Centrum Wiskunde & Informatica

A reversible simulator

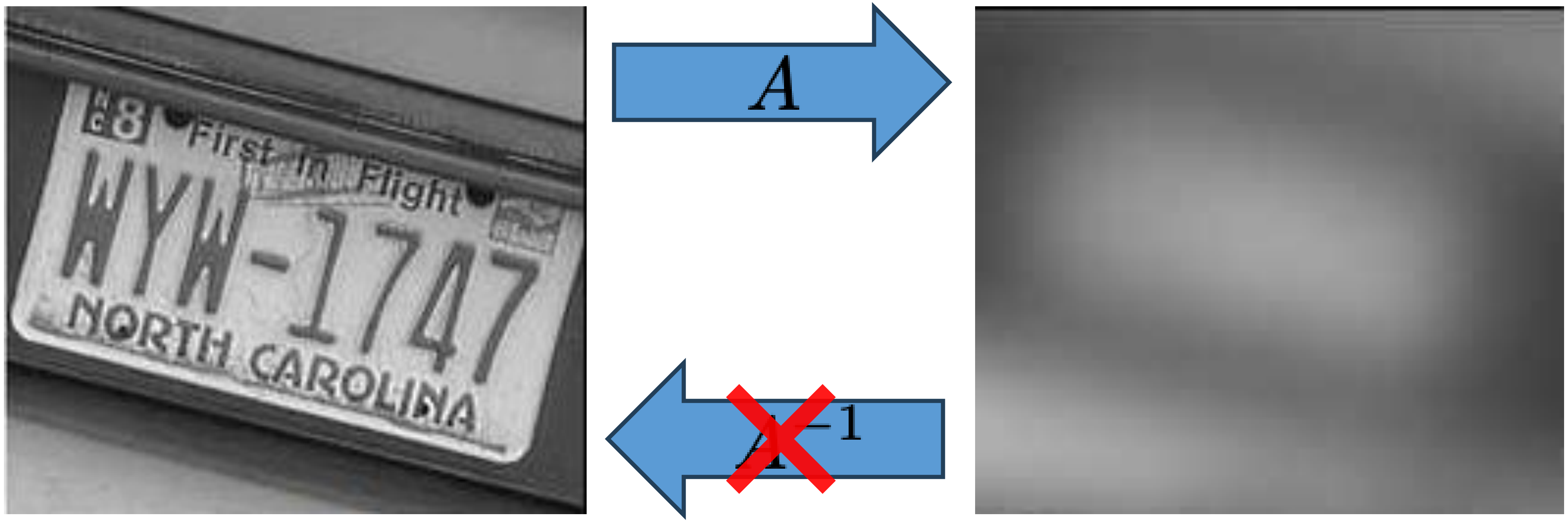
Generative modelling for forward and inverse problems

Tristan van Leeuwen

Christoph Brune, Marcello Carioni (U. Twente)
Felix Herrmann, Shiqin Zeng (Georgia Institute of Technology)

Maths4DL Conference on Inverse Problems and Deep Learning 7-9 July 2025, University of Bath


NC 8
First in Flight
WYW-1747
NORTH CAROLINA


NC 8
First in Flight
WYW-1747
NORTH CAROLINA
A
$\cancel{A^{-1}}$

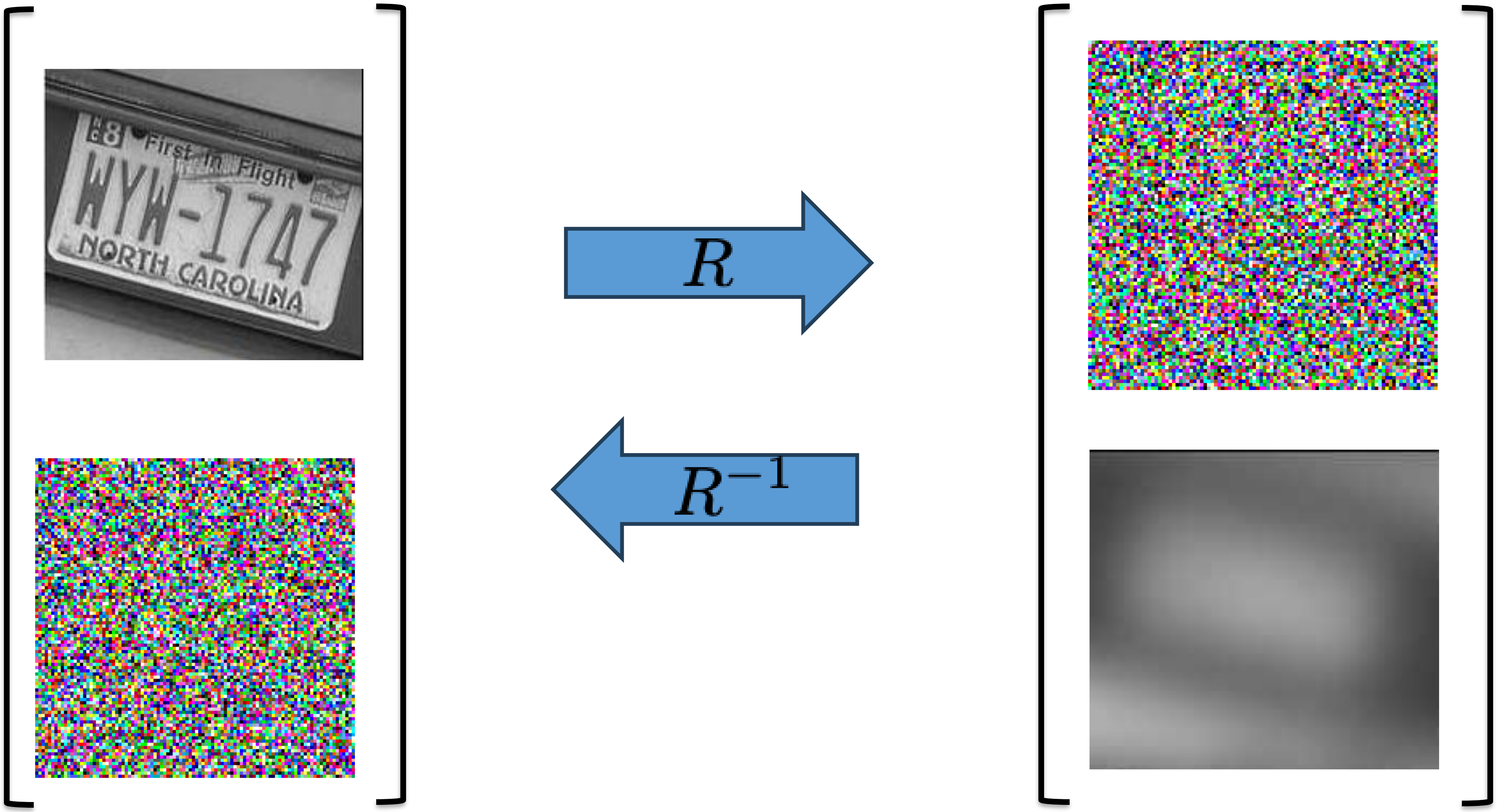
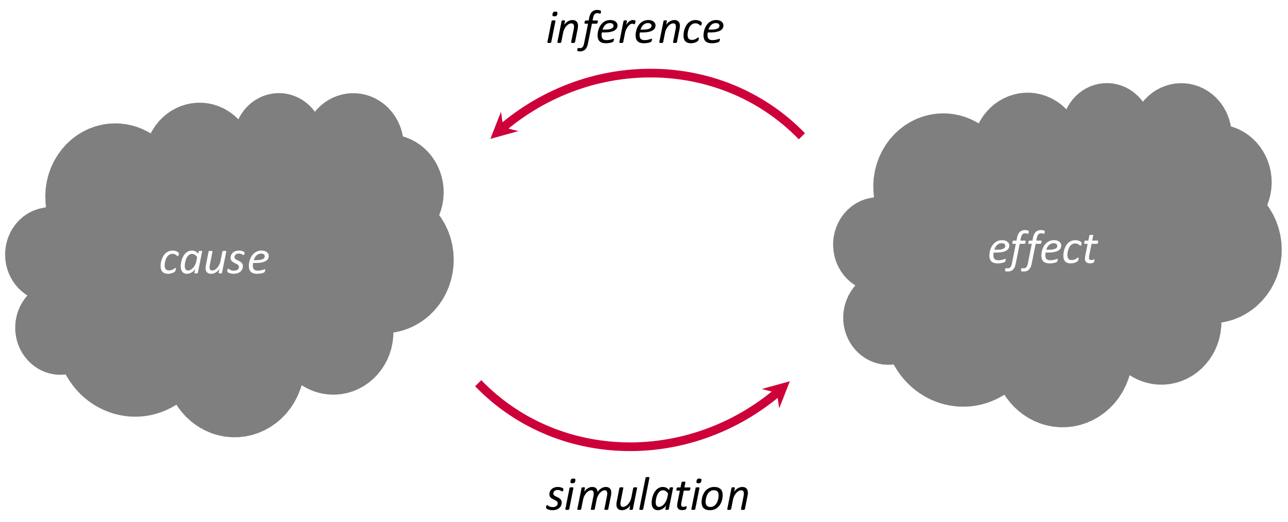

First in Flight
WYW-1747
NORTH CAROLINA
R
R^{-1}

Table of contents

1. The inverse problem
2. A generative model for simulation and inference
3. Some numerical results
4. Wrap-up

The inverse problem



Inverse Problems & Bayesian Statistics

We are given noisy and incomplete measurements

$$\mathbf{f}^\delta \approx \overline{\mathbf{f}}$$

and a model

$$\overline{\mathbf{f}} = K\overline{\mathbf{u}}$$

The goal is to obtain an estimate $\mathbf{u}^\delta$ of $\overline{\mathbf{u}}$

Inverse Problems & Bayesian Statistics

Formulate a *likelihood* to describe the measurements

$$\pi_{\text{like}}(\mathbf{f}|\mathbf{u})$$

a *prior* to capture assumptions on $\mathbf{u}$

$$\pi_{\text{prior}}(\mathbf{u})$$

and use these to construct the *posterior*

$$\pi_{\text{post}}(\mathbf{u}|\mathbf{f}) = c \cdot \pi_{\text{like}}(\mathbf{f}|\mathbf{u})\pi_{\text{prior}}(\mathbf{u})$$

Inverse Problems & Bayesian Statistics

One often computes a MAP estimate

$$\mathbf{u}_{\mathrm{MAP}} = \arg\max_{\mathbf{u}} \pi_{\mathrm{post}}(\mathbf{u}|\mathbf{f})$$

and sometimes characterizes local uncertainty by inspecting the

“Fisher information matrix”

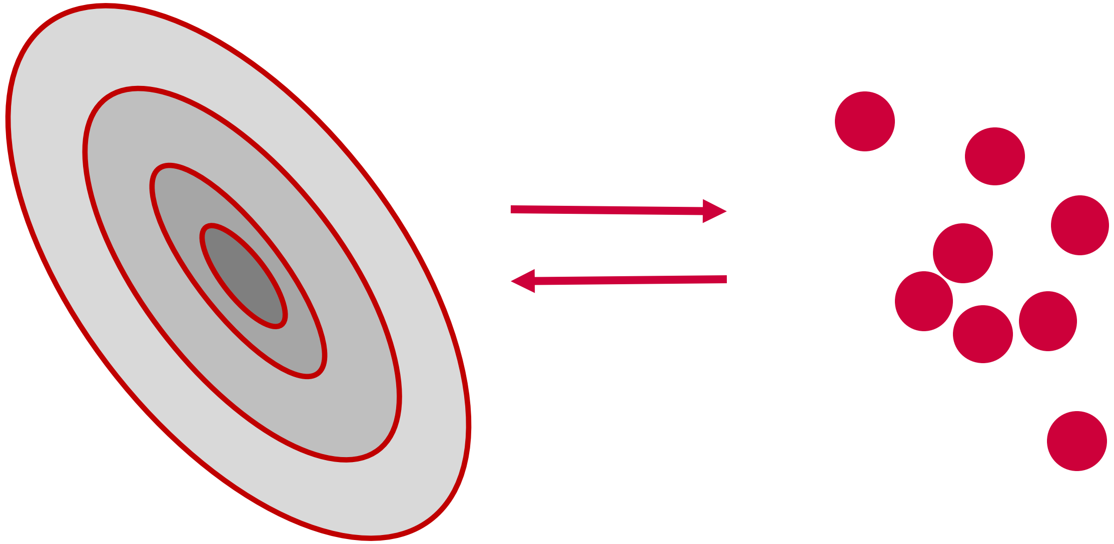
$$-\partial_{\mathbf{u}_i}\partial_{\mathbf{u}_j} \log \pi_{\mathrm{post}}(\mathbf{u}|\mathbf{f})$$

Inverse Problems & Bayesian Statistics

The posterior is the answer to the inverse problem, capturing all prior assumptions and allowing us to give a probabilistic answer.

Both modelling and sampling are challenge tasks and generative models have shown to be a promising tool for both

A generative model for simulation and inference



The basic idea

- If we learn the joint probability $\pi(\mathbf{u}, \mathbf{f})$, we can use it for simulation and inference by conditioning either variable

- By taking system-specific parameters in to account we can make the approach more generic

- Once trained, simulation and inference have similar computational cost, and the model can be used to give probabilistic answers

Implementation using NFs

Represent the sought distribution as

$$\pi(\mathbf{u},\mathbf{f}) = \pi_0(T(\mathbf{u},\mathbf{f}))|\nabla T(\mathbf{u},\mathbf{f})|$$

where T is obtained by solving

$$\min_{T} \mathbb{E}_{(\mathbf{u},\mathbf{f})} \tfrac{1}{2}\|T(\mathbf{u},\mathbf{f})\|_2^2 - \log|\nabla T(\mathbf{u},\mathbf{f})|$$

which requires access to paired samples $\{(\mathbf{u}_i,\mathbf{f}_i)\}_{i=1}^N$

See e.g., Marzouk, 2017

Implementation using NFs

It suffices to consider triangular maps

$$T(\mathbf{u}, \mathbf{f}) = (T_X(\mathbf{u}), T_Y(\mathbf{u}, \mathbf{f}))$$

or

$$T(\mathbf{u}, \mathbf{f}) = (T_X(\mathbf{u}, \mathbf{f}), T_Y(\mathbf{f}))$$

See e.g., Marzouk, 2017

Implementation using NFs

Having trained a lower and upper triangular map, we can approximate the likelihood and prior as

$$\pi_{\text{like}}(\mathbf{f}|\mathbf{u}) = \pi_0(T_{\text{like}}(\mathbf{f};\mathbf{u}))|\nabla T_{\text{like}}(\mathbf{f};\mathbf{u})|$$
$$\pi_{\text{post}}(\mathbf{u}|\mathbf{f}) = \pi_0(T_{\text{post}}(\mathbf{u};\mathbf{f}))|\nabla T_{\text{post}}(\mathbf{u};\mathbf{f})|$$

and sample from them as

$$\mathbf{f} = T_{\text{like}}^{-1}(\mathbf{y};\mathbf{u}), \quad \mathbf{y} \sim \pi_0$$
$$\mathbf{u} = T_{\text{post}}^{-1}(\mathbf{x};\mathbf{f}), \quad \mathbf{x} \sim \pi_0$$

See e.g, Radev et. al. 2022, 2023

Implementation using NFs

We can combine these two maps into an invertible mapping

$$(\mathbf{x}, \mathbf{f}) = S(\mathbf{u}, \mathbf{y})$$

with

$$S(\mathbf{u}, \mathbf{y}) = \left(T_{\text{post}}(\mathbf{u}; T_{\text{like}}^{-1}(\mathbf{y}; \mathbf{u})), T_{\text{like}}^{-1}(\mathbf{y}; \mathbf{u})\right)$$

$$S^{-1}(\mathbf{x}, \mathbf{f}) = \left(T_{\text{post}}^{-1}(\mathbf{x}; \mathbf{f}), T_{\text{like}}(\mathbf{f}; T_{\text{post}}^{-1}(\mathbf{x}; \mathbf{f}))\right)$$

A reversible simulator

- This mapping constitutes a data-driven reversible simulator, which can be used for stochastic simulation and inference

- We can include additional parameters (e.g., noise level, sampling)

$$(\mathbf{x}, \mathbf{f}) = S(\mathbf{u}, \mathbf{y}; \mathbf{s})$$

- Knowledge about the physics can be included in the architecture

$$T_{\text{like}}(\mathbf{f}; \mathbf{u}) = \widetilde{T}_{\text{like}}(\mathbf{f}; K\mathbf{u})$$
$$T_{\text{post}}(\mathbf{u}; \mathbf{f}) = \widetilde{T}_{\text{post}}(\mathbf{u}; K^{\top}\mathbf{f})$$

See e.g., Adler & Öktem 2018, Fluri et. al. 2021, Orozco et. al. 2022

A reversible simulator

This mapping achieves

$$S^{\#}\pi_X\pi_F = \pi_U\pi_Y$$

so it is tempting to train it directly by solving e.g.

$$\min_S \mathrm{KL}(S^{\#}\pi_X\pi_F, \pi_U\pi_Y)$$

but this has trivial 'block diagonal' solutions

$$S(\mathbf{u}, \mathbf{y}) = (S_X(\mathbf{u}), S_F(\mathbf{y}))$$

Some examples

Toy example

Consider a linear inverse problem with zero-mean Gaussian prior and Likelihood

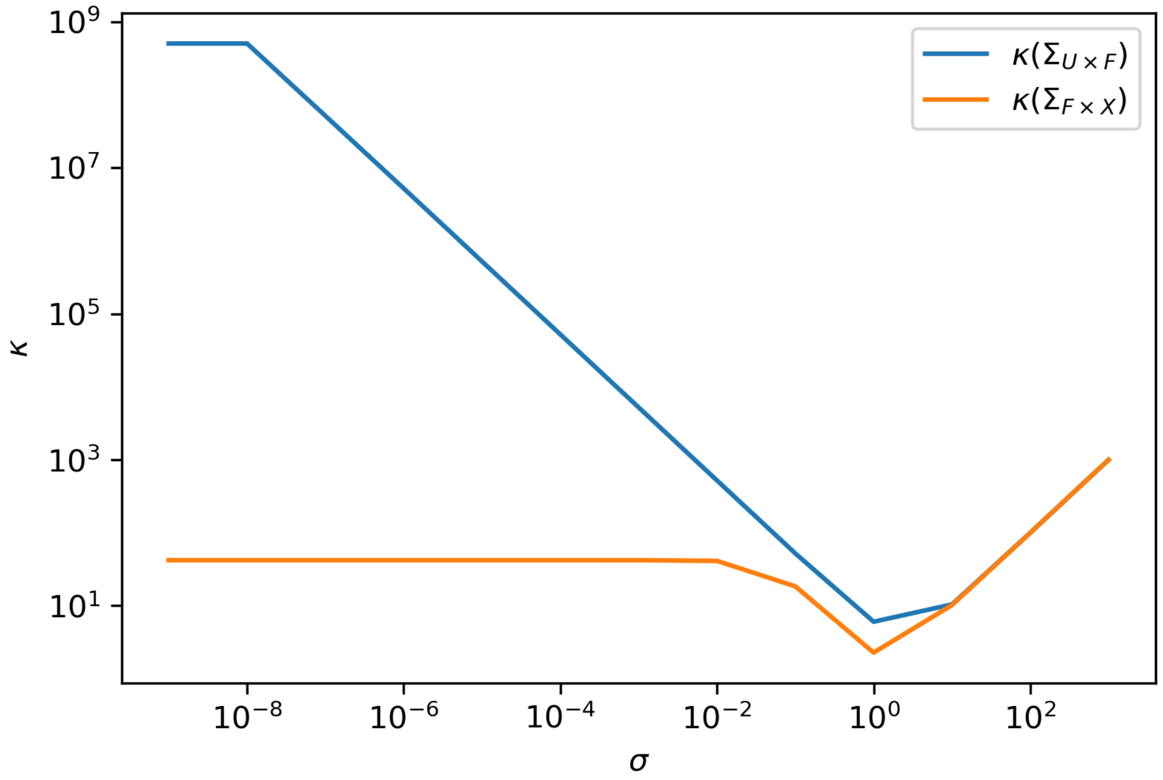
$$\Sigma_{UF} = \begin{pmatrix} \Sigma_u & \Sigma_u K^\top \\ K\Sigma_u & K\Sigma_u K^\top + \Sigma_f \end{pmatrix}$$

The corresponding reversible simulator is represented by

$$S = \begin{pmatrix} \Sigma_{\text{post}}^{1/2}\Sigma_u^{-1} & -\Sigma_{\text{post}}^{1/2} K^\top \Sigma_f^{-1/2} \\ K & \Sigma_f^{1/2} \end{pmatrix},$$

Toy example

Conditioning of the reversible simulator with $\Sigma_f = \sigma^2 I$

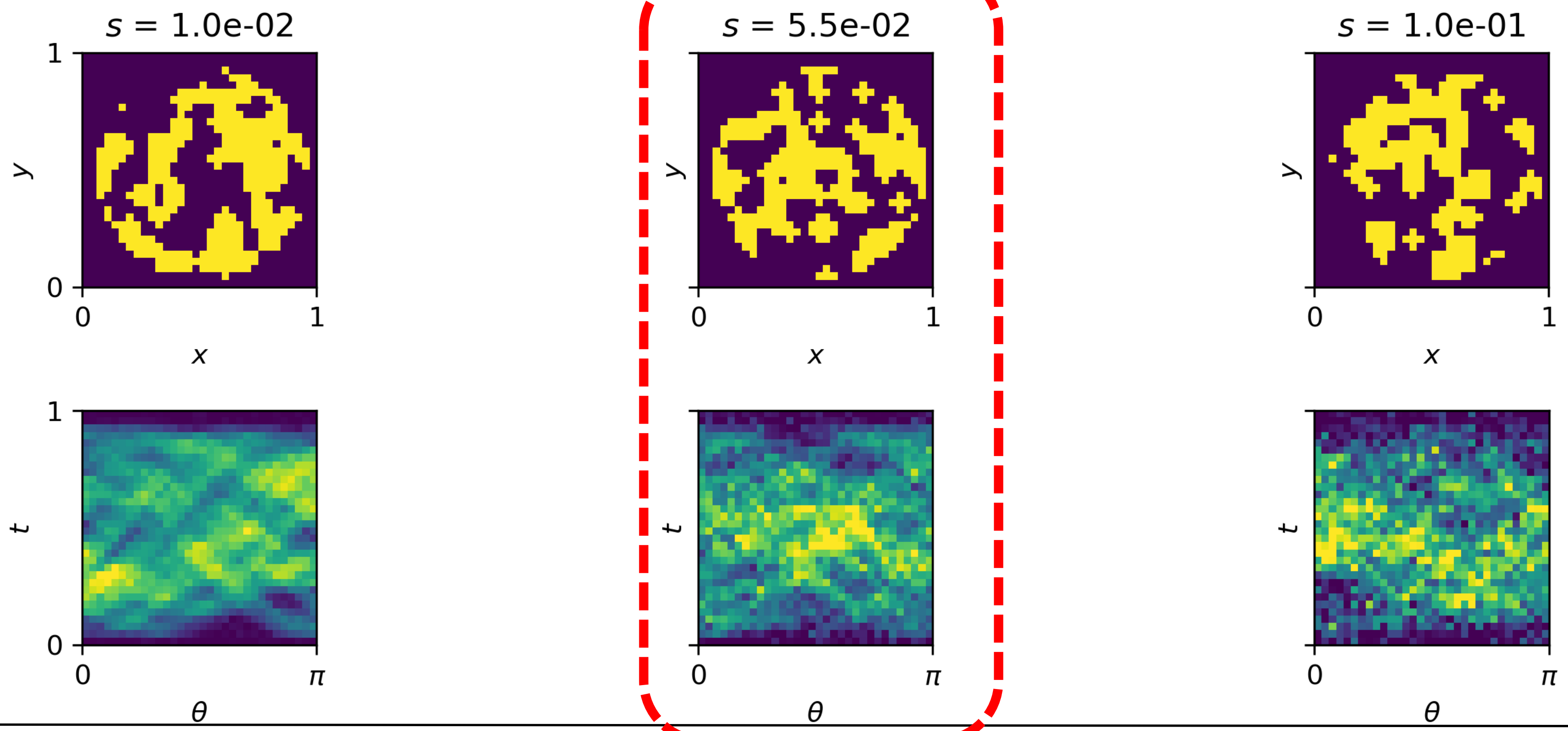


Computed tomography

- Goal is to infer an image $\mathbf{u}$ from a sinogram $\mathbf{f}$
- The measurements include Poisson noise
- Use affine normalizing flows to represent $\pi(\mathbf{u}, \mathbf{f}; \mathbf{s})$ where $\mathbf{s}$ models either noise level or angles

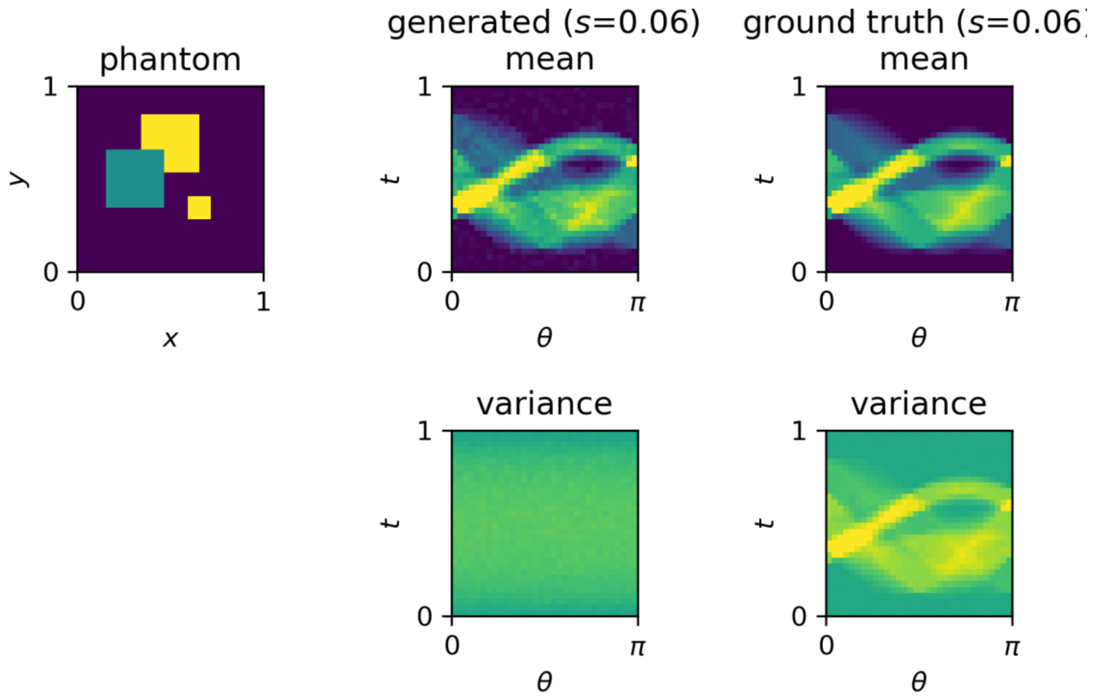
Computed tomography I

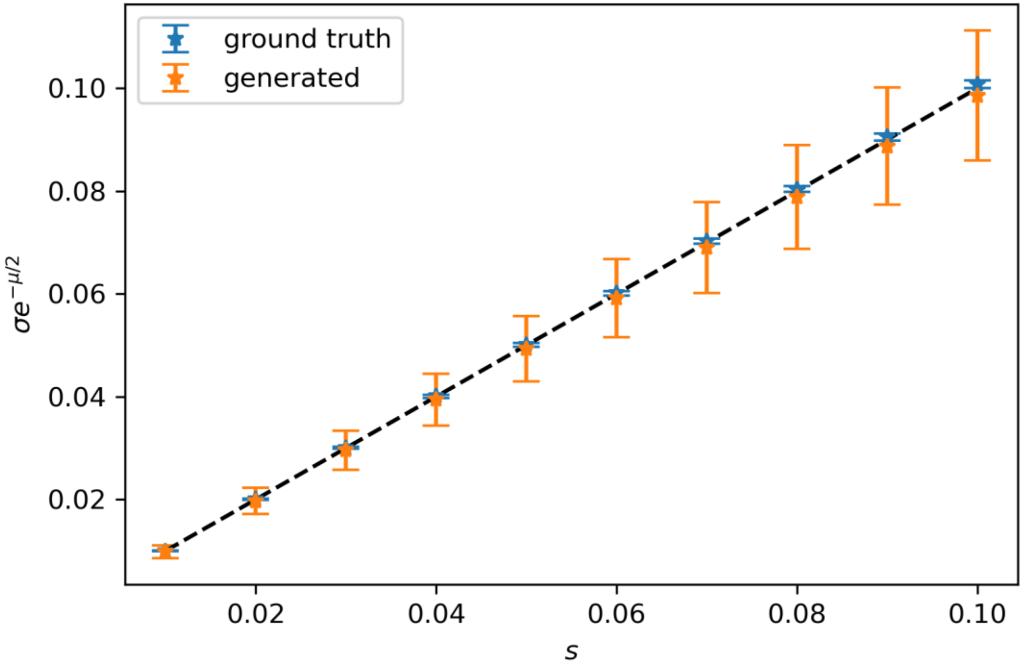
$$T_{\text{like}}(\mathbf{f};\mathbf{u},s) = s^{-1}\widetilde{T}_{\text{like}}(\mathbf{f};\mathbf{u}) \quad T_{\text{post}}(\mathbf{u};\mathbf{f},s) = s^{-1}\widetilde{T}_{\text{post}}(\mathbf{u};\mathbf{f})$$



Computed tomography I

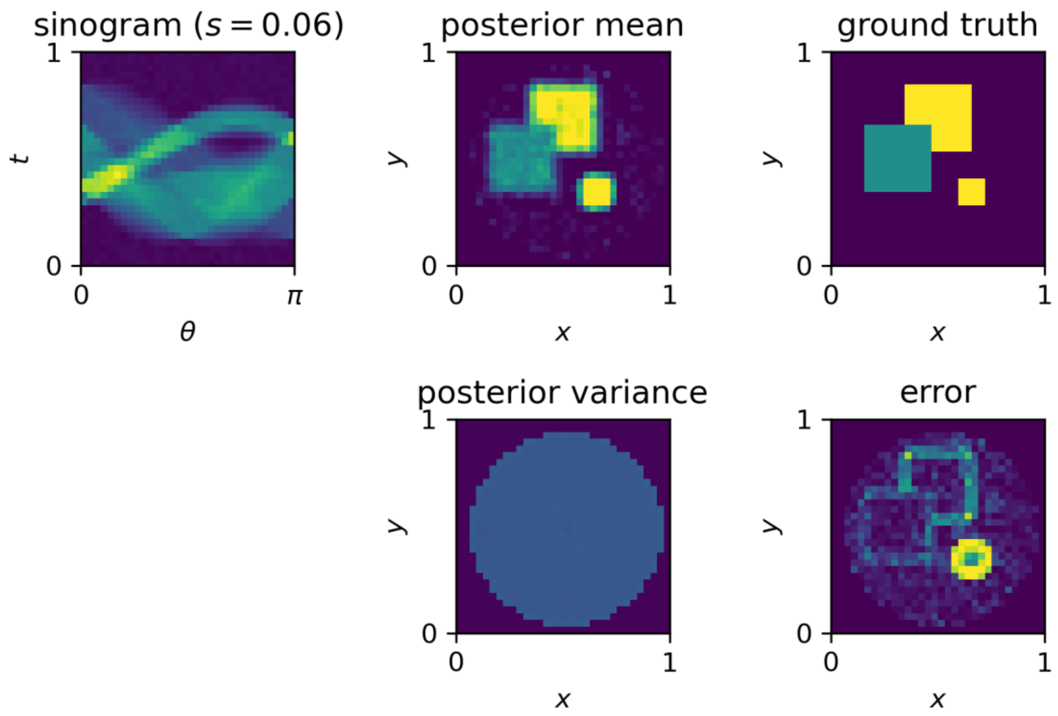
Simulation

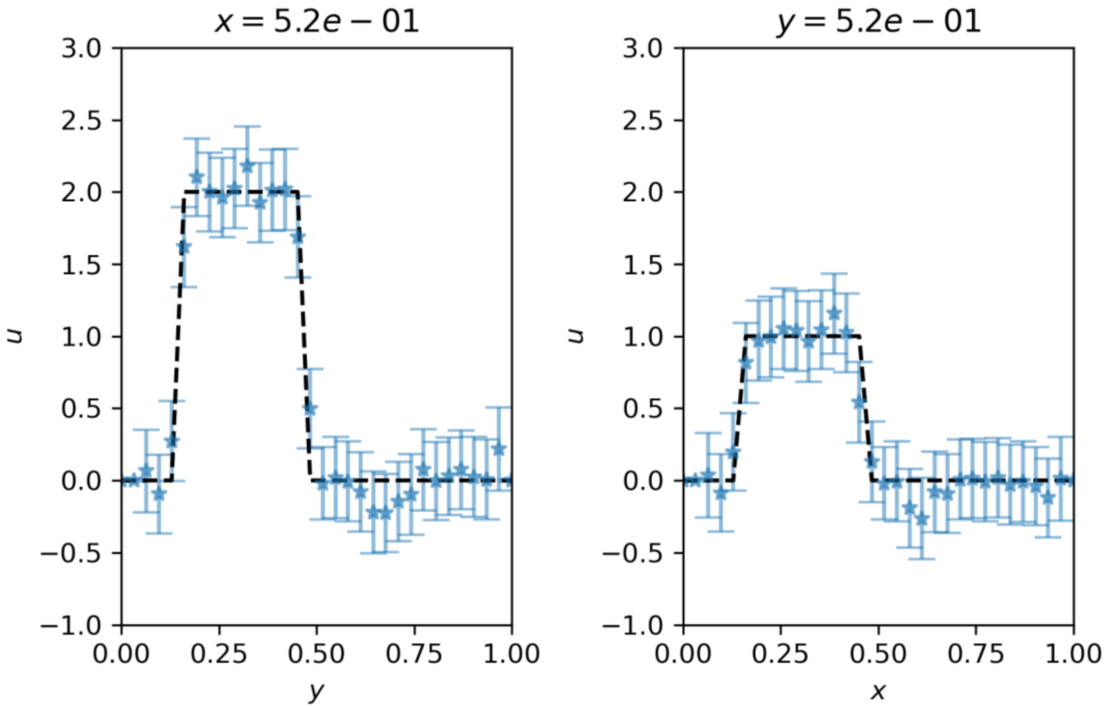




Computed tomography I

Inference

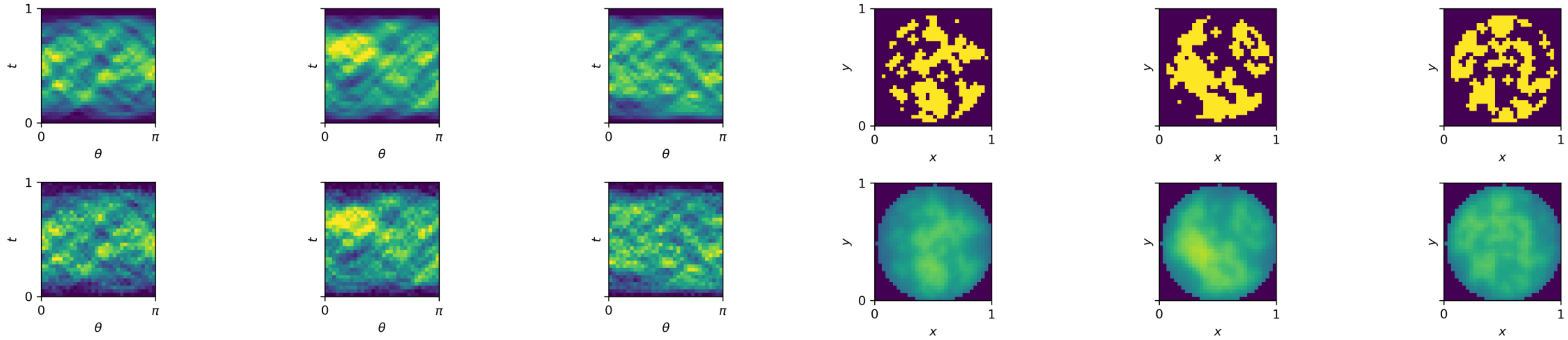




Computed tomography II

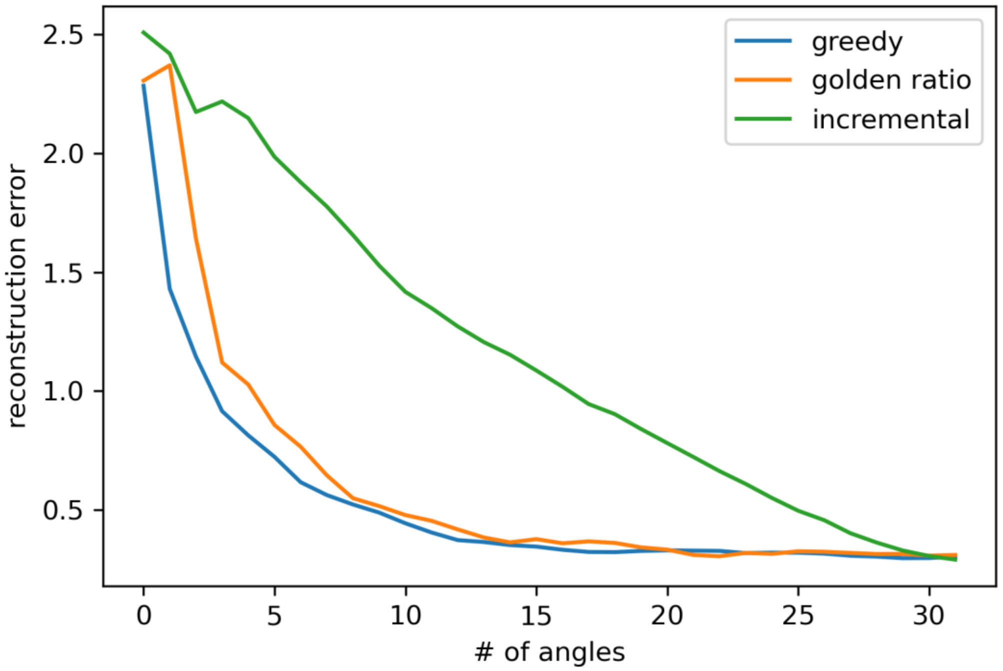
Include forward operator into architecture

$$T_{\text{like}}(\mathbf{f}; \mathbf{u}, \mathbf{s}) = \widetilde{T}_{\text{like}}(\mathbf{f}; K(\mathbf{s})\mathbf{u}) \qquad T_{\text{post}}(\mathbf{u}; \mathbf{f}, \mathbf{s}) = s^{-1}\widetilde{T}_{\text{post}}(\mathbf{u}; K(\mathbf{s})^\top \mathbf{f})$$



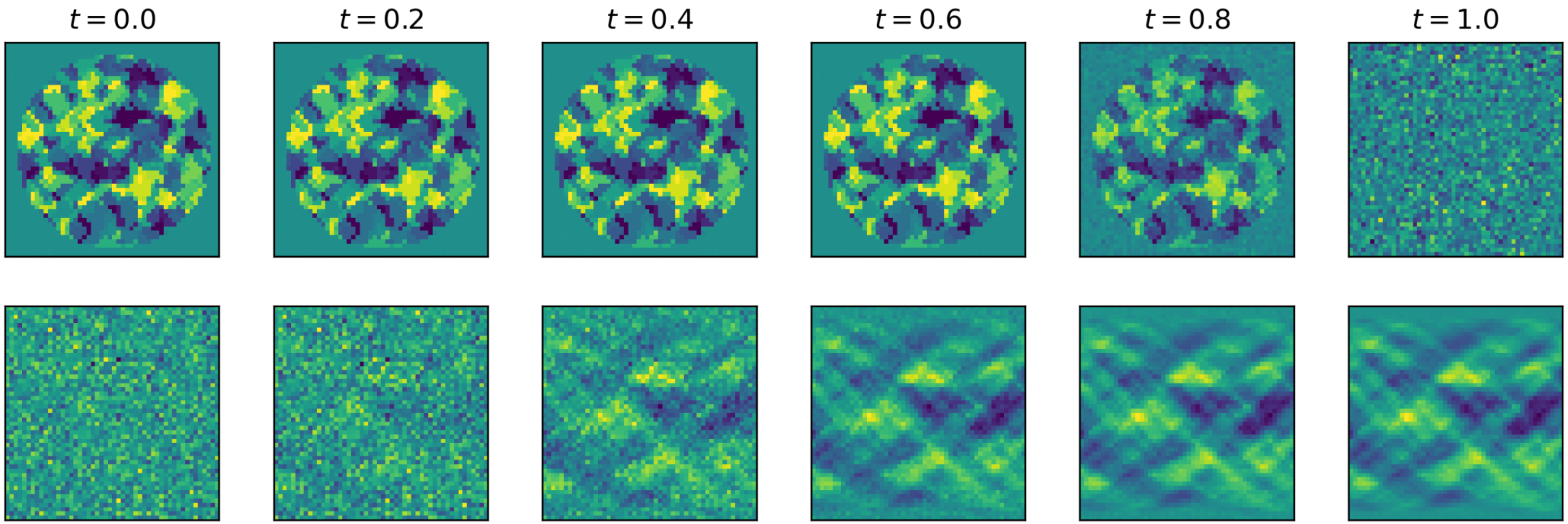
Computed tomography II

Sequentially select angles $\mathbf{s} = [s_1, \ldots, s_m]$ to minimize reconstruction error



Computed tomography III

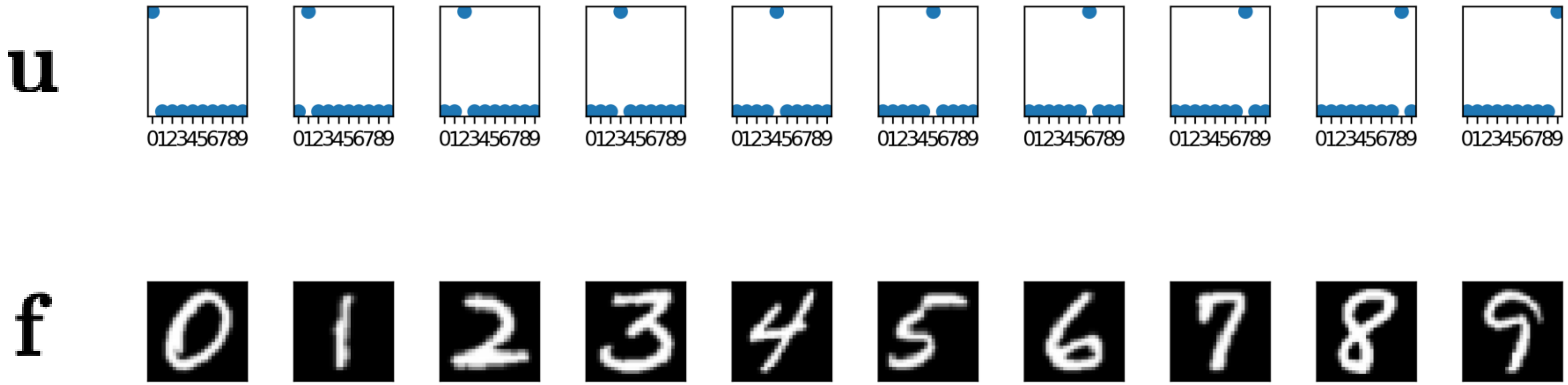
We can also use flow matching as generative model



See e.g., Wang 2024

Classification

Can also be used for generation / classification with the same model

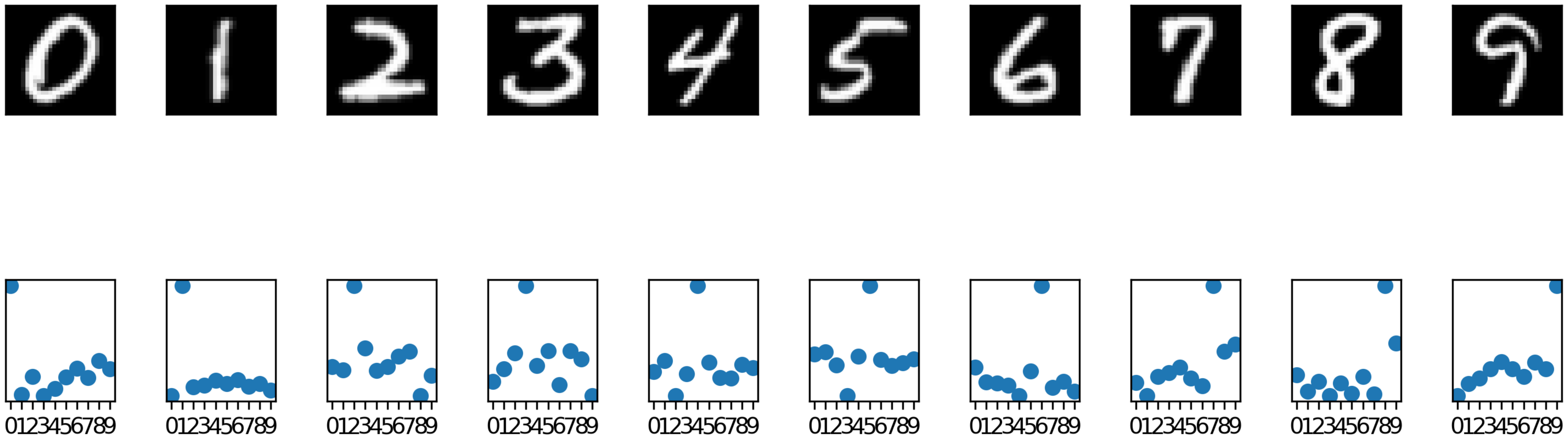


Classification

Generation (conditional mean)

Classification

Classification (conditional mean)



Wrap-up

Wrap-up

- **We can combine generative models for simulation and inference into a single generative model that samples from the underlying conditional distributions**
- **First results are promising, also for experimental design**
- **The approach still has many challenges, including training S directly based on samples from the joint distribution**
- **(Don`t underestimate the power of linear models)**


THANK U

(An incomplete lost of) references

Background, review and concepts

Stuart, Andrew M. "Inverse problems: a Bayesian perspective." *Acta numerica* 19 (2010): 451-559.

Ardizzone, Lynton, et al. "Analyzing Inverse Problems with Invertible Neural Networks." *International Conference on Learning Representations*. 2018.

Arridge, S., Maass, P., Öktem, O., & Schönlieb, C. B. (2019). Solving inverse problems using data-driven models. *Acta Numerica*, *28*(2019), 1–174.

Lavin, Alexander, et al. "Simulation intelligence: Towards a new generation of scientific methods." *arXiv preprint arXiv:2112.03235* (2021).

Ghattas, Omar, and Karen Willcox. "Learning physics-based models from data: perspectives from inverse problems and model reduction." *Acta Numerica* 30 (2021): 445-554.

Kim, Ki-Tae, et al. "hIPPYlib-MUQ: A Bayesian inference software framework for…." *ACM Transactions on Mathematical Software* (2023).

Orozco, R., Siahkoohi, A., Rizzuti, G., van Leeuwen, T., & Herrmann, F. (2023). *Adjoint operators enable fast and amortized machine learning based Bayesian uncertainty quantification*. 56.

Radev, Stefan T., et al. "JANA: Jointly amortized neural approximation of complex Bayesian models." Uncertainty in Artificial Intelligence. PMLR, 2023.

Radev, Stefan T., et al. "BayesFlow: Amortized Bayesian workflows with neural networks." arXiv preprint arXiv:2306.16015 (2023).

Marzouk, Youssef, et al. "Sampling via measure transport: An introduction." Handbook of uncertainty quantification. Springer, Cham, 2017. 785-825.

Adler, Jonas, and Ozan Öktem. "Deep bayesian inversion." arXiv preprint arXiv:1811.05910 1.2 (2018): 7.

Wang, Amy Xiang. "On Conditional Sampling with Joint Flow Matching." ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling. 1905.