Chapter 12
The Reversible Simulator – a data-driven approach for solving forward and inverse problems

Tristan van Leeuwen

Abstract Inverse problems are prevalent across various fields of science and engineering, with applications spanning medical imaging, materials science, non-destructive testing, astrophysics, climate science, and seismology. These problems share a common goal: estimating a quantity of interest from measurements obtained under specific experimental conditions. Traditionally, simulation and inference have been considered separate tasks. However, recent advancements in conditional variational inference offer a promising approach to integrate these tasks within a single framework. This paper reviews some of these advancements in the context of linear inverse problems, focusing on the use of straightforward generative models for inference and experimental design in computed tomography.

12.1 Introduction

Inverse problems are ubiquitous in science and engineering, with applications including medical imaging, materials science, non-destructive testing, astrophysics, climate science, and seismology. What unifies these applications, is that the aim is to estimate a quantity of interest, u, from measurements, f^δ, which are obtained using certain experimental settings, s. We assume that the data-generating process can be simulated by some combination of deterministic (i.e., PDEs, integral operators, ...) and stochastic (i.e., Gaussian process) models. **Simulating the underlying process often entails solving large-scale PDEs, and is computationally expensive.**

A popular framework for modeling, analyzing and solving inverse problems is the Bayesian one [12, 11]. Here, both the underlying data-generating process and prior information are described by probability distributions. The likelihood, describing the data-generating process is denoted by $\pi_{\text{data}}(f^\delta|s,u)$, while prior information on

Tristan van Leeuwen
Centrum Wiskunde & Informatica, Science Park 123, Amsterdam, Netherlands
e-mail: t.van.leeuwen@cwi.nl

u is encoded by $\pi_{\text{prior}}(u)$. Accordingly, the posterior is denoted by

$$\pi_{\text{post}}(u|s, f^{\delta}) \propto \pi_{\text{data}}(f^{\delta}|s, u)\pi_{\text{prior}}(u). \tag{12.1}$$

In a way, this posterior is the answer to the inverse problem; it captures all relevant prior assumptions and allows one to make probabilistic statements about the corresponding quantity of interest.

In practical applications, one often only considers the maximum a-posteriori probability (MAP) estimate

$$\widehat{u}_{\text{MAP}} = \underset{u}{\operatorname{argmin}} - \log \pi_{\text{post}}(u|s, f^{\delta}). \tag{12.2}$$

The resulting optimization problem is typically solved iteratively, where each iteration requires at least one forward simulation of the system. **This makes MAP-estimation at least an order of magnitude more expensive than forward simulation.**

The related uncertainty is sometimes quantified by considering a Gaussian approximation of the posterior with mean $\widehat{u}_{\text{MAP}}$ and covariance

$$\widehat{\Sigma}_{\text{MAP}} = -\left[\nabla_u^2 \log \pi_{\text{post}}(\widehat{u}_{\text{MAP}}|s, f^{\delta})\right]^{-1}. \tag{12.3}$$

To obtain more accurate and more detailed information on the uncertainty, one would need to generate samples from the posterior (12.1). Despite recent developments in Markov Chain Monte Carlo (MCMC) methods, this is not feasible for large-scale applications. An alternative approach is the so-called randomize-then-optimize approach, which solves the inference problem (12.2) for a perturbed data set. Each sample from the posterior in this approach requires computing a MAP estimate. **This currently makes uncertainty quantification (UQ) for large-scale problems at least an order of magnitude more expensive than MAP-estimation, and hence at least two orders of magnitude more expensive than forward simulation.**

This has given rise to recent research on *variational inference* using generative models, which can efficiently generate samples from complex probability distributions, which were fitted to training data. Early work on this topic includes [1], while more recent approaches are described in [8, 10, 5, 4, 7].

These developments allow us to define a *reversible simulator* that can do forward simulation and inference at the same computational cost. Figure 12.1 shows a schematic depiction of this concept. In principle, this lowers the computational cost of UQ by at least an order of magnitude, though this comes at the cost of an off-line training step. Still, for applications where data are collected routinely under similar circumstances this may prove feasible.

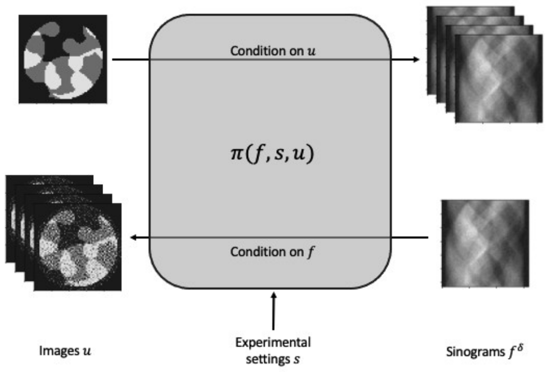


Fig. 12.1: In the forward mode, the Reversible Simulator produces samples from the likelihood; in this case simulating noisy sinograms for a given input image. In the reverse mode, it produces samples from the posterior from which we can compute, for example, the posterior mean and variance.

12.1.1 Approach and challenges

To be able to access the posterior and the likelihood, we need to learn a joint probability density function (pdf) $\pi(f, s, u)$ from examples $\{(f_i, s_i, u_i)\}_{i=1}^N$. This then allows us to simulate measurements by conditioning on (s, u) and to generate samples from the posterior by conditioning it on (f^δ, s).

The main challenges include *i)* including the underlying physics to enable effectively learning the underlying distribution from a limited number of samples, *ii)* parametrization of π to allow for conditional sampling, and *iii)* computational efficiency, to allow for fast simulation and inference.

12.1.2 Contributions and outline

In this paper, we adopt conditional normalizing flows for simulation, inference, and experimental design arising from certain linear inverse problems. In particular, we propose simple architectures for treating linear inverse problems with additive Gaussian noise.

The paper is organized is as follows. In section 12.2, we outline the underlying methodology in more detail. Some illustrative examples are presented in section 12.3. Finally, section 12.4 discusses the results and concludes the paper.

12.2 A reversible simulator

The reversible simulator is represented by an underlying probability distribution $\pi(f, s, u)$. By drawing samples from the conditional distribution for a given (s, u), we can simulate measurements, while conditioning the distribution on a given (f, s) allows us to do inference. Moreover, we can perform Bayesian experimental design by maximizing a utility function over s.

A well-known approach for parametrizing a pdf is through a change of variables

$$\pi(x) = F^{\sharp}\pi_0(x) \equiv \pi_0\left(F(x)\right)|\nabla F(x)|, \tag{12.4}$$

where $F^{\sharp}\pi_0$ denotes the *pull-back* of the reference density π_0, and F is an invertible transport map with inverse $G = F^{-1}$.

Samples from π can be generated as $x = G(y)$, with y drawn from π_0. Here, we denote by x the collection of variables (f, s, u) while the corresponding latent variables are denoted by $y = (g, t, v)$. Likewise, we split the transform and its inverse in 3 blocks corresponding to the 3 variables, $F = (F_g, F_t, F_v)$, $G = (G_f, G_s, G_u)$. Throughout we will assume that the reference density can be written as the product of its marginals, which we will also denote by π_0 in a slight abuse of notation; $\pi_0(y) \equiv \pi_0(g)\pi_0(t)\pi_0(v)$.

Typically, one chooses π_0 to be a standard Normal distribution. In this case, we learn the transport map by minimizing the Kullback–Leibler (KL) divergence between π and $F^{\sharp}\pi_0$:

$$\min_F \mathbb{E}_{x\sim\pi}\, \tfrac{1}{2}\|F(x)\|_2^2 - \log|\nabla F(x)|. \tag{12.5}$$

Having parametrized the mapping in a suitable manner we pose (12.5) as a non-linear optimization problem which we can solve with stochastic gradient descent (SGD). Certain triangular architectures will allow for efficient computation of the required Jacobian, while the invertibility of the transform can be exploited to efficiently compute the gradient of the objective w.r.t. its parameters. See [6] for a recent overview.

The resulting mapping $\widehat{F}$ now represents our reversible simulator. For simulation, we need to sample from the conditional distribution

$$\widehat{\pi}_{\text{like}}(f|s,u) = \frac{\pi_0(\widehat{F}(f,s,u))|\nabla\widehat{F}(f,s,u)|}{\int \pi_0(\widehat{F}(f,s,u))|\nabla\widehat{F}(f,s,u)|\mathrm{d}f}.$$

The procedure for doing this is sketched in Algorithm 1. Note that sampling from this distribution does not require us to compute the normalizing constant. Likewise, samples from the posterior

$$\widehat{\pi}_{\text{post}}(u|s,f) = \frac{\pi_0(\widehat{F}(f,s,u))|\nabla\widehat{F}(f,s,u)|}{\iint \pi_0(\widehat{F}(f,s,u))|\nabla\widehat{F}(f,s,u)|\mathrm{d}s\mathrm{d}u}$$

can be generated according to Algorithm 2.

In both algorithms, the non-linear system of equations can be solved by applying a Newton-like method to $H_{\text{simulation}}(f,t,v) = F(f,s,u) - (g,t,v)$ and $H_{\text{inference}}(g,t,u) = F(f,s,u) - (g,t,v)$ respectively. Alternatively, (accelerated) fixed point iterations can be used if we have access to the inverse map G [3].

Algorithm 1 Simulation

```
Require: $(s,u)$, transport map $F$
Ensure: $f$ drawn from the conditional distribution $\widehat{\pi}(f|s,u)$
  Draw $g$ from $\pi_0$
  Solve for $(f,t,v)$ from $(g,t,v) = F(f,s,u)$
  return $f$, $(t,v)$
```

Algorithm 2 Inference

```
Require: $(f,s)$, transport map $F$
Ensure: $u$ drawn from the posterior distribution $\widehat{\pi}(u|s,f)$
  Draw $v$ from $\pi_0$
  Solve for $(g,t,u)$ from $(g,t,v) = F(f,s,u)$
  return $u$, $(t,g)$
```

Based on these capabilities we can perform Bayesian experimental design (BED) by optimizing some utility function. A well-known example is *D-optimal* design, which aims to maximize the expected information gain (EIG)

$$\max_s \mathbb{E}_{(f,u)\sim\widehat{\pi}_{\text{like}}(f|s,u)\pi_{\text{prior}}(u)} \left[\log\widehat{\pi}_{\text{post}}(u|s,f) - \log\pi_{\text{prior}}(u)\right]. \tag{12.6}$$

where π_{prior} denotes the prior distribution over which we want to optimize the design. An alternative is *A-optimal* design, which minimizes the expected risk:

$$\min_s \mathbb{E}_{\widehat{u}\sim\pi_{\text{post}}(u|s,f),f\sim\pi_{\text{like}}(f|s,u),u\sim\pi_{\text{prior}}} \|\widehat{u}-u\|_2^2. \tag{12.7}$$

In practice, we would estimate the required quantities using a Monte Carlo approximation [9].

12.2.1 Practical implementation with triangular maps

The Algorithms can be greatly simplified if we construct triangular maps (so-called Knothe–Rosenblatt arrangements) [2]. To this end, we introduce an upper and a lower triangular transport map

$$\overrightarrow{F}(f,s,u) = \left(\overrightarrow{F}_g(f,s,u), \overrightarrow{F}_t(s,u), \overrightarrow{F}_v(u)\right),$$

$$\overleftarrow{F}(f,s,u) = \left(\overleftarrow{F}_g(f), \overleftarrow{F}_t(f,s), \overleftarrow{F}_v(f,s,u)\right).$$

A nice feature is that the variational problem (12.5) now separates over the individual components of the triangular maps. Since we are only interested in $\overrightarrow{F}_g$ (for simulation), $\overleftarrow{F}_v$ (for inference), and $\overrightarrow{F}_v$ (for prior sampling), we only need to learn these components by solving

$$\min_{\overrightarrow{F}_g} \mathbb{E}_{(f,s,u)\sim\pi}\, \tfrac{1}{2}\|\overrightarrow{F}_g(f,s,u)\|_2^2 - \log|\nabla_f \overrightarrow{F}_g(f,s,u)|,$$

$$\min_{\overleftarrow{F}_v} \mathbb{E}_{(f,s,u)\sim\pi}\, \tfrac{1}{2}\|\overleftarrow{F}_v(f,s,u)\|_2^2 - \log|\nabla_u \overleftarrow{F}_v(f,s,u)|.$$

$$\min_{\overrightarrow{F}_v} \mathbb{E}_{u\sim\pi}\, \tfrac{1}{2}\|\overrightarrow{F}_v(u)\|_2^2 - \log|\nabla_u \overleftarrow{F}_v(u)|.$$

The estimated conditional distributions needed for simulation and inference are now given by

$$\widehat{\pi}_{\text{like}}(f|s,u) = \pi_0(\overrightarrow{F}_g(f,s,u))|\nabla_f \overrightarrow{F}_g(f,s,u)|,$$

$$\widehat{\pi}_{\text{post}}(u|s,f) = \pi_0(\overleftarrow{F}_v(f,s,u))|\nabla_u \overleftarrow{F}_v(f,s,u)|.$$

Likewise, the prior is given by

$$\widehat{\pi}_{\text{prior}}(u) = \pi_0(\overrightarrow{F}_v(u))|\nabla_u \overrightarrow{F}_v(u)|.$$

The upper triangular map can be used for simulation and reduces Algorithm 1 to solving f from

$$\overrightarrow{F}_g(f,s,u) = g, \quad \text{with } g \text{ drawn from } \pi_0,$$

since this arrangement ensures that the map $f \mapsto \overrightarrow{F}_g(f,s,u)$ is invertible. Likewise, the lower triangular map can be used for inference, reducing Algorithm 2 to solving u from

$$\overleftarrow{F}_v(f,s,u) = v, \quad \text{with } v \text{ drawn from } \pi_0,$$

which is facilitated by the fact that the map $u \mapsto \overleftarrow{F}_v(f,s,u)$ is guaranteed to be invertible.

12.2.2 An architecture for linear inverse problems

A template for linear inverse problems is the additive Gaussian noise model with a Gaussian prior:

$$f \sim \mathcal{N}(Ku, \Sigma_{\text{like}}), \quad u \sim \mathcal{N}(\mu_{\text{prior}}, \Sigma_{\text{prior}}),$$

with $K \in \mathbb{R}^{m\times n}$ denoting the forward operator. The resulting posterior is a normal distribution with mean and covariance

$$\mu_{\text{post}} = \mu_{\text{prior}} + \Sigma_{\text{post}} K^T \Sigma_{\text{like}}^{-1} \left(f - K\mu_{\text{prior}}\right), \quad \Sigma_{\text{post}} = \left(K^T \Sigma_{\text{like}}^{-1} K + \Sigma_{\text{prior}}^{-1}\right)^{-1}.$$

The parameter s appears implicitly in Σ_{like} and/or K. The *true* transport maps would in this case be given by

$$\overrightarrow{F}_g(f,s,u) = \Sigma_{\text{like}}^{-1/2}\left(f - Ku\right), \tag{12.8}$$

$$\overleftarrow{F}_v(f,s,u) = \Sigma_{\text{post}}^{-1/2}\left(u - \mu_{\text{prior}}\right) + \Sigma_{\text{post}}^{1/2} K^T \Sigma_{\text{like}}^{-1}(f - K\mu_{\text{prior}}), \tag{12.9}$$

$$\overrightarrow{F}_v(u) = \Sigma_{\text{prior}}^{-1/2}\left(u - \mu_{\text{prior}}\right). \tag{12.10}$$

We propose the following architecture for the transport maps, which preserve these linear relationships.

$$\overrightarrow{F}_g(f,s,u) = \overrightarrow{A}_{gf}(s)f + \overrightarrow{A}_{gu}(s)u + \overrightarrow{b}_g(s), \tag{12.11}$$

$$\overleftarrow{F}_v(f,s,u) = \overleftarrow{A}_{vf}(s)f + \overleftarrow{A}_{vu}(s)u + \overleftarrow{b}_v(s), \tag{12.12}$$

$$\overrightarrow{F}_v(u) = \overrightarrow{A}_{vu}u + \overrightarrow{b}_v. \tag{12.13}$$

Here, $\overrightarrow{A}_{vu} \in \mathbb{R}^{n\times n}$, $\overrightarrow{A}_{gf}(s) \in \mathbb{R}^{m\times m}$ are invertible upper triangular matrices, $\overleftarrow{A}_{vu}(s) \in \mathbb{R}^{n\times n}$ is an invertible lower triangular matrix. The dependency of the weights on s is adapted to the application at hand, a few examples of which are shown in a later section.

12.3 Numerical examples

The following examples illustrate the workings of the reversible simulator.

12.3.1 A Gaussian example

For the first example, we consider the following data generating process

$$f^\delta \sim \mathcal{N}(Ku, s^2 I),\ u \sim \mathcal{N}(\mu_{\text{prior}}, \Sigma_{\text{prior}}), \tag{12.14}$$

with

$$K = \begin{pmatrix} 1 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{pmatrix}, \quad \mu_{\text{prior}} = \mathbf{1}, \quad \Sigma_{\text{prior}} = 0.1I.$$

12.3.1.1 Parametrizing the transport maps

We define the transport maps as

$$\begin{aligned}\overrightarrow{F}_g(f, s, u) &= s^{-1}(\overrightarrow{A}_{gf}f + \overrightarrow{A}_{gu}u + \overrightarrow{b}_g),\\ \overleftarrow{F}_v(f, s, u) &= s^{-1}(\overleftarrow{A}_{vf}f + \overleftarrow{A}_{vu}u + \overleftarrow{b}_v),\\ \overrightarrow{F}_v(u) &= \overrightarrow{A}_{vu}u + \overrightarrow{b}_v,\end{aligned}$$

which matches the expected behavior as $s \downarrow 0$ (cf. (12.11)-(12.12)).

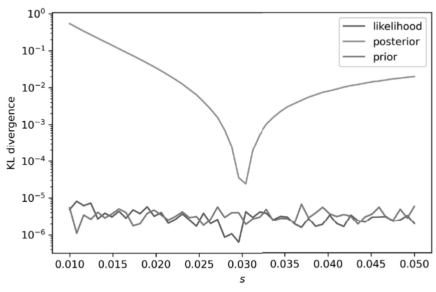
12.3.1.2 Training

We train the transport maps on training data ($N = 10^6$) generated for a fixed value of $s = 0.03$, allowing us to compute the corresponding transport maps directly from the mean and co-variance of the training data. To stabilize the process, we use a shrinkage estimator of the covariance with $\alpha = 10^{-6}$ (see the Appendix for more details).

12.3.2 Validation

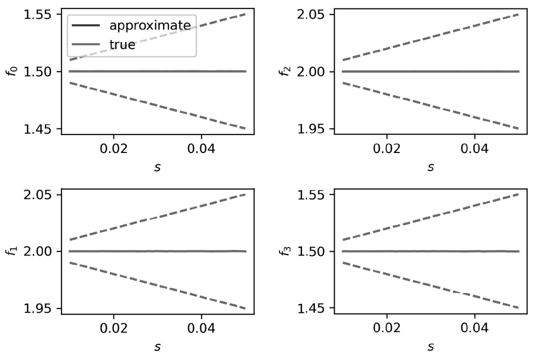
To validate the transport maps, we apply it to test data ($N = 10^6$) generated for $s \in [0.01, 0.05]$. The KL-divergence between the estimated and true likelihood, posterior and prior are shown in 12.2. The errors behave as expected; the prior and likelihood fit with the chosen parametrization of the mappings and give a small error while the error in the posterior is generally larger, with a minimum for $s = 0.03$.

Fig. 12.2: KL-divergence between the estimated and true densities.

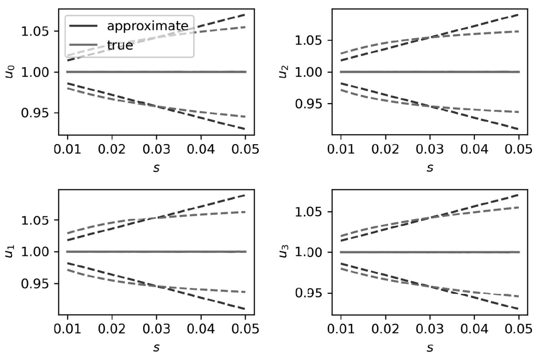


A different view on the error in the likelihood and posterior is shown in figure 12.3. Here, we show the mean and standard deviation of samples of the likelihood and posterior corresponding to $u = \mu_{\text{prior}}$ and $f = K\mu_{\text{prior}}$ respectively. We note that the main source of error is the standard deviation of the posterior samples, stemming from the chosen parametrization of the mappings.

Fig. 12.3: Mean and standard deviation of samples generated from the likelihood (a) and posterior (b).



(a)



(b)

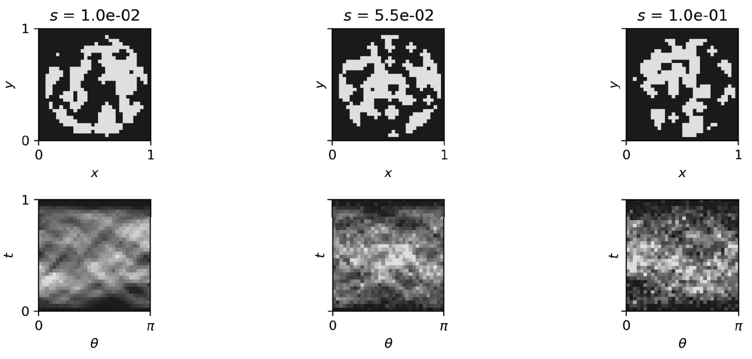
12.3.3 Computed tomography I

For a more realistic example, we consider computed tomography. Here, the measurements are given by

$$f^\delta = -\log\left(s^2 I^\delta\right),\ I^\delta \sim \mathcal{P}(s^{-2}\exp(-Ku)), \tag{12.15}$$

where $u \in \mathbb{R}^{n^2}$ are randomly generated $n \times n$ binary images, $K \in \mathbb{R}^{n^2 \times n^2}$ is a discretization of the Radon transform for a fixed acquisition geometry with n detector pixels and n equispaced angles, and $\mathcal{P}$ denotes the Poisson distribution. The parameter s in this case controls the intensity of the X-ray beam: $I_0 = s^{-2}$. An example of the corresponding data is shown in figure 12.4.

Fig. 12.4: The data consists of images images (top) and their corresponding sinograms (bottom) for $s \in [0.01, 0.1]$.



In the high-dose limit, we can approximate the corresponding likelihood with a Normal distribution with mean Ku and covariance $\mathrm{diag}(s^2 \exp(Ku))$. We define the transport map accordingly in the same way as in the previous example.

12.3.3.1 Training

We train the transport maps on data ($n = 32, N = 10^4$) for $s = 0.055$ with $\alpha = 10^{-6}$ using the procedure described in the Appendix.

12.3.4 Validation

We apply the trained mapping to validation data generated for $s \in [0.01, 0.1]$. The KL divergence is shown in figure 12.5. We note that the error in the prior is smallest, and that the error in the likelihood gets smaller for larger s.

To get more insight in the learned map, we compare samples from the learned likelihood and prior to samples of the true likelihood and prior. Figure 12.6 shows examples samples drawn from the estimated prior. Figure 12.7 shows the mean and variance of samples from the estimated and true likelihood for a given phantom. We see a good correspondence of the mean, while the intensity-dependent variance of the true likelihood is not well-captured as expected. Figure 12.8 shows a summary statistic of the samples drawn from the estimated and true likelihood for varying values of $s \in [0.01, 0.1]$, showing a good correspondence with increasing error for larger s.

In figure 12.9 we show an example of the posterior mean and variance for a given sinogram. Compared to the true phantom the standard deviation provides a useful indication of the error as shown in figure 12.10.

Fig. 12.5: KL divergence between the estimated and true densities.

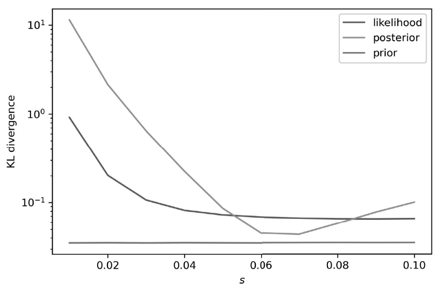
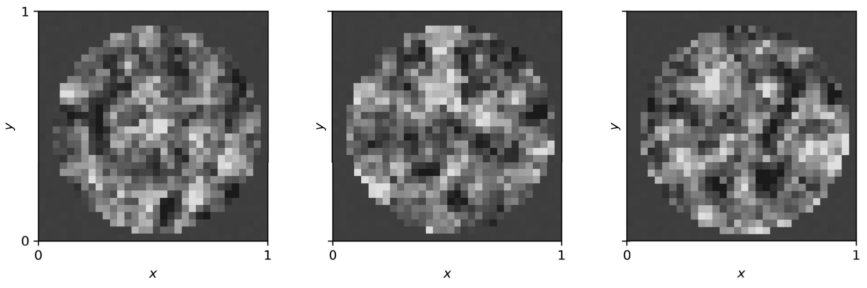


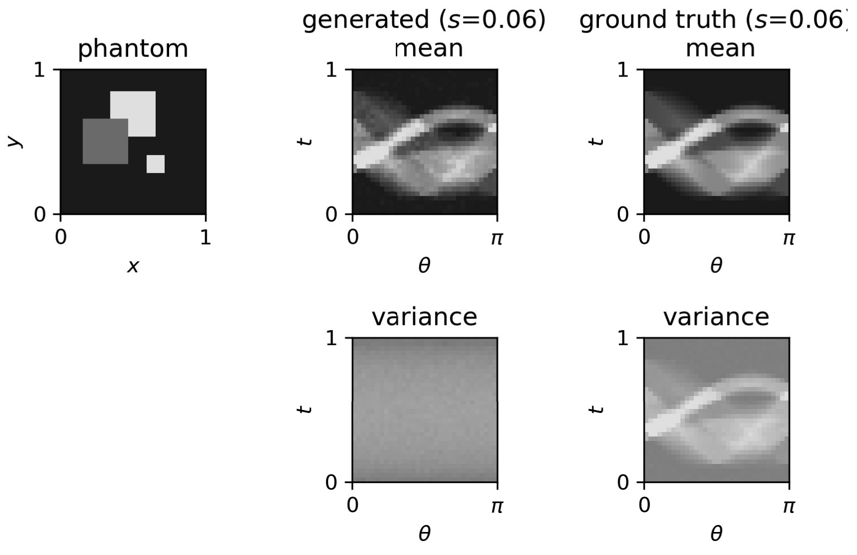
Fig. 12.6: A few generated prior samples.



12.3.5 Computed tomography II

We follow a similar setup as the previous example, except here we fix the noise level and let $s \in \mathbb{R}^m$ denote the $m \leq n$ angles at which the data are acquired:

Fig. 12.7: Comparison in terms of mean and variance of the simulated sinograms to samples of the true likelihood for a phantom consisting of 3 squares.



$$f^{\delta} = -\log\left(I^{\delta}/I_0\right),\ I^{\delta} \sim \mathcal{P}(I_0 \exp(-K(s)u)). \tag{12.16}$$

For moderate noise levels, the likelihood again can be well-approximated by a normal distribution. To account for the non-trivial dependence on s we propose the following parametrization of the transport maps:

$$\overrightarrow{F}_g(f,s,u) = \overrightarrow{A}_{gf} f + \overrightarrow{A}_{gu} K(s)u + \overrightarrow{b}_g,$$

$$\overleftarrow{F}_v(f,s,u) = \overleftarrow{A}_{vf} K(s)^T f + \overleftarrow{A}_{vf} u + \overleftarrow{b}_v.$$

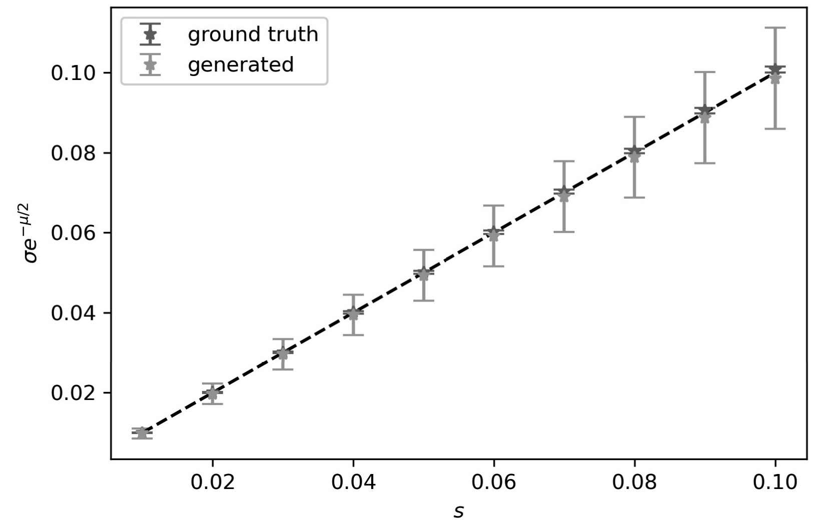
$$\overrightarrow{F}_v(u) = \overrightarrow{A}_{vu} u + \overrightarrow{b}_v.$$

In effect, in known physics is included explicitly through K and K^T [5].

12.3.5.1 Training

We follow the same training procedure as outline above, except for $\overrightarrow{F}_g$ we use pairs $(f_i, K(s)u_i)$ (cf. figure 12.11) while for $\overleftarrow{F}_v$ we use pairs $(K(s)^T f_i, u_i)$ (cf. figure 12.12). For the training we fix s to represent n regularly spaced angles in $[0,\pi]$

Fig. 12.8: Here we plot $\sigma e^{-\mu/2}$, with μ, σ denoting the mean and variance of the likelihood samples as a function of s. For the used likelihood, we expect this summary statistic to follow a straight line.



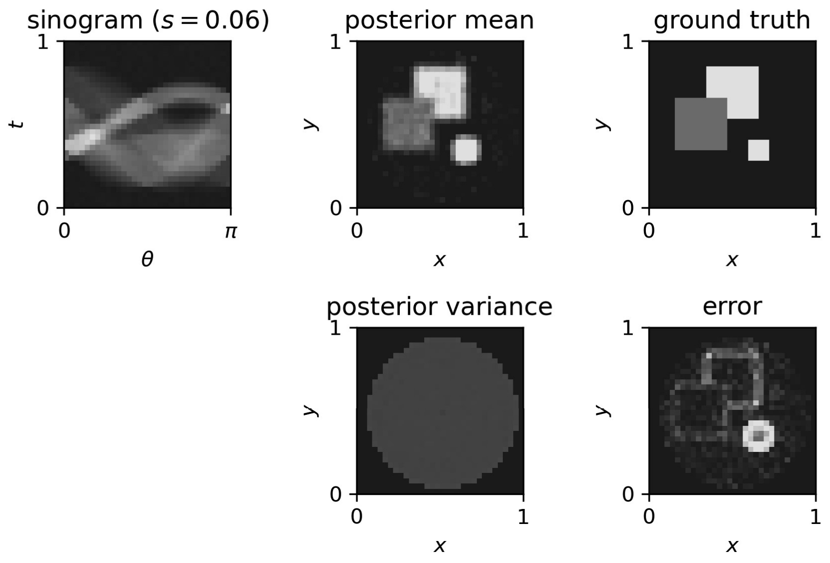
12.3.5.2 Validation

To highlight the ability to perform experimental design with $\overleftarrow{F}_v$, we use it to compute the reconstruction error between a ground-truth image (cf. figure 12.9) and the posterior mean, as a function of s. This corresponds to A-optimal experimental design (cf. equation 12.7). The chosen architecture allows to evaluate the posterior mean for any number of angles, so we can plot the error as we increase the number of angles. In particular, we consider three scenarios: adding angles sequentially (i.e., $s = (0), s = (0, \Delta s), s = (0, \Delta s, 2\Delta s)$, etc.) with $\Delta s = \pi/n$ (incremental) or $\Delta s = 0.618\pi$ (golden ratio), and a greedy procedure where each next angle is chosen amongst all possible candidates to minimize the reconstruction error. The result is shown in figure 12.13. We note that the error decreases fasted for the greedy procedure, followed closely by the golden ratio approach.

12.4 Conclusion and Discussion

We reviewed the use of (triangular) normalizing flows for simulation, inference, and experimental design in the context of linear inverse problems. In a few case studies

Fig. 12.9: Example of the posterior mean and variance for a given sinogram, as compared to the underlying true phantom and the corresponding reconstruction error.



we have shown that simple affine transformations can yield good approximations for linear inverse problems, even when we depart from additive Gaussian noise. The proposed architectures involve affine transformations and are amenable to application on large-scale inverse problems. In particular, tools from numerical linear algebra can be brought to bear on this problem and yield efficient ways to approximate the affine transformations. We expect that further refinement of the results can be achieved by composing the affine transformations with (point-wise) non-linear operations.

Acknowledgements

I want to express my gratitude to Gabrio Rizutti, Felix Herrmann and Rafael Orozco with whom I have had numerous fruitful discussions on this topic in the past years.

Fig. 12.10: Detailed view of the posterior mean and standard deviation as compared to the true phantom.

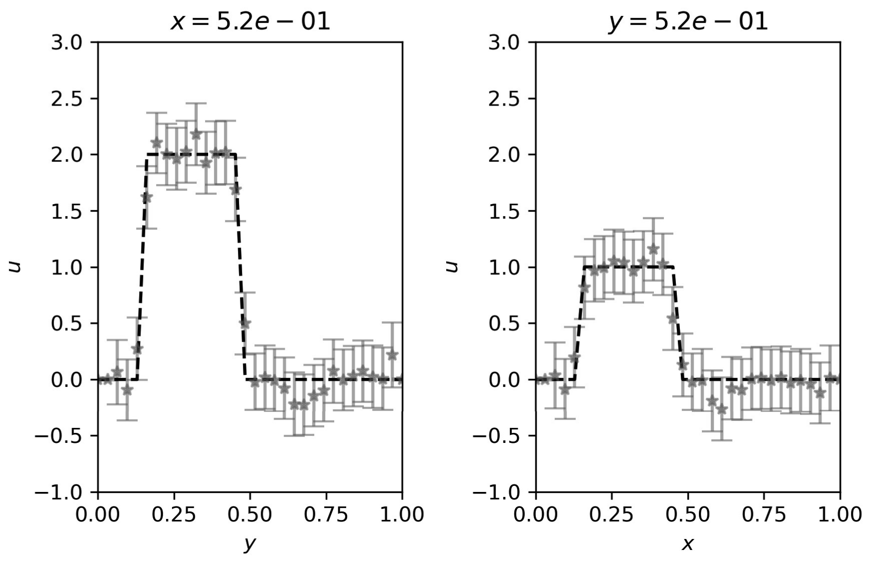


Fig. 12.11: The data consists of forward projected images (top) and their corresponding noisy sinograms (bottom).

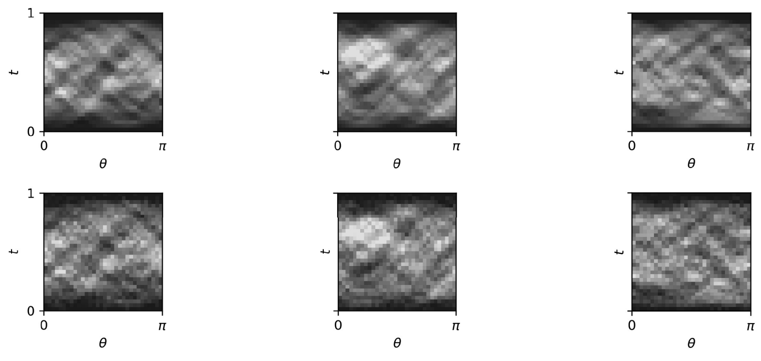


Fig. 12.12: The data consists of images (top) and their corresponding backprojected sinograms (bottom).

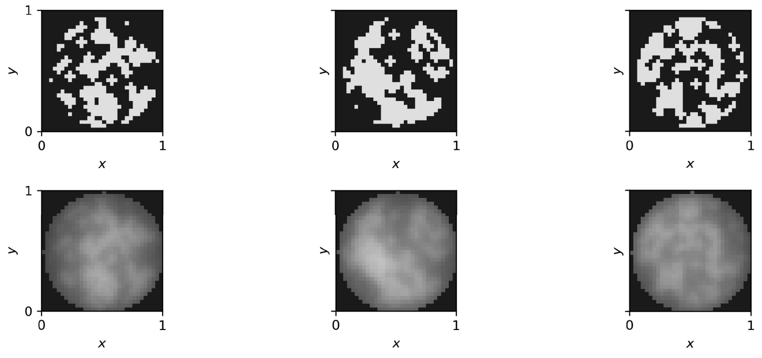
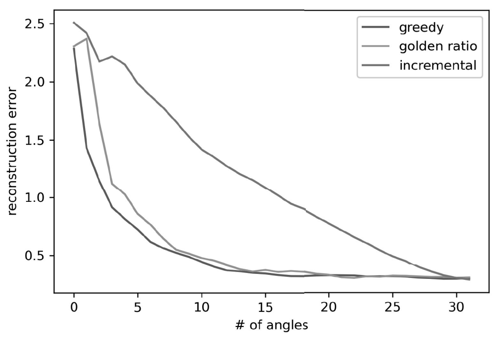


Fig. 12.13: Reconstruction error as a function of the number of angles for three angle-selection procedures.



References

1. Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*, 2018.
2. Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. An introduction to sampling via measure transport. 2 2016.
3. Vincent Moens, Aivar Sootla, Haitham Bou Ammar, and Jun Wang. Viscos Flows: Variational Schur Conditional Sampling With Normalizing Flows. pages 1–18, 2021.
4. Rafael Orozco, Mathias Louboutin, Ali Siahkoohi, Gabrio Rizzuti, Tristan van Leeuwen, and Felix Herrmann. Amortized Normalizing Flows for Transcranial Ultrasound with Uncertainty Quantification. (2020):1–18, 2023.
5. Rafael Orozco, Ali Siahkoohi, Gabrio Rizzuti, Tristan van Leeuwen, and Felix Herrmann. Adjoint operators enable fast and amortized machine learning based Bayesian uncertainty

quantification. page 56, 2023.
6. George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.
7. Stefan T Radev, Marvin Schmitt, Valentin Pratz, Umberto Picchini, Ullrich Köthe, and Paul Christian Bürkner. JANA: Jointly Amortized Neural Approximation of Complex Bayesian Models. In *Proceedings of Machine Learning Research*, volume 216, pages 1695–1706, 2023.
8. Stefan T. Radev, Andreas Voss, Ulf K. Mertens, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning complex stochastic models with invertible neural networks. *arXiv*, 2020.
9. Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern Bayesian Experimental Design. (1), 2023.
10. Dominik Strutz and Andrew Curtis. Variational bayesian experimental design for geophysical applications. 7 2023.
11. A. M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, may 2010.
12. Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. Society for Industrial Mathematics, 2005.

Appendix

Training and validation

The triangular maps arise by fitting a Gaussian to the data $\{(f_i, u_i)\}_{i=1}^N$, and computing a Cholesky decomposition of the Schur complements of the resulting covariance matrix. The core of this involves estimating the mean and covariance of a set of samples, and validating the fit.

Training

Given a set of n−dimensional training data $\{x_i\}_{i=1}^N$ drawn from π, we aim to fit a Normal distribution to it. The resulting mean and (shrinkage) covariance estimates are

$$\widehat{\mu} = N^{-1}\sum_{i=1}^N x_i,$$

$$\widehat{\Sigma}_\alpha = (1-\alpha)\widehat{\Sigma} + \alpha n^{-1}\mathrm{trace}(\widehat{\Sigma})I,$$

with

$$\widehat{\Sigma} = N^{-1}\sum_{i=1}^N (x_i - \widehat{\mu})(x_i - \widehat{\mu})^T.$$

The corresponding mapping is then defined as

$$F(x) = \widehat{\Sigma}_\alpha^{-1/2}(x - \widehat{\mu}),$$

where the matrix square root is to be interpreted in terms of a Cholesky decomposition, leading an upper or lower triangular matrix.

Validation

To validate the trained mapping we would ideally like to compute the difference between the true distribution π and its estimated counter part.

If we have access to the underlying true pdf π we could compute the KL-divergence

$$\mathrm{KL}(\pi, F^{\sharp}\pi_0) \approx N^{-1}\sum_{i}^{N} \log \pi(x_i) - \log(\pi_0(F(x_i))|\nabla F(x_i)|),$$

with $x_i \sim \pi$ representing the validation data. We can alternatively compute this as

$$\mathrm{KL}(\pi, F^{\sharp}\pi_0) = \mathrm{KL}(F_{\sharp}\pi, \pi_0) \approx N^{-1}\sum_{i}^{N} \log \pi(G(y_i)|\nabla G(y_i)|) - \log \pi_0(y_i),$$

with $y_i = F(x_i)$. When the distributions involved are Gaussian (i.e., $\pi = \mathcal{N}(\mu, \Sigma)$, $\pi_0 = \mathcal{N}(0, I)$), we can compute this as

$$\mathrm{KL}(\pi, F^{\sharp}\pi_0) \approx \tfrac{1}{2}\left(\|\widehat{\mu}_0\|_2^2 + \mathrm{trace}\left(\widehat{\Sigma}_0\right) - n - \log|\widehat{\Sigma}_0|\right), \tag{12.17}$$

where $\widehat{\mu}_0, \widehat{\Sigma}_0$ are the sample mean and covariance of the normalized samples $\{y_i\}_{i=1}^{N}$. This effectively tests how far (in distribution) the normalized samples y_i are to π_0.

Even when π is not a normal distribution, this is a useful way of measuring how well the normalized samples fit π_0 as in that case we expect $\widehat{\mu}_0 \approx \mathbf{0}$, $\widehat{\Sigma}_0 \approx \boldsymbol{I}$.