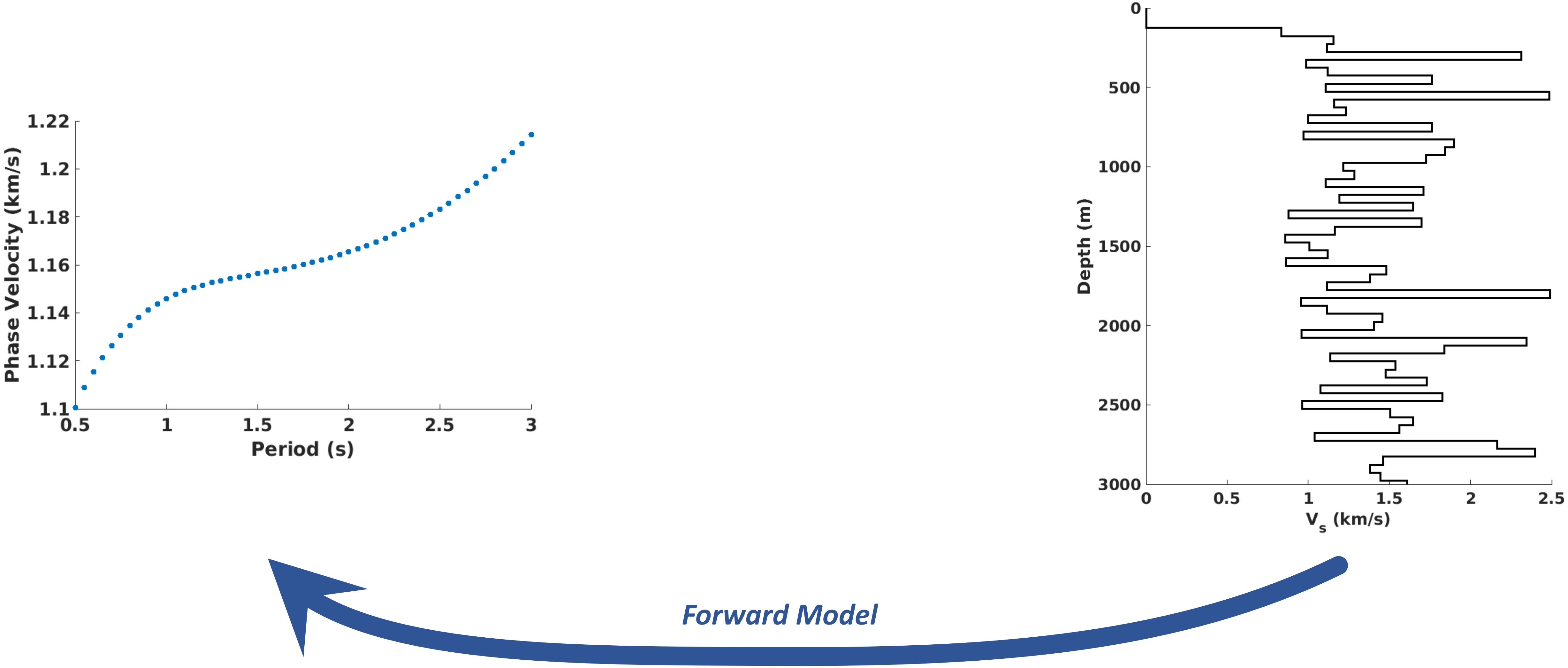


Method



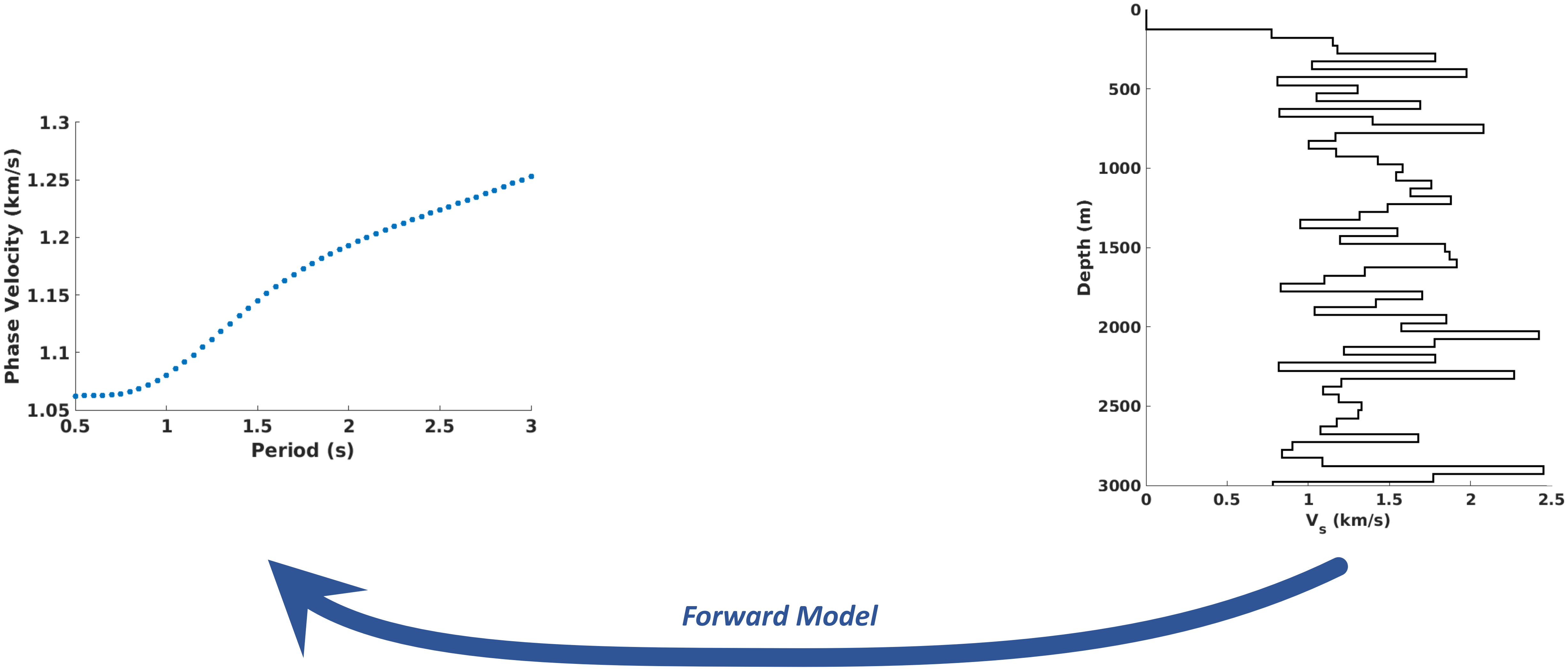
1.42
1.4
1.38
1.36
1.34
1.32
1.3
1.28
Phase Velocity (km/s)
0.5
1
1.5
2
2.5
3
Period (s)
0
500
1000
1500
2000
2500
3000
Depth (m)
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

Method

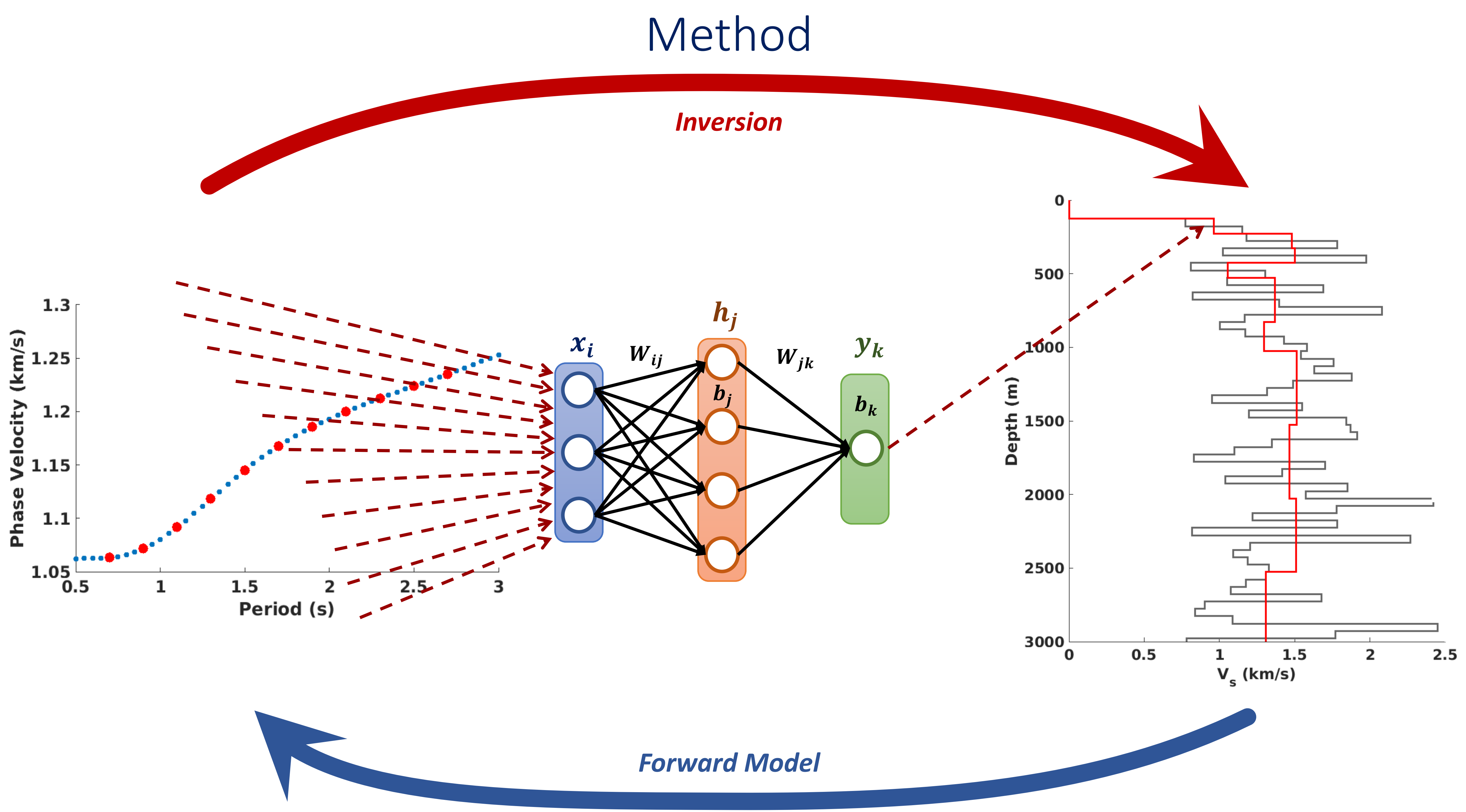


Phase Velocity (km/s)
1.22
1.2
1.18
1.16
1.14
1.12
1.1
0.5
1
1.5
2
2.5
3
Period (s)
Depth (m)
0
500
1000
1500
2000
2500
3000
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

Method

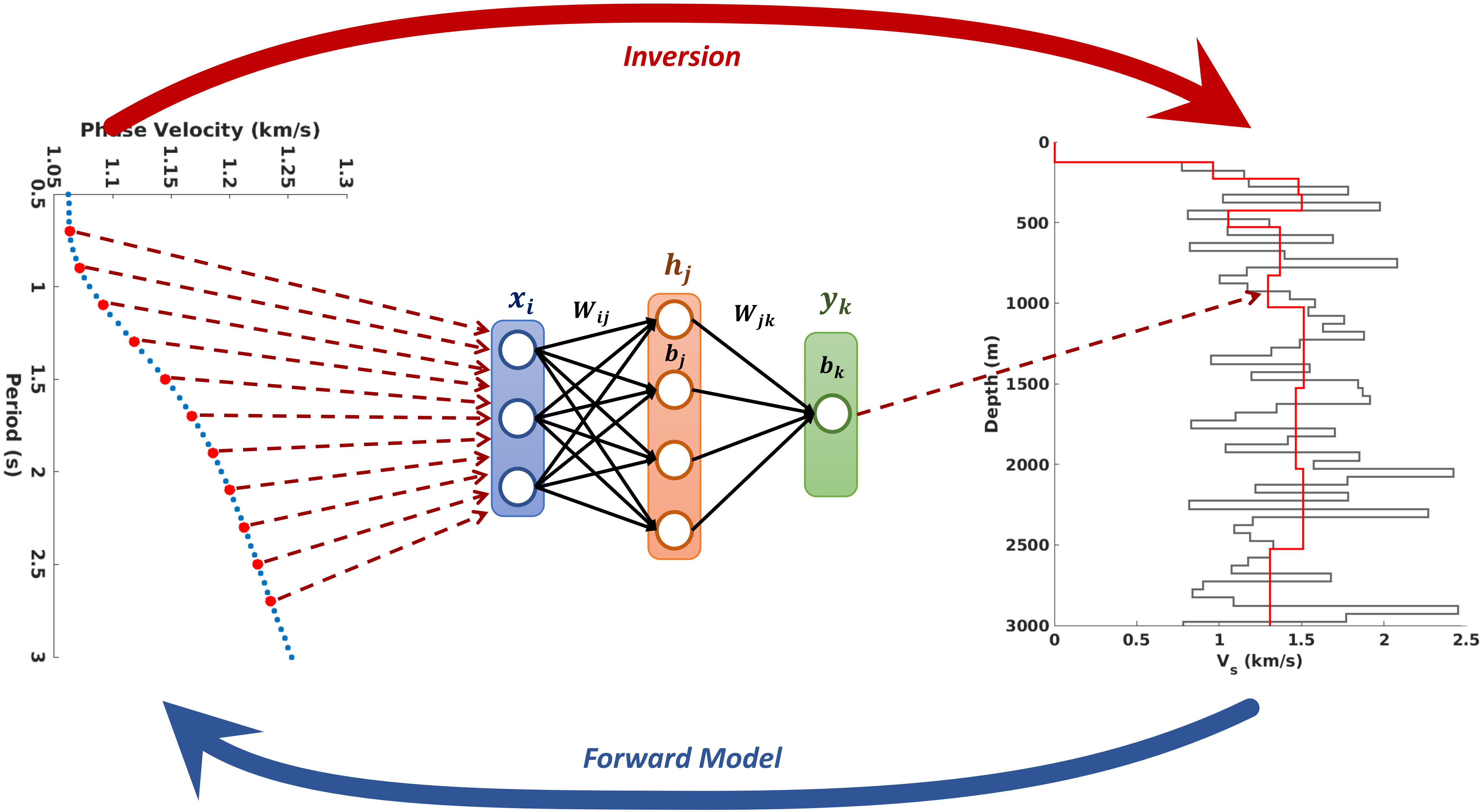


Phase Velocity (km/s)
1.3
1.25
1.2
1.15
1.1
1.05
0.5
1
1.5
2
2.5
3
Period (s)
Depth (m)
0
500
1000
1500
2000
2500
3000
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model



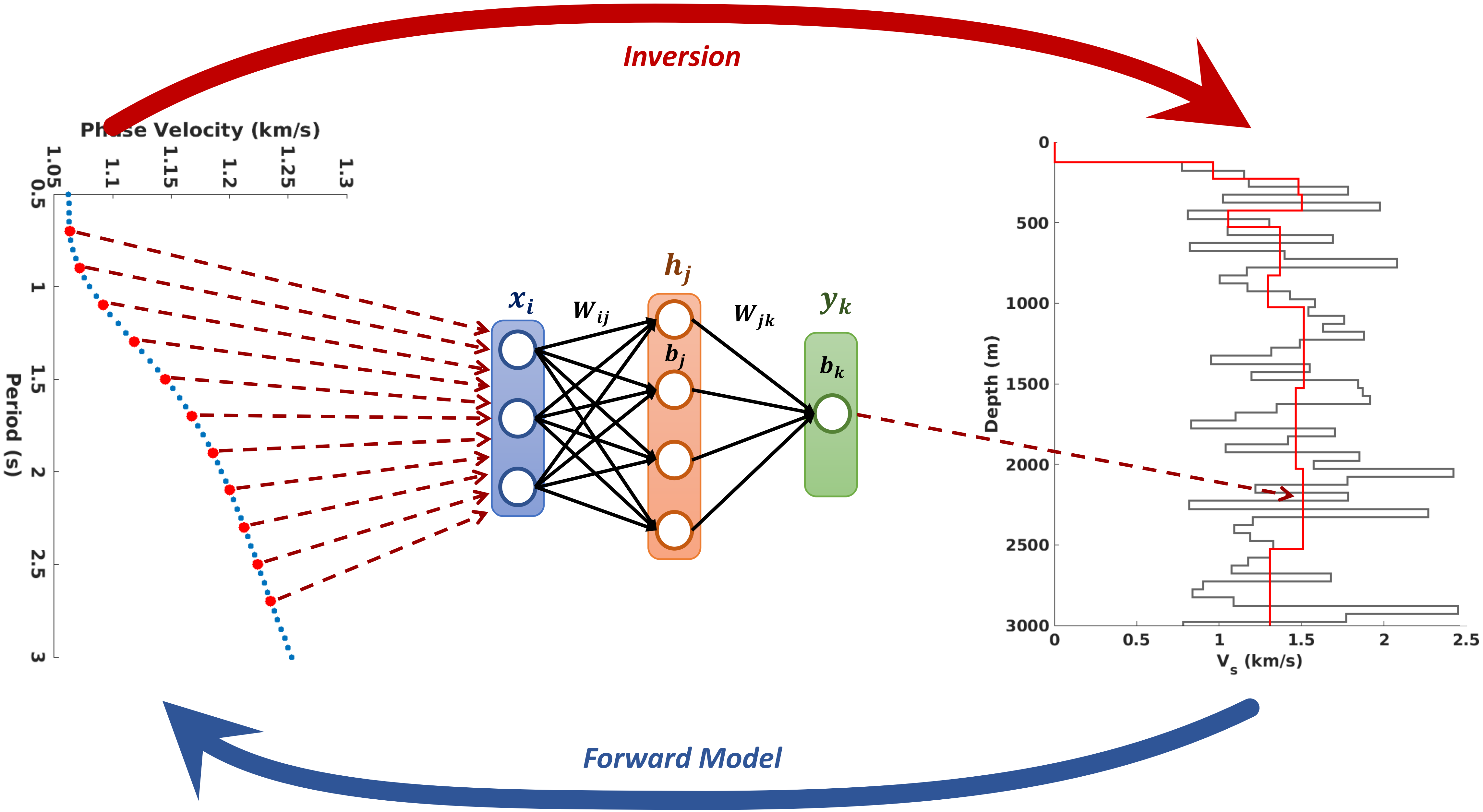
Method
Inversion
Forward Model
Phase Velocity (km/s)
Period (s)
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
V_s (km/s)

Method



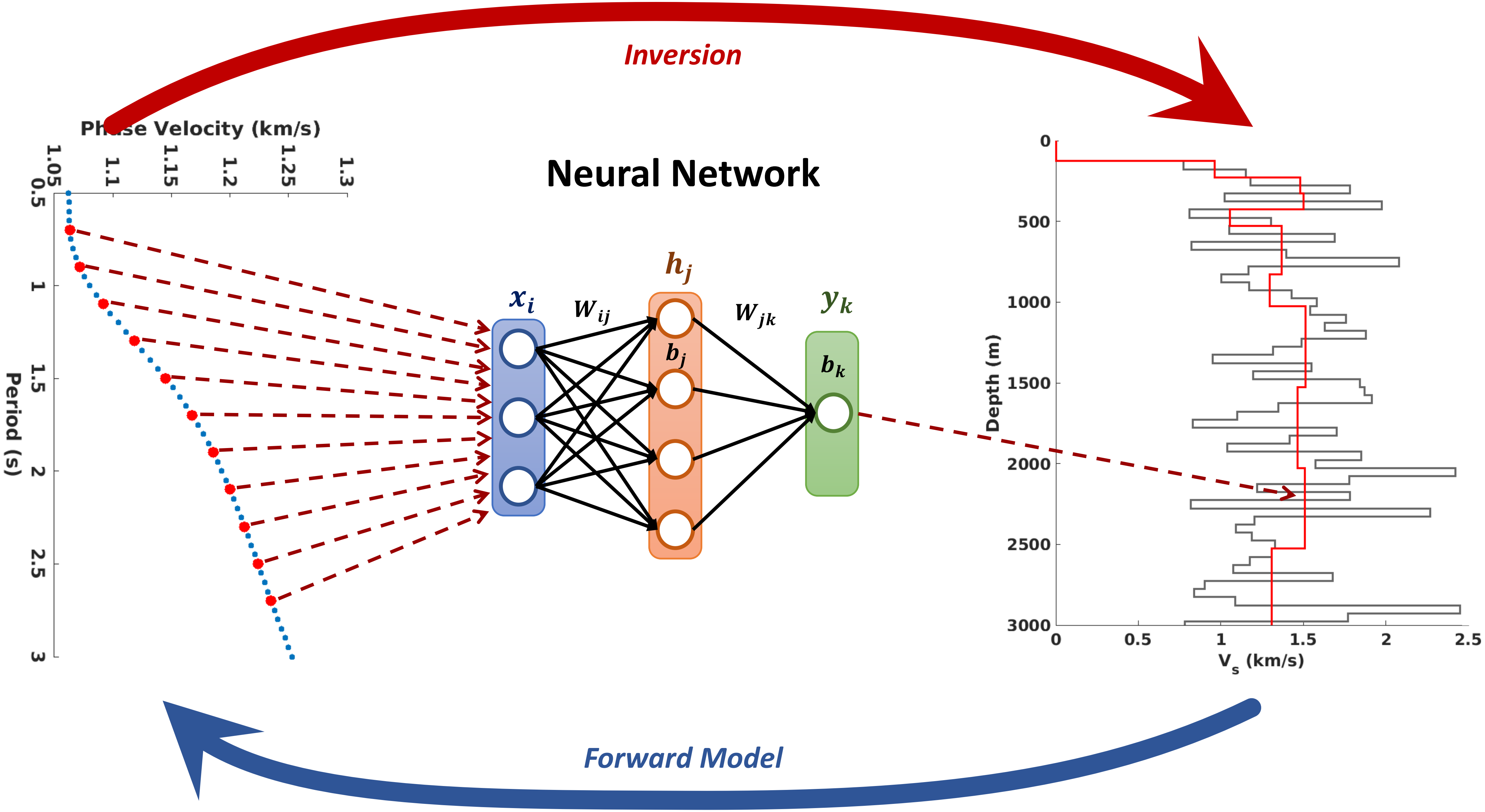
Inversion
Phase Velocity (km/s)
1.05
1.1
1.15
1.2
1.25
1.3
0.5
1
1.5
2
2.5
3
Period (s)
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
0
500
1000
1500
2000
2500
3000
Depth (m)
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

Method



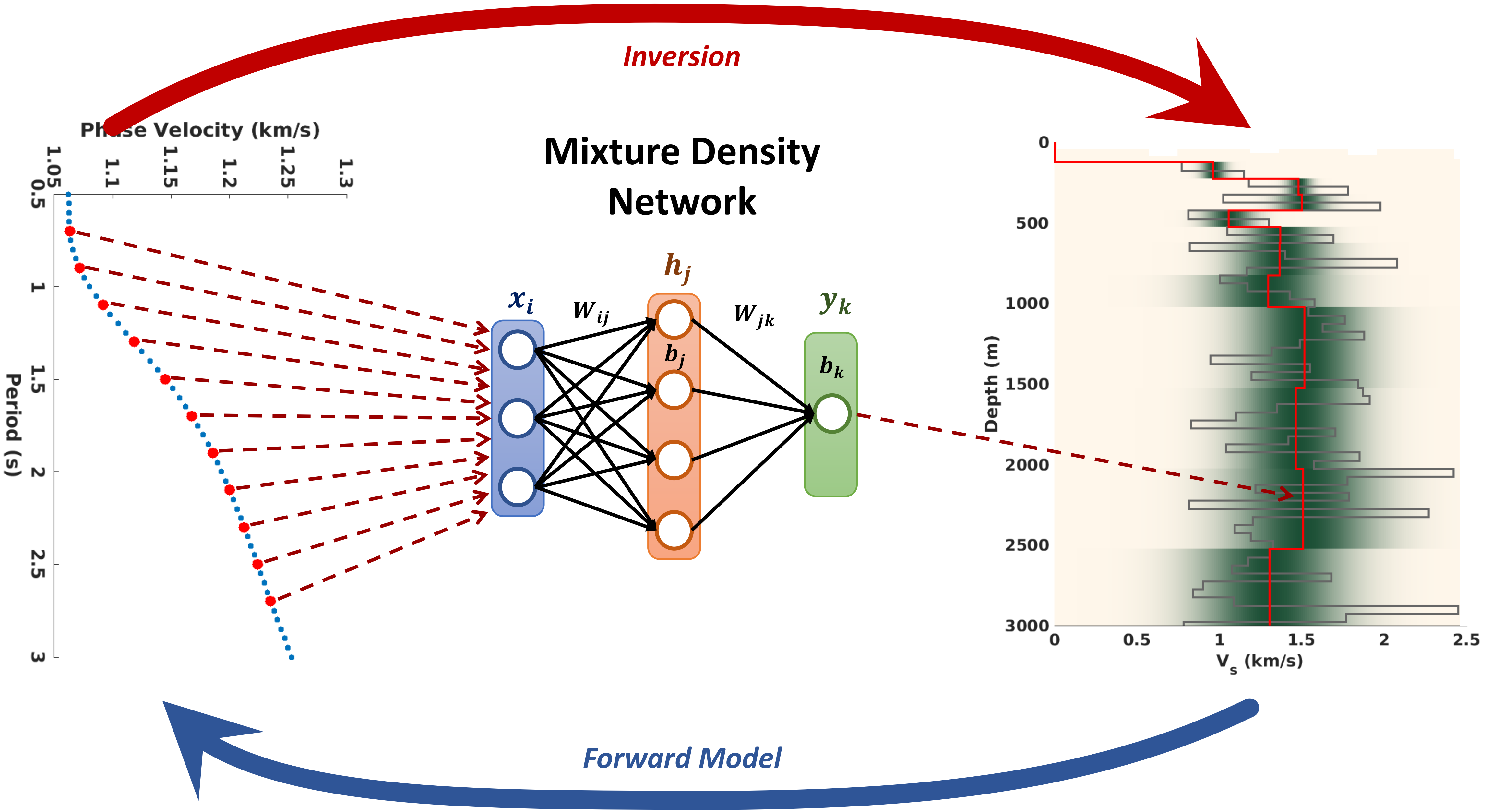
Inversion
Phase Velocity (km/s)
1.05
1.1
1.15
1.2
1.25
1.3
0.5
1
1.5
2
2.5
3
Period (s)
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
0
500
1000
1500
2000
2500
3000
Depth (m)
0
0.5
1
1.5
2
2.5
V_s (km/s)
Forward Model

Method



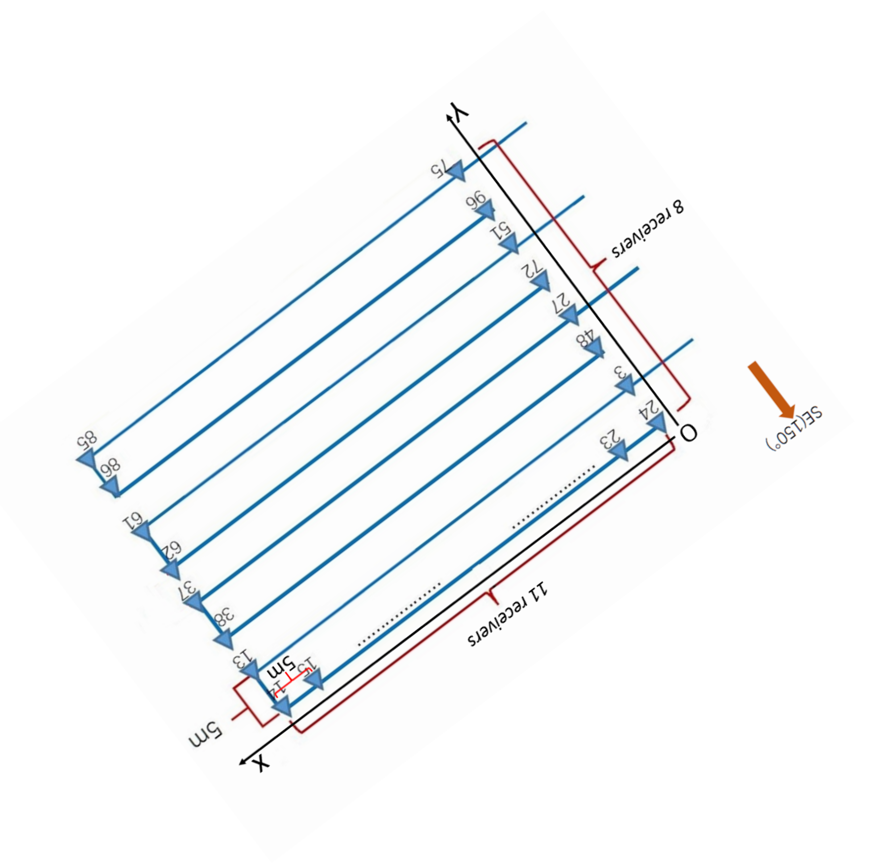
Inversion
Phase Velocity (km/s)
Period (s)
Neural Network
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
V_s (km/s)
Forward Model

Method

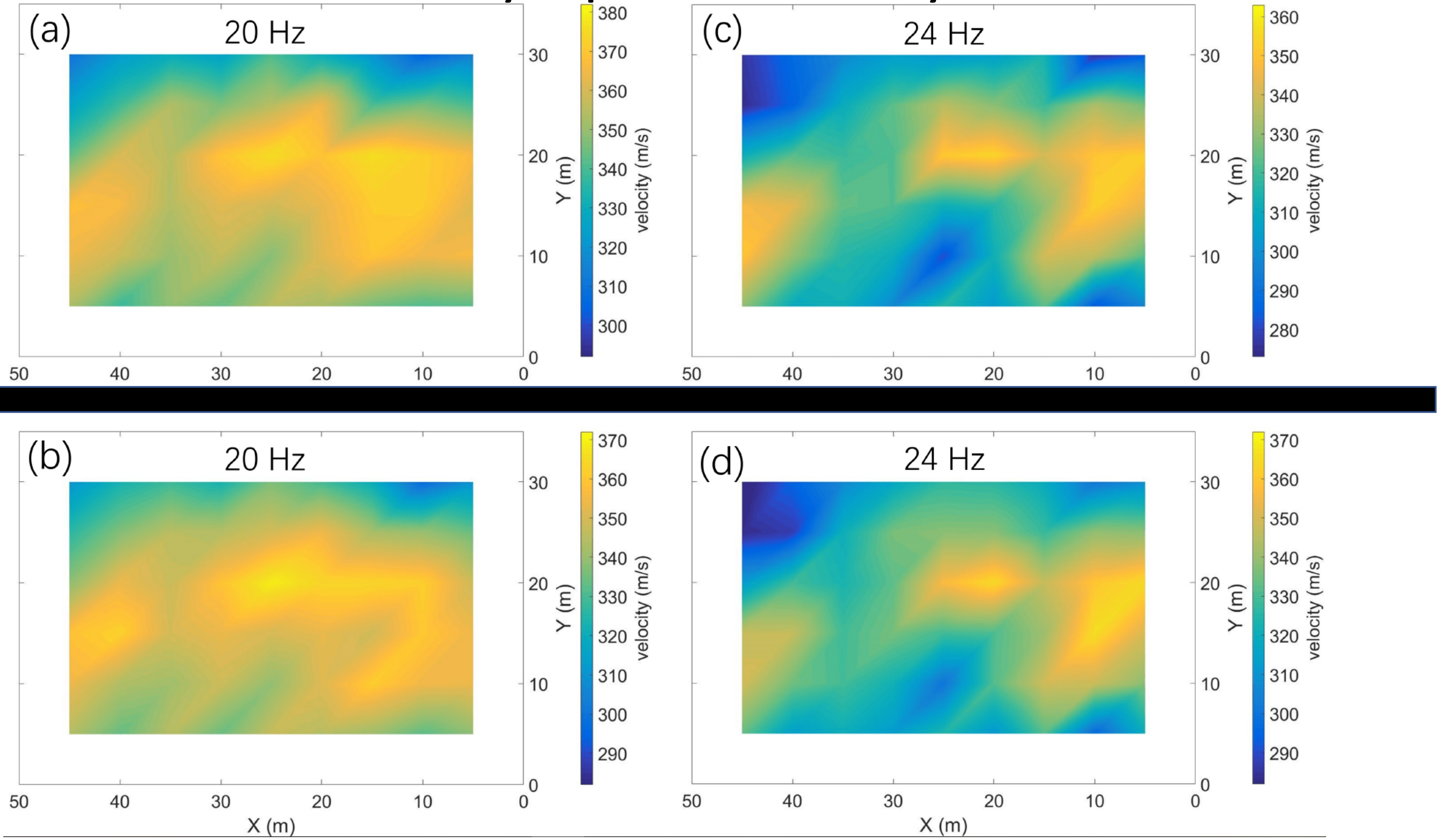


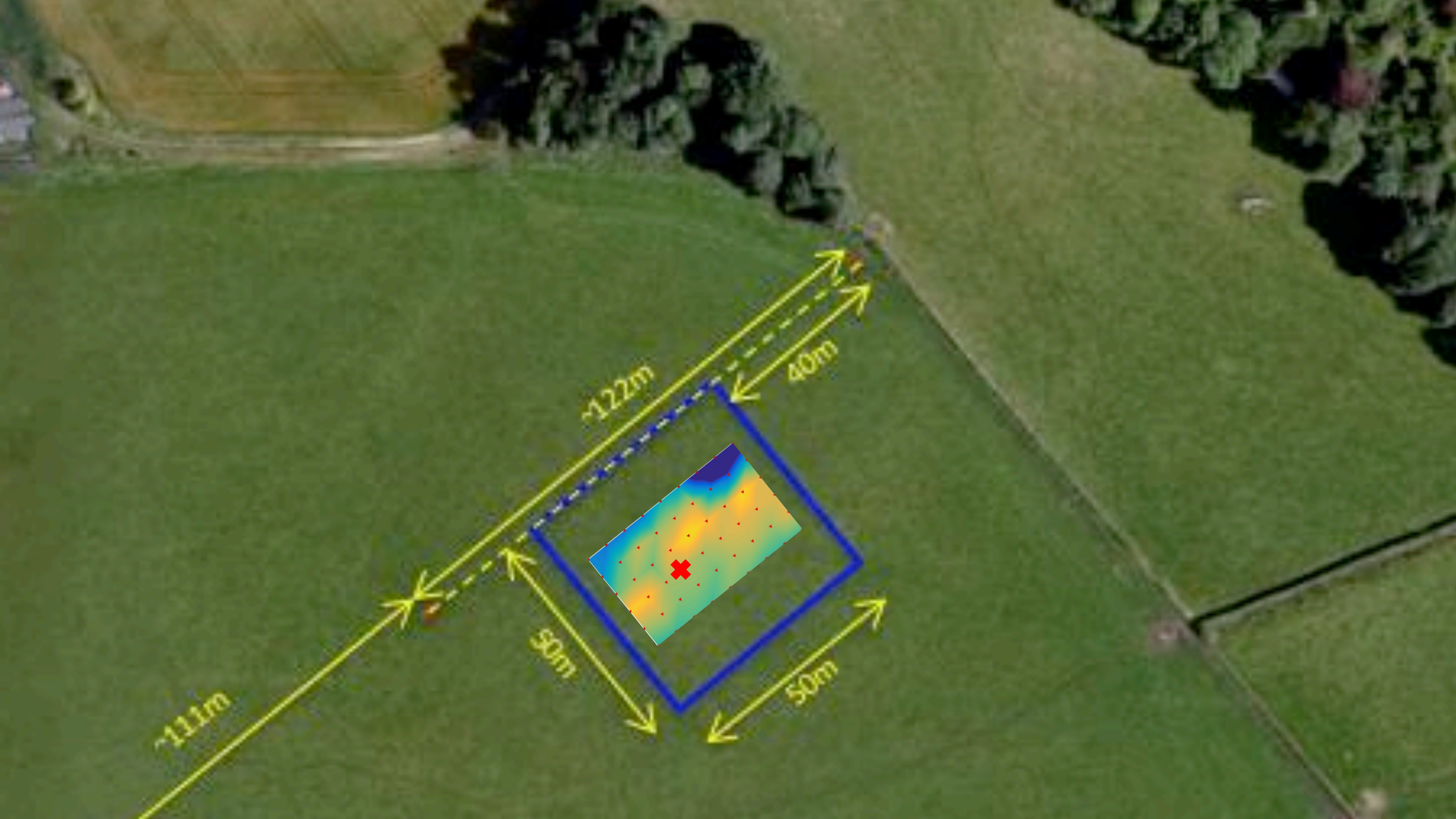
Inversion
Velocity (km/s)
Period (s)
Mixture Density Network
x_i
W_{ij}
h_j
b_j
W_{jk}
y_k
b_k
Depth (m)
V_s (km/s)
Forward Model

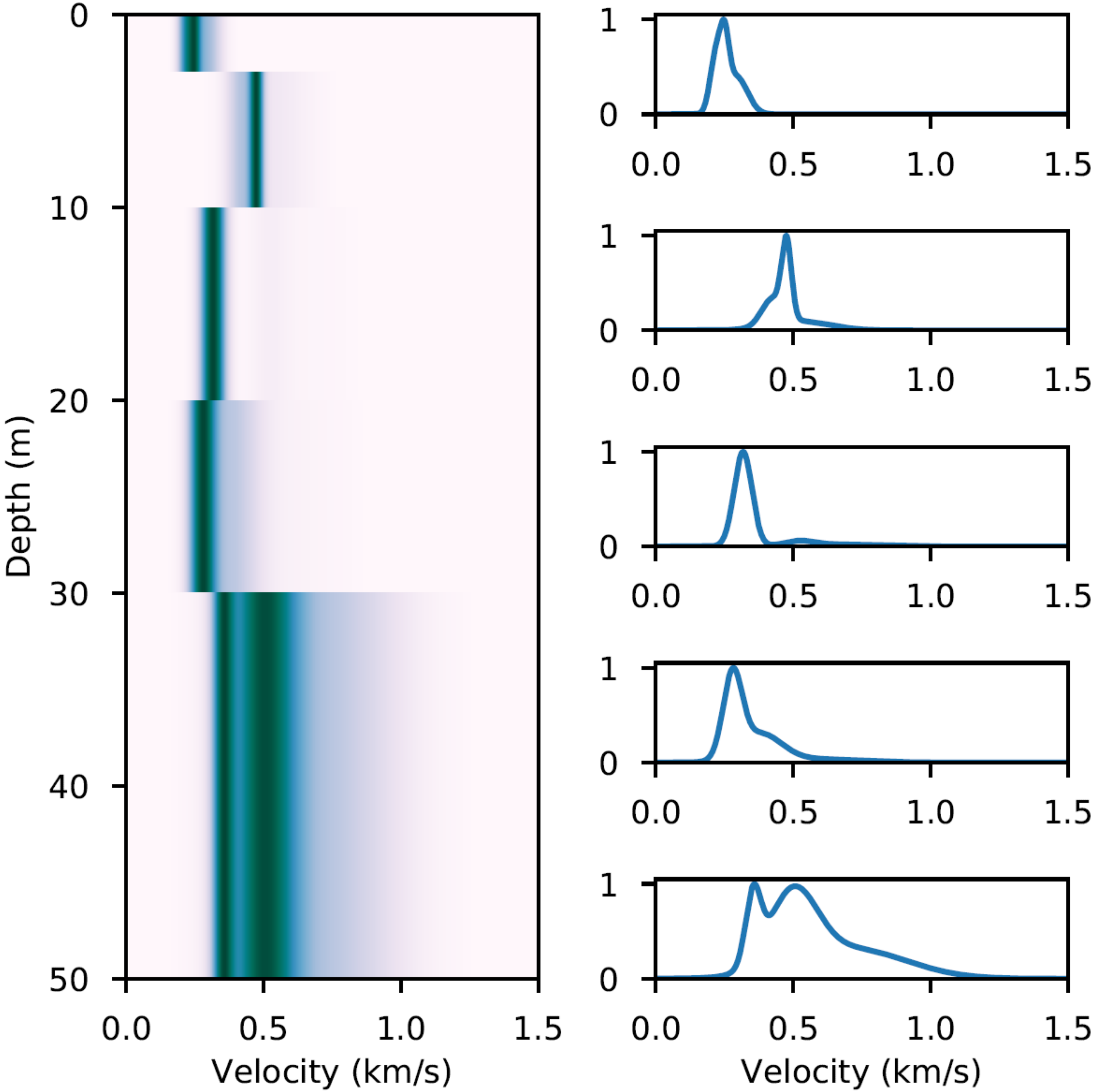
Field over a Landfill dump near the Edinburgh Ring-Road



Phase Velocity Maps from Gradiometry




~122m
40m
90m
50m
~111m



Depth (m)
0
10
20
30
40
50
0.0
0.5
1.0
1.5
Velocity (km/s)
1
0
0.0
0.5
1.0
1.5
Velocity (km/s)

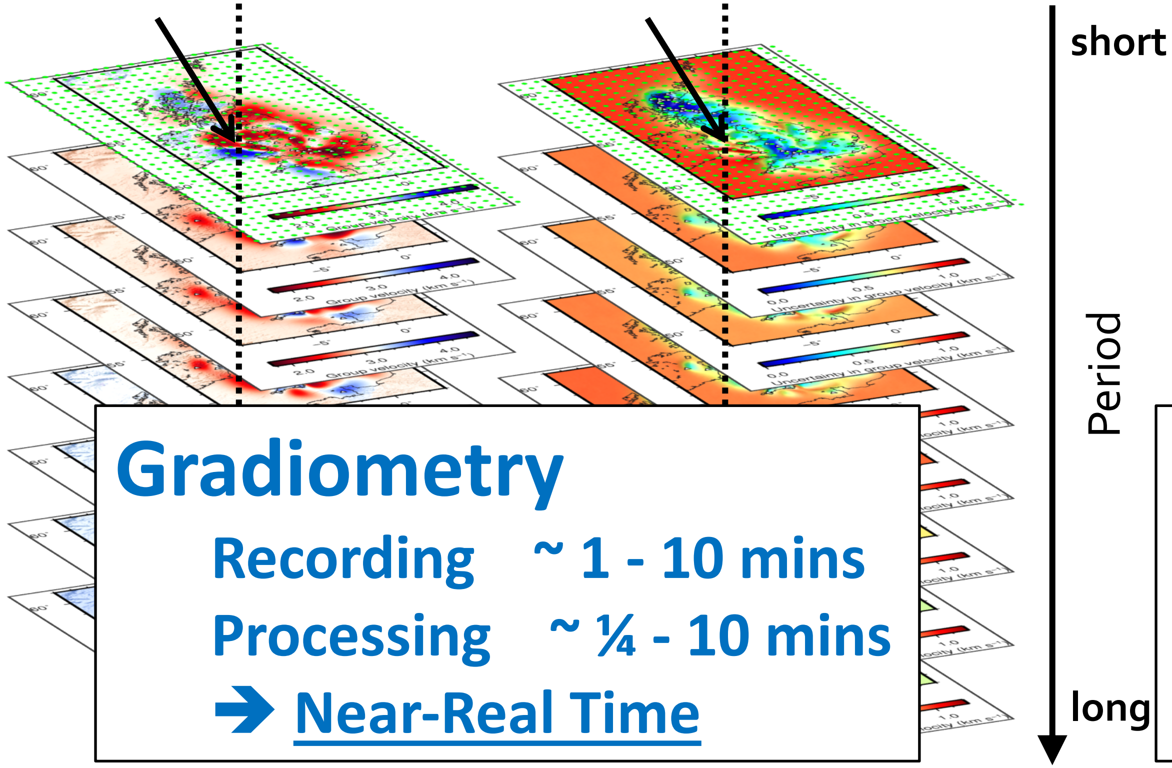
Seismic Surface Wave Tomography : typical workflow

- **Step 1**: construct *m* x 2D phase/group velocity maps
- **Step 2**: 1D depth inversion at each grid point
  Repeat for *n* grid points ➔ 3D model

➔ **Decomposition**: 3D tomog = *m* x 2D + *n* x 1D tomog's

Phase Velocity and Uncertainty map

Step 1

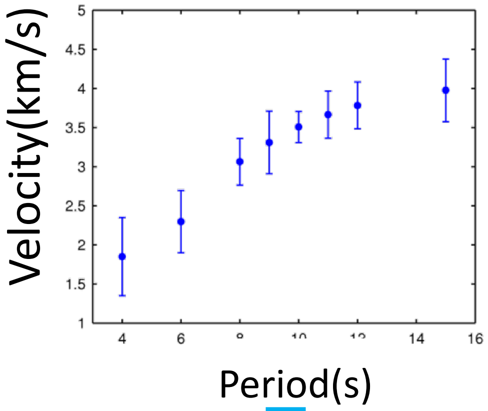


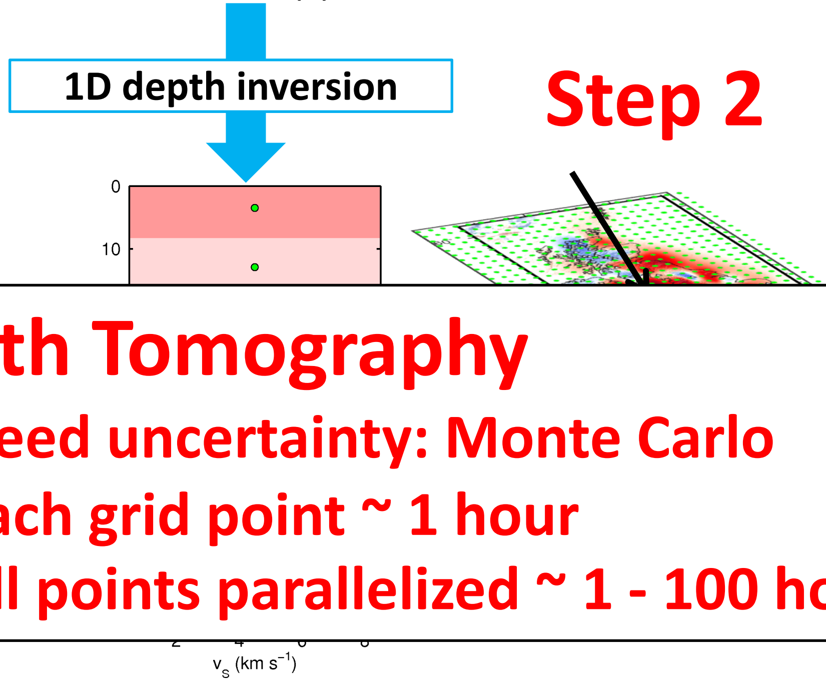
Gradiometry

Recording ~ 1 - 10 mins

Processing ~ ¼ - 10 mins

➔ Near-Real Time





Step 2

Depth Tomography

Need uncertainty: Monte Carlo

Each grid point ~ 1 hour

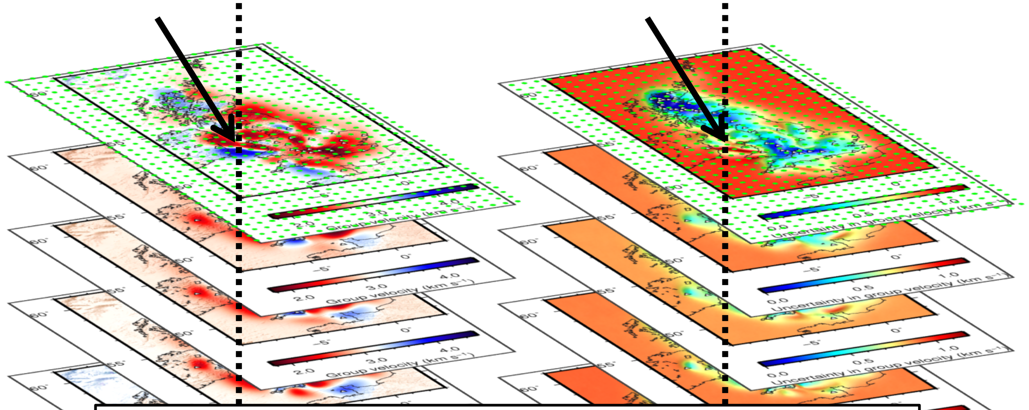
All points parallelized ~ 1 - 100 hours

Seismic Surface Wave Tomography : typical workflow

- **Step 1**: construct *m* x 2D phase/group velocity maps
- **Step 2**: 1D depth inversion at each grid point
  Repeat for *n* grid points ➔ 3D model

➔ **Decomposition**: 3D tomog = *m* x 2D + *n* x 1D tomog's

Phase Velocity and Uncertainty map

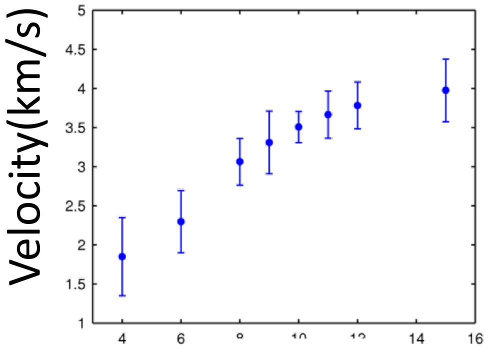


short

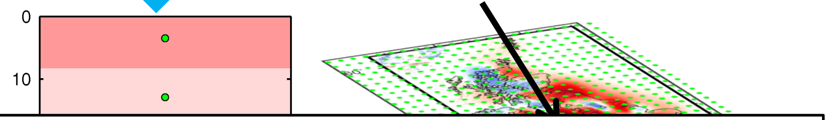
Period

long

Step 1



1D depth inversion



Step 2

Gradiometry

Recording ~ 1 - 10 mins

Processing ~ ¼ - 10 mins

➔ Near-Real Time

Neural Networks

Each grid point ~ 1 second

All points parallelized ~ 1 - 100 secs

➔ Near-Real Time

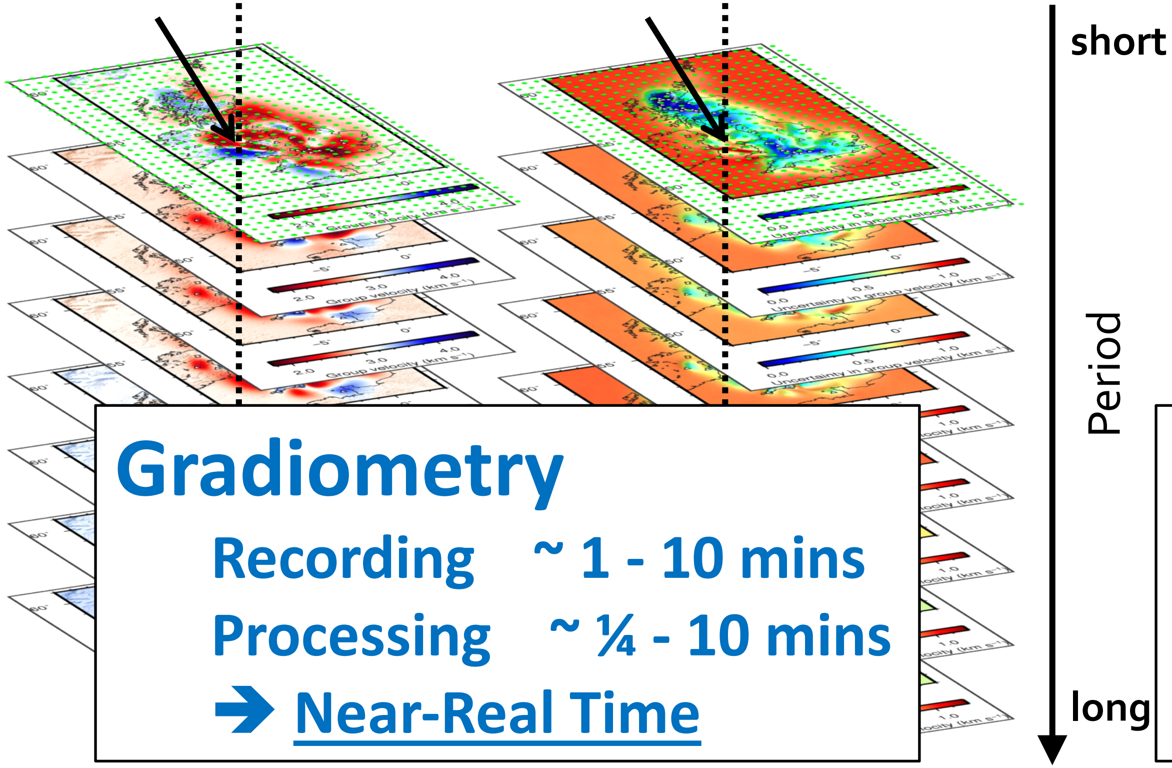
Seismic Surface Wave Tomography : typical workflow

- **Step 1**: construct *m* x 2D phase/group velocity maps
- **Step 2**: 1D depth inversion at each grid point
  Repeat for *n* grid points ➔ 3D model

➔ **Decomposition**: 3D tomog = *m* x 2D + *n* x 1D tomog's

Phase Velocity and Uncertainty map

Step 1

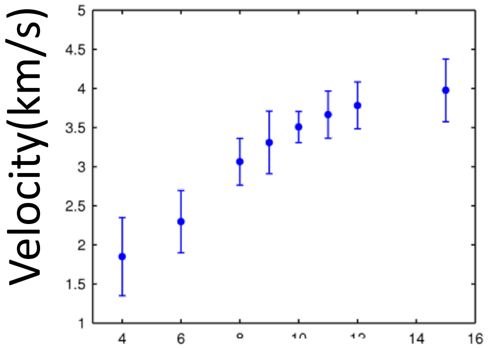


Gradiometry

Recording ~ 1 - 10 mins

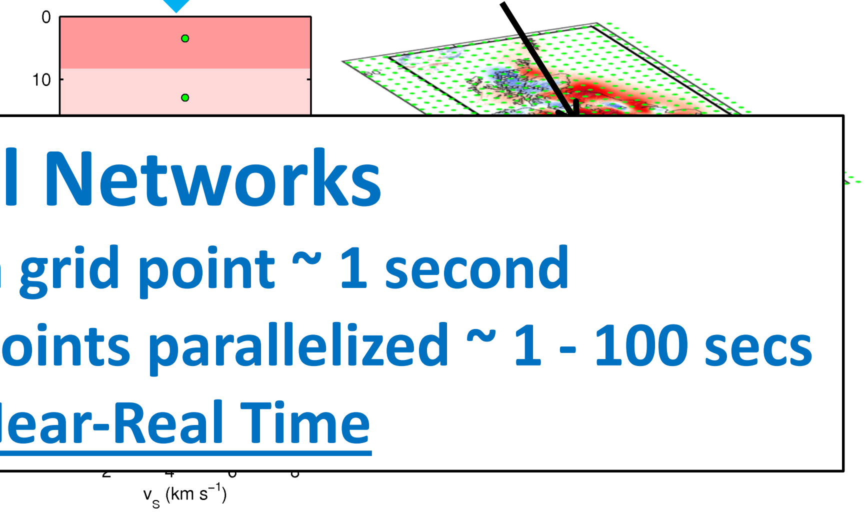
Processing ~ ¼ - 10 mins

➔ Near-Real Time



1D depth inversion

Step 2



Neural Networks

Each grid point ~ 1 second

All points parallelized ~ 1 - 100 secs

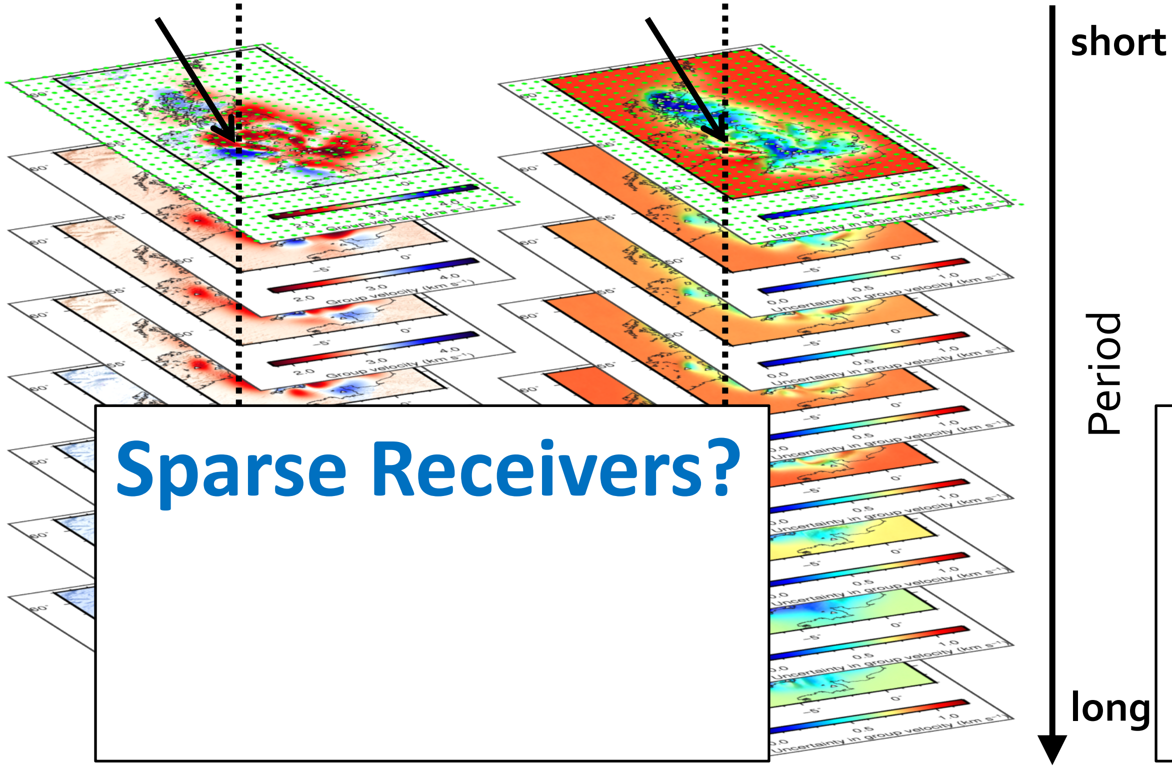
➔ Near-Real Time

Seismic Surface Wave Tomography : typical workflow

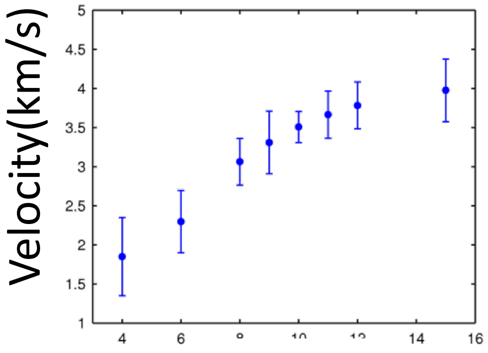
- **Step 1**: construct *m* x 2D phase/group velocity maps
- **Step 2**: 1D depth inversion at each grid point
  Repeat for *n* grid points ➔ 3D model

➔ **Decomposition**: 3D tomog = *m* x 2D + *n* x 1D tomog's

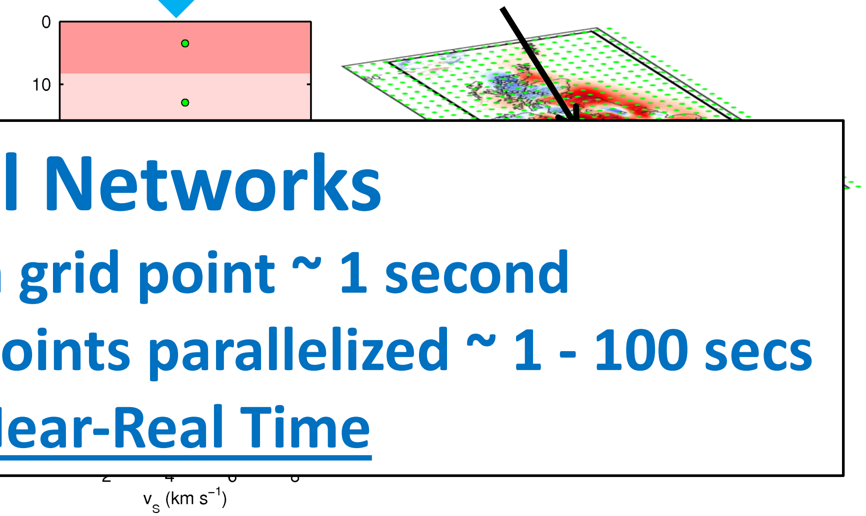
Phase Velocity and Uncertainty map



Step 1



1D depth inversion



Step 2

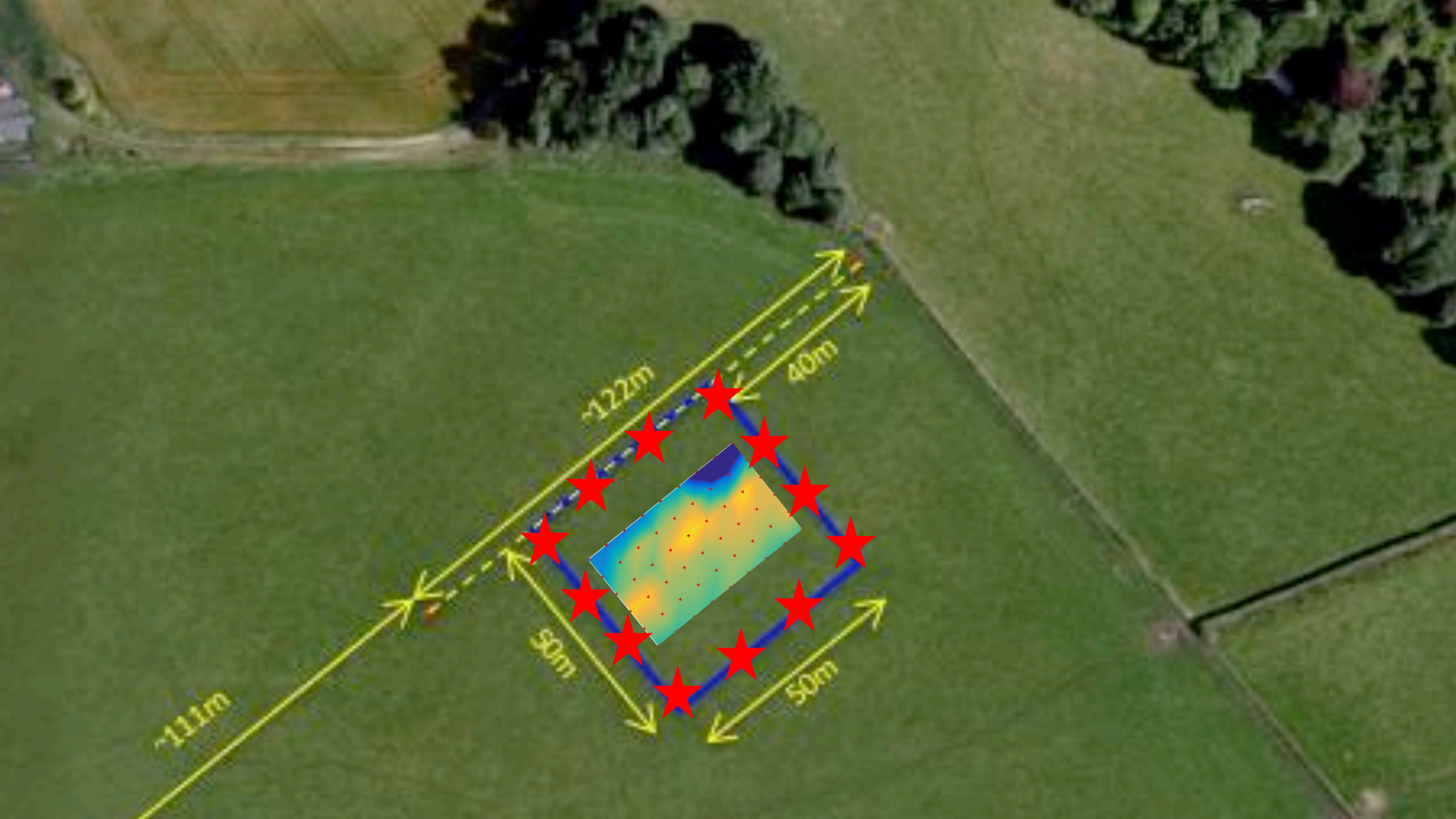
Sparse Receivers?

Neural Networks

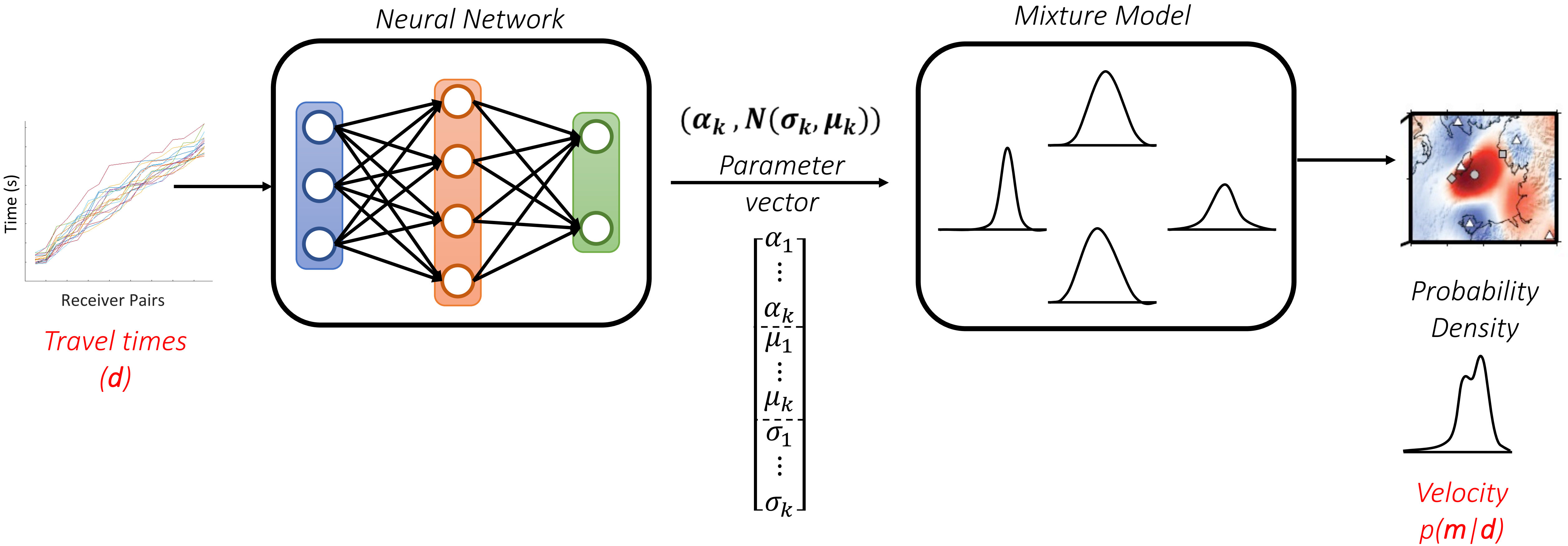
Each grid point ~ 1 second

All points parallelized ~ 1 - 100 secs

➔ **Near-Real Time**

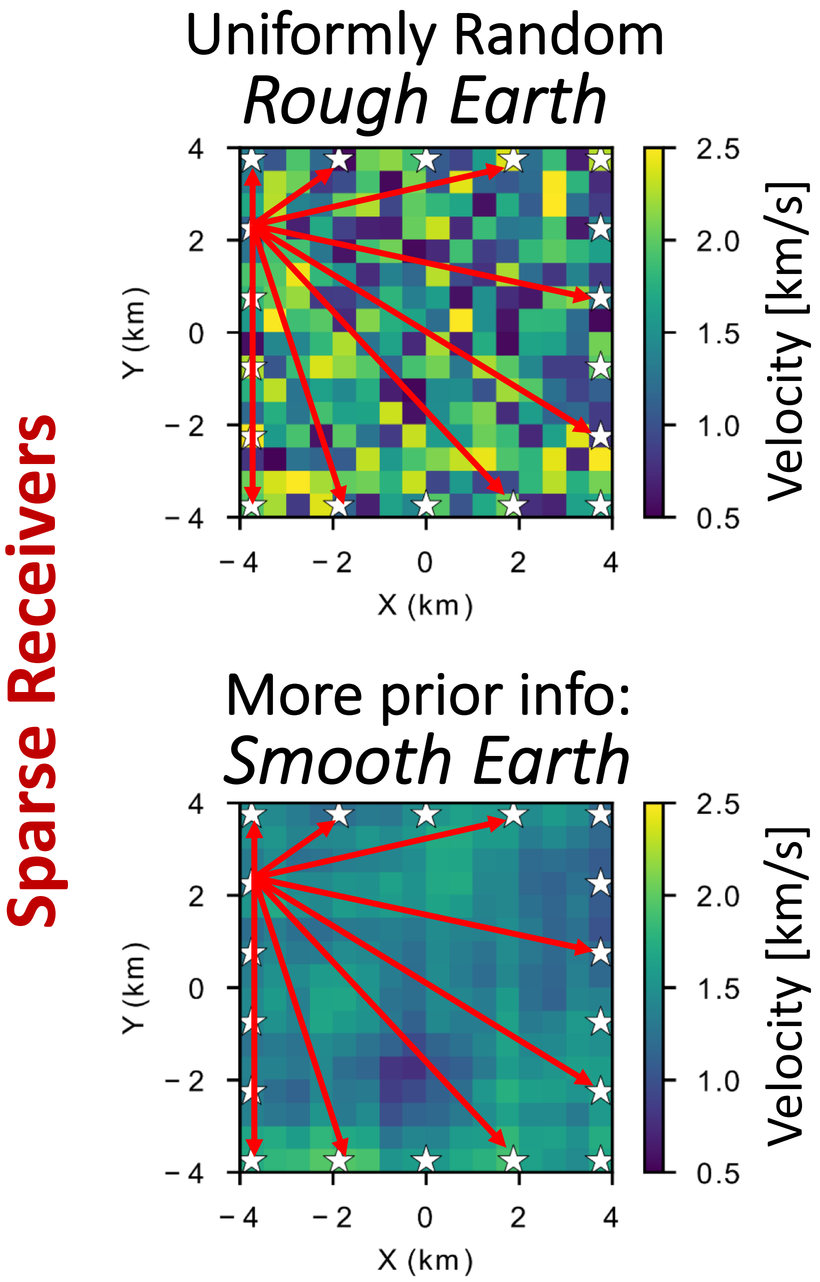

~122m
40m
90m
50m
~111m

Mixture Density Network (MDN)



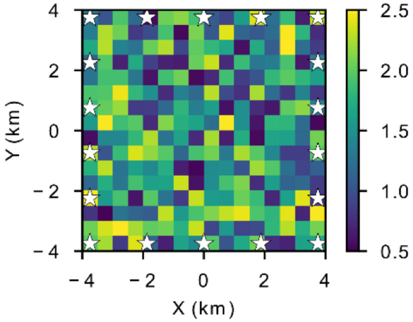
Training Set

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

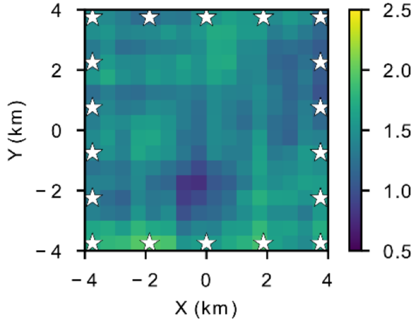


EAGE Abstract + arXiv: Earp & Curtis, 2019

Rough Prior

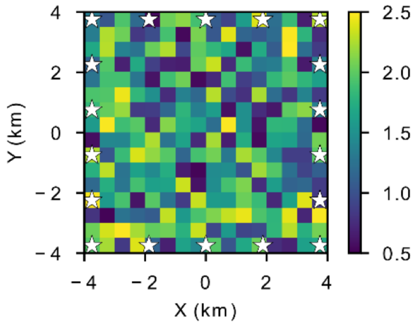


Smooth Prior

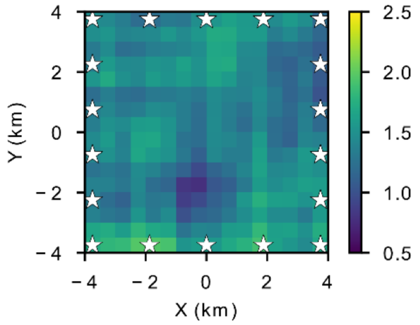


EAGE Abstract + arXiv: Earp & Curtis, 2019

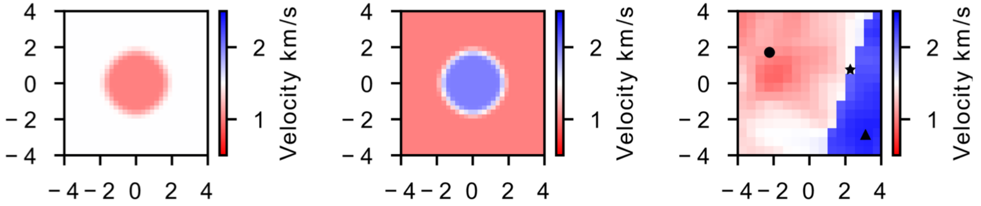
Rough Prior



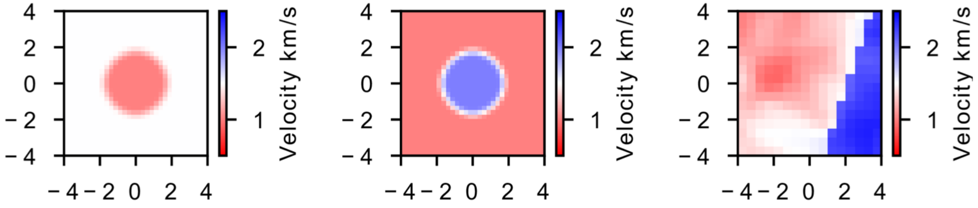
Smooth Prior



True Model

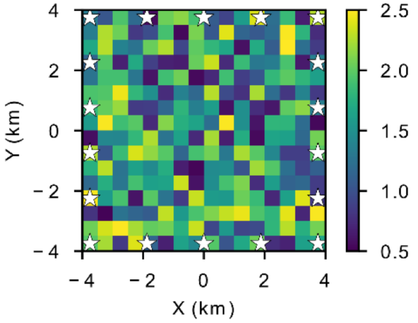


True Model

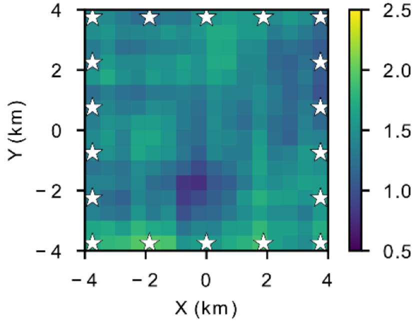


EAGE Abstract + arXiv: Earp & Curtis, 2019

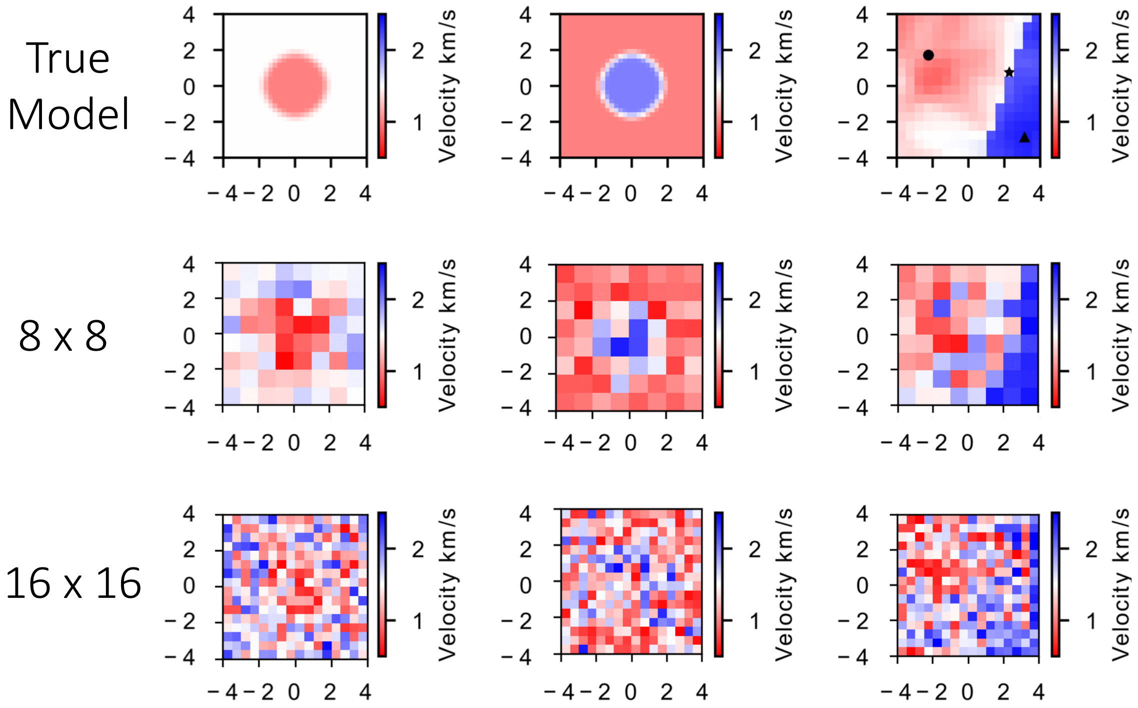
Rough Prior

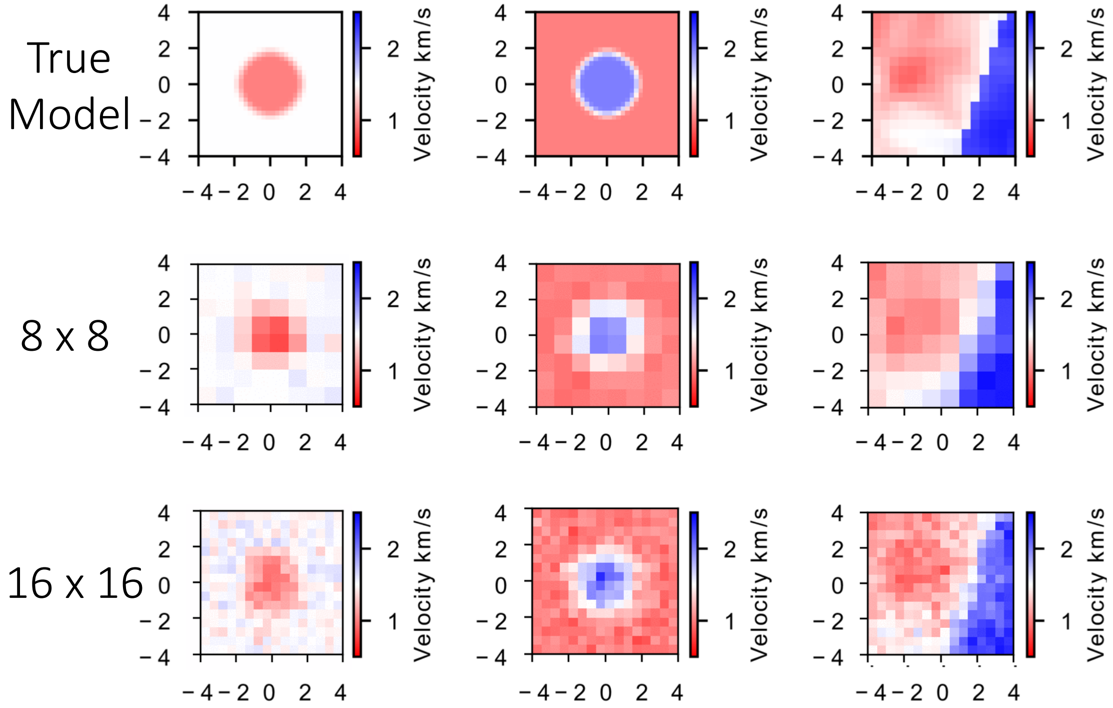


Smooth Prior



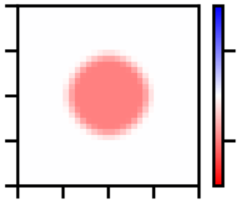
Samples from posterior marginal pdf's



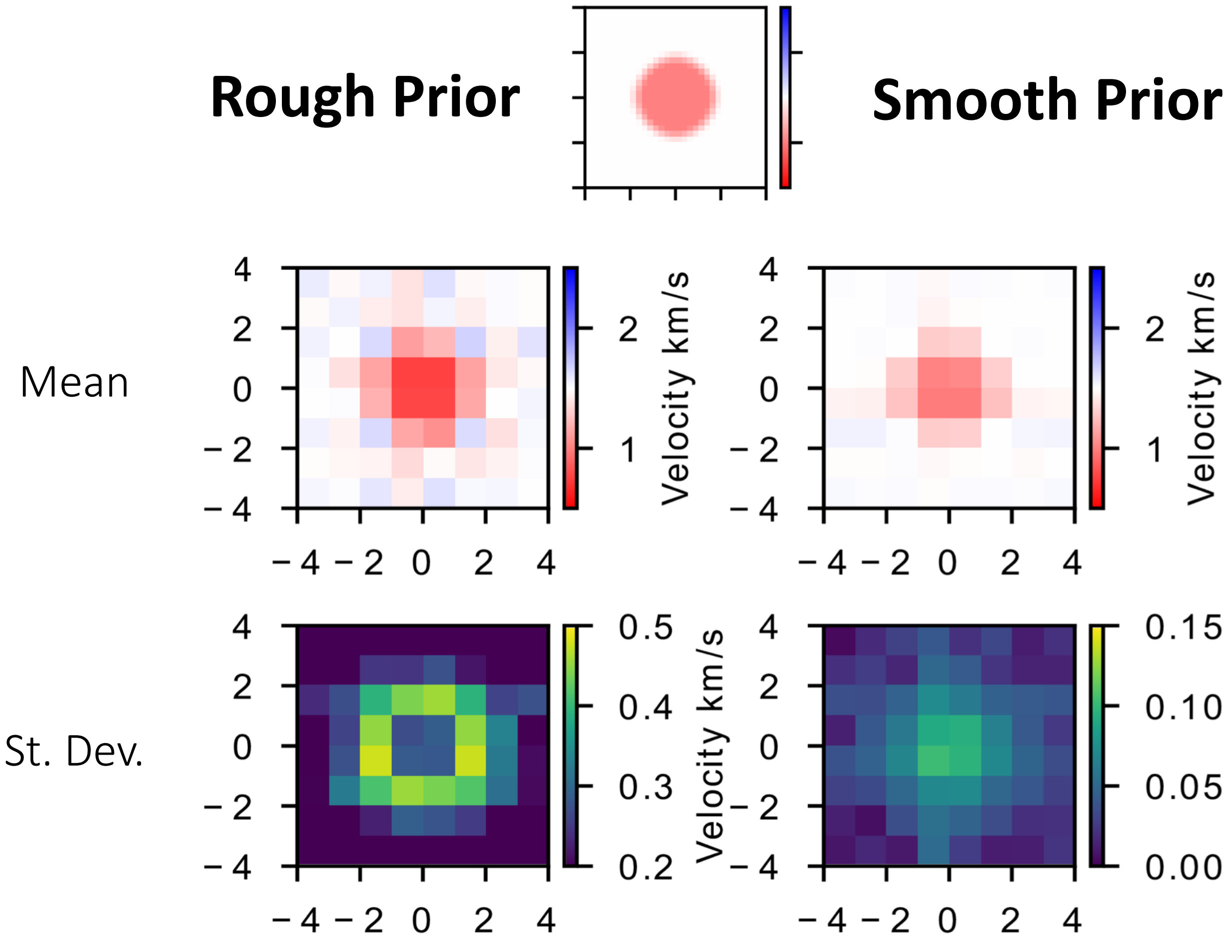


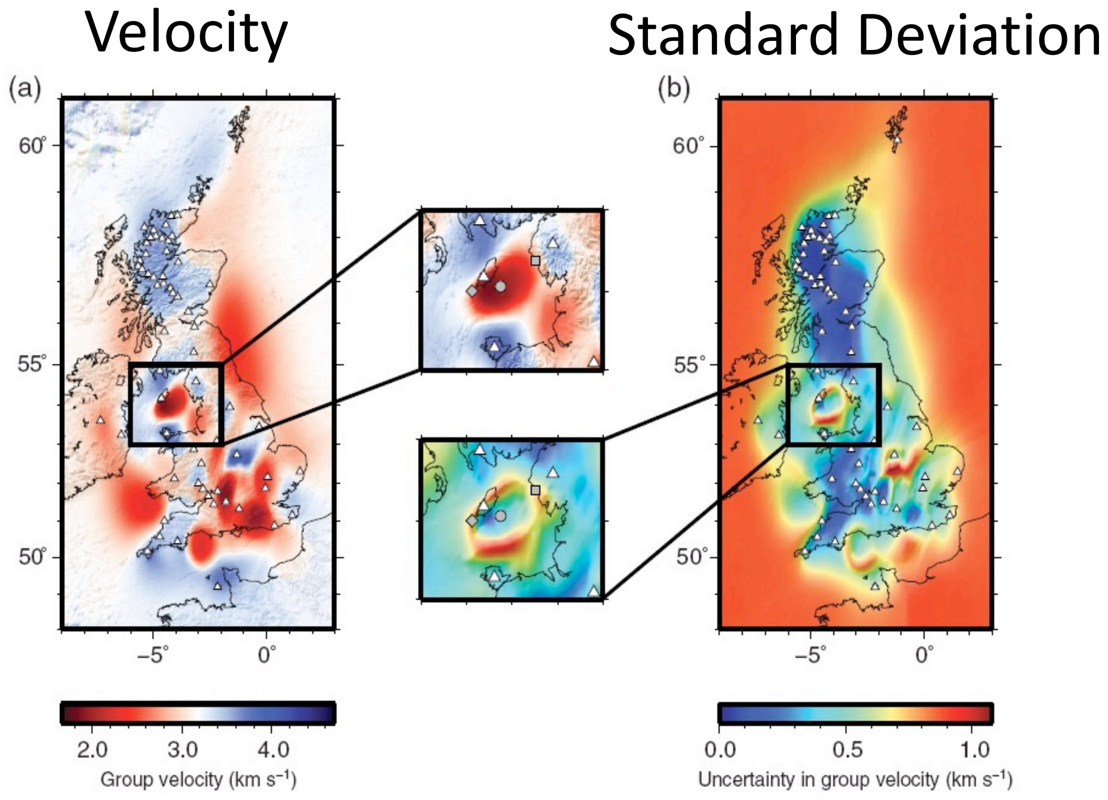
EAGE Abstract + arXiv: Earp & Curtis, 2019

Rough Prior

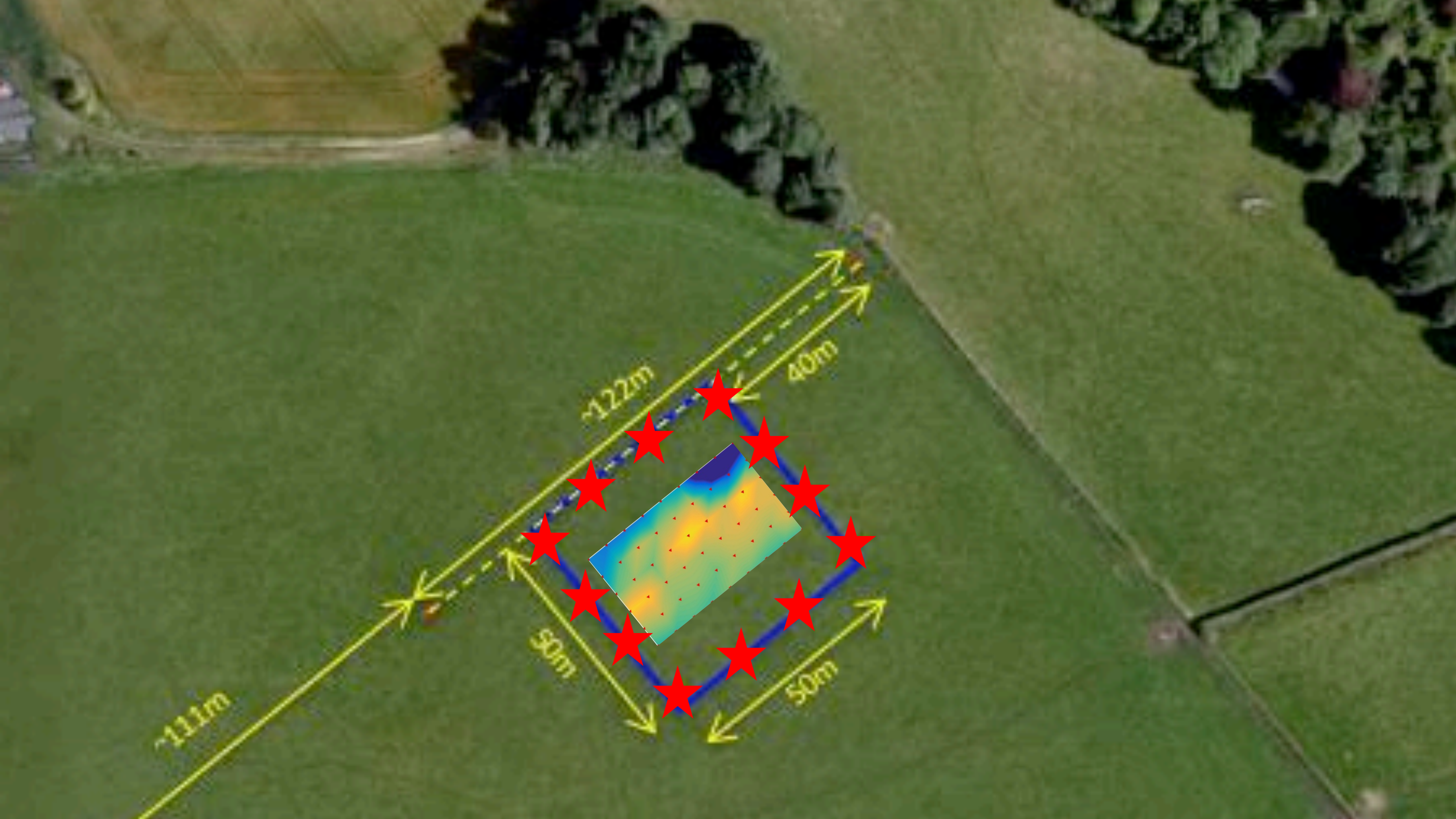


Smooth Prior





Galetti et al. Phys. Rev. Lett. 2015

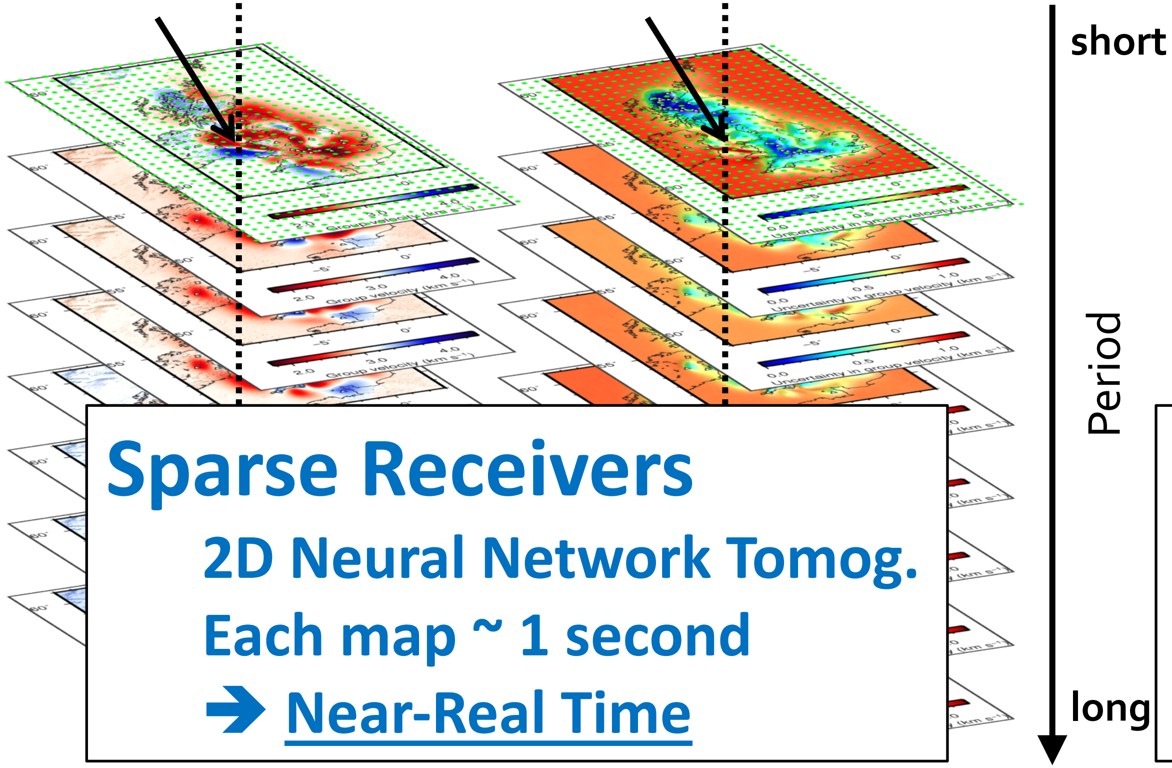

~122m
40m
90m
50m
~111m

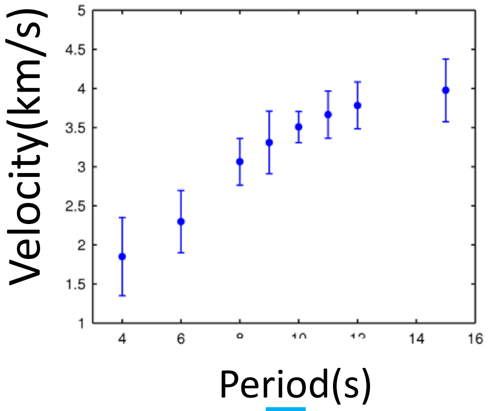
Seismic Surface Wave Tomography : typical workflow

- **Step 1**: construct *m* x 2D phase/group velocity maps
- **Step 2**: 1D depth inversion at each grid point
  Repeat for *n* grid points ➔ 3D model

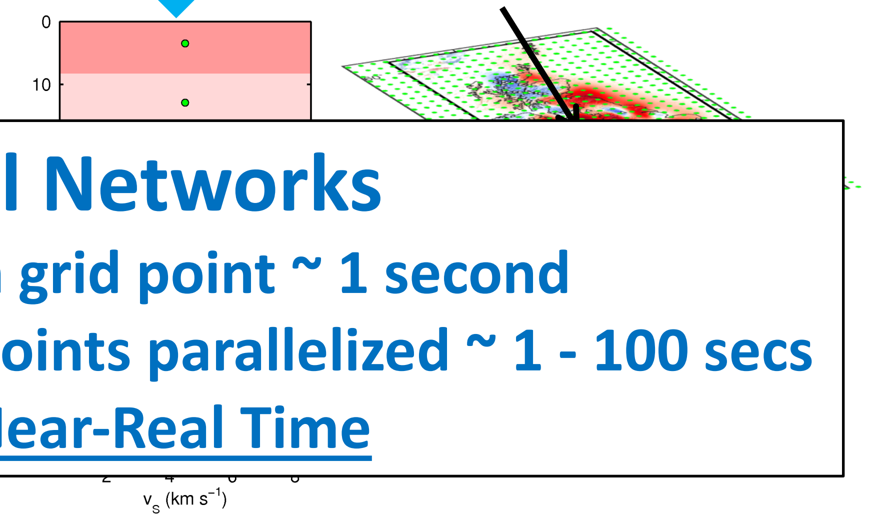
➔ **Decomposition**: 3D tomog = *m* x 2D + *n* x 1D tomog's

Phase Velocity and Uncertainty map





1D depth inversion



Step 1

Step 2

Sparse Receivers

2D Neural Network Tomog.

Each map ~ 1 second

➔ Near-Real Time

Neural Networks

Each grid point ~ 1 second

All points parallelized ~ 1 - 100 secs

➔ Near-Real Time

Near-Real Time 3D Tomography

- *Bayesian Nonlinear Ambient Noise Tomography in near-real time*

- This is possible! (or at least is a reasonable goal)

Real-time applications

➔ Monitoring induced seismicity/deformation

➔ Earthquake/volcano early warning

➔ Subsurface leak/pollution detection

➔ Rolling Arrays

Achieved by **OPTIMISATION**

Near-Real Time 3D Tomography

- *Bayesian Nonlinear Ambient Noise Tomography in near-real time*

- This is possible! (or at least is a reasonable goal)

- Generalised methods – (*hardly investigated in Geophysics*)

Generalised **Optimisation** for Probability Distributions

- **Variational Bayesian Inference**
- Synthetic tests
- Application to Grane array data

arXiv: Zhang & Curtis, 2019

Variational Inference

- Bayesian solution

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d})}$$

$\mathbf{d}$=data, $\mathbf{m}$=parameters

- Variational methods replace **stochastic sampling** with **optimisation** of functions

- Strategy 1: **fit** semi-analytic functions to $p(\mathbf{m}|\mathbf{d})$
  - Choose family of functions $q(\mathbf{m}, \varphi)$, φ=parameters [c.f. φ=Gaussian mean & covar]
  - Optimise φ s.t. $q(\mathbf{m}|\varphi) \cong p(\mathbf{m}|\mathbf{d})$
- Define a measure of difference between q and p; then minimize it.

- Strategy 2: generate a set of **samples** of $p(\mathbf{m}|\mathbf{d})$ by **optimisation**

arXiv: Zhang & Curtis, 2019

Variational Inference

- **Kullback-Liebler divergence** measures difference between q and p:

log(evidence) : intractable

$$KL[q||p] = E_{q(\mathbf{m})}[\log q(\mathbf{m}|\varphi)] - E_{q(\mathbf{m})}[\log p(\mathbf{m}|\mathbf{d})] + \boxed{\log p(\mathbf{d})}$$

- KL ≥ 0 and KL $= 0$ when $q = p$. Rearrange…

log(evidence) : constant w.r.t. q

$$\underline{KL[q||p]} + \underline{E_{q(\mathbf{m})}[\log p(\mathbf{m}|\mathbf{d})] - E_{q(\mathbf{m})}[\log q(\mathbf{m}|\varphi)]} = \boxed{\log p(\mathbf{d})}$$

- Evidence lower bound (ELBO)

Bayes Rule, prior, likelihood
→ Efficient, case-specific analytical methods (Nawaz & Curtis 2018/19)

$$ELBO(q) = E_{\boxed{q(\mathbf{m})}}[\log \boxed{p(\mathbf{m}|\mathbf{d})}] - E_{\boxed{q(\mathbf{m})}}[\log q(\mathbf{m}|\varphi)]$$

Expectations w.r.t. q which we choose

→ Maximise ELBO → minimises KL divergence (difference) between q and p.

Automatic Differential Variational Inference (ADVI)

- Transform constrained parameter $\mathbf{m}$ to real space: $\boldsymbol{\theta} = T(\mathbf{m})$
- In transformed space assume a Gaussian distribution:

$$q(\boldsymbol{\theta}; \varphi) = Normal(\boldsymbol{\theta}|\boldsymbol{\mu}, \mathbf{L}\mathbf{L}^T)$$

- Standardize the normal distribution: $\boldsymbol{\eta} = S_\varphi(\boldsymbol{\theta})$

$$q(\boldsymbol{\eta}) = Normal(\boldsymbol{\eta}|\mathbf{0}, \mathbf{I})$$

- Gradient of ELBO $\mathcal{L}$ can be calculated:

$$\nabla_{\boldsymbol{\mu}}\mathcal{L} = E_{N(\boldsymbol{\eta})}[\nabla_{\mathbf{m}}\log p(\mathbf{d}, \mathbf{m})\nabla_{\boldsymbol{\theta}}T^{-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\log|det\mathbf{J}_{T^{-1}}(\boldsymbol{\theta})|]$$

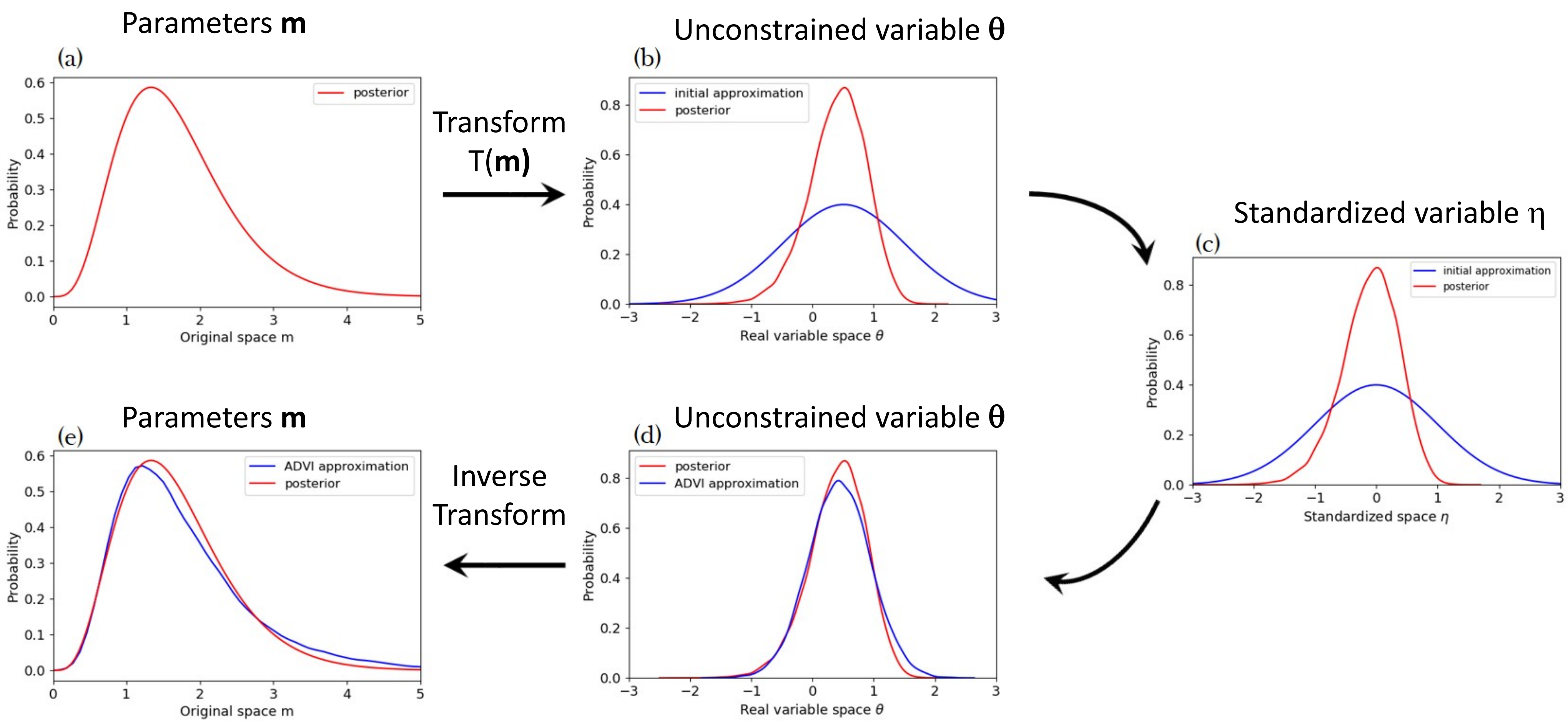
$$\nabla_{\mathbf{L}}\mathcal{L} = E_{N(\boldsymbol{\eta})}[\nabla_{\mathbf{m}}\log p(\mathbf{d}, \mathbf{m})\nabla_{\boldsymbol{\theta}}T^{-1}(\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\log|det\mathbf{J}_{T^{-1}}(\boldsymbol{\theta})|\,\boldsymbol{\eta}^T] + (\mathbf{L}^{-1})^T$$

Expectations calculated by MC. But sampling N(0,1) → low number of samples (even 1)

- Maximize ELBO $\mathcal{L}$ using **gradient ascent**

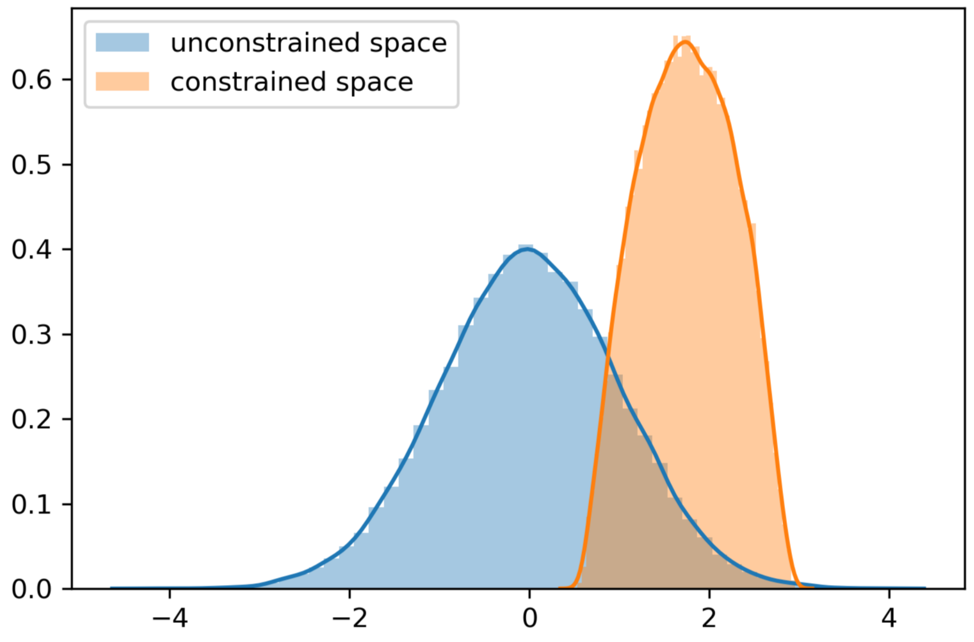
Kucukelbir et al., 2017 JMLR

Automatic Differential Variational Inference (ADVI)



Transform

$\theta_i = \log(m_i - a) - \log(b - m_i)$ where a and b are lower and upper bounds



Variational Inference based on **Invertible Transforms**

- Approximate posterior p using a series of transforms:

$$q_n = T_n \cdots T_1 T_0(q_0)$$

- Optimize transform T_i by minimizing KL divergence.

- *Normalizing flow, Stein variational methods, etc.*

Stein Variational Gradient Descent(SVGD)

- Assume a transformation $T(\mathbf{m}) = \mathbf{m} + \epsilon\phi(\mathbf{m})$

$$\nabla_\epsilon KL[q_{[T]}||p]|_{\epsilon=0} = -E_{\mathbf{m}\sim q}[trace\left(A_p\phi(\mathbf{m})\right)]$$

where $A_p\phi(\mathbf{m}) = \nabla_\mathbf{m}\log p(\mathbf{m}|\mathbf{d})\phi(\mathbf{m})^T + \nabla_\mathbf{m}\phi(\mathbf{m})$

SAMPLES

Standard gradients (as used in linearised inversion)

- ϕ^* that maximize the negative gradient:

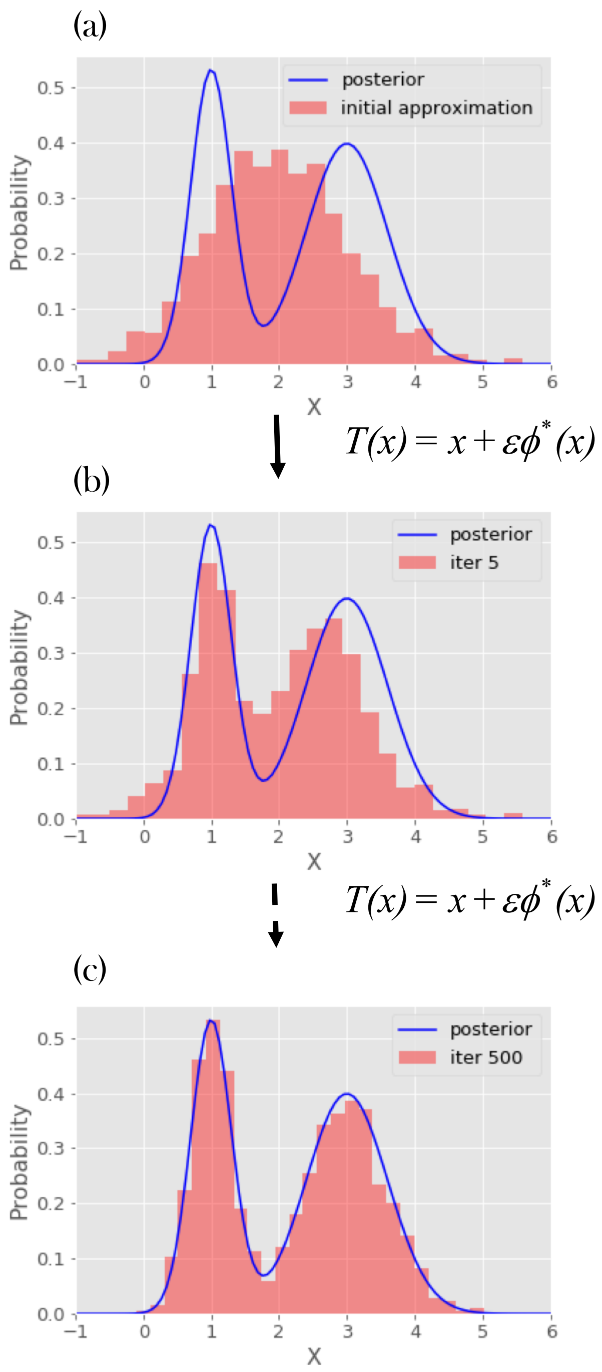
$$\phi^*_{q,p}(\mathbf{m}) = E_{\mathbf{m}'\sim q}[k(\mathbf{m}',\mathbf{m})\nabla_{\mathbf{m}'}\log p(\mathbf{m}'|\mathbf{d}) + \nabla_{\mathbf{m}'}k(\mathbf{m}',\mathbf{m})]$$

where k is a kernel, e.g. a RBF kernel:

$$k(\mathbf{m},\mathbf{m}') = \exp(-\frac{1}{h}\|\mathbf{m}-\mathbf{m}'\|^2)$$

Liu and Wang, 2016 arXiv.

SVGD

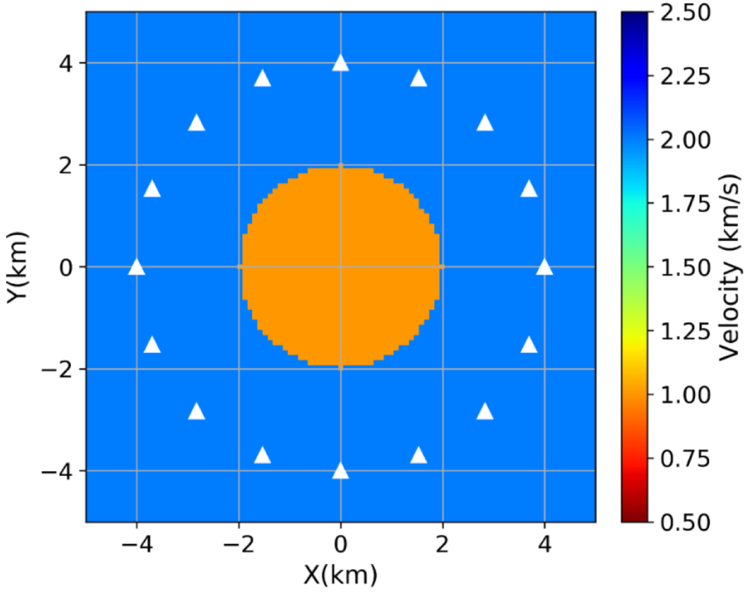


- Target probability
- Histogram of particles

Try this:

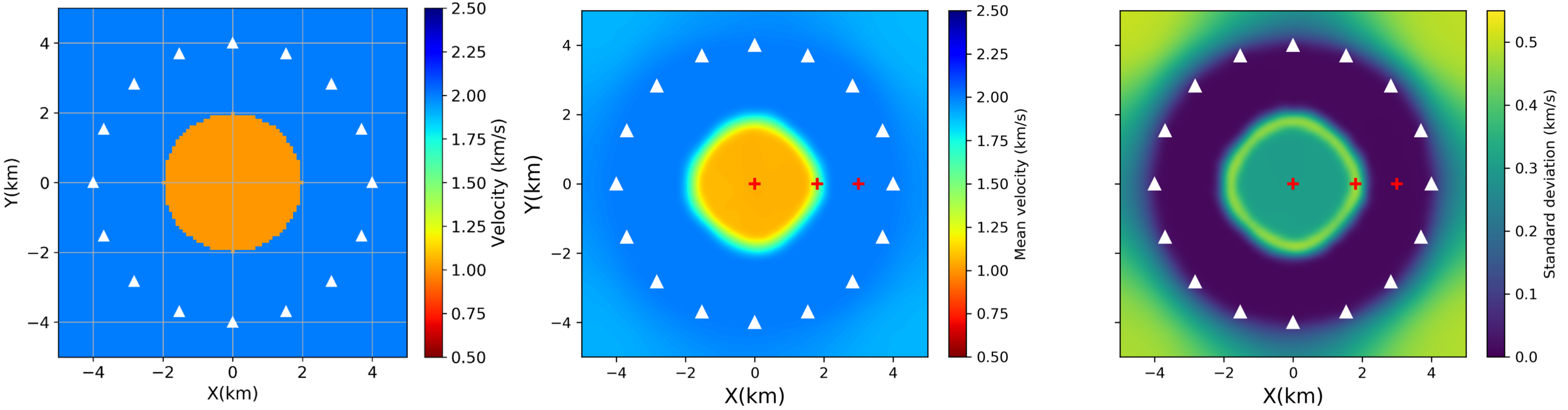
https://chi-feng.github.io/mcmc-demo/app.html

Synthetic tests



Reversible jump McMC

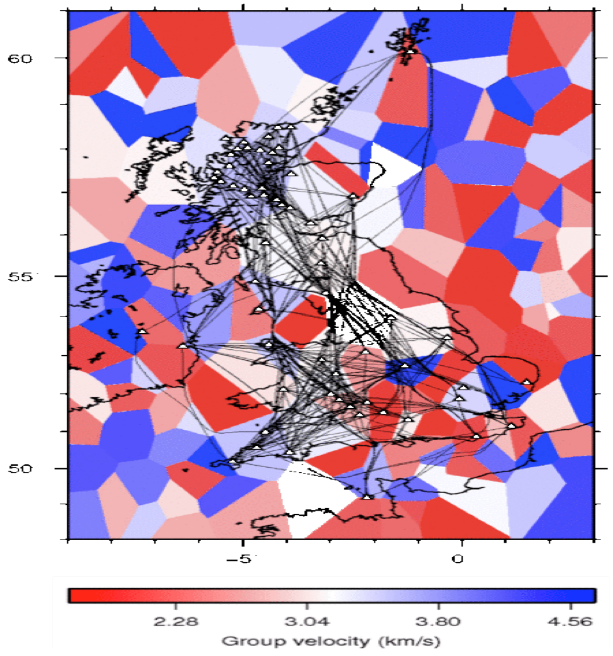
True model | Mean | Stdev



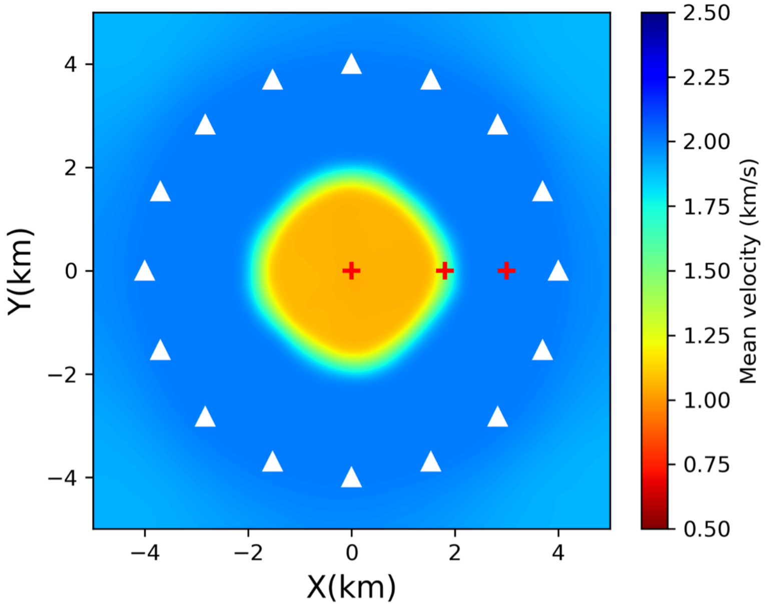
Parameterized by Voronoi cells

Reversible jump McMC

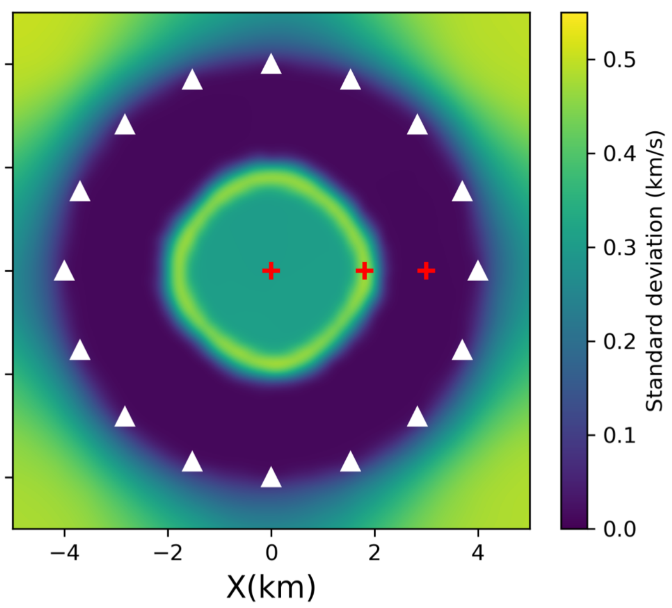
True model



Mean



Stdev



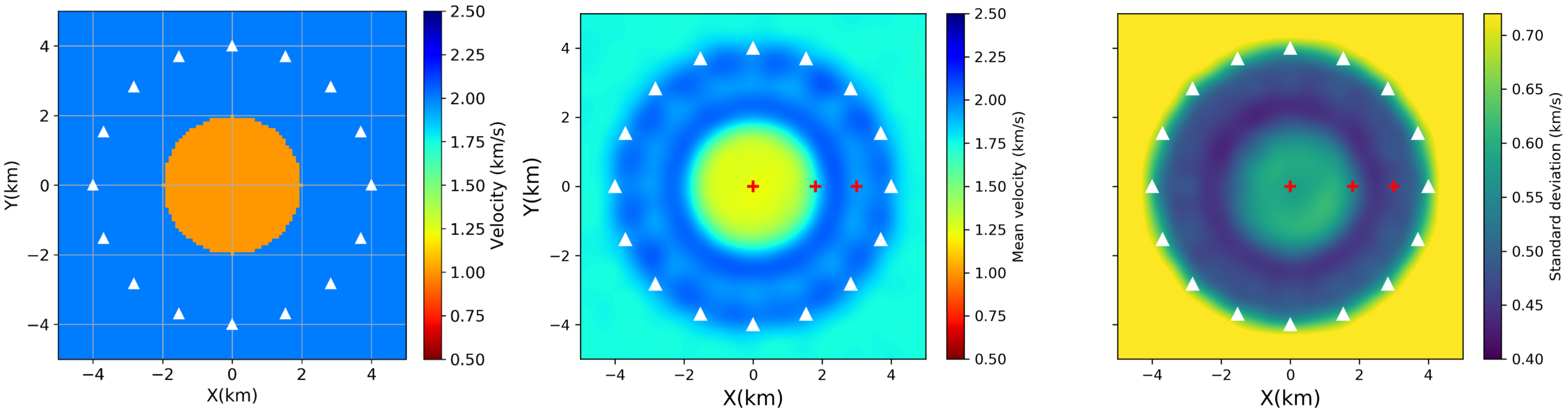
Parameterized by Voronoi cells

ADVI results

True model

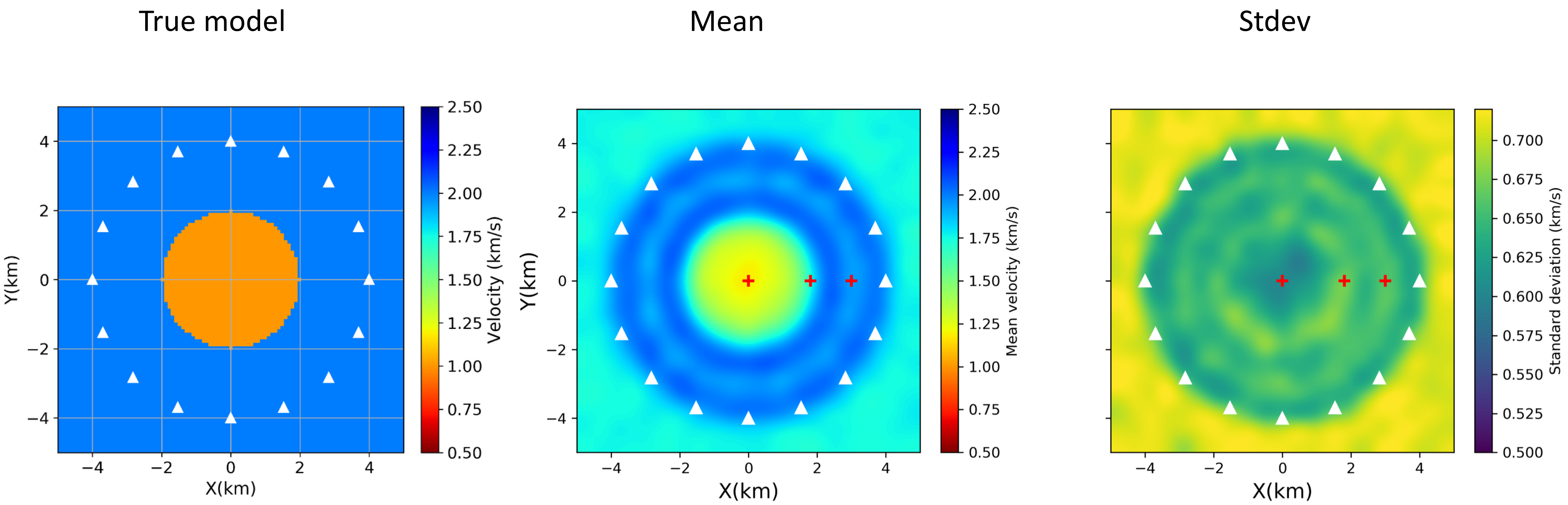
Mean

Stdev



Parameterized by a 21*21 grid

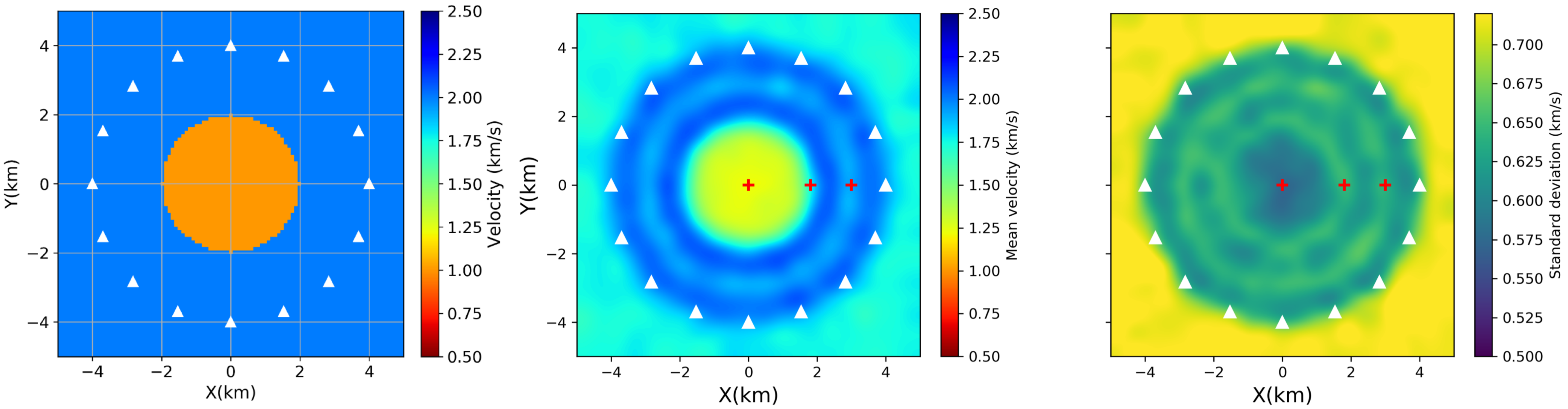
SVGD results



Parameterized by a 21*21 grid

Metropolis-Hastings McMC

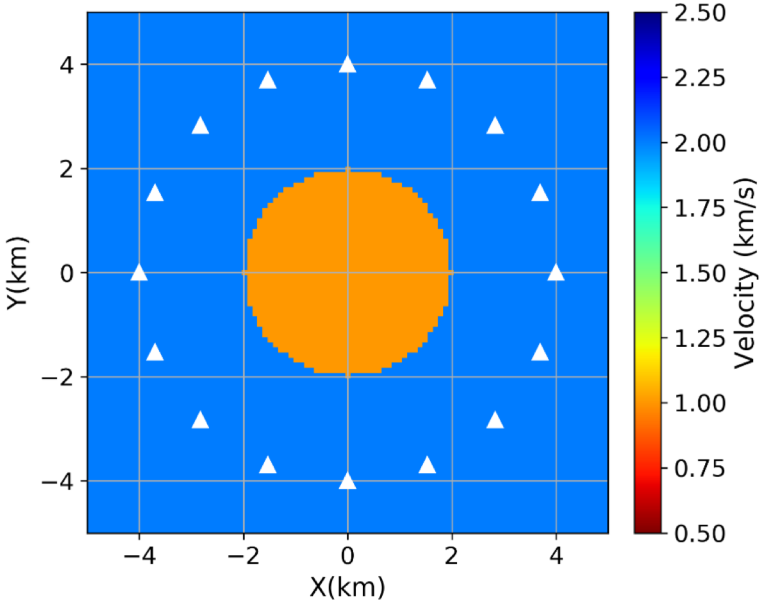
True model Mean Stdev



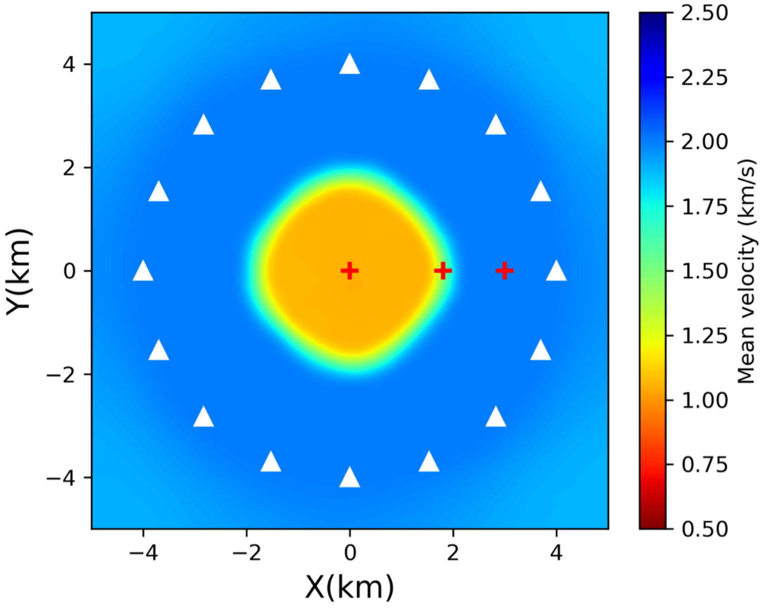
Parameterized by a 21*21 grid

Reversible jump McMC

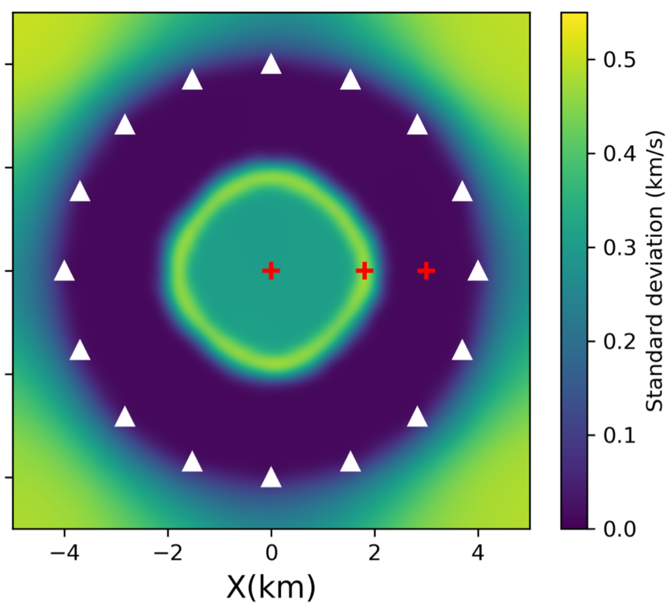
True model



Mean



Stdev



Parameterized by Voronoi cells

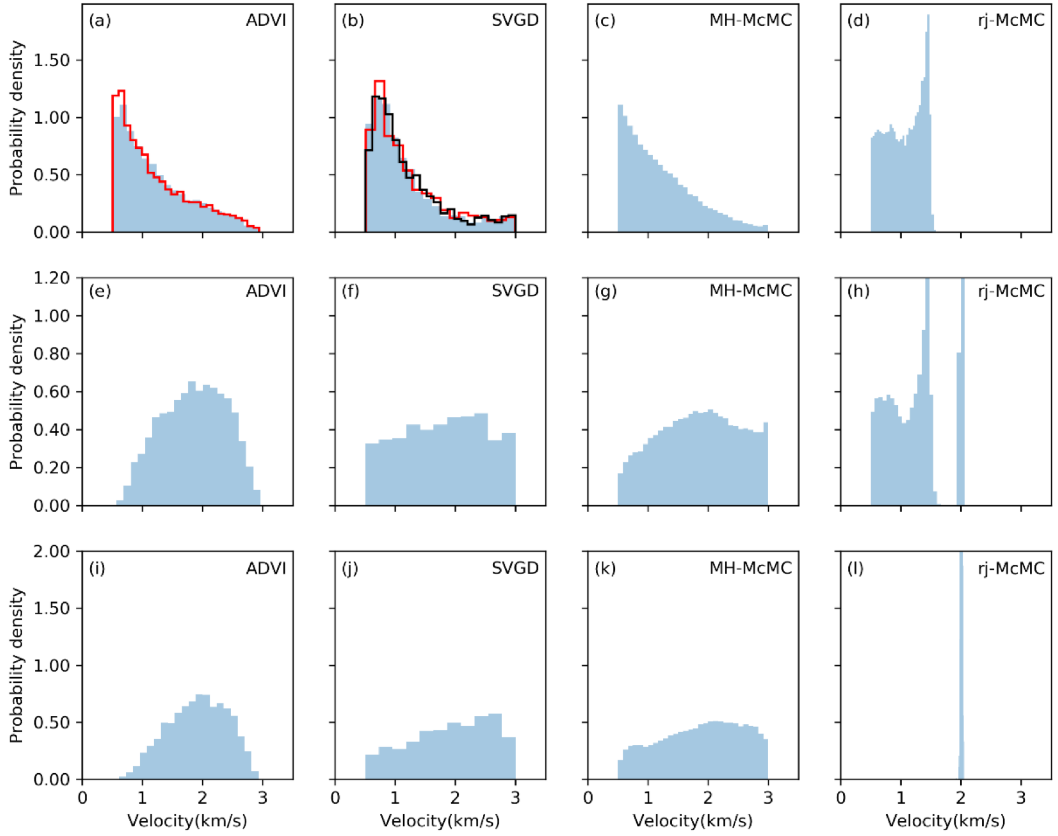
Marginal distribution

ADVI **SVGD** **MH-McMC** **rj-McMC**

Interior
Point (0.0,0)

Boundary
Point (1.8,0)

Exterior
Point (3.0,0)



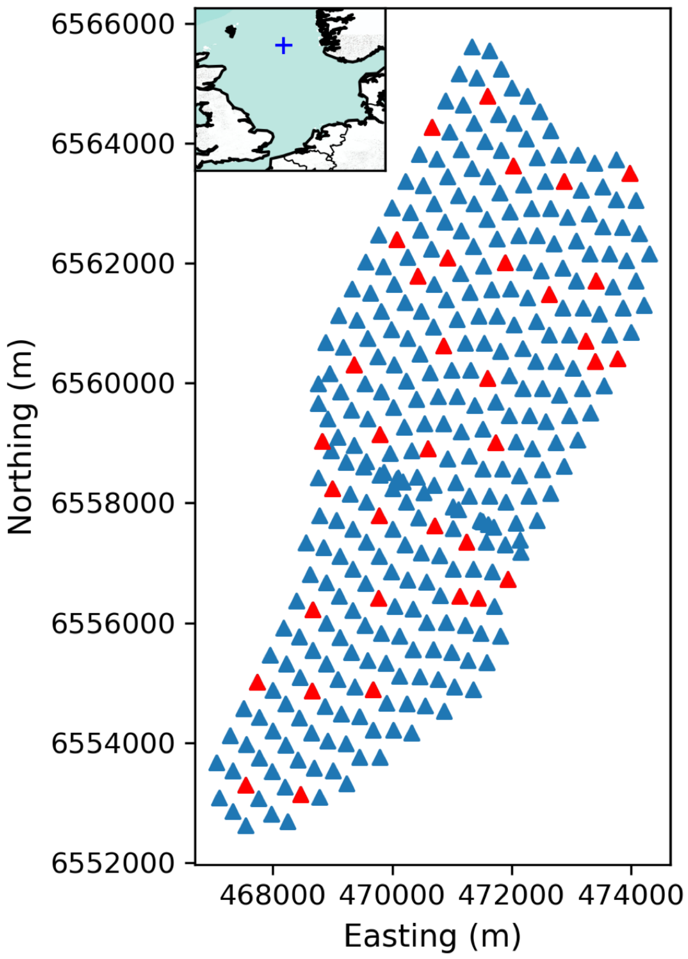
Computational cost

Methods	Number of simulations	CPU hours	Real time (hours)
ADVI	10,000	0.45	0.45
SVGD	400,000	8.53	0.97
MH-McMC	12,000,000	410.3	68.4
Rj-McMC	3,000,000	102.6	17.1

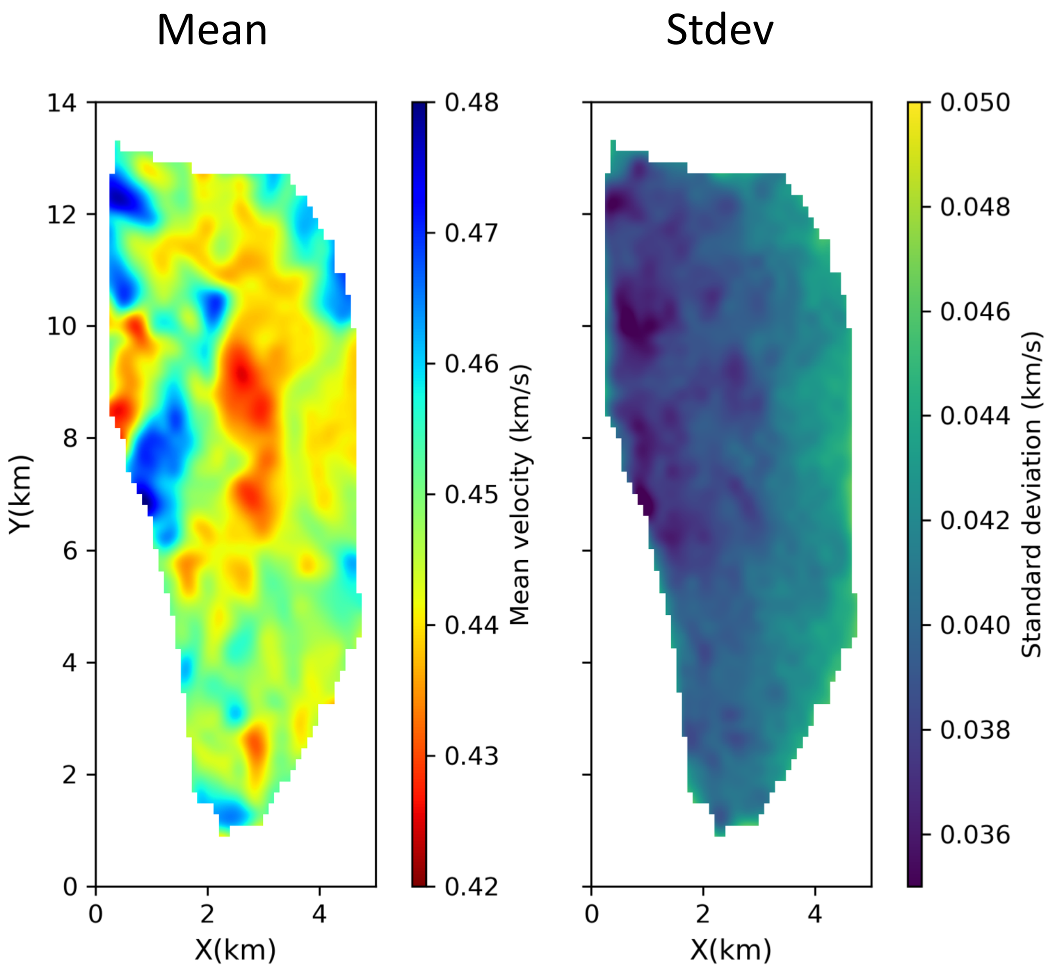
Application to Grane field

346 receivers

35 virtual sources

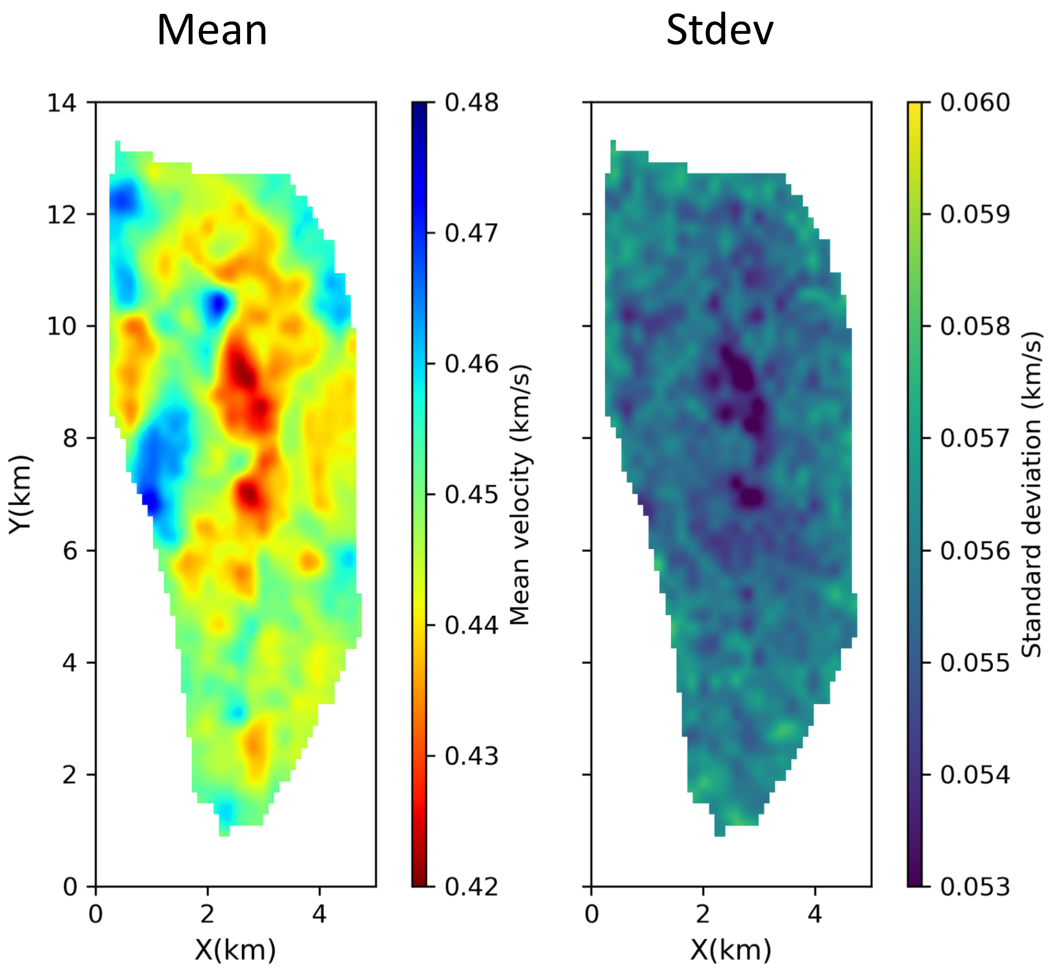


ADVI



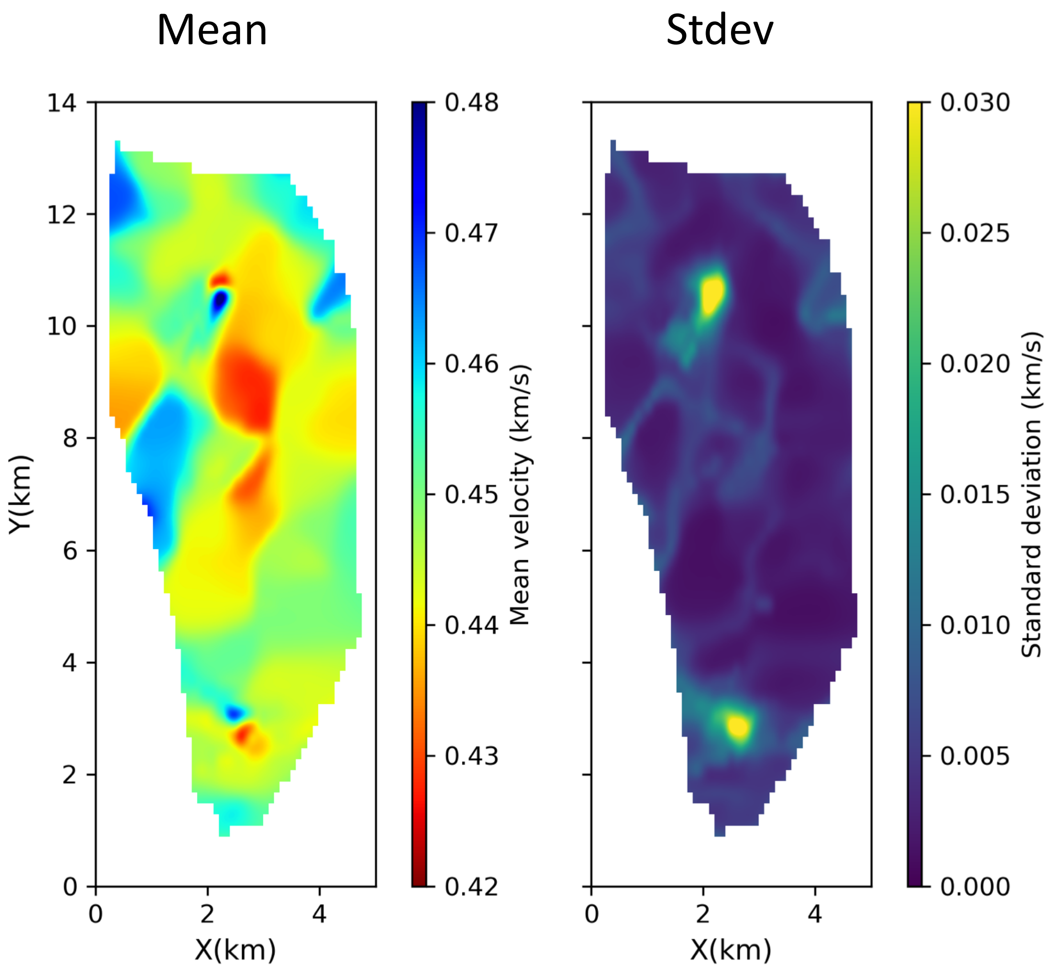
10,000 forward modelling, 5.1 hours

SVGD



500,000 forward modelling, 12.1 hours parallelized using 12 cores

rj-McMC



12,800,000 forward modelling, 5 days running on 16 cores

Summary

- Introduced two variational inference methods to seismic tomography
  - Automatic differential variational inference (ADVI)
  - Stein variational gradient descent (SVGD)

- Compared with Metropolis-Hastings and reversible jump McMC
  - Variational methods provide efficient alternatives to McMC

- Variational methods almost unexplored in Geophysics – **Try them!**

→FOR PAPERS ON ALL TOPICS SEE: www.ed.ac.uk/homes/acurtis

→Or email Andrew.Curtis@ed.ac.uk