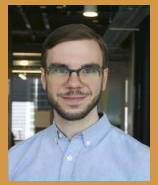
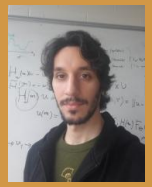
Seismic Imaging with Uncertainty Quantification
Sampling from the Posterior with Generative Networks

Ali Siahkoohi[1], Philipp A. Witte[2], Mathias Louboutin[1], Felix J. Herrmann[1], and **Gabrio Rizzuti**[1]

SLIM

[1] School of Computational Science and Engineering, Georgia Tech
[2] Microsoft Azure

SIAM IS20

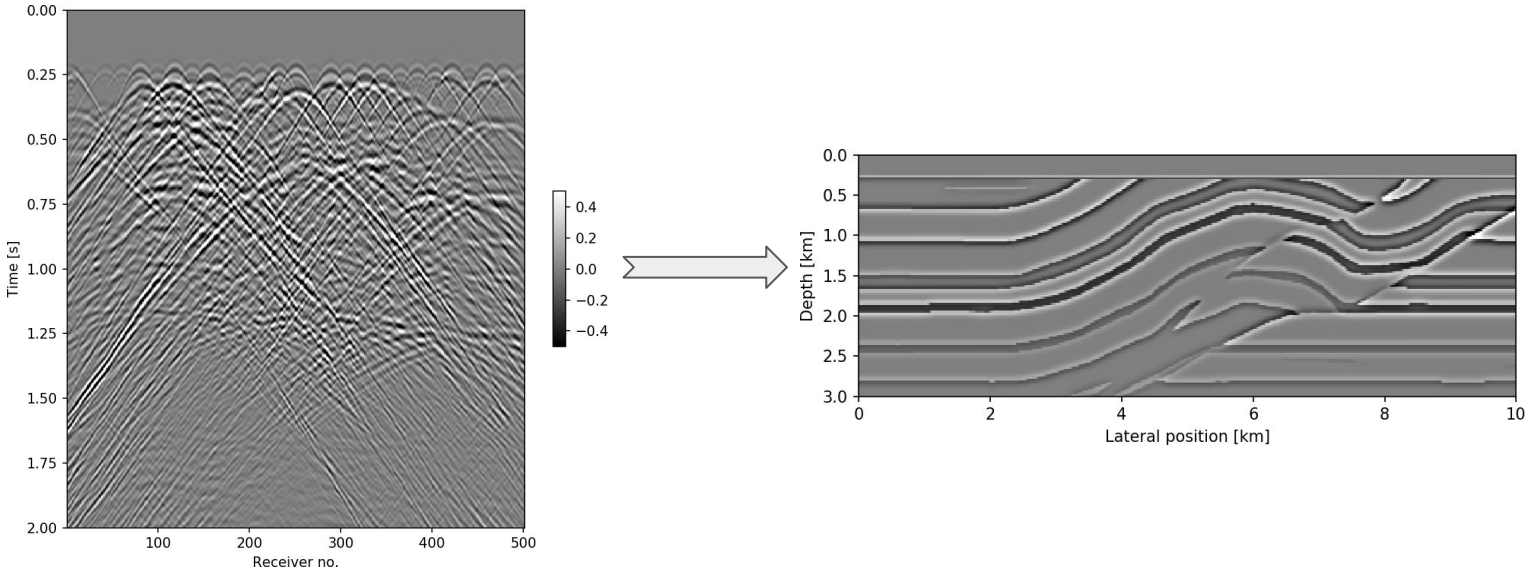
Seismic imaging

Challenges:

- Computational cost (cannot afford 10s of data passes)
- Very large scale (multiple TB of data, ~1 billion unknown parameters)
- State variables: ~**10k** time steps x no. of model parameters (**10 TB** per source location)
- Evaluate anisotropic/elastic forward/adjoint modeling operators
- Mathematical challenges (ill-conditioning, inconsistency, noise, etc.)

Seismic imaging with neural networks?

- data-driven or combined model-data driven ("loop unrolling")?
- implicit regularization ("deep prior")?
- uncertainty quantification?



Agenda

- Speeding up imaging via loop unrolling
- Uncertainty quantification with networks and MCMC (deep priors and task UQ)
- UQ: sampling from the posterior w/ invertible networks

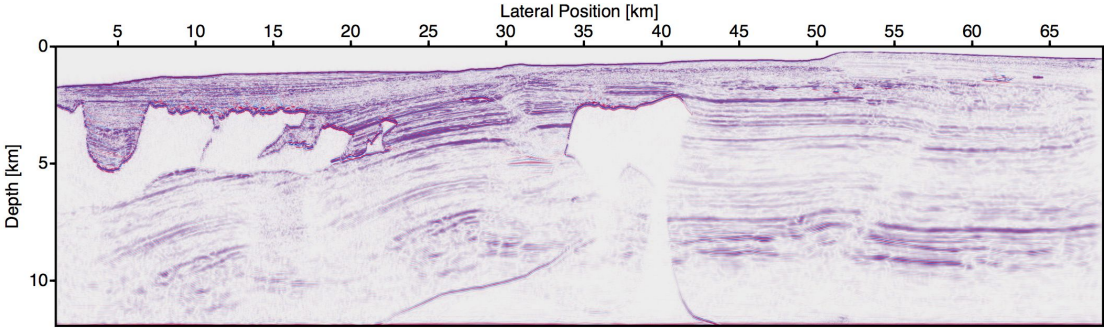
Least-squares imaging & loop unrolling

Philipp A. Witte, Mathias Louboutin, Fabio Luporini, Gerard J. Gorman, and Felix J. Herrmann, Compressive least-squares migration with on-the-fly Fourier transforms, GEOPHYSICS 2019 84:5, R655-R672

Sparsity-promoting least-squares RTM

$$\underset{\delta\mathbf{m}}{\text{minimize}} \quad \lambda \left\|\mathbf{C}\,\delta\mathbf{m}\right\|_1 + \frac{1}{2}\left\|\mathbf{C}\,\delta\mathbf{m}\right\|_2^2$$

$$\text{subject to:} \quad \sum_{i=1}^{n_s}\sum_{j=1}^{n_f}\left\|\mathbf{M}_l^{-1}\mathbf{J}\mathbf{M}_r^{-1}\,\delta\mathbf{m} - \mathbf{M}_l^{-1}\bar{\mathbf{d}}_{ij}\right\| \leq \sigma$$



Philipp A. Witte, Mathias Louboutin, Fabio Luporini, Gerard J. Gorman, and Felix J. Herrmann, Compressive least-squares migration with on-the-fly Fourier transforms, GEOPHYSICS 2019 84:5, R655-R672
Luporini, Fabio, et al. "Architecture and performance of Devito, a system for automated stencil computation." *arXiv preprint arXiv:1807.03032* (2018).
Louboutin, Mathias, et al. "Devito: an embedded domain-specific language for finite differences and geophysical exploration." *arXiv preprint arXiv:1808.01995* (2018).

Least-squares imaging & loop unrolling

Strategies:

- Sparsity-promoting LS-RTM + on-the-fly Fourier transforms
- Save cost by working with subsets of random shots/frequencies (can afford 1 or 2 data passes)
- Automatic code generation using Devito
- Data + model driven imaging via **loop unrolling**

Loop unrolling (example w/ gradient descent)

```
function Λ_θ(J, d)
    x_0 = 0, s = 0
    for j = 1, ..., n
        g = J^⊤ (J x_{j-1} − d)
        u = [x_{j-1}, g, s]
        u = ReLU(u * w_1 + b_1)
        u = ReLU(u * w_2 + b_2)
        u = u * w_3 + b_3
        s = ReLU(u[:, 1:end-1, :, :])
        Δx = u[:, end, :, :]
        x_j = x_{j-1} − Δx
    end
    return x_n
end
```

gradient computation

gradient processing via nn

model update

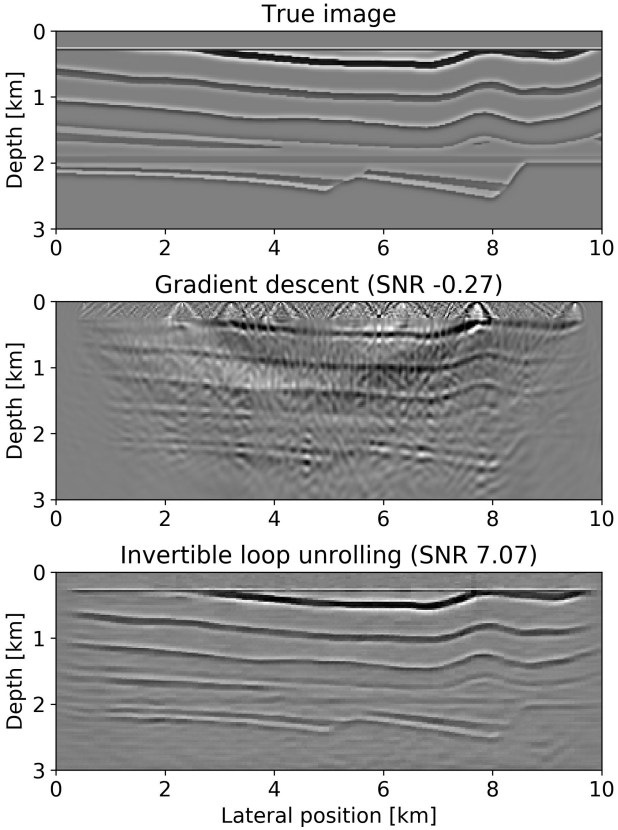
Loop unrolled gradient descent

Loop unrolling w/ invertible networks

Invertible version of loop-unrolled network:

- Loop unrolling: memory scales w/ network depth
- Instead: architecture based on invertible nets

```
function G(J, d)
    x = 0
    for j = 1, ..., n
        x = Qx
        x'_1 = x_1
        g = J^T (J x'_1[1] − d)
        s', t = NN([g, x'_1[2:end]])
        s = σ(s')
        x'_2 = x_2 ⊙ s + t
        x = Q^T x'
    end
    return x
end
```



Discussion and conclusions

- Loop unrolling has the potential to speed up imaging, when data and ground truth are jointly available
- What about UQ?

Seismic imaging w/ deep priors and uncertainty quantification with tasks (automatic horizon tracking)

- Ali Siahkoohi, Gabrio Rizzuti, and Felix J. Herrmann. "A deep-learning based Bayesian approach to seismic imaging and uncertainty quantification." arXiv preprint arXiv:2001.04567 (2020).
- Ali Siahkoohi, Gabrio Rizzuti, and Felix J. Herrmann. "Uncertainty quantification in imaging and automatic horizon tracking: a Bayesian deep-prior based approach." arXiv preprint arXiv:2004.00227 (2020).

Why uncertainty quantification?

Uncertainties in **seismic imaging**:

- Noisy/partial observations
- Ill-conditioning
- Confidence in prior model

Goals:

- Determine how likely a certain result is
- Analyze results “equally” likely
- How the uncertainty reflects on subsequent **tasks**? (e.g. horizon tracking)

J. Besag (1994). "Comments on "Representations of knowledge in complex systems" by U. Grenander and MI Miller". *Journal of the Royal Statistical Society, Series B*. **56**: 591–592

Ingredient 1/2: MCMC methods

E.g., Metropolis-adjusted Langevin algorithm

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \frac{\varepsilon_n}{2}\nabla_{\mathbf{m}_n}\log\pi + \eta_n, \quad \eta_n \sim N(0, \varepsilon_n I)$$

$$\alpha = \min(1, \frac{\pi(\mathbf{m}_{n+1})q(\mathbf{m}_n|\mathbf{m}_{n+1})}{\pi(\mathbf{m}_n)q(\mathbf{m}_{n+1}|\mathbf{m}_n)})$$

$$\alpha \geq u \sim U[0,1]? \quad \text{Accept } \mathbf{m}_{n+1}$$

π : target distribution

q : transition density

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky (2018). "Deep image prior." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
J. Kruse et al. (2019). “HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference”. https://arxiv.org/abs/1905.10687

Ingredient 2/2: Generative networks

$$G : \mathcal{Z} \to \mathcal{M}$$

$$\mathbf{z} \sim N(0, I), \quad \mathbf{m} = G(\mathbf{z}) \sim p_{\text{post}}(\mathbf{m}|\mathbf{d})$$

Potential advantages:

- Inductive bias (implicit **“deep” prior**)
- Problem-dependent bias via **loop-unrolling**
- Encode distribution via **push-forward** action

$\mathcal{Z}$: latent space

$\mathcal{M}$: model space

Ulyanov, Dmitry, Andrea Vedaldi, and Victor Lempitsky. "Deep image prior." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

Seismic imaging w/ deep priors

- Implicit **regularization** via network

$$
\begin{aligned}
\min_\theta \quad & -\log p_{\text{post}}(\theta|\delta\mathbf{d}) \\
& = -\log p_{\text{like}}(\delta\mathbf{d}|\theta) - \log p_{\text{prior}}(\theta) + \dots \\
& = \frac{1}{2\sigma^2}\sum_{i=1}^{N_s} ||\delta\mathbf{d}_i - J(\mathbf{m}_0, \mathbf{q}_i)G_\theta(\mathbf{z})||^2 + \frac{\lambda^2}{2}||\theta||^2 + \dots
\end{aligned}
$$

fixed background fixed input

$\delta\mathbf{d}$	data residual
$J(\mathbf{m}, \mathbf{q})$	linearized forward operator
$G_\theta : \mathcal{Z} \to \mathcal{M}$	neural network
λ	weighting

Welling, Max, and Yee W. Teh. "Bayesian learning via stochastic gradient Langevin dynamics." Proceedings of the 28th international conference on machine learning (ICML-11). 2011.
Li, Chunyuan, et al. "Preconditioned stochastic gradient Langevin dynamics for deep neural networks." Thirtieth AAAI Conference on Artificial Intelligence. 2016.

Sampling the posterior distribution θ

- Preconditioned **Stochastic Gradient Langevin Dynamics**

$$\theta_{k+1} = \theta_k - \frac{\varepsilon}{2}\mathbf{M}_k \nabla_{\theta_k} L + \eta_k, \quad \eta_k \sim N(0, \varepsilon \mathbf{M}_k)$$

$$L = \frac{N_s}{n} \sum_{i=1}^{n} \frac{1}{2\sigma^2} ||\delta \mathbf{d}_i - J(\mathbf{m}_0, \mathbf{q}_i) G_\theta(\mathbf{z})||^2 + \frac{\lambda^2}{2} ||\theta||^2$$

stochastic approximation

$\mathbf{M}_k$ adaptive diagonal preconditioning

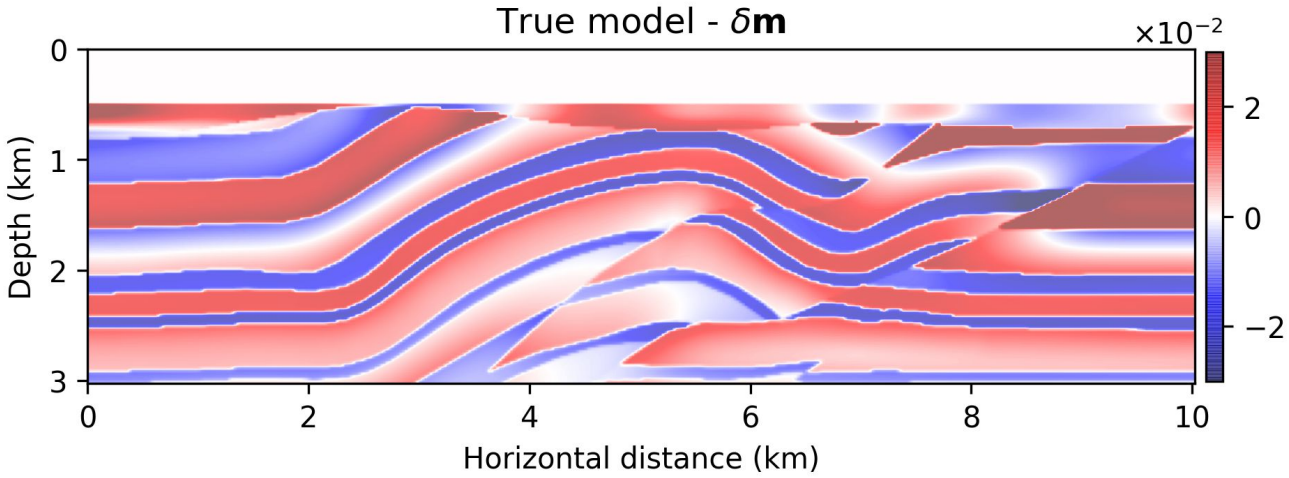
ε steplength

Conditional estimators: from θ to m

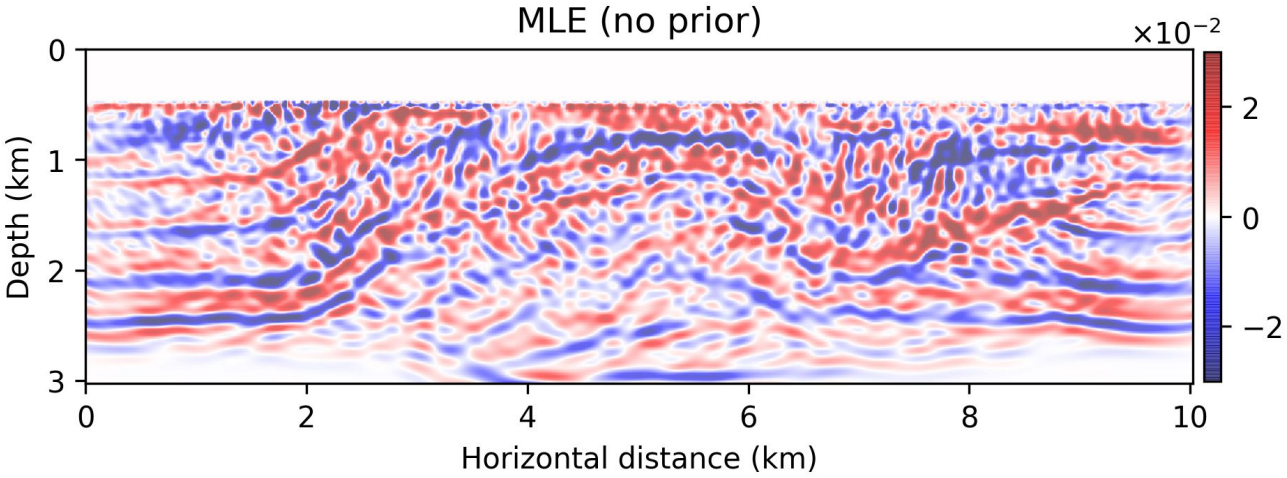
$$\mathbb{E}(\theta|\delta\mathbf{d}) = \int p_{\text{post}}(\theta|\delta\mathbf{d})\,\theta\,\mathrm{d}\theta \;\rightarrow\; \mathbb{E}_{(G)}(\delta\mathbf{m}|\delta\mathbf{d}) = \int p_{\text{post}}(\theta|\delta\mathbf{d})\,G_\theta(\mathbf{z})\,\mathrm{d}\theta$$

- Conditional mean $$\mathbb{E}_{(G)}(\delta\mathbf{m}|\delta\mathbf{d}) \approx \delta\hat{\mathbf{m}} = \frac{1}{n}\sum_{i=1}^{n} G_{\theta_i}(\mathbf{z})$$
- Conditional pointwise variance $$\mathbb{E}_{(G)}((\delta\mathbf{m} - \delta\hat{\mathbf{m}})^{\cdot 2}|\delta\mathbf{d}) \approx \frac{1}{n}\sum_{i=1}^{n}(G_{\theta_i}(\mathbf{z}) - \delta\hat{\mathbf{m}})^{\cdot 2}$$

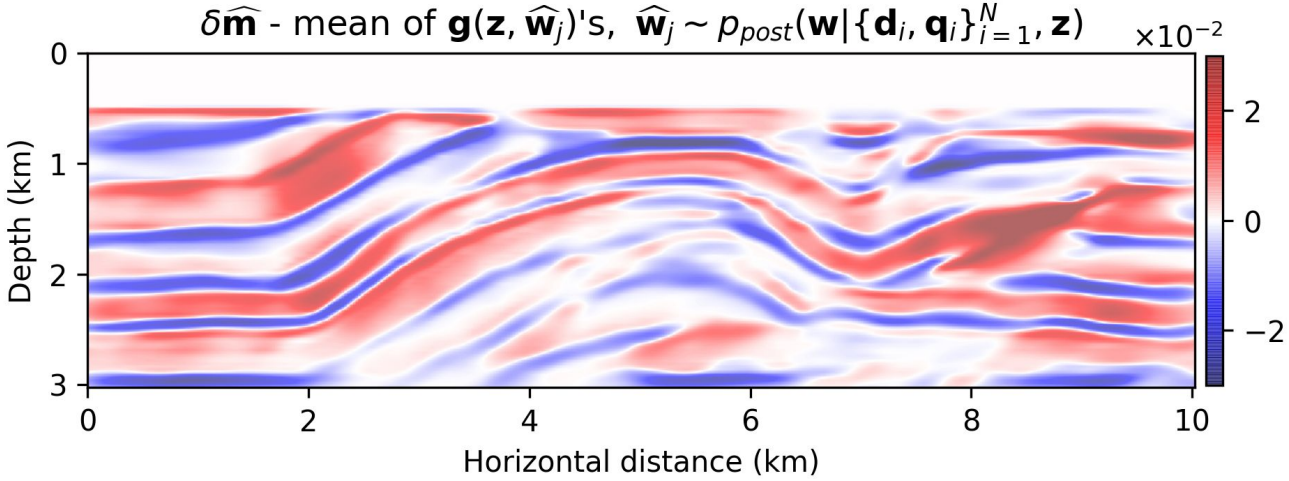
True model: Overthrust model



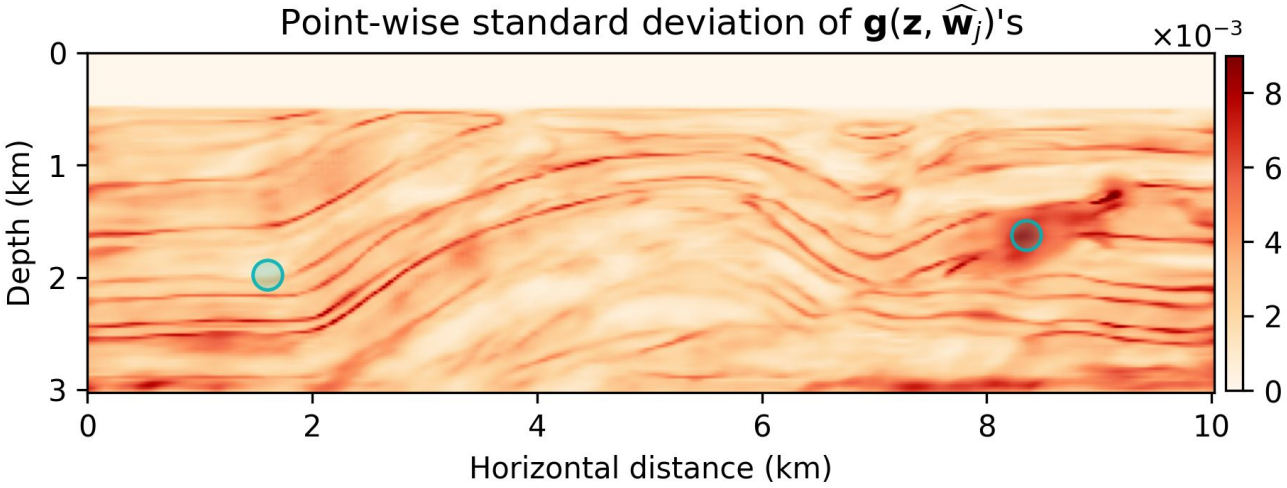
Using no prior



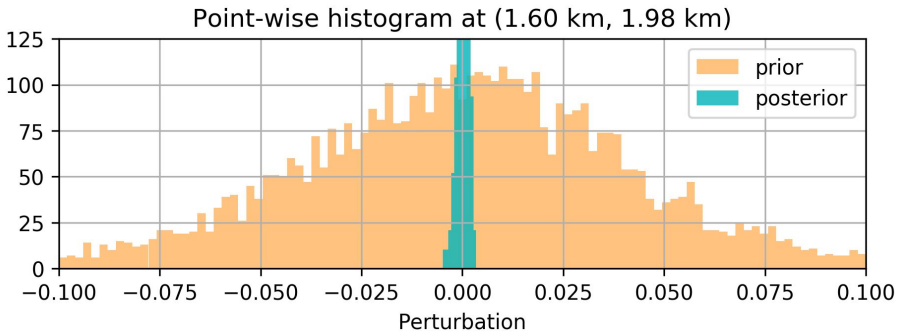
Conditional mean

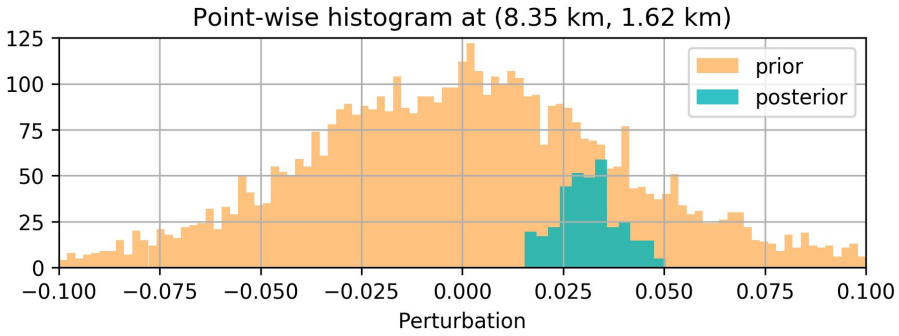


UQ: conditional pointwise standard deviation



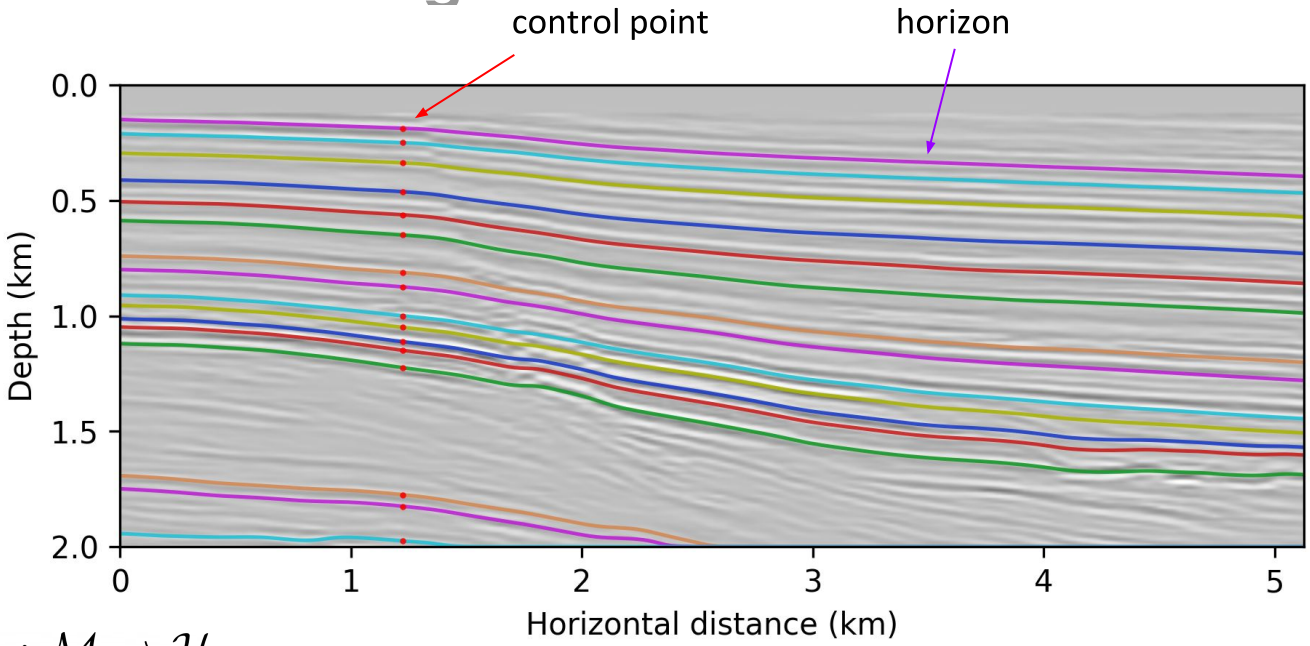
UQ: pointwise histograms





Wu, Xinming, and Sergey Fomel. "Least-squares horizons with local slopes and multigrid correlations." Geophysics 83.4 (2018): IM29-IM40.

Horizon tracking



$$H_{\text{track}} : \mathcal{M} \rightarrow \mathcal{H}$$

Pushing model uncertainty to horizons

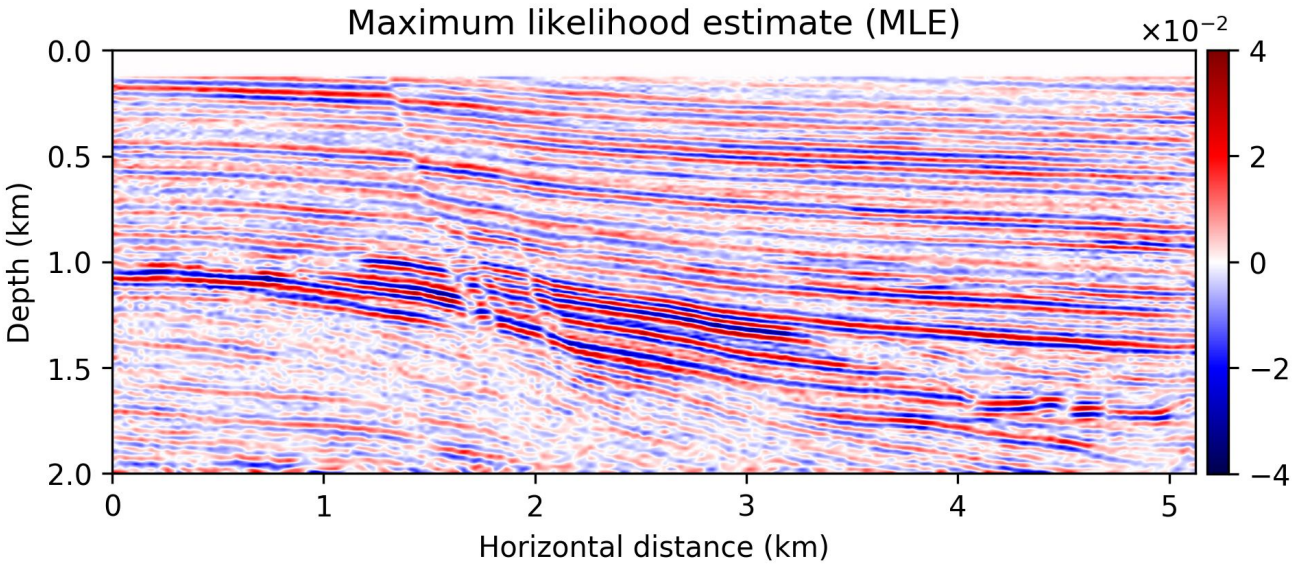
Neural net parameters | Physical models | Horizons

$$\theta \sim p_{\text{post}}(\theta|\delta\mathbf{d}) \quad \rightarrow \quad \delta\mathbf{m} \sim \tilde{p}_{\text{post}}(\delta\mathbf{m}|\delta\mathbf{d}) \quad \rightarrow \quad \mathbf{H} \sim \tilde{p}_{\text{post}}(\mathbf{H}|\delta\mathbf{d})$$

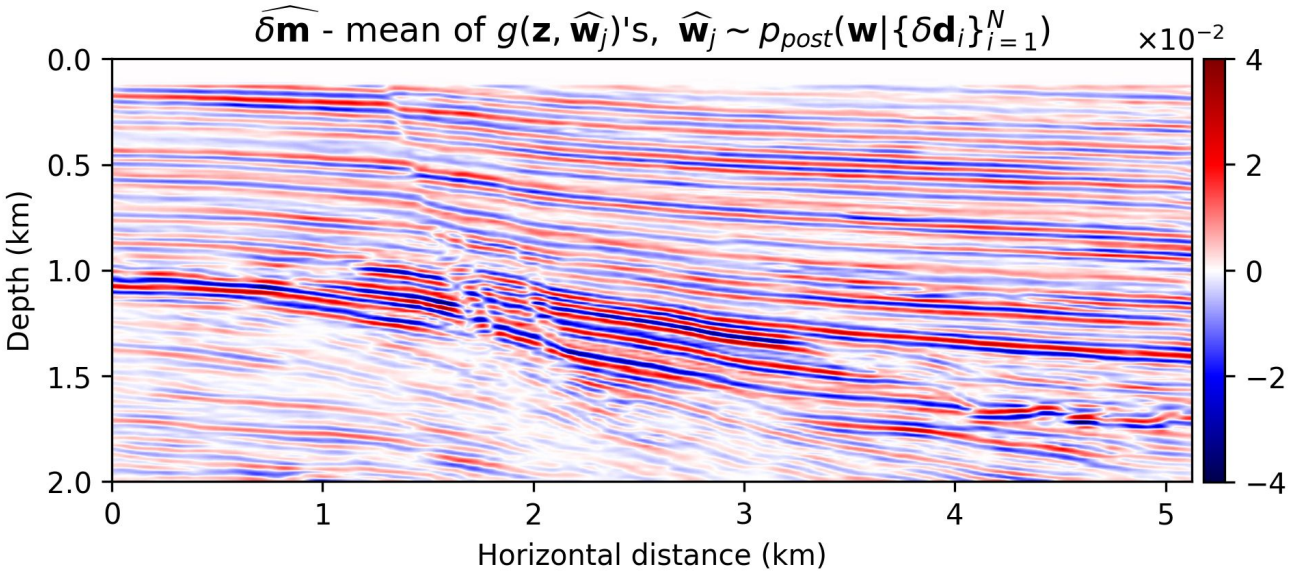
$$G_{\cdot}(\mathbf{z}) : \Theta \rightarrow \mathcal{M} \qquad\qquad H_{\text{track}} : \mathcal{M} \rightarrow \mathcal{H}$$

Parihaka dataset:
Veritas, 2005, Parihaka 3D Marine Seismic Survey - Acquisition and Processing Report: Technical Report New Zealand Petroleum Report 3460, New Zealand Petroleum & Minerals, Wellington.

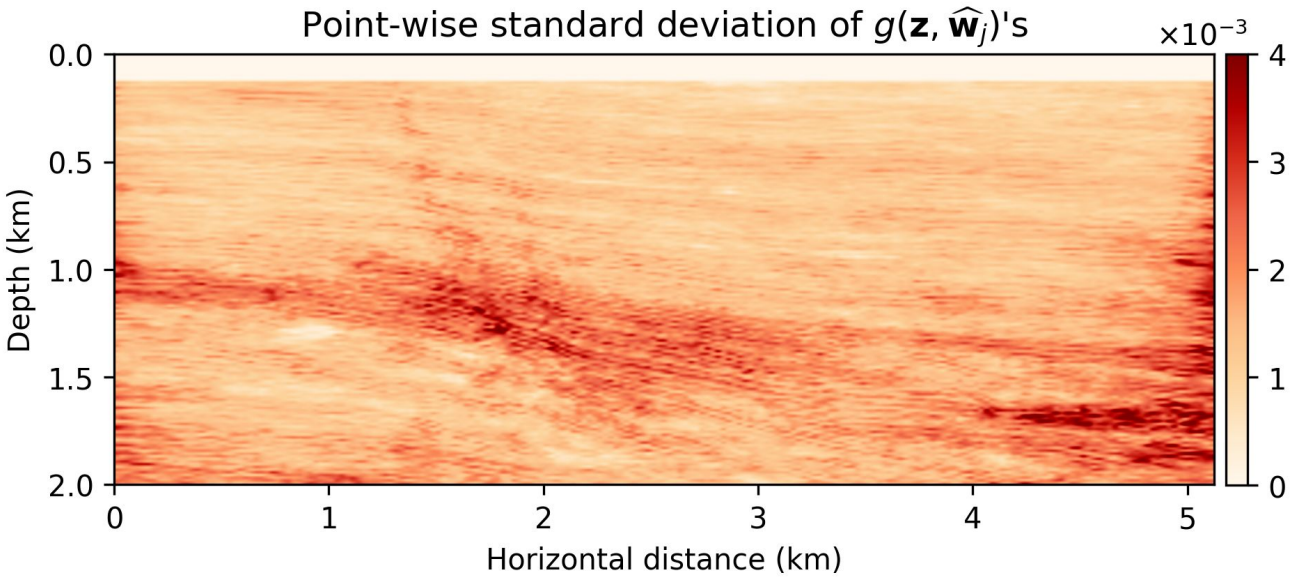
MLE



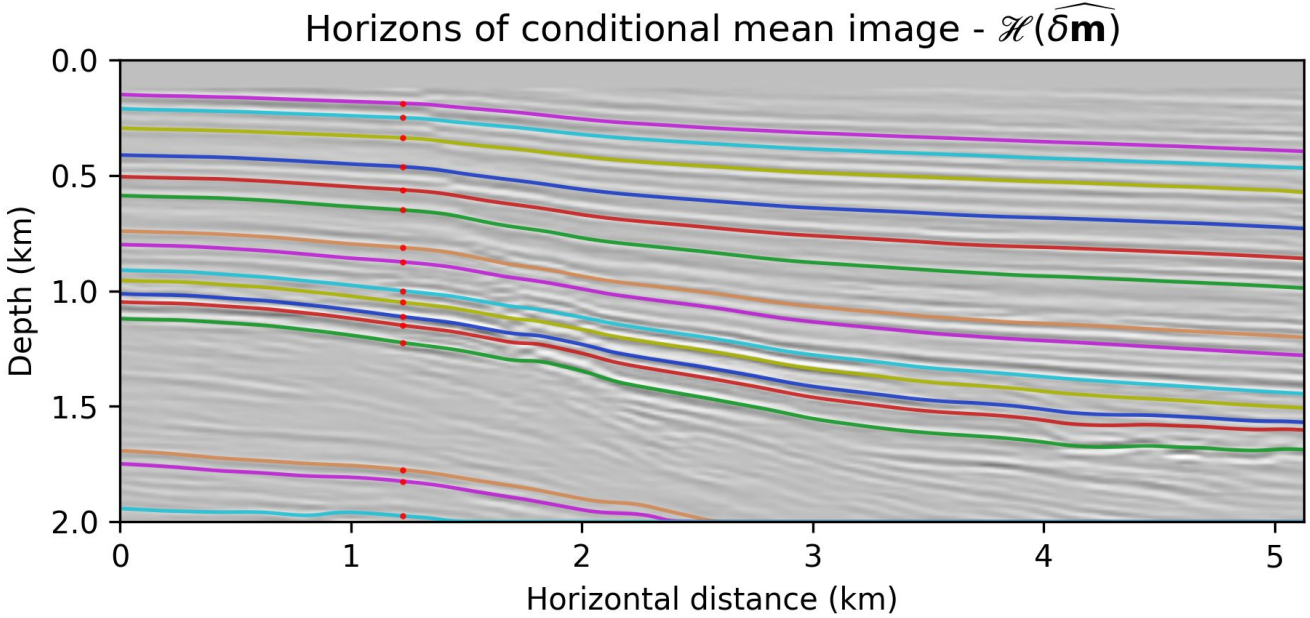
Conditional mean



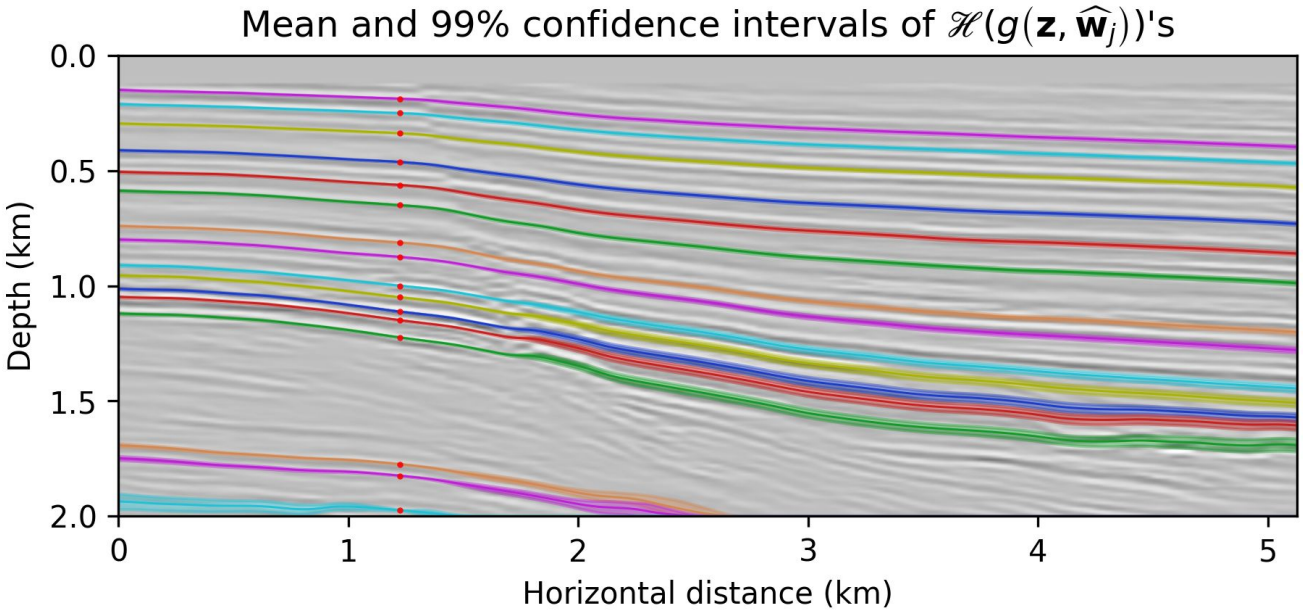
UQ: conditional pointwise standard deviation



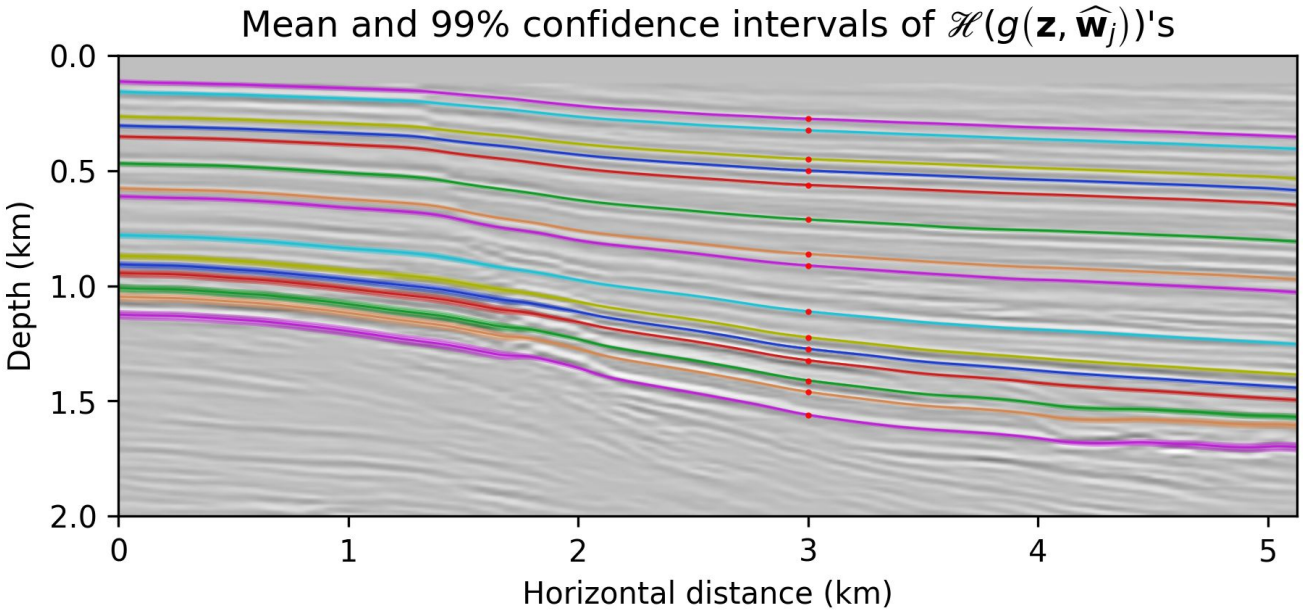
Horizon tracking on conditional mean



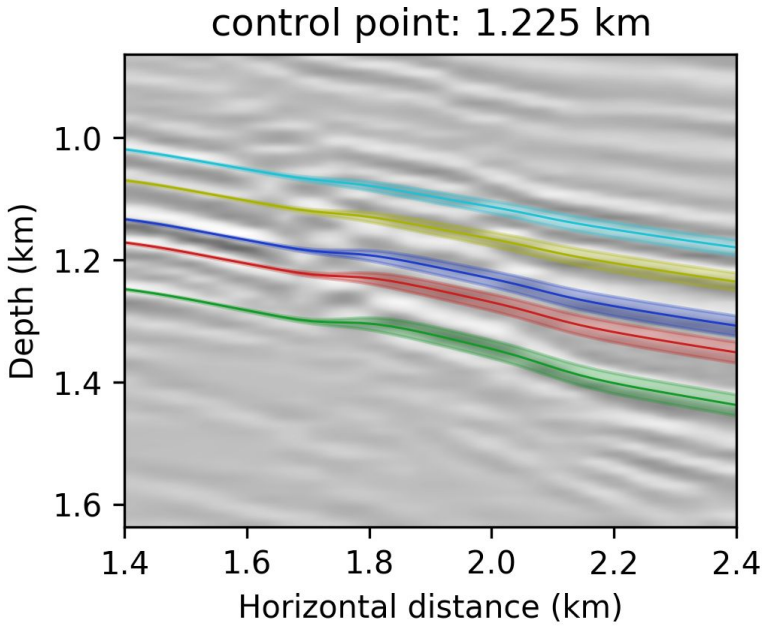
Horizon tracking and UQ

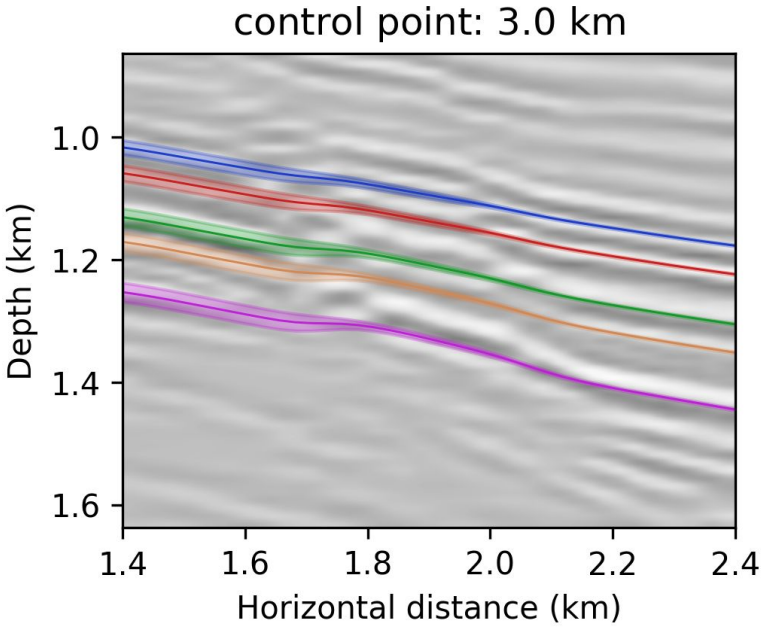


Horizon tracking and UQ



Horizon tracking and UQ





Discussion and conclusions

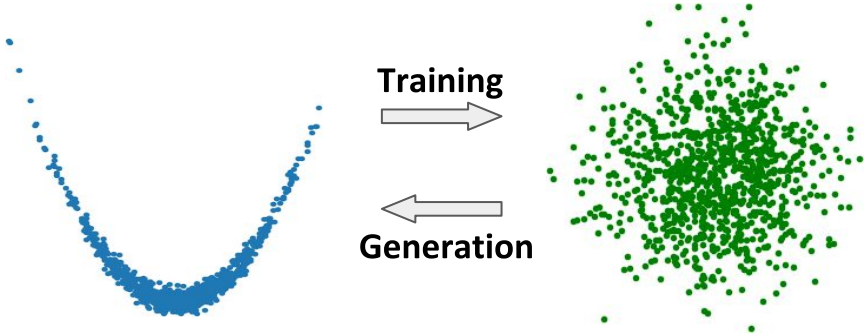
- Networks provide a beneficial inductive bias for inverse problems and their UQ (comprising tasks)
- MCMC methods tend to be expensive...

Invertible neural networks and uncertainty quantification

- Gabrio Rizzuti, Ali Siahkoohi, Philipp A. Witte, and Felix J. Herrmann. "Parameterizing uncertainty by deep invertible neural networks, an application to reservoir characterization." arXiv preprint arXiv:2004.07871 (2020).

Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. “Normalizing Flows: An Introduction and Review of Current Methods”, arXiv:1908.09257

Normalizing flows



$\mathbf{x} \sim p_X(\mathbf{x})$ $\mathbf{z} \sim p_Z(\mathbf{z})$

$$G_\theta : \mathcal{X} \to \mathcal{Z}$$

Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. “Normalizing Flows: An Introduction and Review of Current Methods”, arXiv:1908.09257

Normalizing flows

- **Invertible** → constant memory complexity

- Generate samples from target distribution

$$\mathbf{z} \sim p_Z(\mathbf{z}), \quad \mathbf{x} = G_\theta^{-1}(\mathbf{z})$$

when trained via MLE (no adversarial approach as GANS)

$$\begin{aligned} \log p_X(\mathbf{x}) &\approx \log p_{X,\theta}(\mathbf{x}) \qquad \text{“pull-back”} \\ &= \log p_Z(G_\theta(\mathbf{x})) + \log|\det J_{G_\theta}(\mathbf{x})| \\ &= \frac{1}{2}||G_\theta(\mathbf{x})||^2 + \log|\det J_{G_\theta}(\mathbf{x})| \end{aligned}$$

Jakob Kruse, Gianluca Detommaso, Robert Scheichl, and Ullrich Köthe. “HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference”. arXiv:1905.10687

“Unsupervised” imaging and UQ

Given data pertaining to a **single** subsurface model,

$$\begin{aligned}\min_\theta \quad & \mathrm{KL}(p_\theta(\mathbf{m})||p_{\mathrm{post}}(\mathbf{m}|\mathbf{d})) \\ & = \mathbb{E}_{\mathbf{z}\sim p_Z(\mathbf{z})}\frac{1}{2\sigma^2}||\mathbf{d} - F(G_\theta(\mathbf{z}))||^2 - \log p_{\mathrm{prior}}(G_\theta(\mathbf{z})) - \log|\det J_{G_\theta}(\mathbf{z})|\end{aligned}$$

$F : \mathcal{M} \to \mathcal{D}$ forward modeling operator

$G_\theta : \mathcal{Z} \to \mathcal{M}$

Jakob Kruse, Gianluca Detommaso, Robert Scheichl, and Ullrich Köthe. “HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference”. arXiv:1905.10687

Supervised imaging and UQ

Given joint pairs (**m**, **d**),

$$\begin{aligned} \min_\theta \quad & \mathrm{KL}(p_{\mathrm{joint}}(\mathbf{m},\mathbf{d}) || p_\theta(\mathbf{m},\mathbf{d})) \\ & = \mathbb{E}_{\mathbf{m},\mathbf{d}\sim p_{\mathrm{joint}}(\mathbf{m},\mathbf{d})} \frac{1}{2} ||G_\theta(\mathbf{m},\mathbf{d})||^2 - \log|\det J_{G_\theta}(\mathbf{m},\mathbf{d})| \end{aligned}$$

+ special **conditional structure** on G

Jakob Kruse, Gianluca Detommaso, Robert Scheichl, and Ullrich Köthe. “HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference”. arXiv:1905.10687

Supervised imaging and UQ

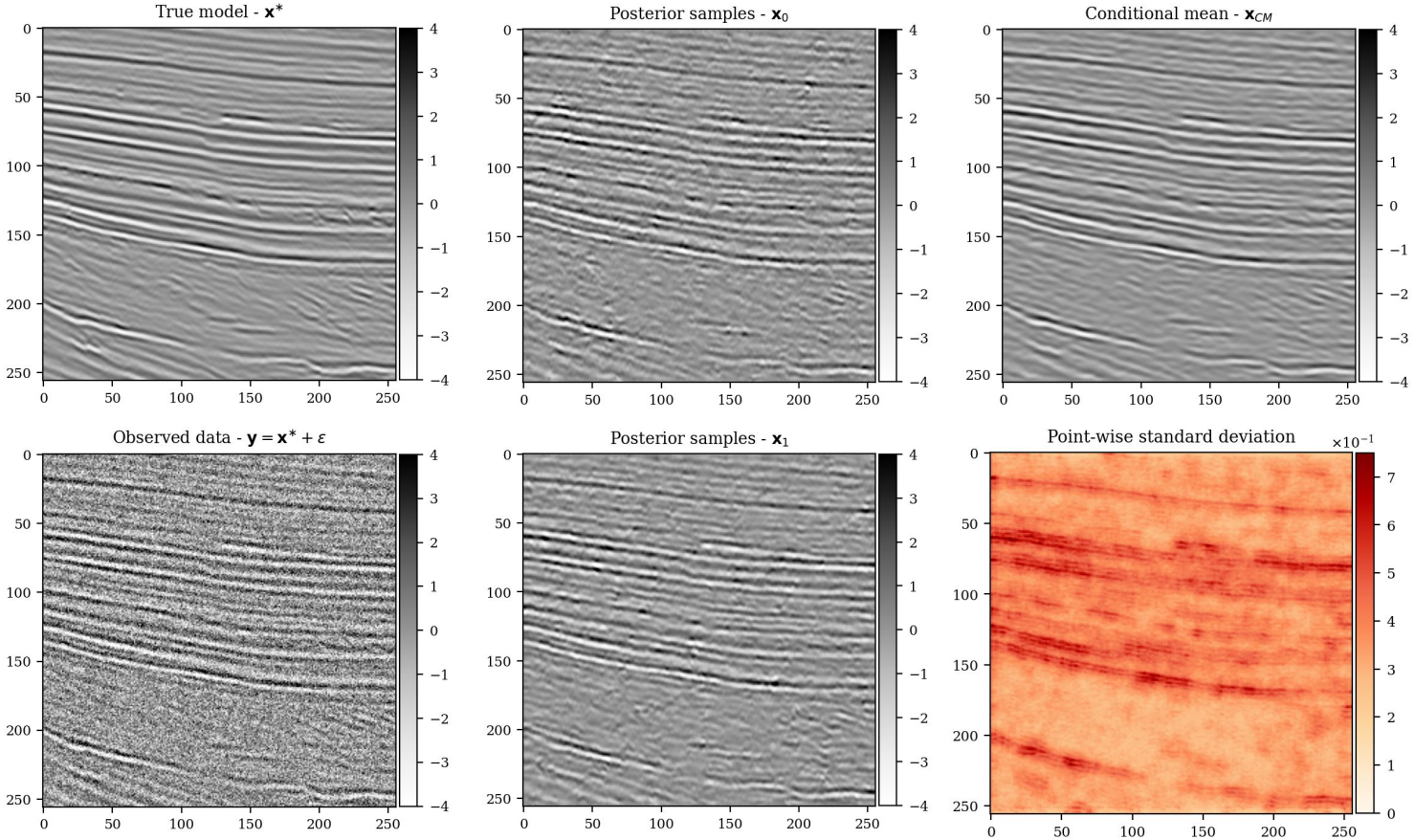
Special **conditional structure** on G

$$\begin{aligned}(\mathbf{z}_m, \mathbf{z}_d) &= G_\theta(\mathbf{m}, \mathbf{d}) \\ &= (G_\theta^m(\mathbf{m}, \mathbf{d}), G_\theta^d(\mathbf{d}))\end{aligned}$$

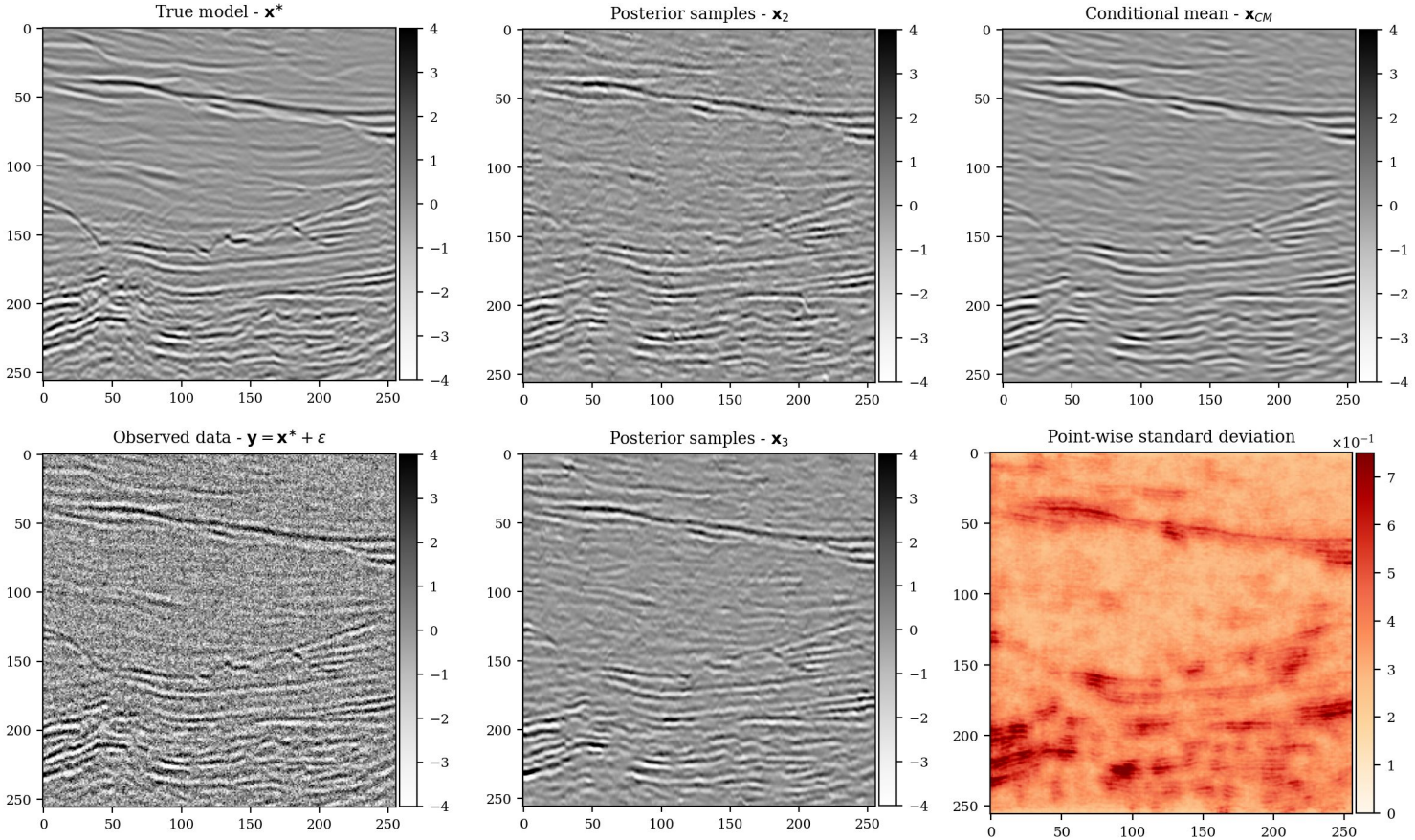
Posterior sampling w/ conditional nets:

$$\begin{aligned}&\mathbf{z}_d := G_\theta^d(\mathbf{d}), \mathbf{z}_m \sim p_Z(\mathbf{z}_m), \\ &(\mathbf{m}, \mathbf{d}) = G_\theta^{-1}(\mathbf{z}_m, \mathbf{z}_d) \Leftrightarrow \mathbf{m} \sim p_{\text{post}}(\mathbf{m}|\mathbf{d})\end{aligned}$$

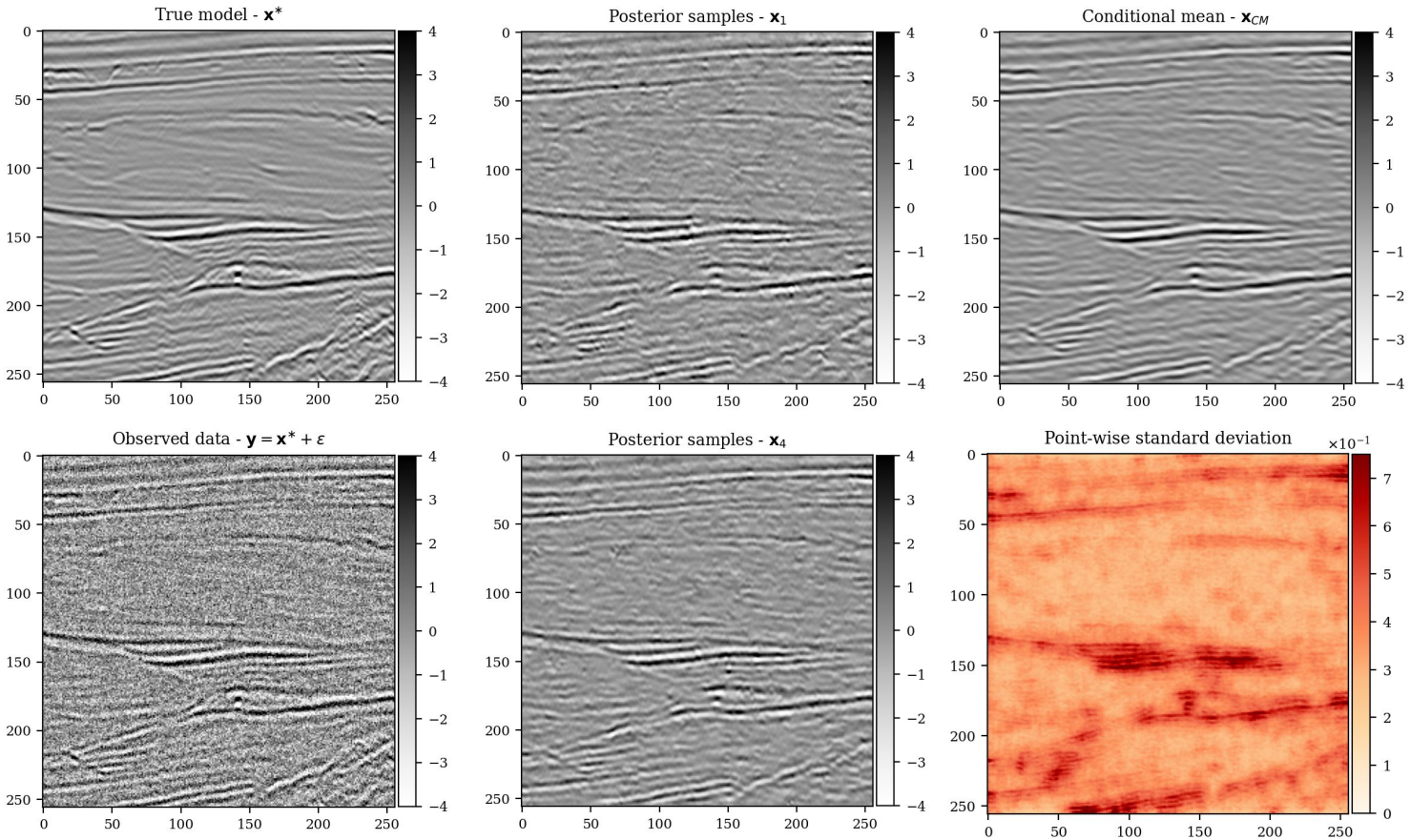
Supervised UQ (pseudo-imaging example)



Supervised UQ (pseudo-imaging example)



Supervised UQ (pseudo-imaging example)



Discussion and conclusions

- Invertible nets offer a new way to think about UQ (w/ potential advantages over classical MCMC methods)
- Imaging for large scale problems will have to rely on memory savvy networks
- Training might be expensive...

Luporini, Fabio, et al. "Architecture and performance of Devito, a system for automated stencil computation." *arXiv preprint arXiv:1807.03032* (2018).
Louboutin, Mathias, et al. "Devito: an embedded domain-specific language for finite differences and geophysical exploration." *arXiv preprint arXiv:1808.01995* (2018).
Wu, Xinming, and Sergey Fomel. "Least-squares horizons with local slopes and multigrid correlations." Geophysics 83.4 (2018): IM29-IM40.

Implementation details

Wave-equation operators via Devito:

- *https://github.com/devitocodes/devito*

Integrating Devito into PyTorch:

- *https://github.com/slimgroup/Devito4PyTorch*

Automatic horizon tracking:

- *https://github.com/xinwucwp/mhe*

Software

***InvertibleNetworks.jl*: a scalable Julia package for invertible nets**

- Networks w/o need to store forward states during training
- Manually implemented derivatives that make use of invertibility
- Unit testing (adjoints, gradients, invertibility)
- *https://github.com/slimgroup/InvertibleNetworks.jl* (MIT license)

Software

Class structure:

```
struct CouplingLayerGlow <: NeuralNetLayer
    C::Conv1x1
    RB::ResidualBlock
    logdet::Bool
    forward::Function
    inverse::Function
    backward::Function
end
```

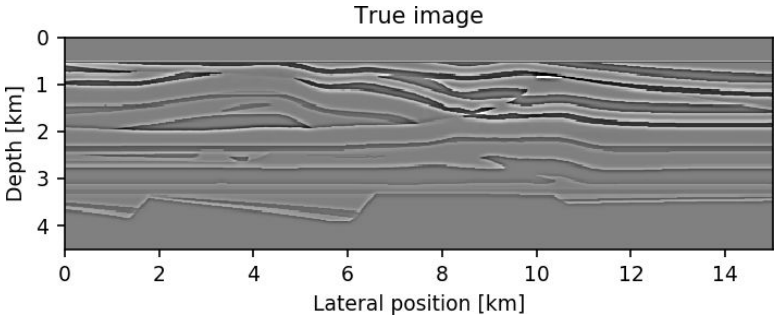
Usage:

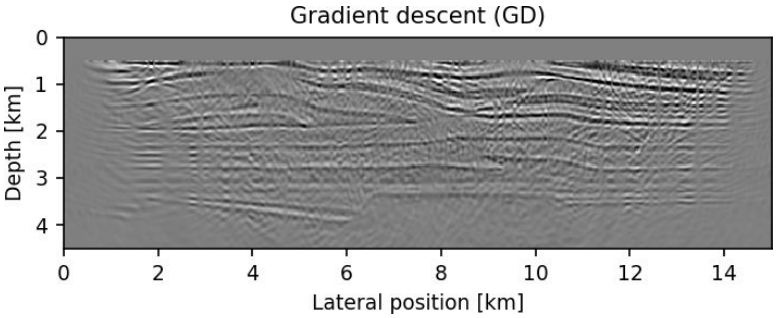
```
# Create Layer
LayerGlow = CouplingLayerGlow(nx, ny, nchannel, nhidden, batchsize)

# Operations
Y = LayerGlow.forward(X)
X_ = LayerGlow.inverse(Y)
ΔX, X_ = LayerGlow.backward(ΔY, Y)
```

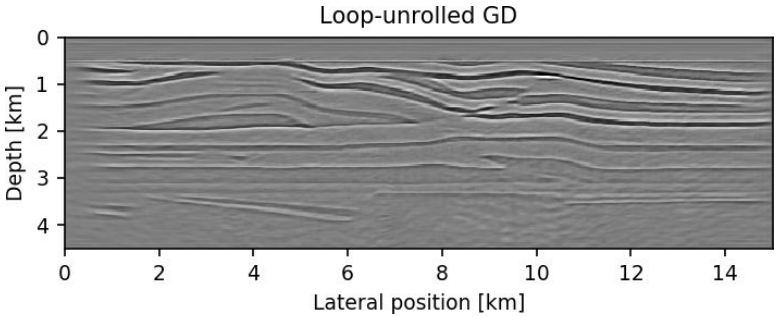
Loop unrolled seismic imaging

Training loop unrolled network:

- 1000 pairs of seismic data + true image
- Single epoch w/ ADAM







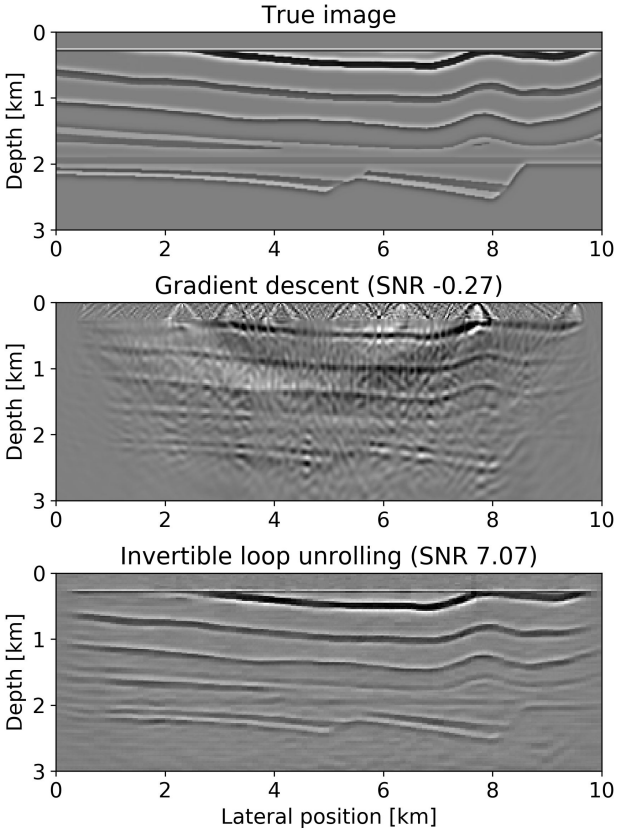
Loop unrolling w/ invertible networks

Invertible version of loop-unrolled network:

- Loop unrolling: memory scales w/ network depth
- Instead: architecture based on invertible nets
- Invertible + tractable logdet

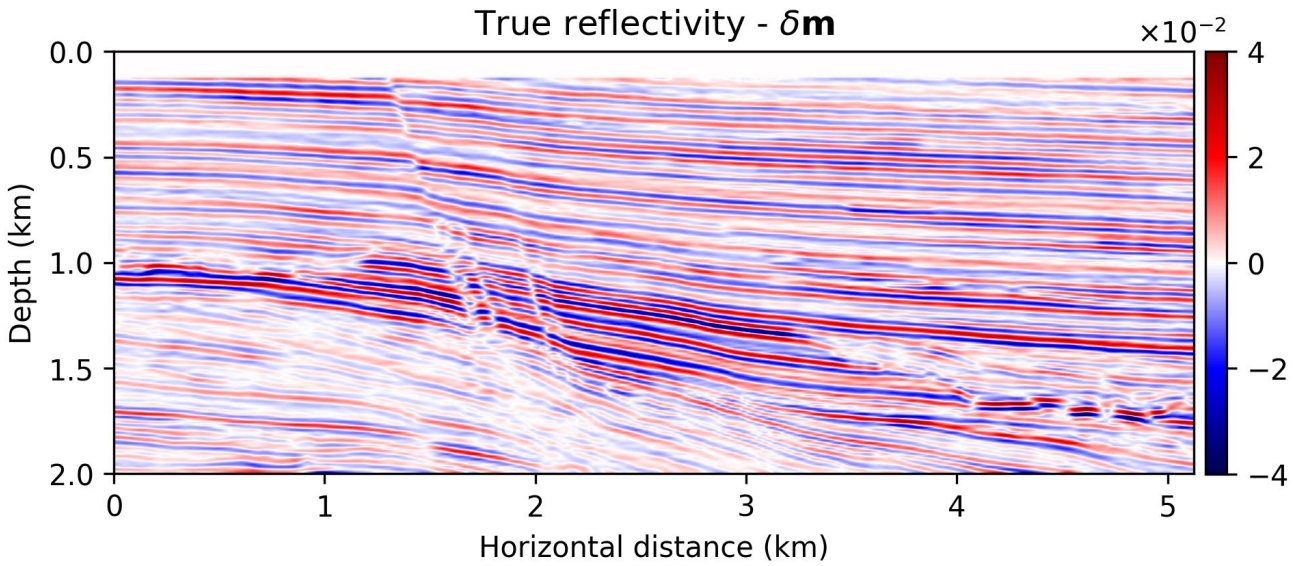
---

function $\mathcal{G}(\mathbf{J}, \mathbf{d})$
  $\mathbf{x} = 0$
  for $j = 1, ..., n$
    $\mathbf{x} = \mathbf{Q}\mathbf{x}$
    $\mathbf{x}_1' = \mathbf{x}_1$
    $\mathbf{g} = \mathbf{J}^\top\big(\mathbf{J}\mathbf{x}_1'[1] - \mathbf{d}\big)$
    $\mathbf{s}', \mathbf{t} = \mathrm{NN}\big([\mathbf{g}, \mathbf{x}_1'[2{:}\mathrm{end}]]\big)$
    $\mathbf{s} = \sigma\big(\mathbf{s}'\big)$
    $\mathbf{x}_2' = \mathbf{x}_2 \odot \mathbf{s} + \mathbf{t}$
    $\mathbf{x} = \mathbf{Q}^\top\mathbf{x}'$
  end
  return $\mathbf{x}$
end



Veritas, 2005, Parihaka 3D Marine Seismic Survey - Acquisition and Processing Report: Technical Report New Zealand Petroleum Report 3460, New Zealand Petroleum & Minerals, Wellington.

True model: Parihaka dataset



Confidence intervals empirical verification