# Preconditioned training of normalizing flows for variational inference in inverse problems

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

[1]Georgia Institute of Technology

[2]Microsoft

SLIM
Georgia Institute of Technology

# Inverse problems



Find $\boldsymbol{x}$ such that:

$$\boldsymbol{y} = F(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

observed data, $\boldsymbol{y}$

measurement noise, $\boldsymbol{\epsilon} \sim \mathrm{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$

expensive (nonlinear) forward operator, $F$

limited access to training data (marginal or joint)

# Variational inference with normalizing flows



$$\min_{\theta} \mathbb{E}_{\boldsymbol{z}\sim \mathrm{N}(\boldsymbol{0},\boldsymbol{I})} \left[ \frac{1}{2\sigma^2} \left\| F\big(T_\theta(\boldsymbol{z})\big) - \boldsymbol{y} \right\|_2^2 - \log \pi_{\text{prior}}\big(T_\theta(\boldsymbol{z})\big) - \log \left| \det \nabla_{z} T_\theta(\boldsymbol{z}) \right| \right]$$

no training data required

cheap posterior sampling: $T_\theta(\boldsymbol{z}) \sim \pi_\theta(\boldsymbol{x} \mid \boldsymbol{y}) \approx \pi_{\text{post}}(\boldsymbol{x} \mid \boldsymbol{y})$

requires a prior density, $\pi_{\text{prior}}(\boldsymbol{x})$

repeated evaluations of $F$ and $\nabla F^\top$

specific to one observation, $\boldsymbol{y}$

---

Qiang Liu and Dilin Wang. "Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm". In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016, pp. 2378–2386. URL: `https://proceedings.neurips.cc/paper/2016/file/b3ba8f1bee1238a2f37603d90b58898d-Paper.pdf`.

# Variational inference with conditional normalizing flows



$$\min_{\phi} \mathbb{E}_{\boldsymbol{y},\boldsymbol{x}\sim\widehat{\pi}_{y,x}(\boldsymbol{y},\boldsymbol{x})} \left[\frac{1}{2}\left\|G_\phi(\boldsymbol{y},\boldsymbol{x})\right\|^2 - \log\left|\det \nabla_{y,x} G_\phi(\boldsymbol{y},\boldsymbol{x})\right|\right],$$

$$\text{where} \quad G_\phi(\boldsymbol{y},\boldsymbol{x}) = \begin{bmatrix} G_{\phi_y}(\boldsymbol{y}) \\ G_{\phi_x}(\boldsymbol{y},\boldsymbol{x}) \end{bmatrix}, \ \phi = \{\phi_y, \phi_x\}$$

cheap posterior sampling for all $\boldsymbol{y}$: $G_{\phi_x}^{-1}(G_{\phi_y}(\boldsymbol{y}), \boldsymbol{z}) \sim \pi_\phi(\boldsymbol{x} \mid \boldsymbol{y})$
does not involve the forward operator, $F$
does not require a prior density, $\pi_{\text{prior}}(\boldsymbol{x})$

heavily relies on access to training pairs, $\widehat{\pi}_{y,x} \neq \pi_{y,x}$

J. Kruse et al. *HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference*. 2019. arXiv: 1905.10687 [stat.ML].

Ricardo Baptista, Olivier Zahm, and Youssef Marzouk. *An adaptive transport framework for joint and conditional density estimation*. 2020. arXiv: 2009.10303 [stat.ML].

# Variational inference with the preconditioned scheme



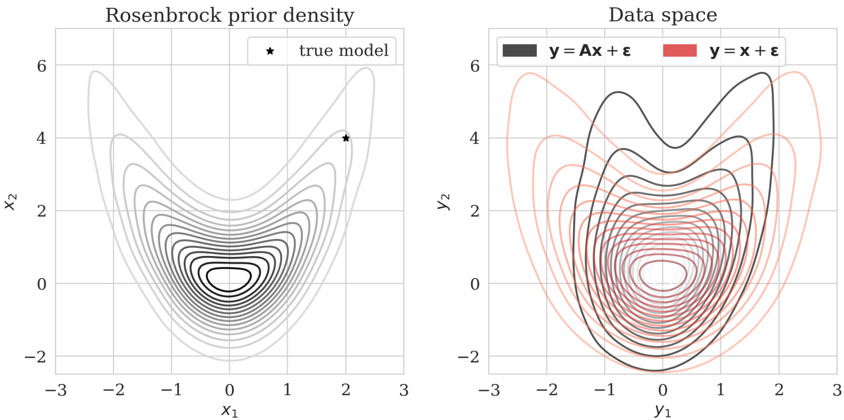
$$\min_{\phi_x} \mathbb{E}_{\boldsymbol{z}\sim \mathrm{N}(\mathbf{0},\boldsymbol{I})} \left[ \frac{1}{2\sigma^2} \left\| F\big(T_{\phi_x}(\boldsymbol{z})\big) - \boldsymbol{y} \right\|_2^2 - \log \pi_G\big(T_{\phi_x}(\boldsymbol{z})\big) - \log \left| \det \nabla_z T_{\phi_x}(\boldsymbol{z}) \right| \right]$$

transfer learning:

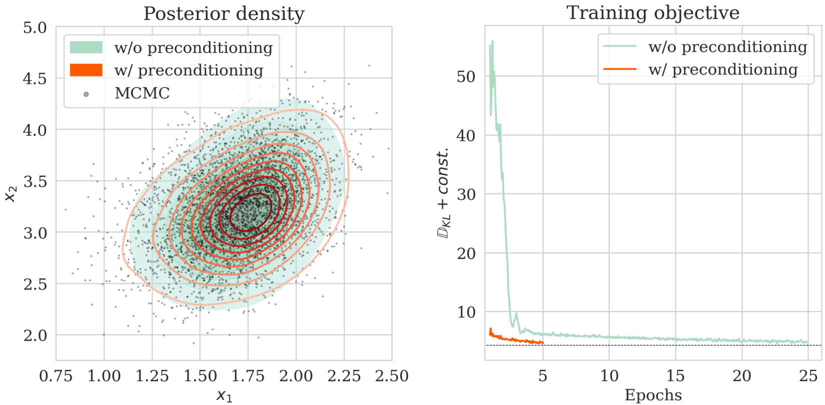
- correct for low-fidelity training data, $\widehat{\pi}_{y,x} \neq \pi_{y,x}$
- less evaluations of $F$ and $\nabla F^\top$

learned prior density, $\pi_G(\boldsymbol{x}) := \pi_{\boldsymbol{z}}\big(G_{\phi_x}(\boldsymbol{y}, \boldsymbol{x})\big) \left| \det \nabla_x G_{\phi_x}(\boldsymbol{y}, \boldsymbol{x}) \right|$

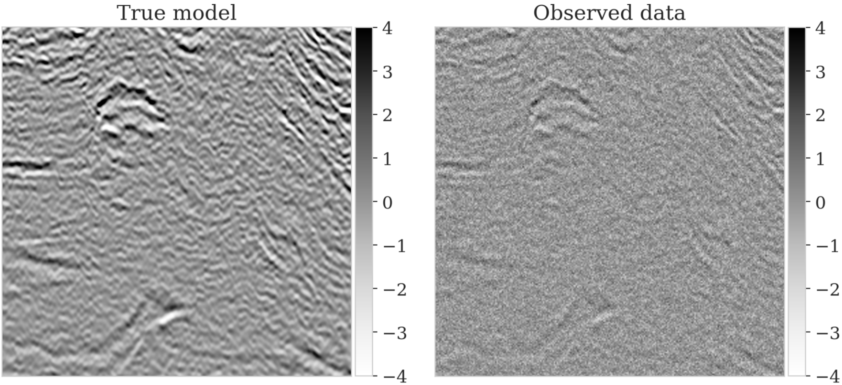
fast to adapt to new observation



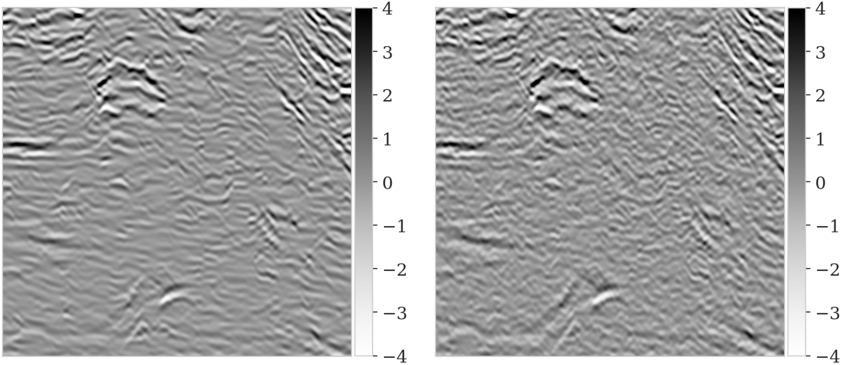

(left) prior, (right) low- and high-fidelity data




(left) w/o vs w/ preconditioning vs MCMC, (right) loss w/ and w/o preconditioning




(left) true model, (right) observed data




conditional mean before (left) and after (right) transfer learning

# Contributions

Take full advantage of existing training data to:

speed-up variational inference

train a learned (conditional) prior density

https://github.com/slimgroup/Software.siahkoohi2021AABIpto