Parameterizing uncertainty by deep invertible networks

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



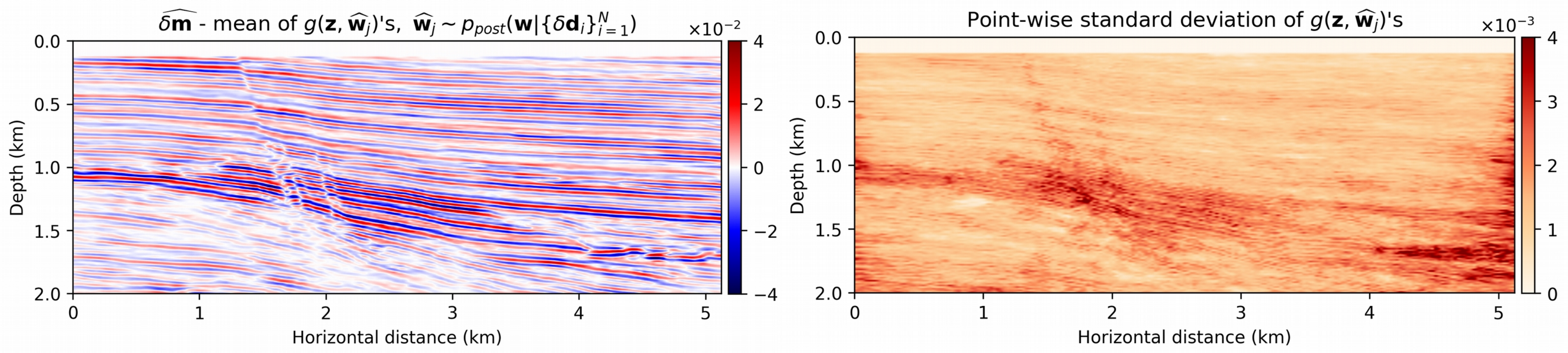

[1] Georgia Institute of Technology, School of Computational Science and Engineering
[2] Microsoft Research

SLIM

SEG 2020

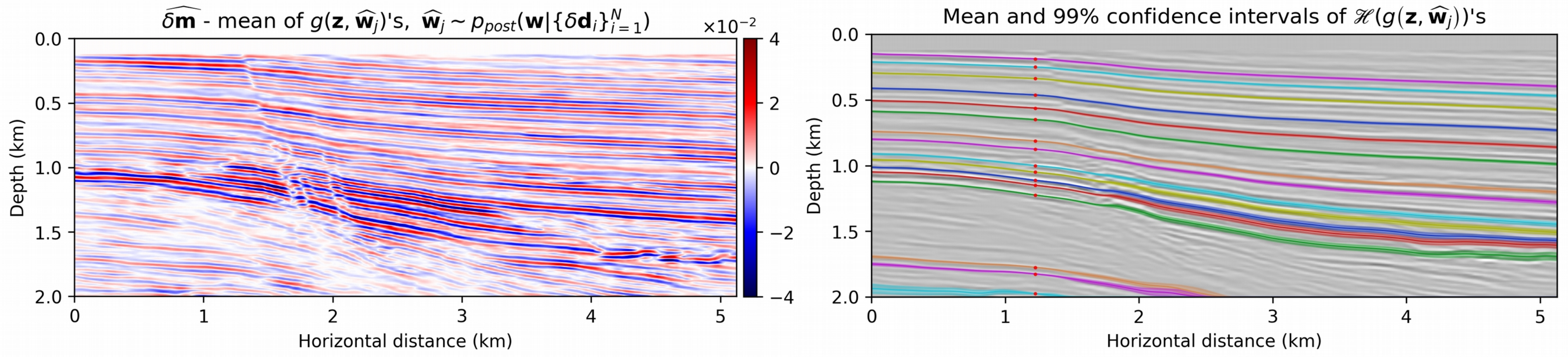
- Siahkoohi, A., Rizzuti, G., and Herrmann, F. J., 2020, *Uncertainty quantification in imaging and automatic horizon tracking: a Bayesian deep-prior based approach*

Uncertainty quantification for imaging



- Siahkoohi, A., Rizzuti, G., and Herrmann, F. J., 2020, *Uncertainty quantification in imaging and automatic horizon tracking: a Bayesian deep-prior based approach*

Uncertainty quantification for imaging



Bayesian framework

$$\mathbf{d} \sim p_{\text{like}}(\mathbf{d}|\mathbf{m})$$

$$\mathbf{m} \sim p_{\text{prior}}(\mathbf{m})$$

$$\mathbf{m} \sim p_{\text{post}}(\mathbf{m}|\mathbf{d}) \propto p_{\text{like}}(\mathbf{d}|\mathbf{m})\, p_{\text{prior}}(\mathbf{m})$$

Bayesian framework

$$-\log p_{\mathrm{like}}(\mathbf{d}|\mathbf{m}) = \frac{1}{2\sigma_{\mathrm{d}}^2}||\mathbf{d} - \mathcal{F}(\mathbf{m})||^2$$

$$-\log p_{\mathrm{prior}}(\mathbf{m}) = R(\mathbf{m})$$

$$-\log p_{\mathrm{post}}(\mathbf{m}|\mathbf{d}) = \frac{1}{2\sigma_{\mathrm{d}}^2}||\mathbf{d} - \mathcal{F}(\mathbf{m})||^2 + R(\mathbf{m})$$

- Welling, M, and Teh, Y. W., 2011, *Bayesian Learning via Stochastic Gradient Langevin Dynamics*
- Brosse, N., Moulines, E., and Durmus A., 2018, *The promises and pitfalls of Stochastic Gradient Langevin Dynamics*

SLIM

MCMC

Langevin Dynamics

$$\mathbf{m}_{t+\Delta t} = \mathbf{m}_t + \Delta t \, \nabla_{\mathbf{m}} \log p_{\text{post}}(\,\cdot\,|\mathbf{d})|_{\mathbf{m}=\mathbf{m}_t} + \mathcal{N}(\mathbf{0}, 2\Delta t)$$

Computationally inefficient → stochastic gradient LD?

$$-\log p_{\text{post}}(\mathbf{m}|\mathbf{d}) \approx \frac{1}{2\sigma_{\mathrm{d}}^2} \sum_{i=1}^{n_s} \frac{N_s}{n_s} ||\mathbf{d}_i - \mathcal{F}(\mathbf{m})_i||^2 + R(\mathbf{m})$$

- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D., 2017, *Variational Inference: A Review for Statisticians*
- Zhang, X., and Curtis, A., 2019, *Seismic tomography using variational inference methods*
- Siahkoohi, Ali, Rizzuti, G., Witte, P. A., and Herrmann, F. J., 2020, *Faster Uncertainty Quantification for Inverse Problems with Conditional Normalizing Flows*

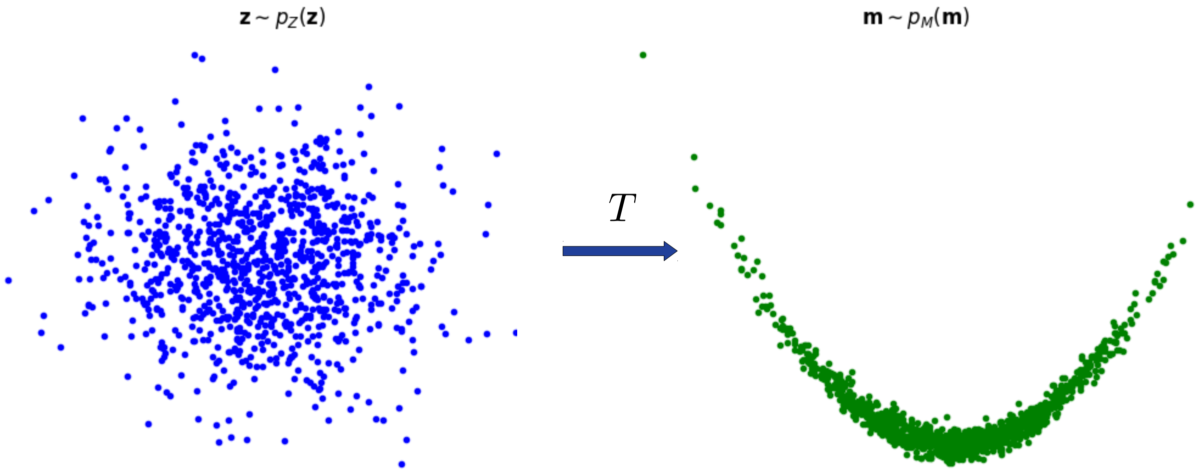
Variational Inference

$$p_{\text{post}}(\mathbf{m}|\mathbf{d}) \approx p_\theta(\mathbf{m}), \quad \theta \in \Theta$$

- Density encoded by transport maps (e.g. neural networks)
- Stochastic VI vs MCMC
  - small-batch optimization
  - parallelization
  - preconditioning/transfer learning

Transport Maps

$$\mathbf{z} \sim p_Z(\mathbf{z}), \quad T : Z \to M, \quad T_{\#}p_Z \approx p_{\text{post}}(\,\cdot\,|\mathbf{d})$$



Variational inference w/ TMs

$$\min_{T} \; \mathrm{KL}(T_{\#}p_Z || p_{\mathrm{post}}) =$$

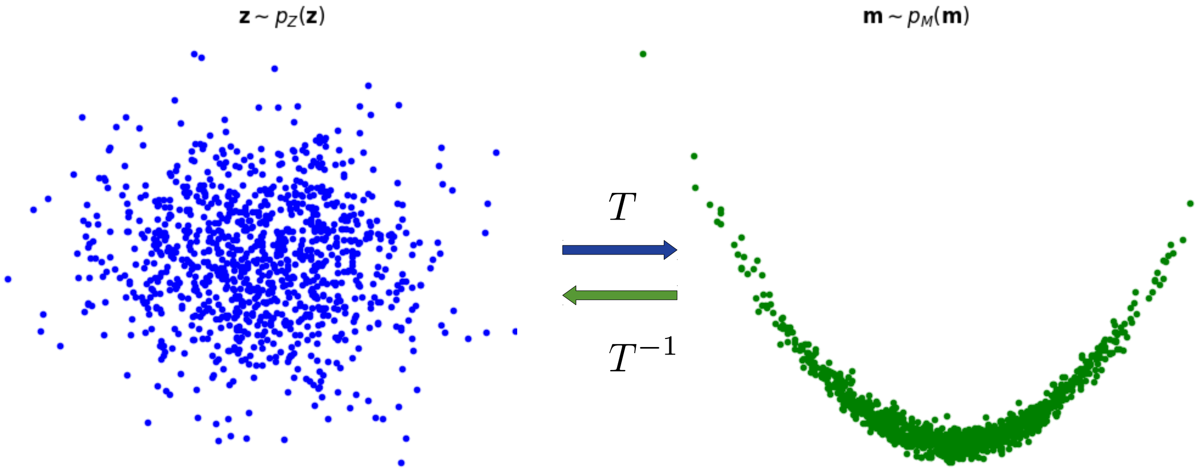
$$= \mathbb{E}_{\mathbf{z}\sim p_Z(\mathbf{z})} \frac{1}{2\sigma_d^2} ||\mathbf{d} - \mathcal{F}(T(\mathbf{z}))||^2 + R(T(\mathbf{z})) - H(T_{\#}p_Z)$$

- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B., 2019, *Normalizing Flows for Probabilistic Modeling and Inference*
- Kobyzev, I., Prince, S. J. D., Brubaker, M. A., 2020, *Normalizing Flows: An Introduction and Review of Current Methods*
- Kingma, D., and Welling, M., 2014. *Auto-encoding variational Bayes*

Normalizing Flows

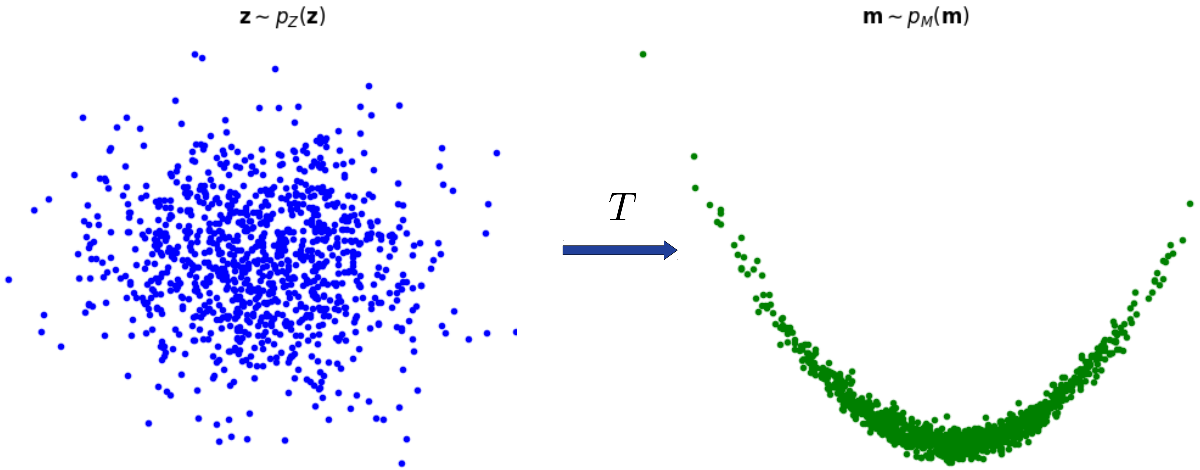
Advantages: memory efficient, change of variable formula

$$p_T(\mathbf{m}) = p_Z(T^{-1}(\mathbf{m}))|\det J_T(T^{-1}(\mathbf{m}))|^{-1}$$

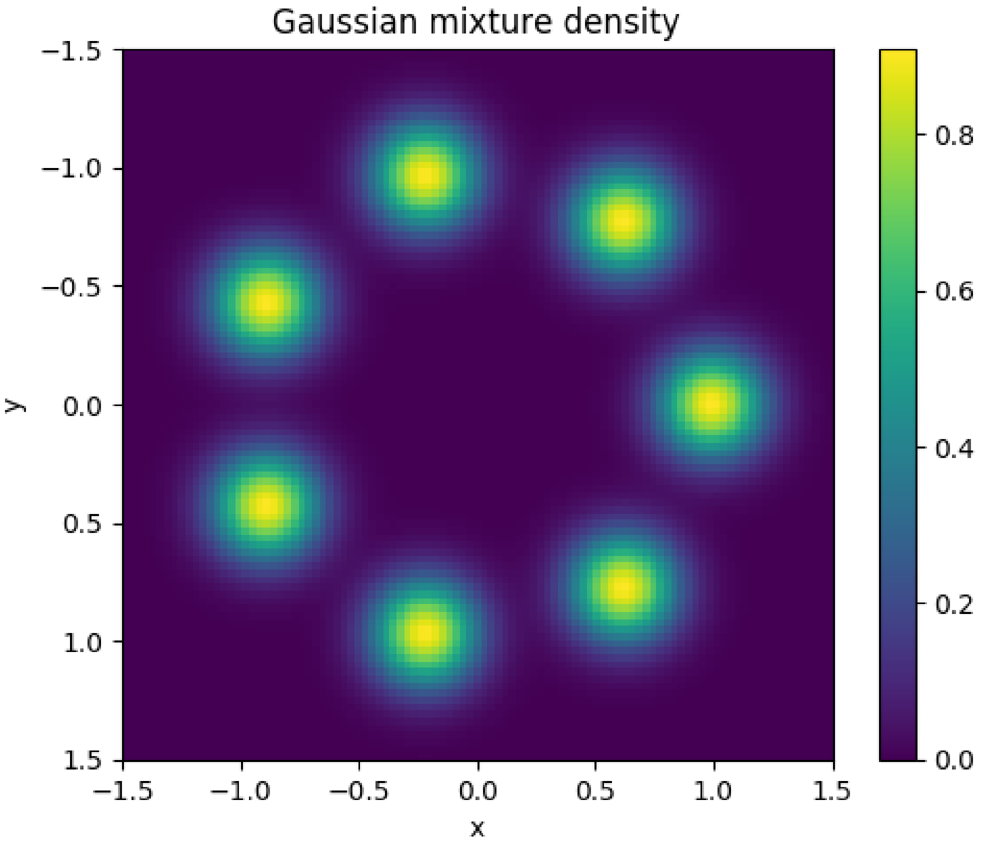


Training and testing UQ w/ NFs

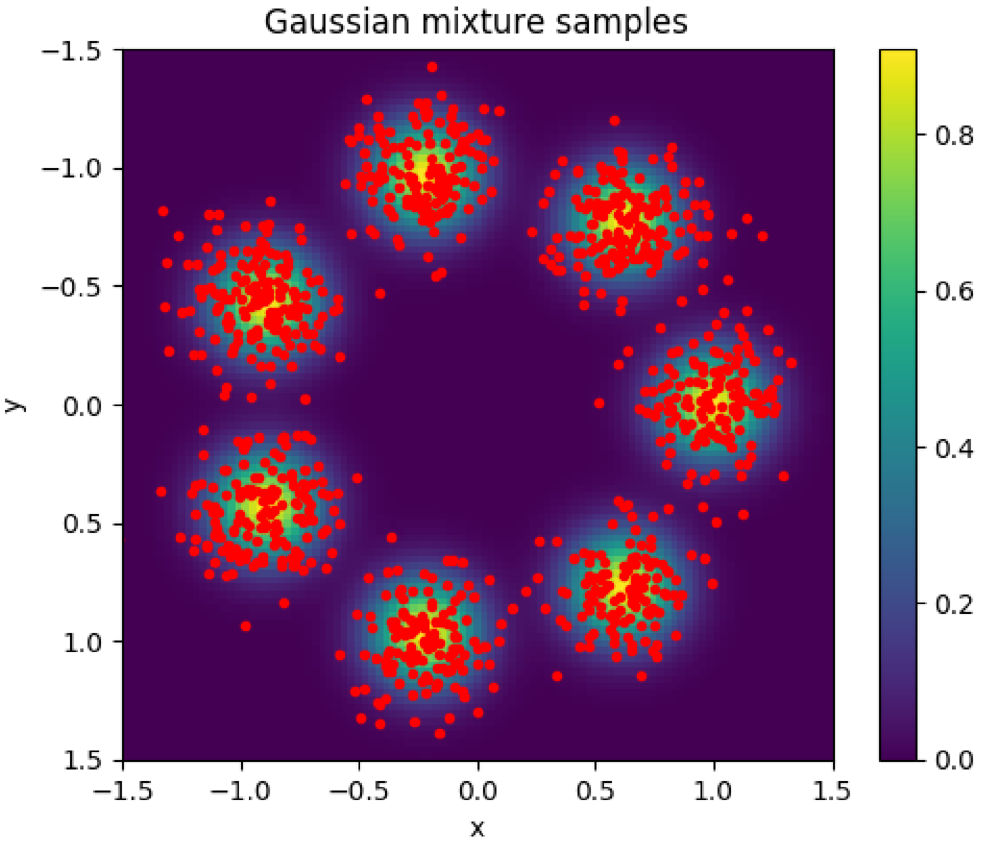
$$\min_{\theta} \; \mathbb{E}_{\mathbf{z}\sim p_Z(\mathbf{z})} \; \frac{1}{2\sigma_d^2}||\mathbf{d} - \mathcal{F}(T_\theta(\mathbf{z}))||^2 + R(T_\theta(\mathbf{z})) - \log|\det J_{T_\theta}(\mathbf{z})|$$



Example (2D)

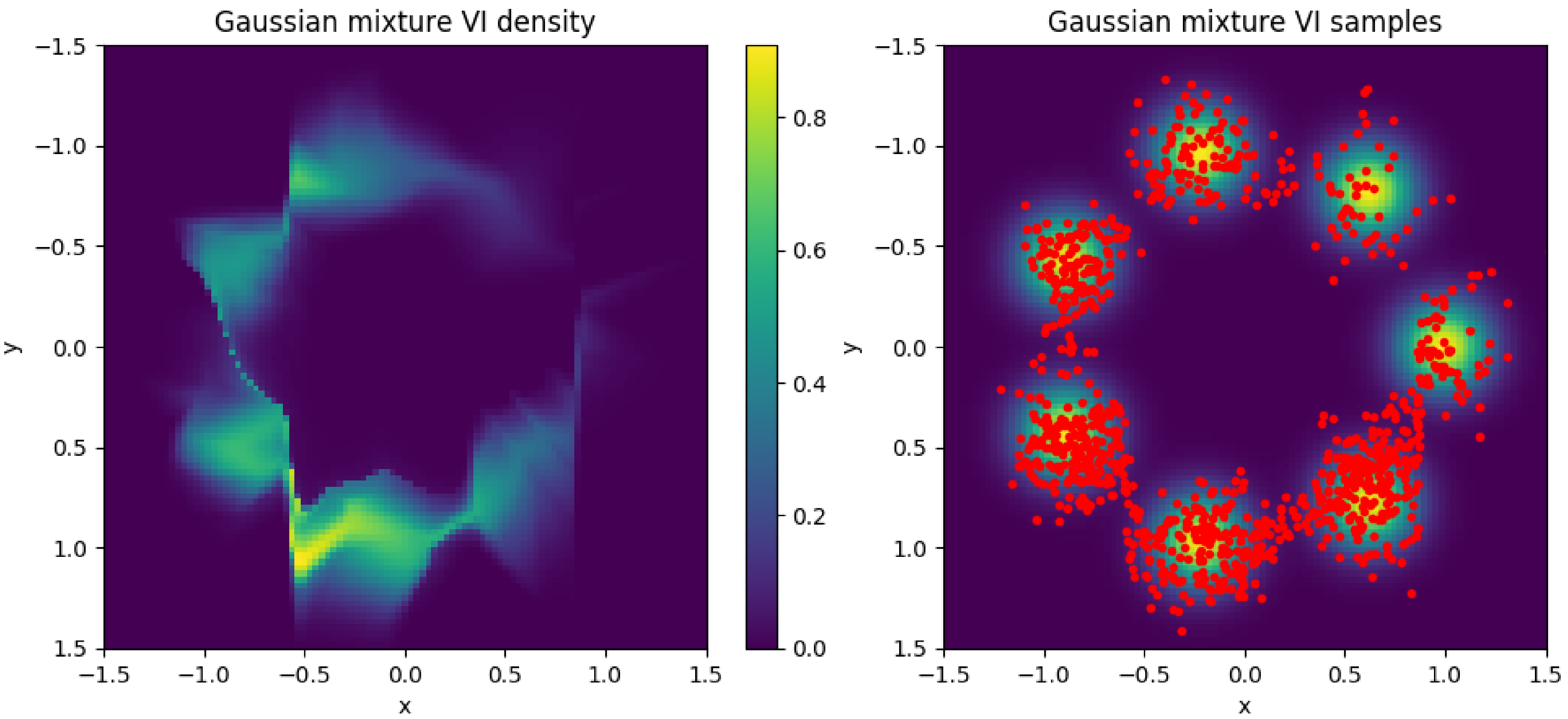


Gaussian mixture density
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
y
x
0.8
0.6
0.4
0.2
0.0



Gaussian mixture samples
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
y
x
0.8
0.6
0.4
0.2
0.0

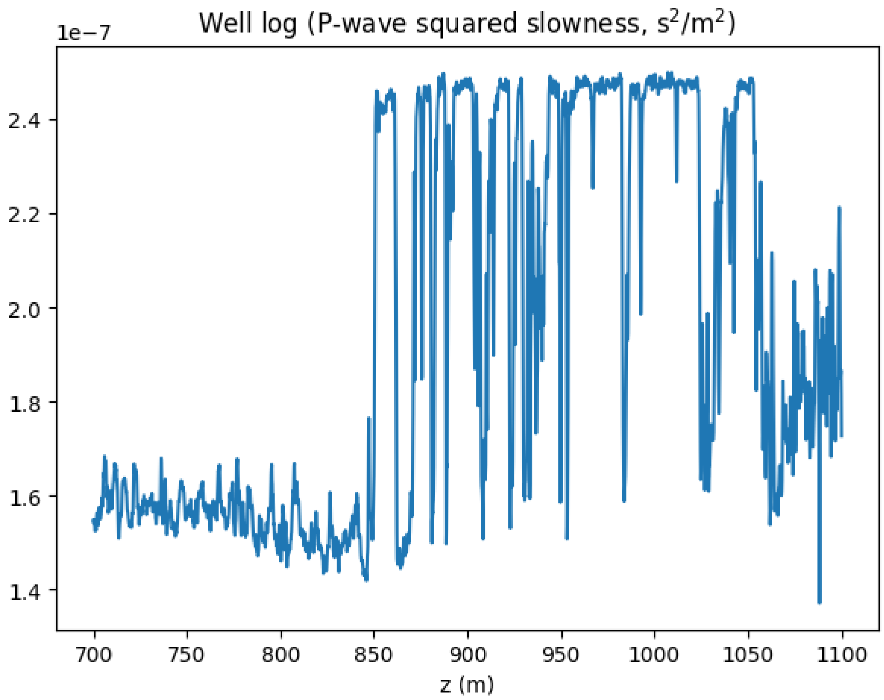
Example (2D)

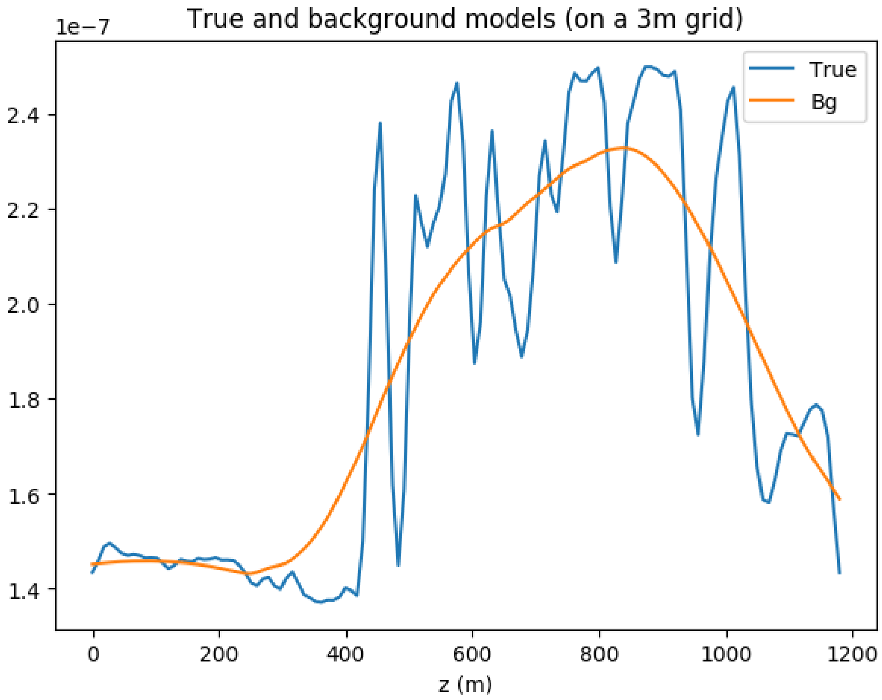


Gaussian mixture VI density
Gaussian mixture VI samples
y
x
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
0.0
0.2
0.4
0.6
0.8

Example (AVP)

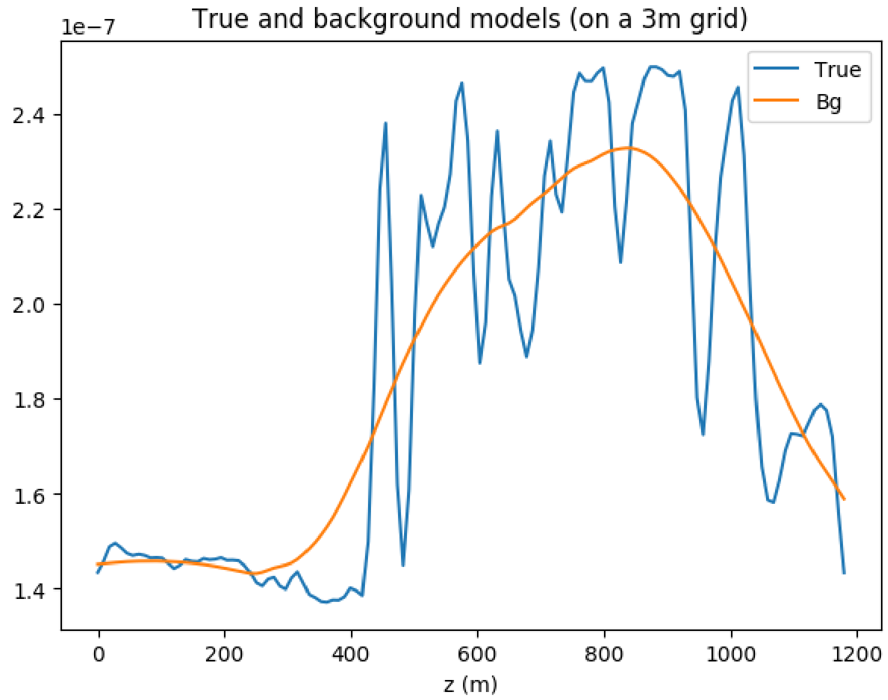
- Sleipner data, `co2datashare.org/`, thanks to Equinor!

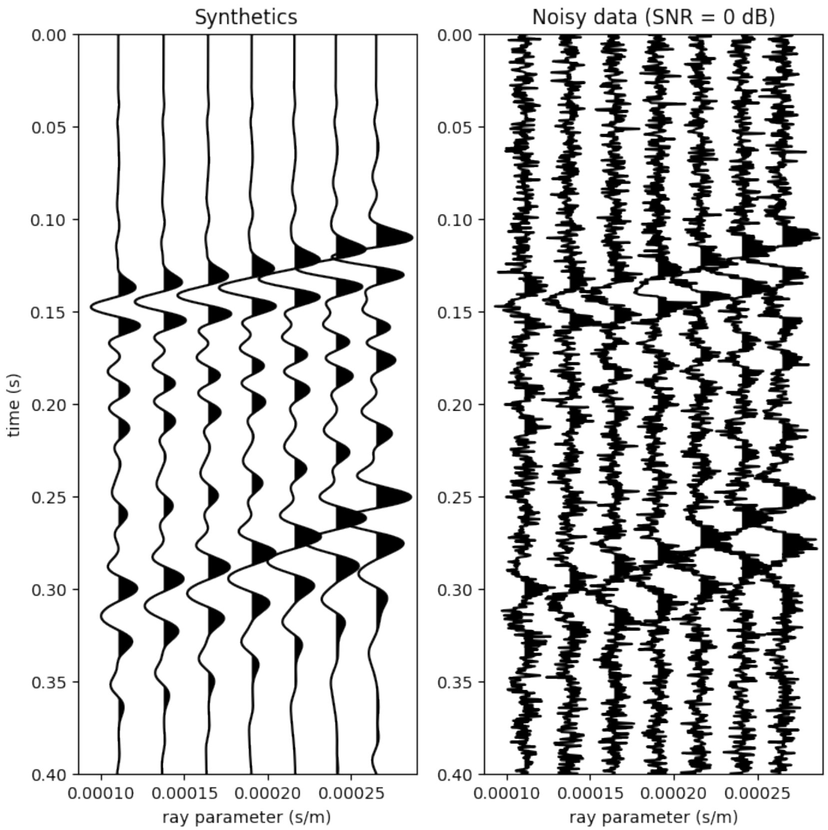




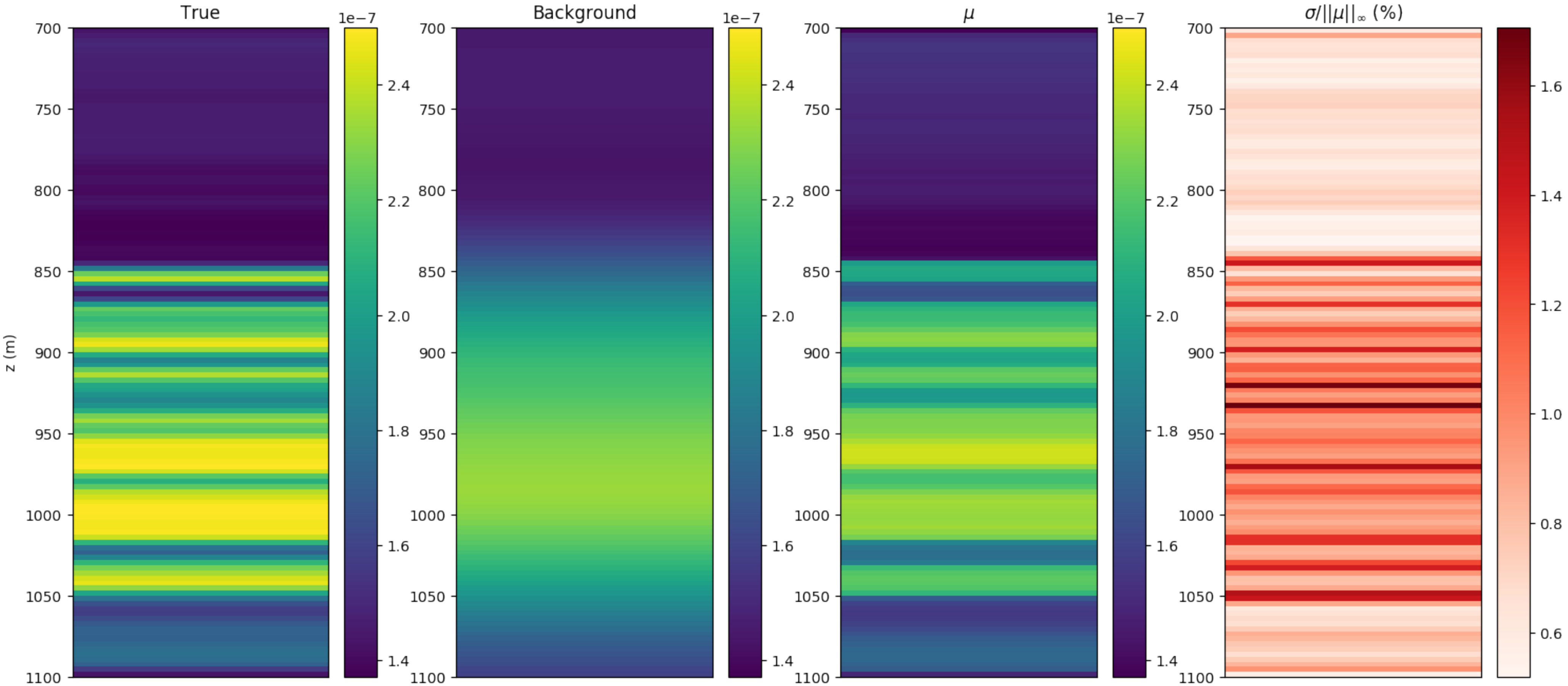
> Sleipner data, `co2datashare.org/`, thanks to Equinor!

Example (AVP)

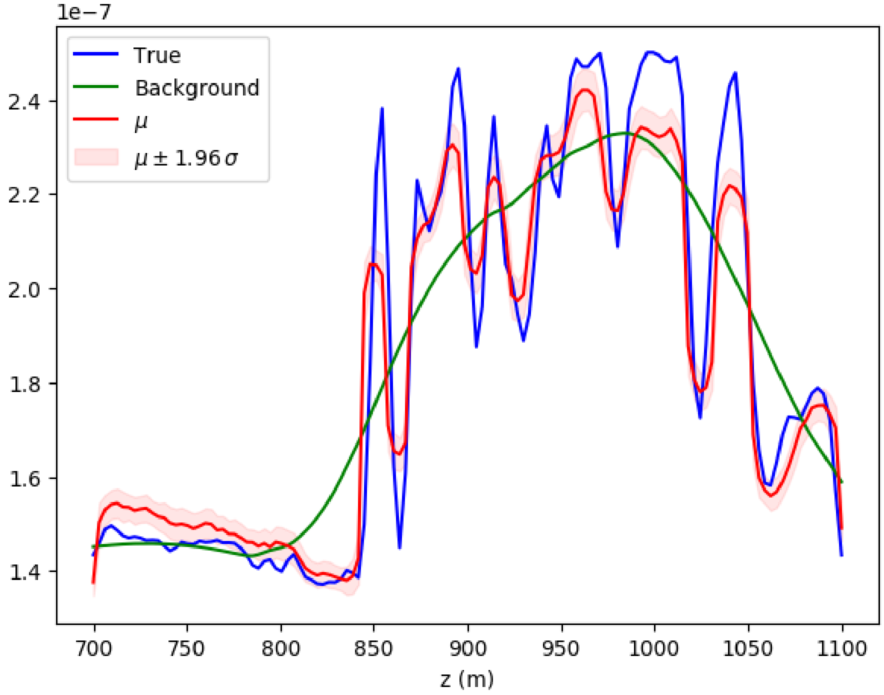




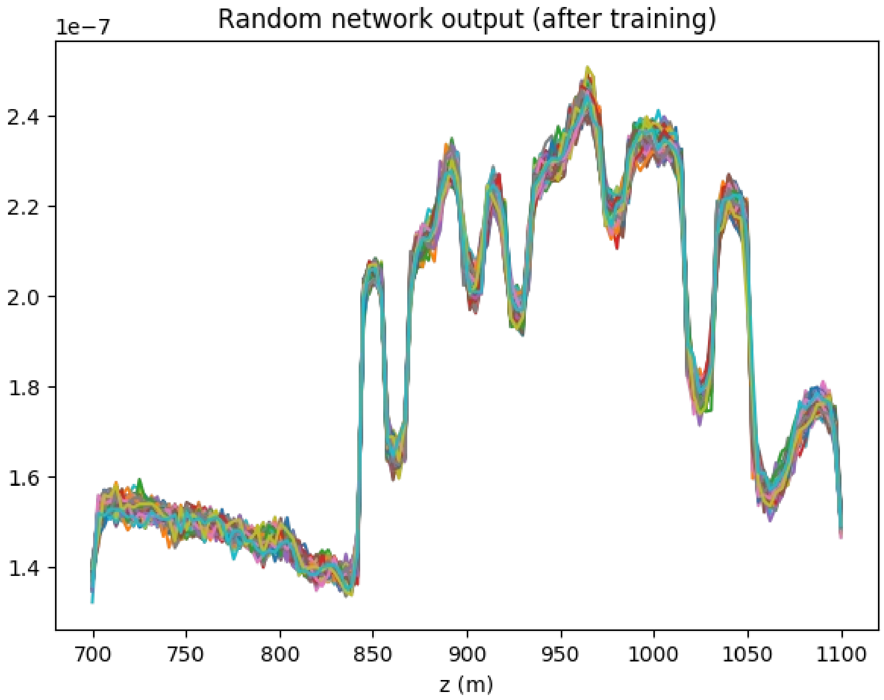
Example (AVP)



Example (AVP)



1e−7
True
Background
μ
$\mu \pm 1.96\,\sigma$
2.4
2.2
2.0
1.8
1.6
1.4
700 750 800 850 900 950 1000 1050 1100
z (m)



Random network output (after training)
1e−7
2.4
2.2
2.0
1.8
1.6
1.4
700 750 800 850 900 950 1000 1050 1100
z (m)

github.com/slimgroup/InvertibleNetworks.jl

github.com/slimgroup/Software.SEG2020/

```julia
# Glow network
G = NetworkGlow(nx, ny, n_in, batchsize, n_hidden, L, K) #|>gpu

# Objective function
function loss(X)
    Y, logdet = G.forward(X)
    f = .5f0/batchsize*norm(Y)^2 - logdet
    ΔX, X_ = G.backward(1f0./batchsize*Y, Y)
    return f
end

# Evaluate loss
f = loss(X)

# Update weights
opt = Flux.ADAM()
Params = get_params(G)
for p in Params
    update!(opt, p.data, p.grad)
end
clear_grad!(G)
```

- Behrmann, J, Vicol, P., Wang, K.-C., Grosse, R., and Jabobsen, J.-H., 2020, *Understanding and mitigating exploding inverses in invertible neural networks*
- Siahkoohi, Ali, Rizzuti, G., Witte, P. A., and Herrmann, F. J., 2020, *Faster Uncertainty Quantification for Inverse Problems with Conditional Normalizing Flows*

Discussion and conclusions

- Variational inference vs MCMC for seismic inversion
- Non-smooth priors? Hard constraints?
- Extension to 2D/3D w/ invertible networks
- Problem-specific inductive bias (loop unrolling)?
- Supervised+unsupervised via transfer learning?