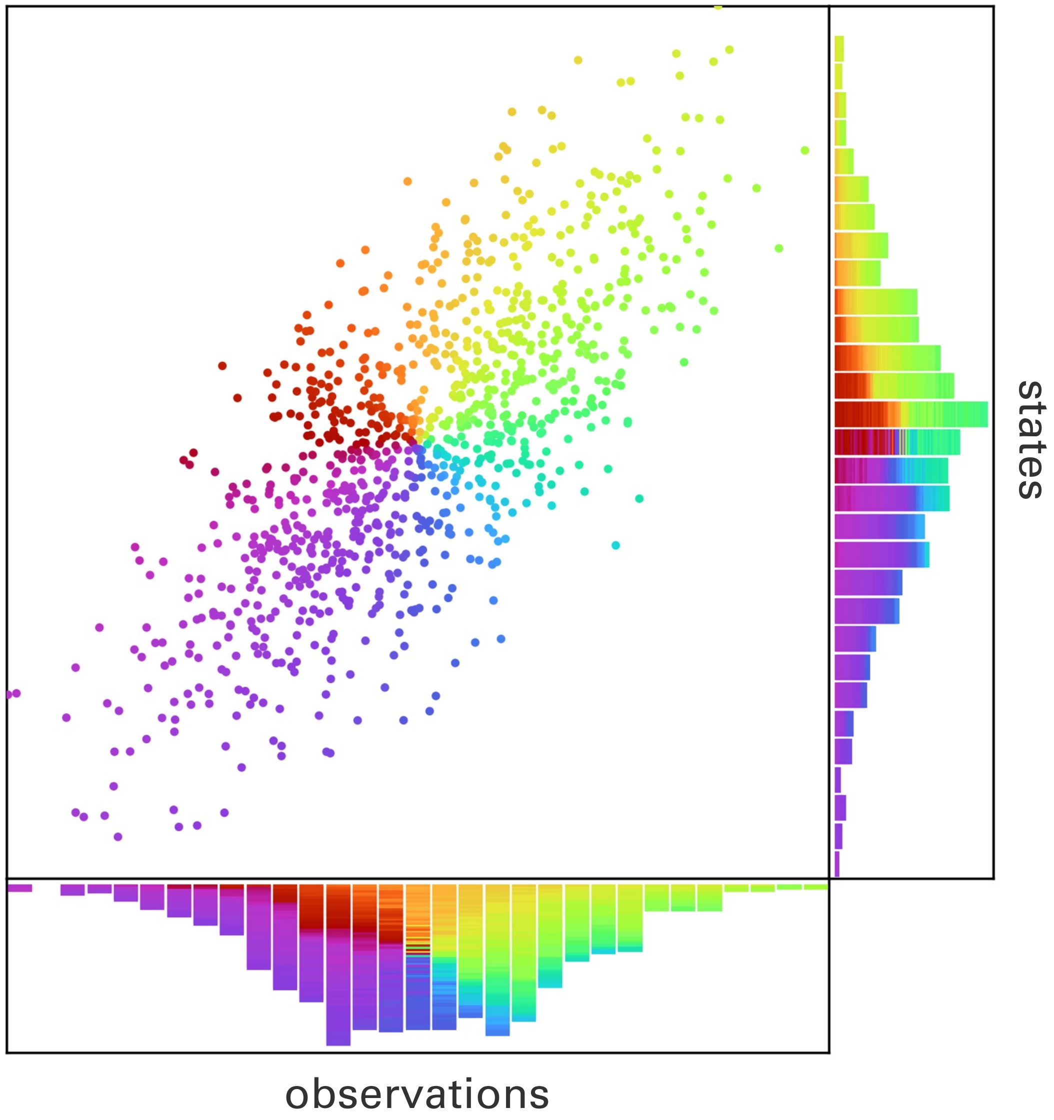
Adaptive and scalable triangular measure transport filters

Maximilian Ramgraber[1] and **Berent Lunde**[2]

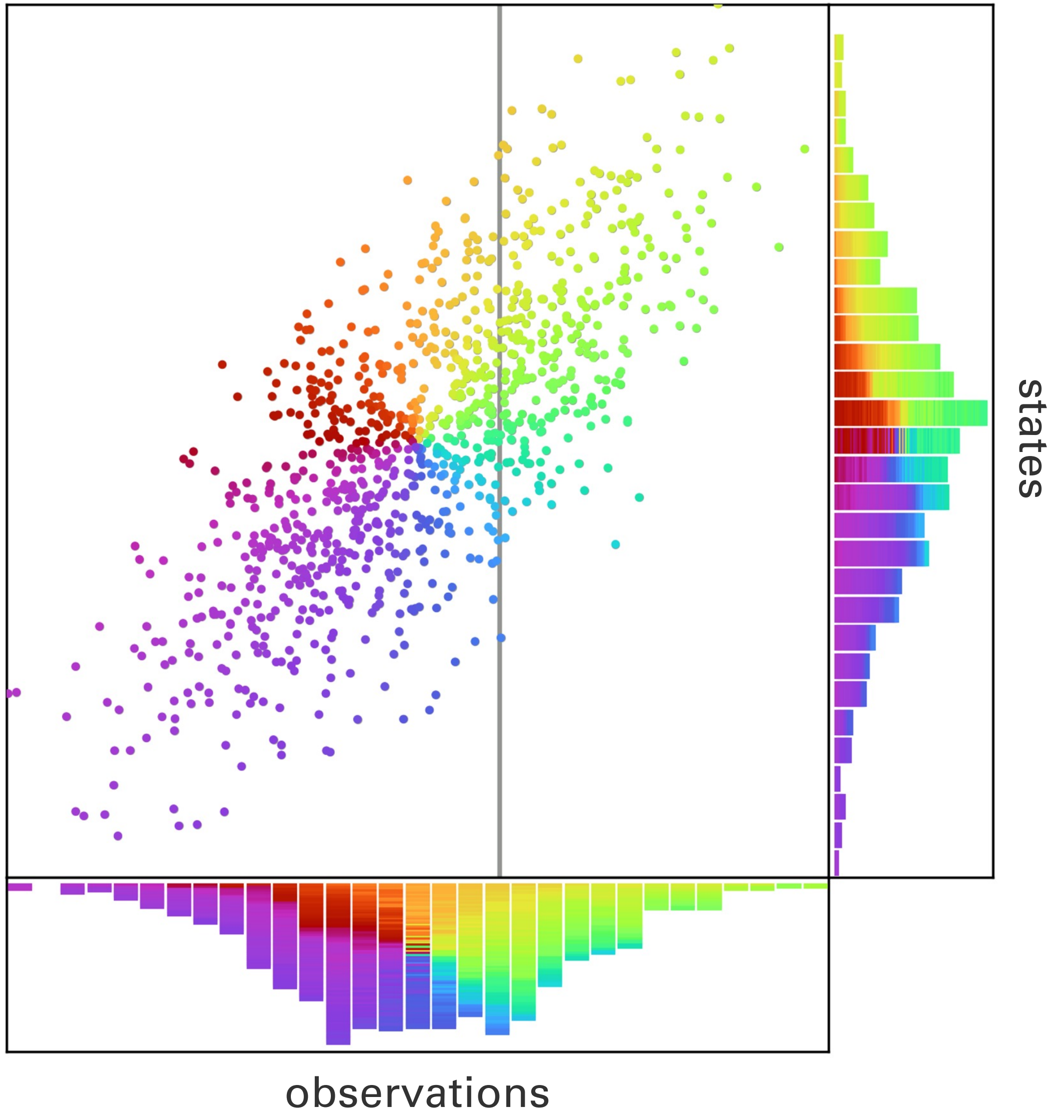
[1] Department of Geoscience and Engineering, TU Delft, The Netherlands
[2] Equinor, Norway

Ullensvang
June 18th 2025



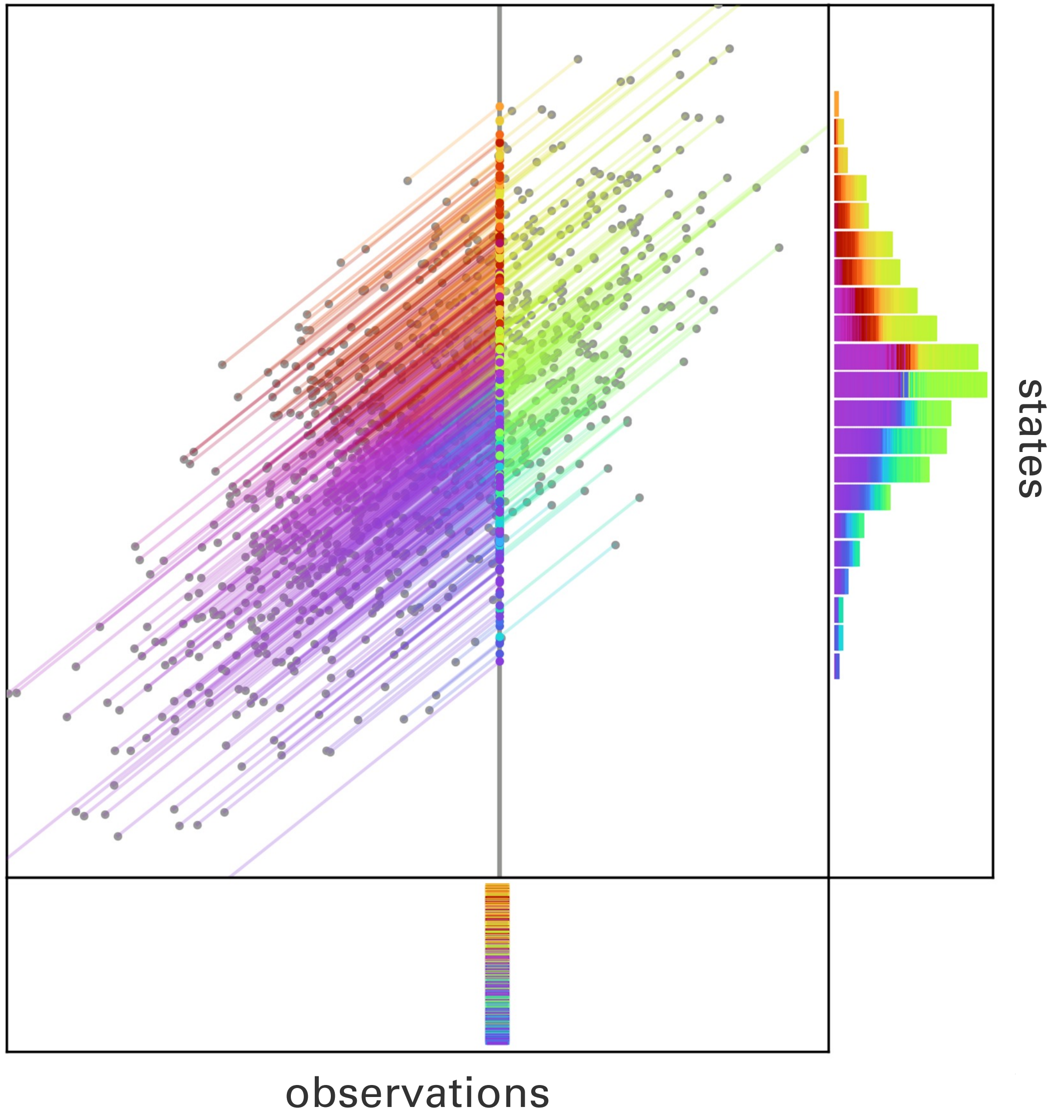
Data assimilation updates characterize **conditional** pdfs of a joint pdf between states and observables.

Linear and Nonlinear Data Assimilation



Data assimilation updates characterize **conditional** pdfs of a joint pdf between states and observables.

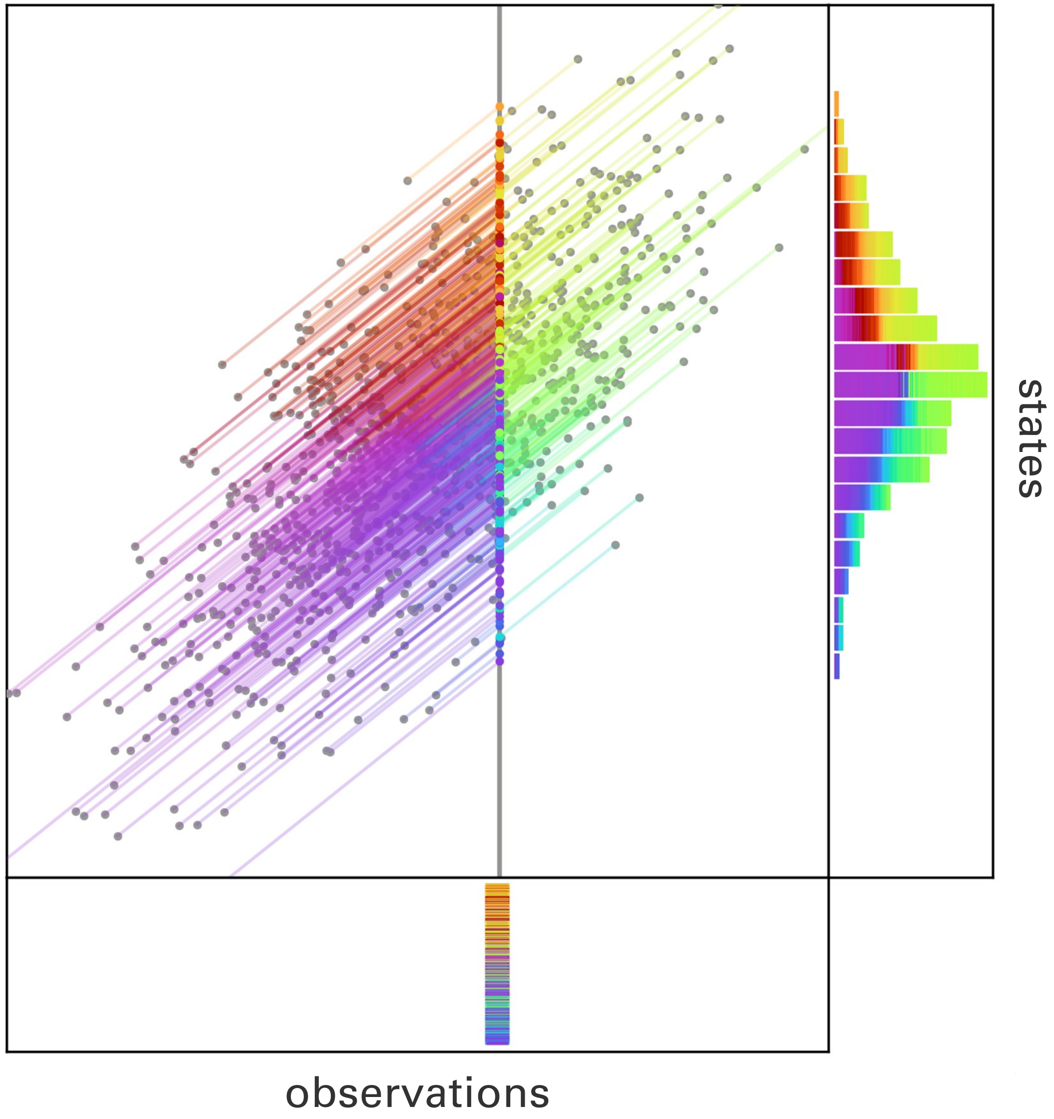
Linear and Nonlinear Data Assimilation



Data assimilation updates characterize **conditional** pdfs of a joint pdf between states and observables.

Kalman-type algorithms rely on a highly efficient **linear transformation**.

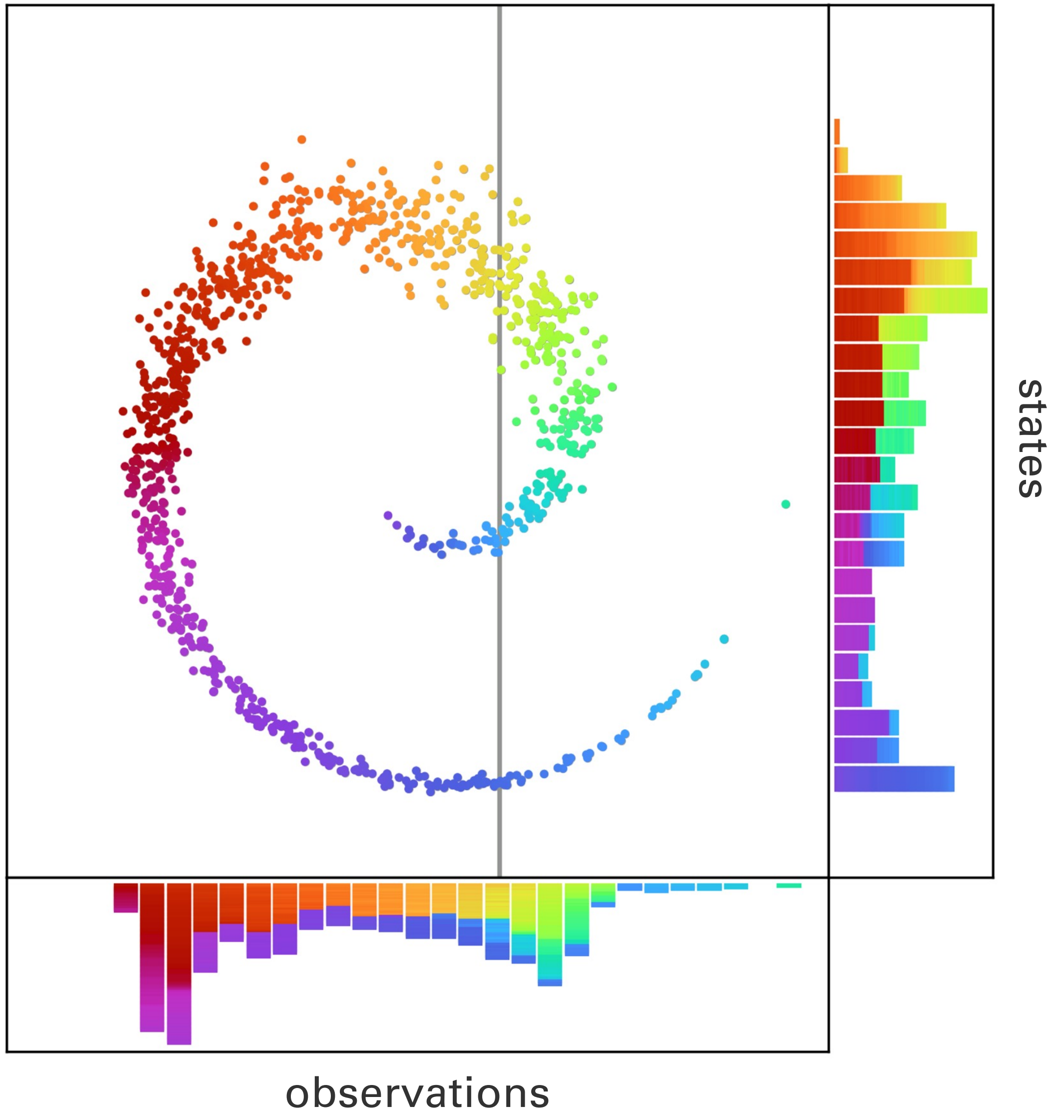
Linear and Nonlinear Data Assimilation



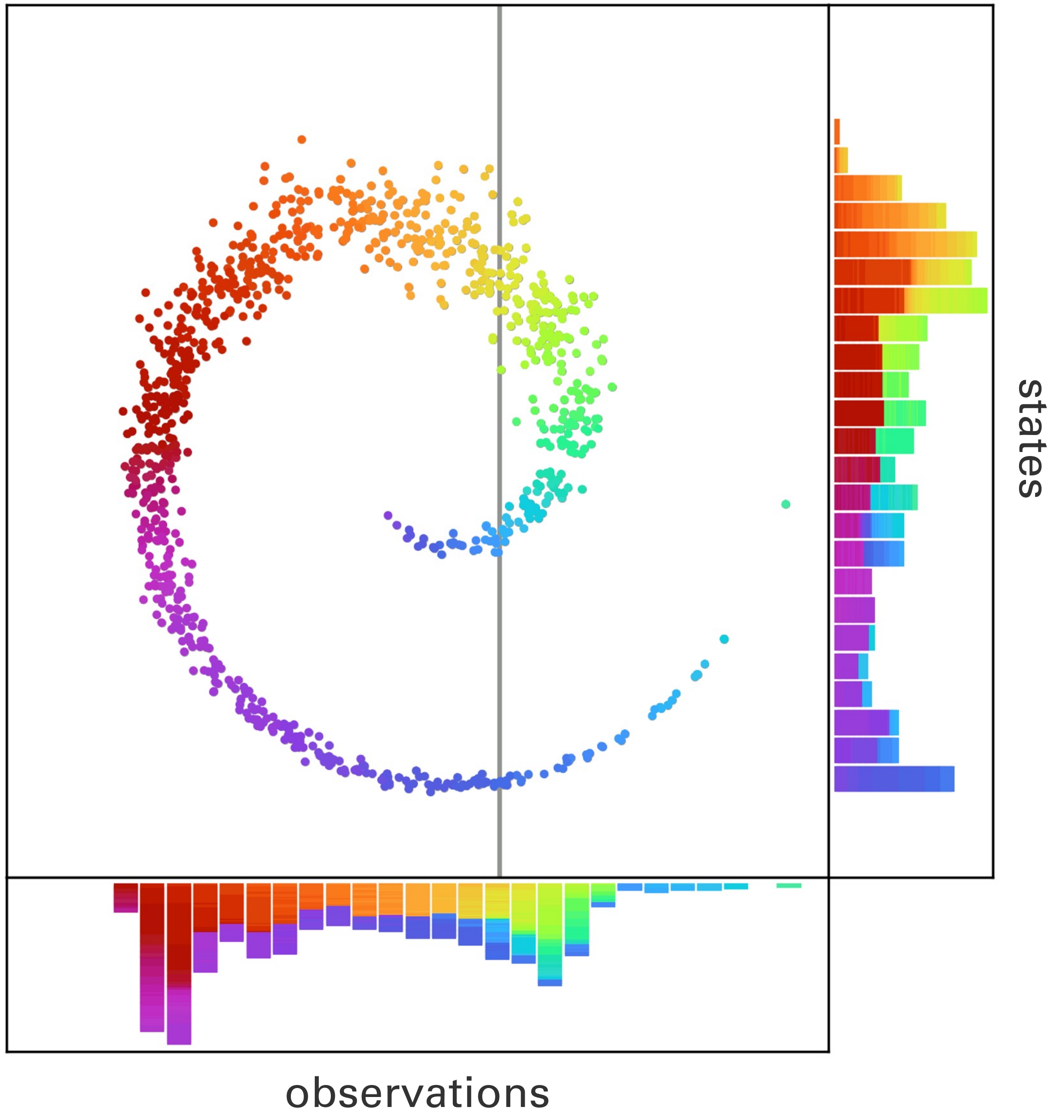
Data assimilation updates characterize **conditional** pdfs of a joint pdf between states and observables.

Kalman-type algorithms rely on a highly efficient **linear transformation**. Unfortunately, this update is only optimal for **Gaussian distributions**.

Linear and Nonlinear Data Assimilation

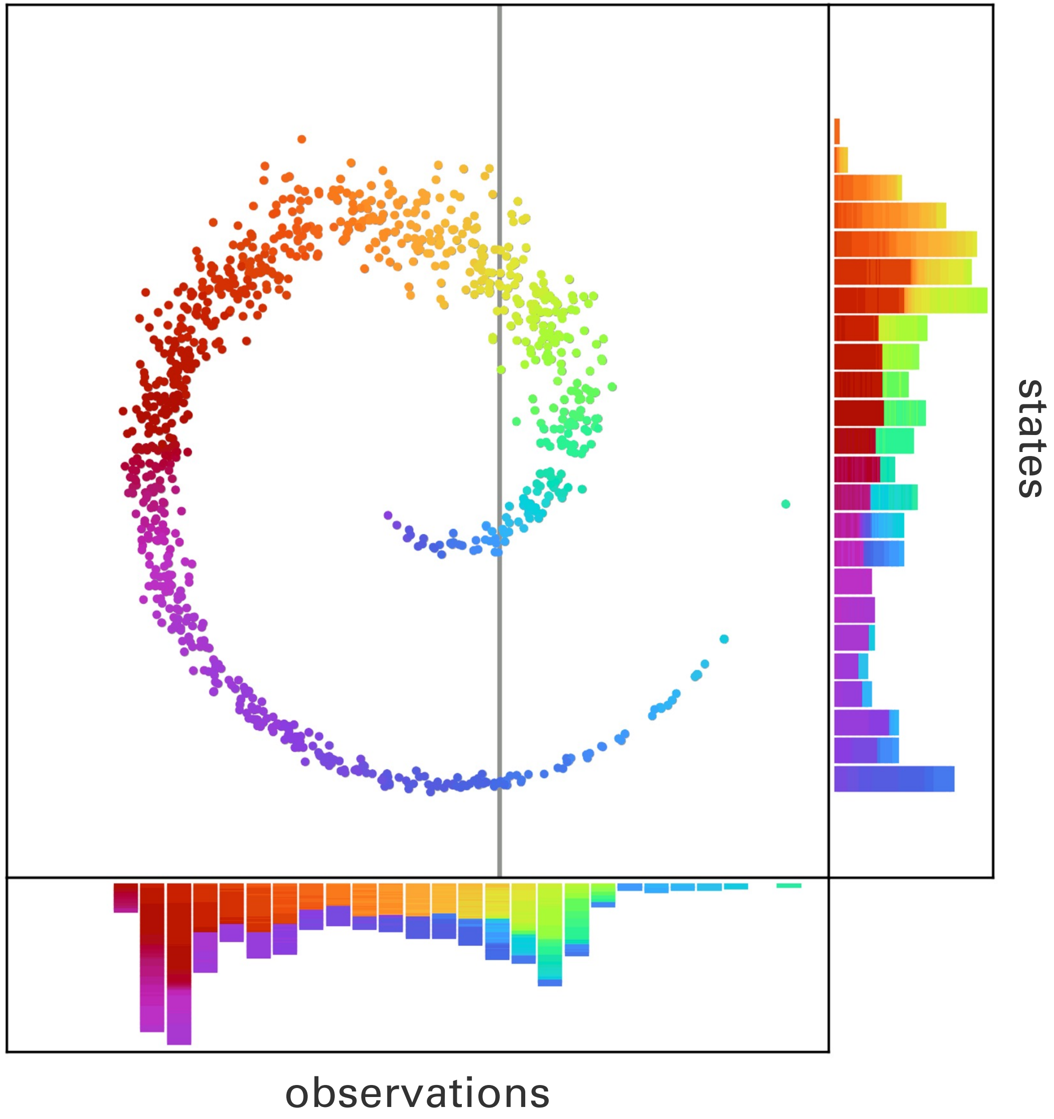


Linear and Nonlinear Data Assimilation



Bayesian inference for non-Gaussian distributions requires **nonlinear** updates.

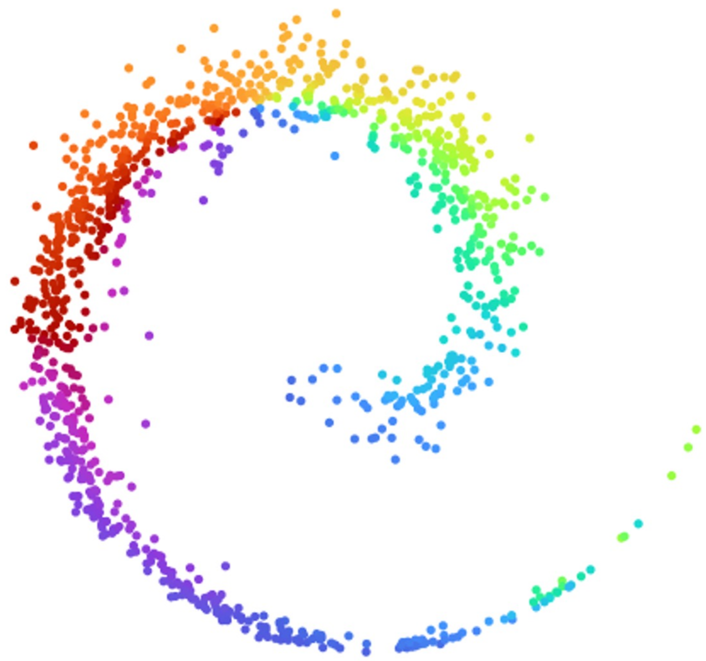
Linear and Nonlinear Data Assimilation



Bayesian inference for non-Gaussian distributions requires **nonlinear** updates. **Triangular transport** is a versatile method for nonlinear data assimilation.

A brief introduction to triangular measure transport

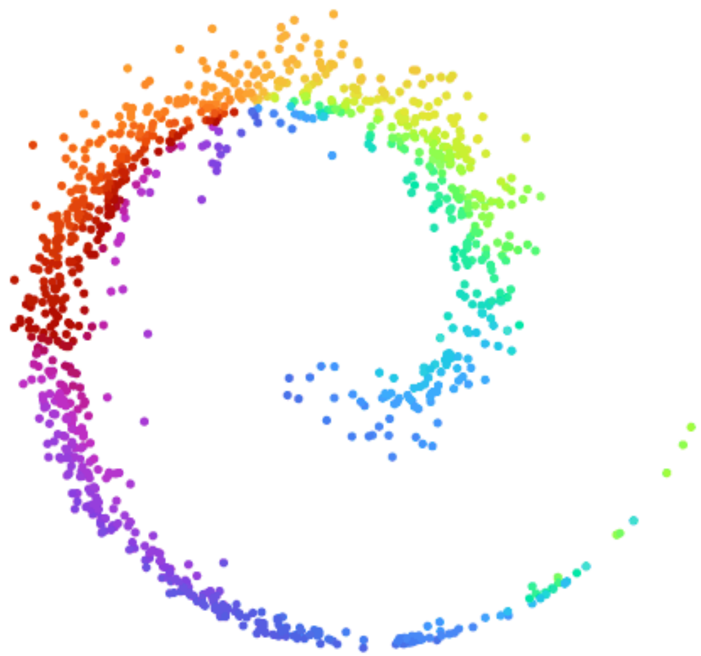
Transport methods seek a monotone, invertible **transport map** $\mathbf{S}$ from a target distribution π to a reference distribution η.



Ramgraber, M., Sharp, D., Provost, M. L., & Marzouk, Y. (2025). A friendly introduction to triangular transport. *arXiv preprint arXiv:2503.21673*.

A brief introduction to triangular measure transport

Transport methods seek a monotone, invertible **transport map** $\mathbf{S}$ from a target distribution π to a reference distribution η.



Ramgraber, M., Sharp, D., Provost, M. L., & Marzouk, Y. (2025). A friendly introduction to triangular transport. *arXiv preprint arXiv:2503.21673*.

A brief introduction to triangular measure transport

We specifically seek a **triangular** transport map $\mathbf{S}$:

(e.g., Spantini et al. 2022)

$$\mathbf{S}(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ \vdots \\ S_K(x_1, \dots, x_K) \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_K \end{bmatrix} = \mathbf{z}$$

$$\mathbf{S} : \mathbb{R}^K \to \mathbb{R}^K \qquad S_k : \mathbb{R}^k \to \mathbb{R}$$

The map comprises of map components S_k :

- Each S_k depends at most on the **first** k **entries** of the target vector $x_1, \dots, x_k$
- Each S_k must be **monotone in its last argument** x_k , i.e. $\partial_{x_k} S_k > 0$

Ramgraber, M., Sharp, D., Provost, M. L., & Marzouk, Y. (2025). A friendly introduction to triangular transport. *arXiv preprint arXiv:2503.21673*.

A brief introduction to triangular measure transport

Inverse map from reference distribution η to target distribution π:

$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \\ S_3^{-1}(z_3; x_1, x_2) \\ \vdots \\ S_K^{-1}(z_K; x_1, \dots, x_{K-1}) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_K \end{bmatrix} = \mathbf{x} \quad \vdots$$

Transport maps decompose the target distribution π: (Spantini et al. 2022)

$$\pi(\mathbf{x}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1,x_2)\dots\pi(x_K|x_1,\dots,x_{K-1})$$

A brief introduction to triangular measure transport

We can manipulate this inversion to characterize **conditionals** of the target distribution π:

$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \\ S_3^{-1}(z_3; x_1, x_2) \\ S_4^{-1}(z_4; x_1, x_2, x_3) \\ \vdots \\ S_K^{-1}(z_K; x_1, x_2, x_3, \dots, x_{K-1}) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_K \end{bmatrix} = \mathbf{x}$$

Transport maps decompose the target distribution π: (Spantini et al. 2022)

$$\pi(\mathbf{x}) = \cancel{\pi(x_1)}\,\cancel{\pi(x_2|x_1)}\,\pi(x_3|x_1, x_2) \dots \pi(x_K|x_1, \dots, x_{K-1})$$

A brief introduction to triangular measure transport

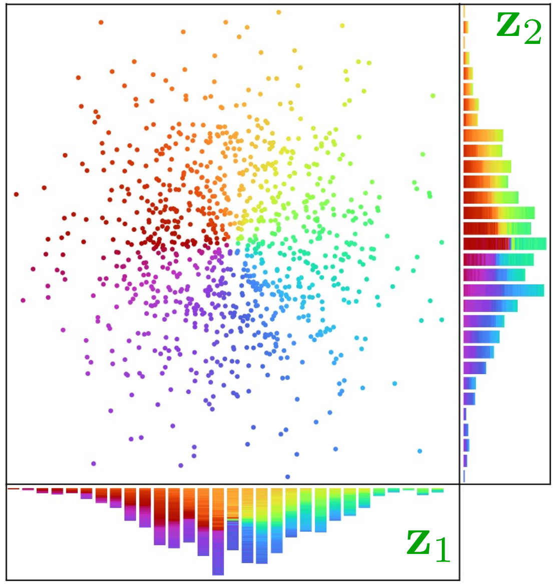
We can manipulate this inversion to characterize **conditionals** of the target distribution π :

$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1^*) \\ S_3^{-1}(z_3; x_1^*, x_2^*) \\ S_4^{-1}(z_4; x_1^*, x_2^*, x_3^*) \\ \vdots \\ S_K^{-1}(z_K; x_1^*, x_2^*, x_3^*, \dots, x_{K-1}^*) \end{bmatrix} = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \\ \vdots \\ x_K^* \end{bmatrix} = \mathbf{x}$$

This partial inversion factorizes a **conditional** of the target distribution π :

$$\pi(\mathbf{x}_{3:K}|\mathbf{x}_{1:2}^*) = \pi(x_3|\mathbf{x}_{1:2}^*)\pi(x_4|\mathbf{x}_{1:2}^*, x_3)\pi(x_5|\mathbf{x}_{1:2}^*, \mathbf{x}_{3:4}) \dots \pi(x_K|\mathbf{x}_{1:2}^*, \mathbf{x}_{3:K-1})$$

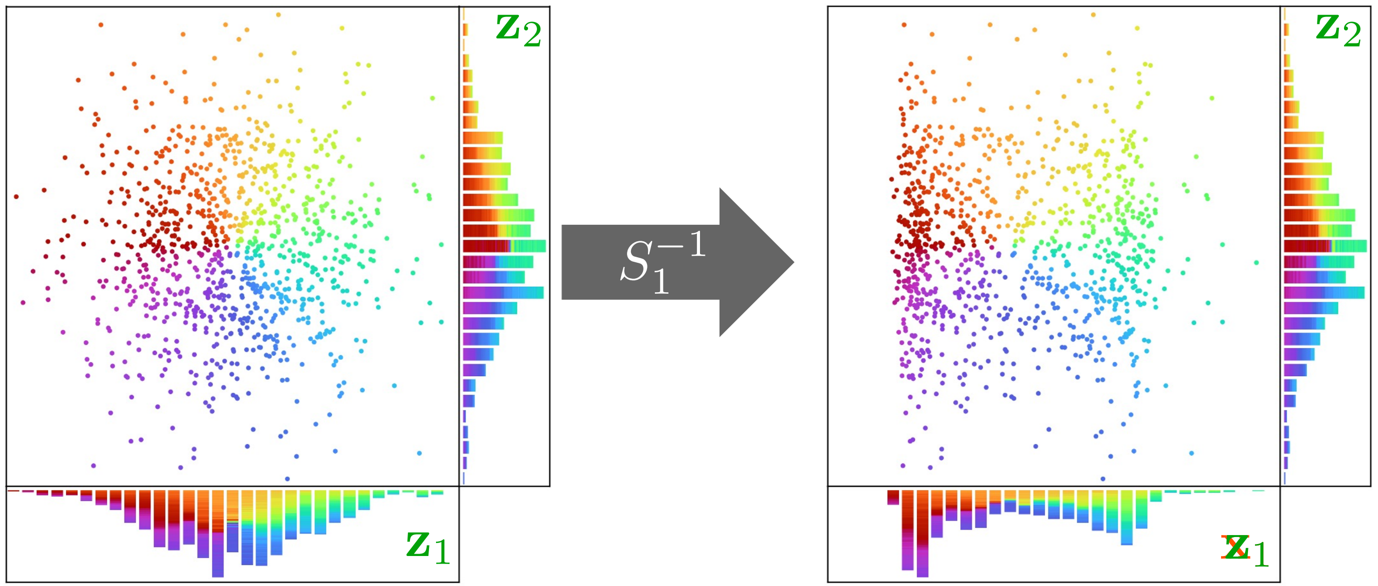
A brief introduction to triangular measure transport



The **map inversion** proceeds one component at a time:

$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

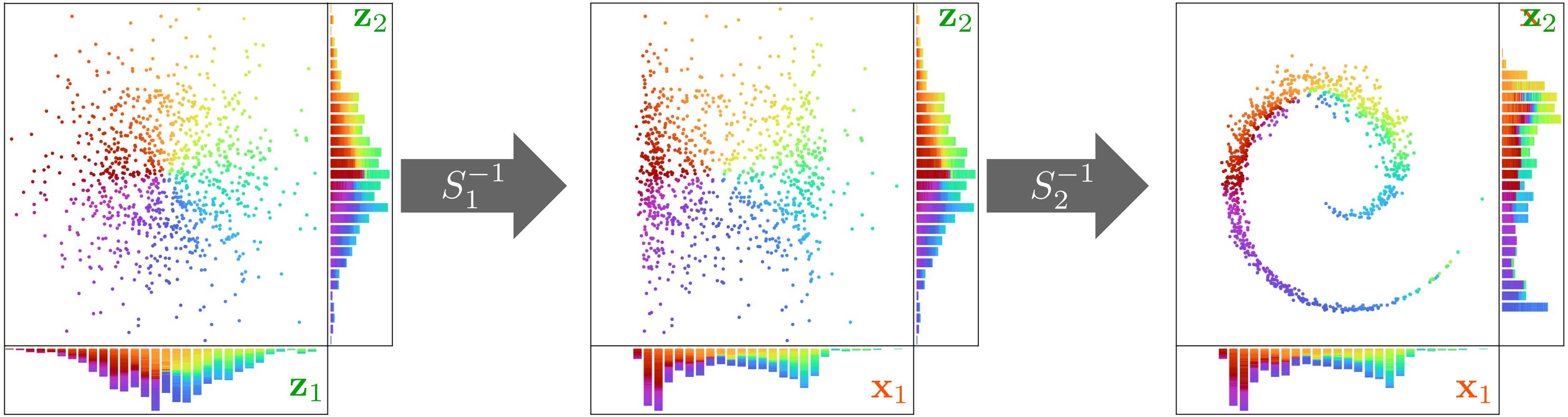
A brief introduction to triangular measure transport



The **map inversion** proceeds one component at a time:

$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

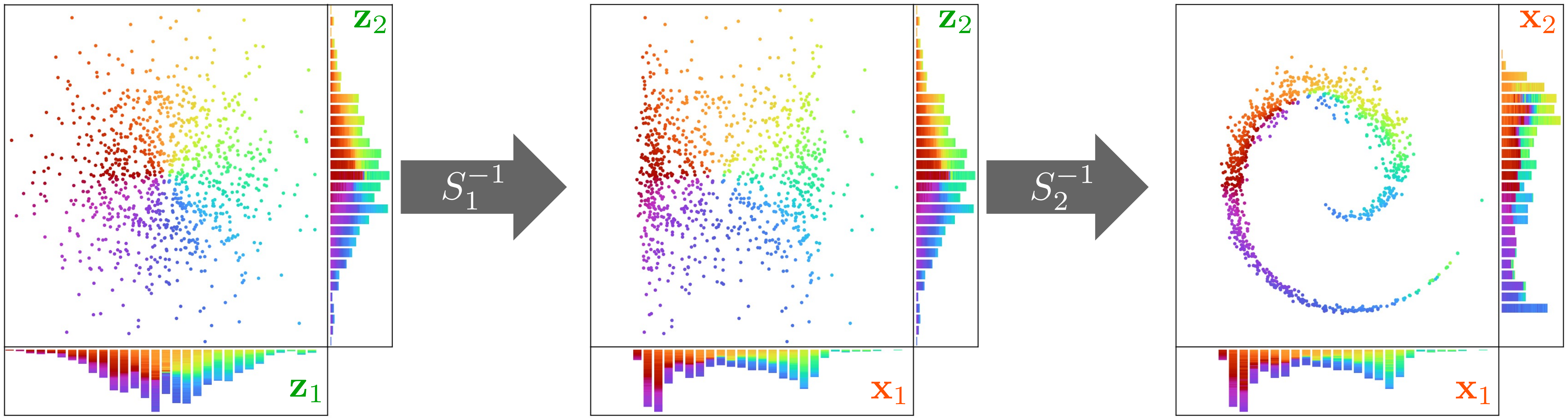
A brief introduction to triangular measure transport



The **map inversion** proceeds one component at a time:

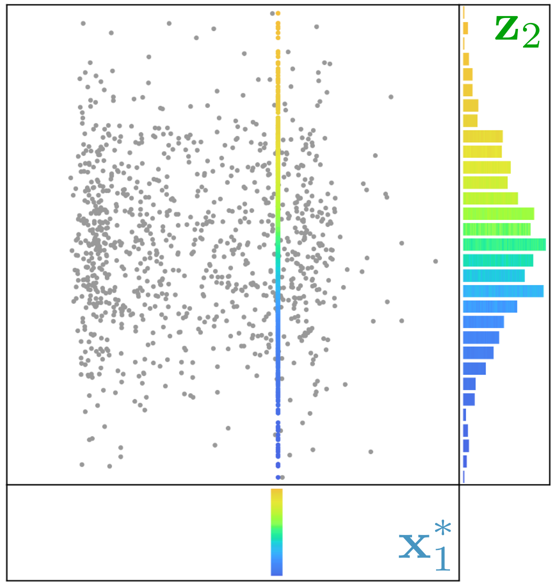
$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

A brief introduction to triangular measure transport

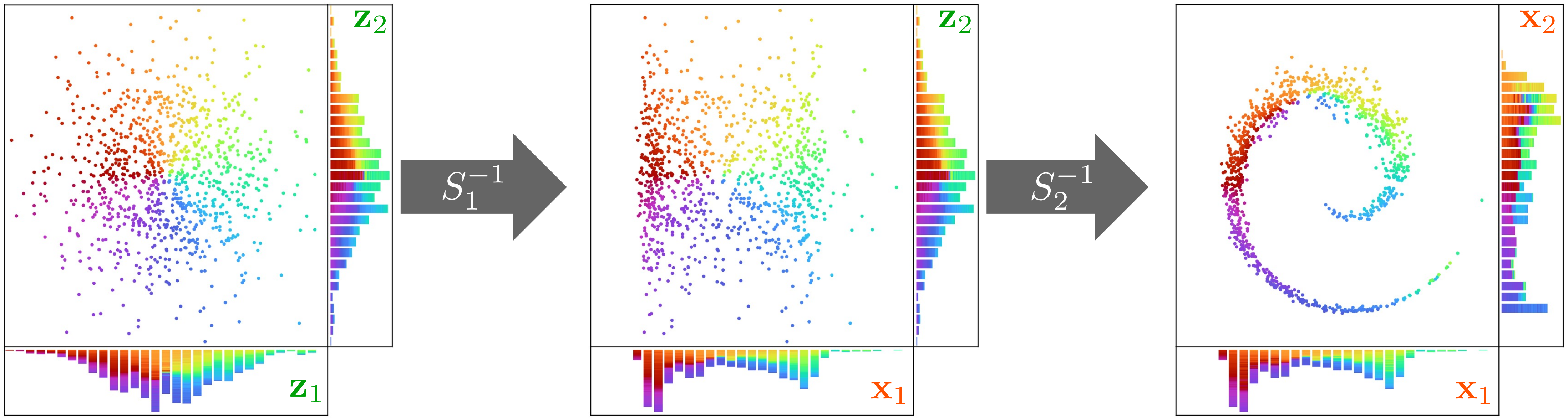


Conditional inversion manipulates the inversion process:

$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1^*) \end{bmatrix} = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$$

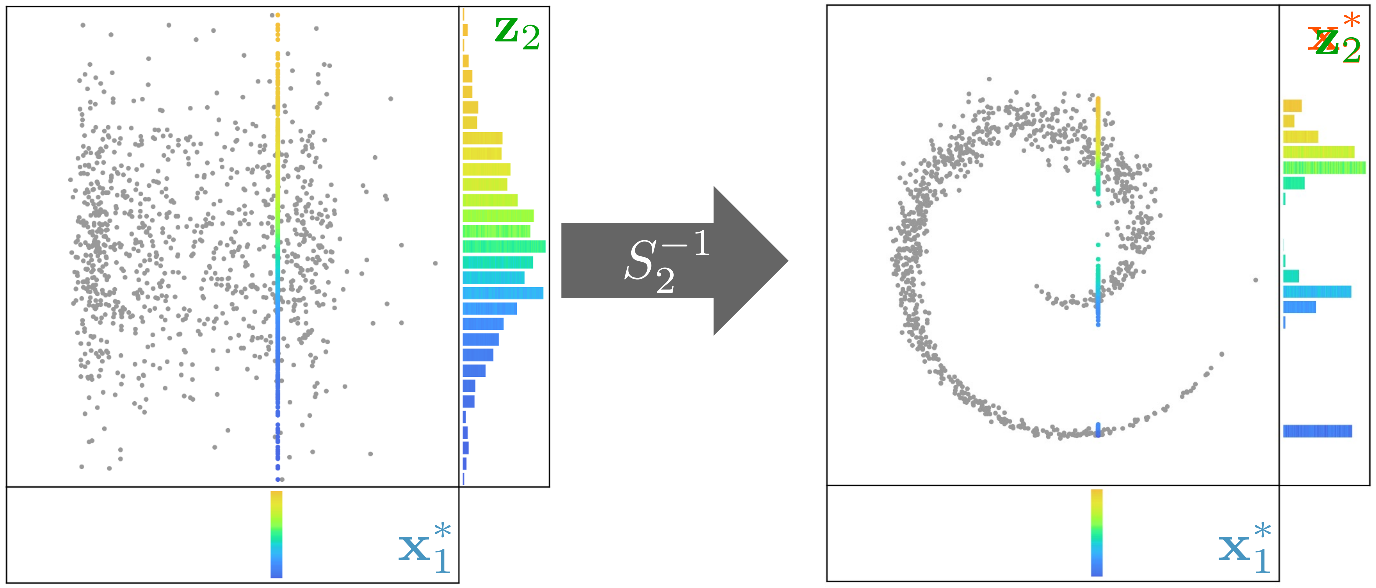


A brief introduction to triangular measure transport



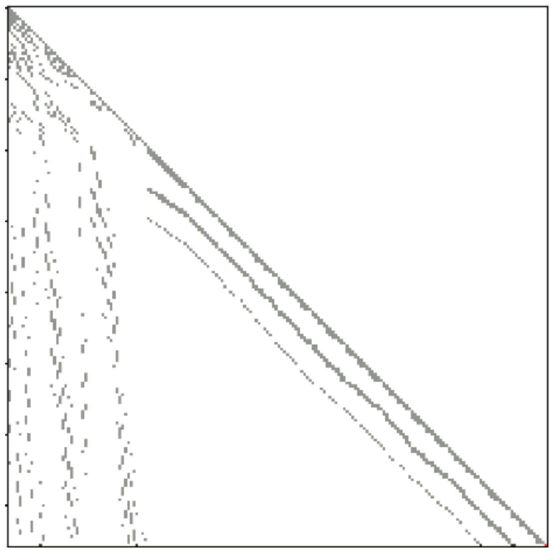
Conditional inversion manipulates the inversion process:

$$\mathbf{S}^{-1}(\mathbf{z}) = \begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1^*) \end{bmatrix} = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$$

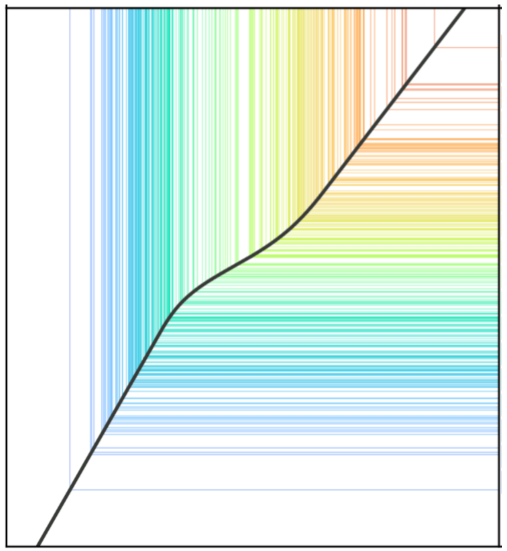


Making triangular maps useful

Applying triangular maps to real systems requires two things:



Sparsity
numerical efficiency
in high-dimensional systems



Parsimony
identifying the
optimal degree of nonlinearity

Scaling triangular measure transport

Many systems are **extremely high-dimensional**.

How can we make triangular maps **scalable**?

$$\mathbf{S}(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ S_3(x_1, x_2, x_3) \\ S_4(x_1, x_2, x_3, x_4) \\ \vdots \\ S_{999,999}(x_1, \dots, x_{999,999}) \\ S_{1,000,000}(x_1, \dots, x_{999,999}, x_{1,000,000}) \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ z_{999,999} \\ z_{1,000,000} \end{bmatrix} = \mathbf{z}$$

Scaling triangular measure transport

Important feature: Triangular transport maps can naturally capitalize on conditional independence by removing variable dependencies.

$$\pi(\mathbf{x}_{1:4}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1,x_2)\pi(x_4|x_1,x_2,x_3)$$

Map structure

$$\begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2;x_1) \\ S_3^{-1}(z_3;x_1,x_2) \\ S_4^{-1}(z_4;x_1,x_2,x_3) \end{bmatrix}$$

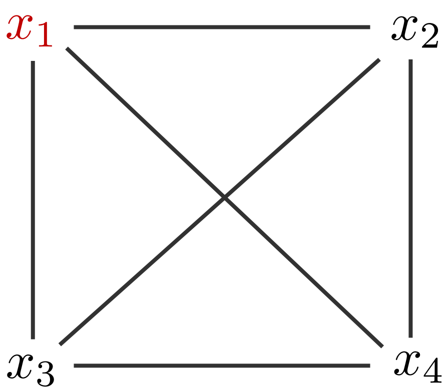
Factorization

$$\begin{array}{l} \pi(x_1) \\ \pi(x_2|x_1) \\ \pi(x_3|x_1,x_2) \\ \pi(x_4|x_1,x_2,x_3) \end{array}$$

Scaling triangular measure transport

Important feature: Triangular transport maps can naturally capitalize on conditional independence by removing variable dependencies.

$$\pi(\mathbf{x}_{1:4}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1, x_2)\pi(x_4|x_1, x_2, x_3)$$



Map structure

$$\begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \\ S_3^{-1}(z_3; x_1, x_2) \\ S_4^{-1}(z_4; x_1, x_2, x_3) \end{bmatrix}$$

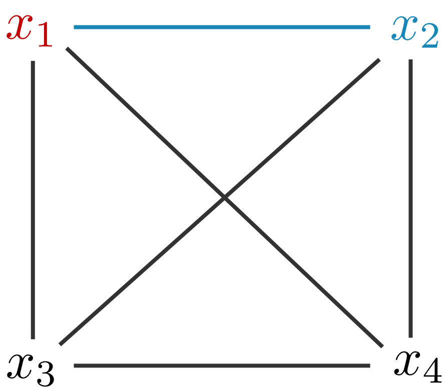
Factorization

$$\begin{array}{l} \pi(x_1) \\ \pi(x_2|x_1) \\ \pi(x_3|x_1, x_2) \\ \pi(x_4|x_1, x_2, x_3) \end{array}$$

Scaling triangular measure transport

Important feature: Triangular transport maps can naturally capitalize on conditional independence by removing variable dependencies.

$$\pi(\mathbf{x}_{1:4}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1,x_2)\pi(x_4|x_1,x_2,x_3)$$



Map structure

$$\begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \\ S_3^{-1}(z_3; x_1, x_2) \\ S_4^{-1}(z_4; x_1, x_2, x_3) \end{bmatrix}$$

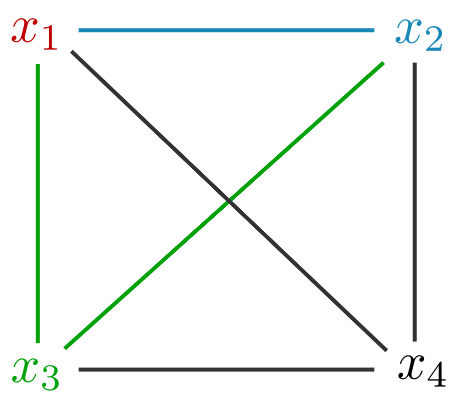
Factorization

$$\pi(x_1)$$
$$\pi(x_2|x_1)$$
$$\pi(x_3|x_1, x_2)$$
$$\pi(x_4|x_1, x_2, x_3)$$

Scaling triangular measure transport

Important feature: Triangular transport maps can naturally capitalize on conditional independence by removing variable dependencies.

$$\pi(\mathbf{x}_{1:4}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1,x_2)\pi(x_4|x_1,x_2,x_3)$$



Map structure

$$\begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2;x_1) \\ S_3^{-1}(z_3;x_1,x_2) \\ S_4^{-1}(z_4;x_1,x_2,x_3) \end{bmatrix}$$

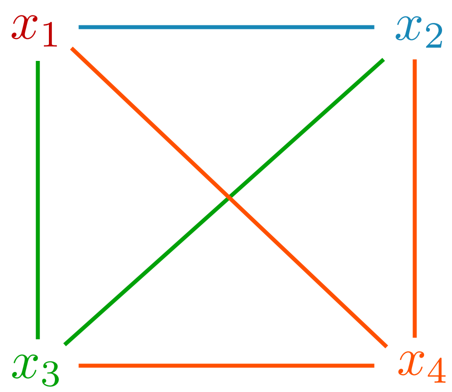
Factorization

$$\begin{matrix} \pi(x_1) \\ \pi(x_2|x_1) \\ \pi(x_3|x_1, x_2) \\ \pi(x_4|x_1, x_2, x_3) \end{matrix}$$

Scaling triangular measure transport

Important feature: Triangular transport maps can naturally capitalize on conditional independence by removing variable dependencies.

$$\pi(\mathbf{x}_{1:4}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1,x_2)\pi(x_4|x_1,x_2,x_3)$$



Map structure

$$\begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \\ S_3^{-1}(z_3; x_1, x_2) \\ S_4^{-1}(z_4; x_1, x_2, x_3) \end{bmatrix}$$

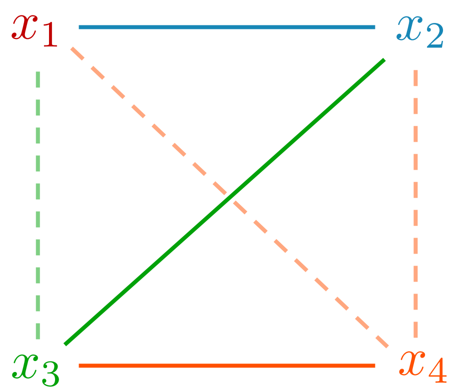
Factorization

$$\begin{aligned} &\pi(x_1) \\ &\pi(x_2|x_1) \\ &\pi(x_3|x_1, x_2) \\ &\pi(x_4|x_1, x_2, x_3) \end{aligned}$$

Scaling triangular measure transport

Important feature: Triangular transport maps can naturally capitalize on conditional independence by removing variable dependencies.

$$\pi(\mathbf{x}_{1:4}) = \pi(x_1)\pi(x_2|x_1)\pi(x_3|x_1, x_2)\pi(x_4|x_1, x_2, x_3)$$



Map structure

$$\begin{bmatrix} S_1^{-1}(z_1) \\ S_2^{-1}(z_2; x_1) \\ S_3^{-1}(z_3; x_1, x_2) \\ S_4^{-1}(z_4; x_1, x_2, x_3) \end{bmatrix}$$

Factorization

$$\begin{array}{l} \pi(x_1) \\ \pi(x_2|x_1) \\ \pi(x_3|x_1, x_2) \\ \pi(x_4|x_1, x_2, x_3) \end{array}$$

! $x_3 \perp\!\!\!\perp x_1 \mid x_2$

! $x_4 \perp\!\!\!\perp x_1, x_2 \mid x_3$

Exploiting **conditional independence**:

- Is a powerful form of **localization**
- **Reduces the computational complexity** of the map

Scaling triangular measure transport

Many systems are **extremely high-dimensional**.

$$\mathbf{S}(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ S_3(x_1, x_2, x_3) \\ S_4(x_1, x_2, x_3, x_4) \\ \vdots \\ S_{999{,}999}(x_1, \dots, x_{999{,}999}) \\ S_{1{,}000{,}000}(x_1, \dots, x_{999{,}999}, x_{1{,}000{,}000}) \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ z_{999{,}999} \\ z_{1{,}000{,}000} \end{bmatrix} = \mathbf{z}$$

Scaling triangular measure transport

Many systems are **extremely high-dimensional**.

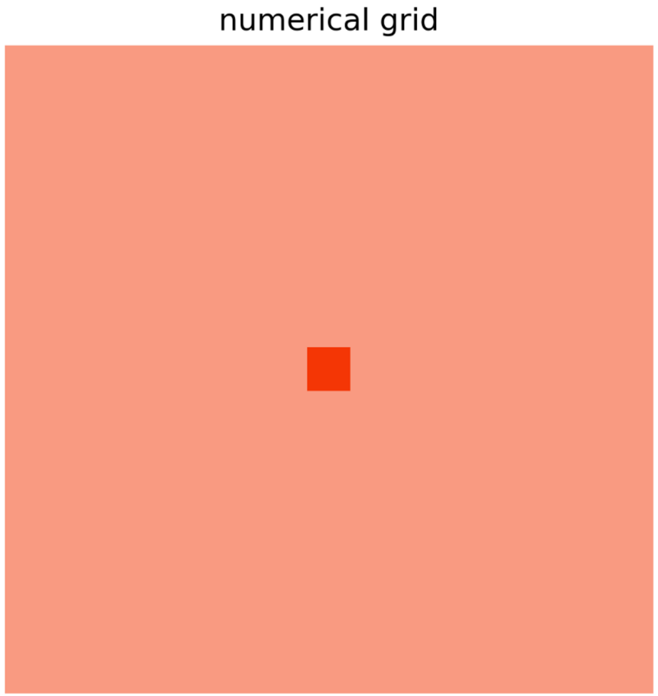
With Markov properties, triangular maps **scale linearly** with dimension!

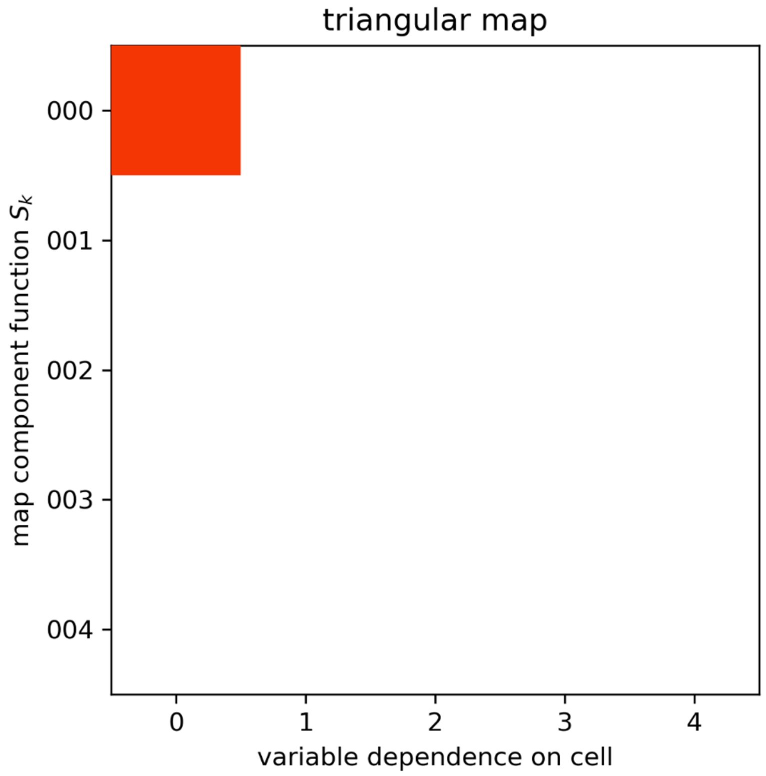
$$\mathbf{S}(\mathbf{x}) = \begin{bmatrix} S_1(x_1) \\ S_2(x_1, x_2) \\ S_3(x_2, x_3) \\ S_4(x_3, x_4) \\ \vdots \\ S_{999,999}(x_{999,998}, x_{999,999}) \\ S_{1,000,000}(x_{999,999}, x_{1,000,000}) \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ z_{999,999} \\ z_{1,000,000} \end{bmatrix} = \mathbf{z}$$

Scaling triangular measure transport

Finding **conditional independence** with the Schäfer algorithm:

1. Initiate an empty set Ω
2. Select the state x_i farthest from the Ω or the boundary
3. Set the minimum distance as a neighbourhood radius r
4. This state depends on previous states $x_j \in \Omega$ with radius $2r$.
5. Add x_i to Ω. Go to step 2.



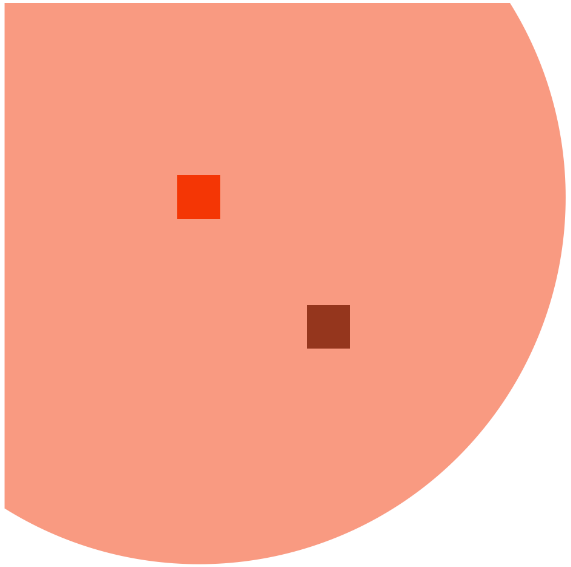


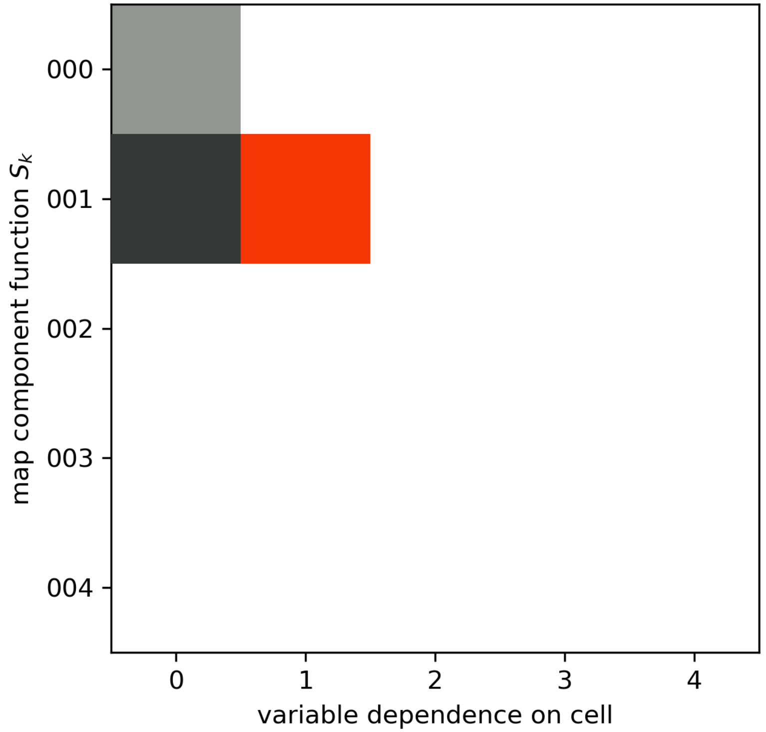
Schäfer, F., Katzfuss, M., & Owhadi, H. (2021). Sparse Cholesky Factorization by Kullback--Leibler Minimization. *SIAM Journal on scientific computing*, *43*(3).

Scaling triangular measure transport

Finding **conditional independence** with the Schäfer algorithm:

1. Initiate an empty set Ω
2. Select the state x_i farthest from the Ω or the boundary
3. Set the minimum distance as a neighbourhood radius r
4. This state depends on previous states $x_j \in \Omega$ with radius $2r$.
5. Add x_i to Ω. Go to step 2.



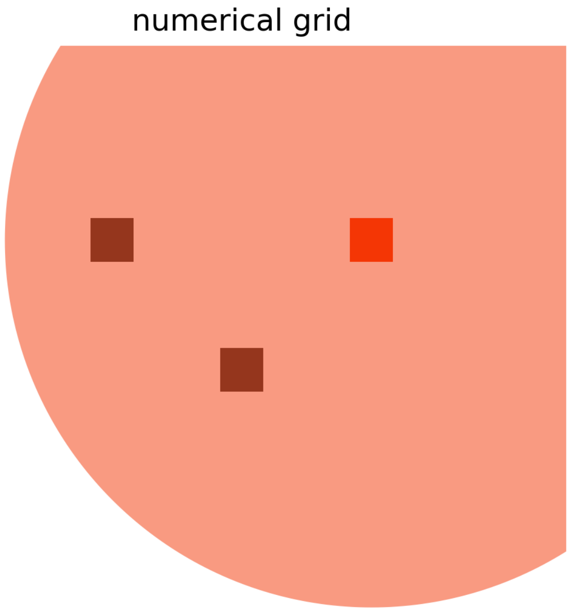


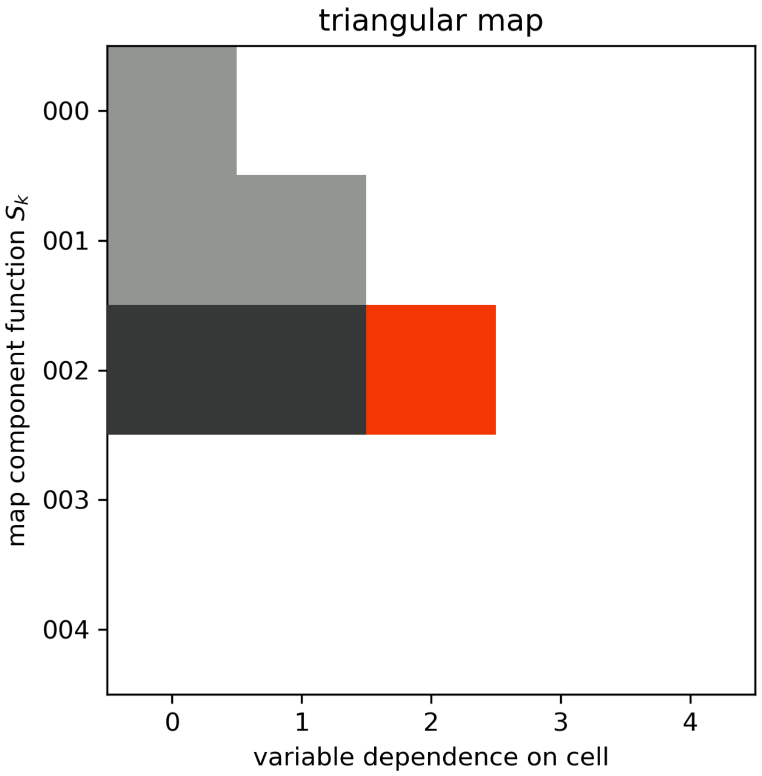
Schäfer, F., Katzfuss, M., & Owhadi, H. (2021). Sparse Cholesky Factorization by Kullback--Leibler Minimization. *SIAM Journal on scientific computing*, *43*(3).

Scaling triangular measure transport

Finding **conditional independence** with the Schäfer algorithm:

1. Initiate an empty set Ω
2. Select the state x_i farthest from the Ω or the boundary
3. Set the minimum distance as a neighbourhood radius r
4. This state depends on previous states $x_j \in \Omega$ with radius $2r$.
5. Add x_i to Ω. Go to step 2.



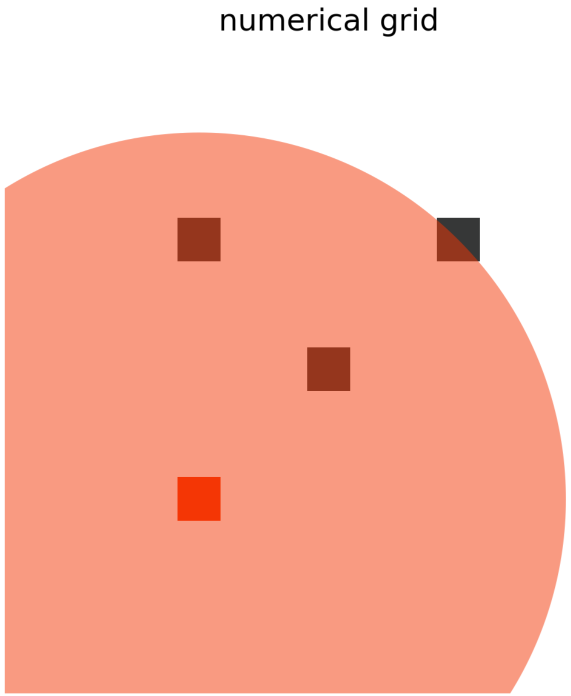


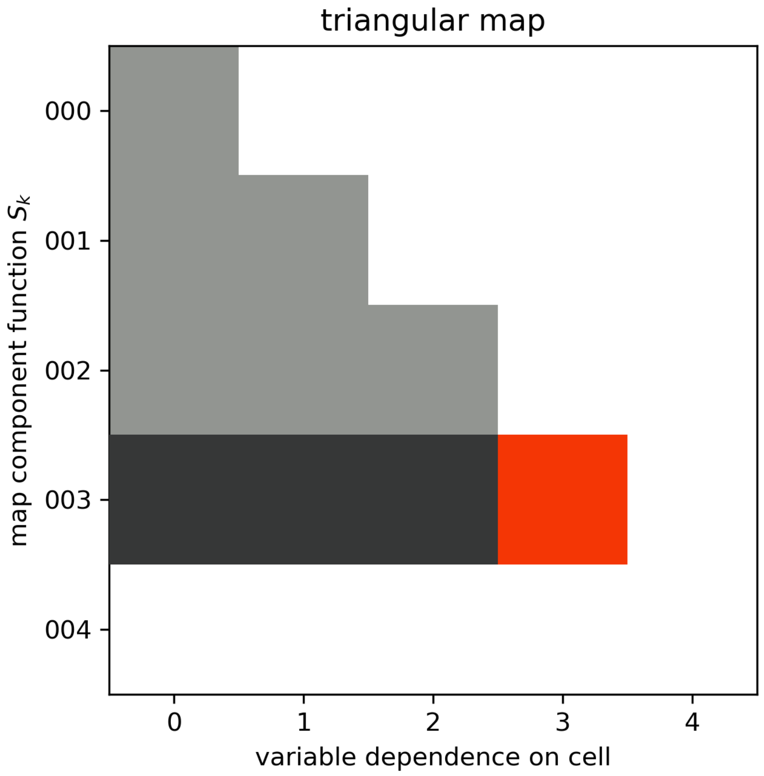
Schäfer, F., Katzfuss, M., & Owhadi, H. (2021). Sparse Cholesky Factorization by Kullback--Leibler Minimization. *SIAM Journal on scientific computing*, *43*(3).

Scaling triangular measure transport

Finding **conditional independence** with the Schäfer algorithm:

1. Initiate an empty set Ω
2. Select the state x_i farthest from the Ω or the boundary
3. Set the minimum distance as a neighbourhood radius r
4. This state depends on previous states $x_j \in \Omega$ with radius $2r$.
5. Add x_i to Ω. Go to step 2.



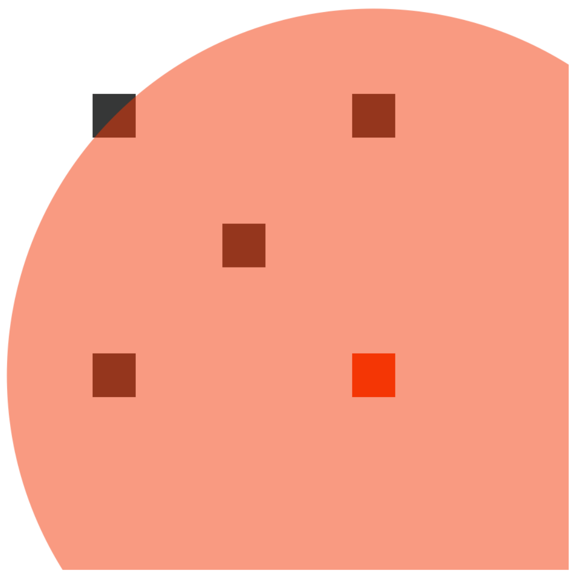


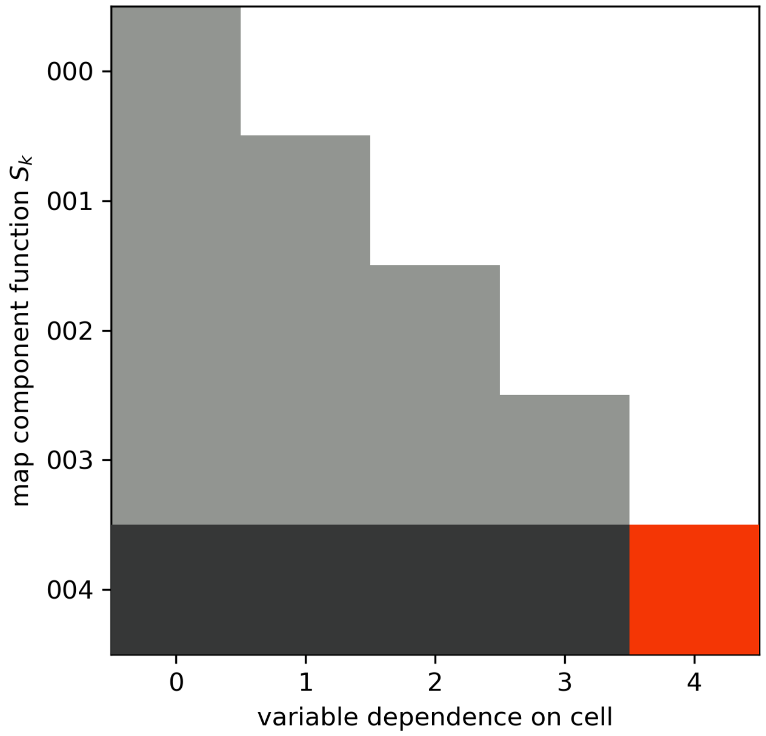
Schäfer, F., Katzfuss, M., & Owhadi, H. (2021). Sparse Cholesky Factorization by Kullback--Leibler Minimization. *SIAM Journal on scientific computing*, *43*(3).

Scaling triangular measure transport

Finding **conditional independence** with the Schäfer algorithm:

1. Initiate an empty set Ω
2. Select the state x_i farthest from the Ω or the boundary
3. Set the minimum distance as a neighbourhood radius r
4. This state depends on previous states $x_j \in \Omega$ with radius $2r$.
5. Add x_i to Ω. Go to step 2.



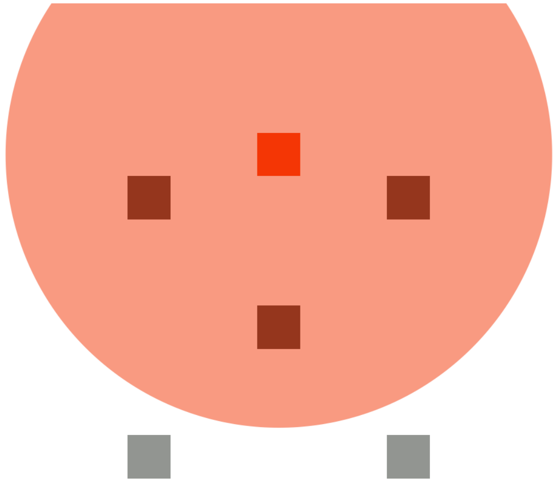


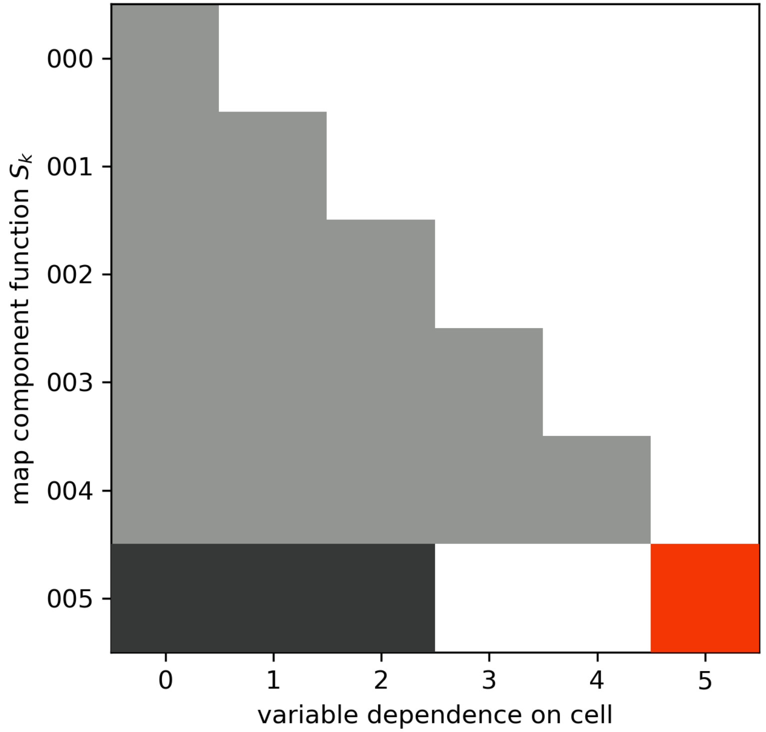
Schäfer, F., Katzfuss, M., & Owhadi, H. (2021). Sparse Cholesky Factorization by Kullback--Leibler Minimization. *SIAM Journal on scientific computing*, *43*(3).

Scaling triangular measure transport

Finding **conditional independence** with the Schäfer algorithm:

1. Initiate an empty set Ω
2. Select the state x_i farthest from the Ω or the boundary
3. Set the minimum distance as a neighbourhood radius r
4. This state depends on previous states $x_j \in \Omega$ with radius $2r$.
5. Add x_i to Ω. Go to step 2.



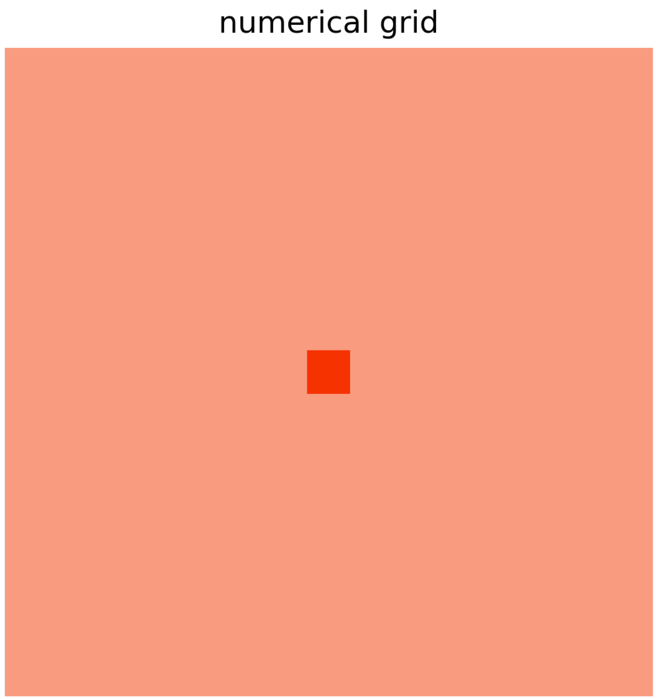


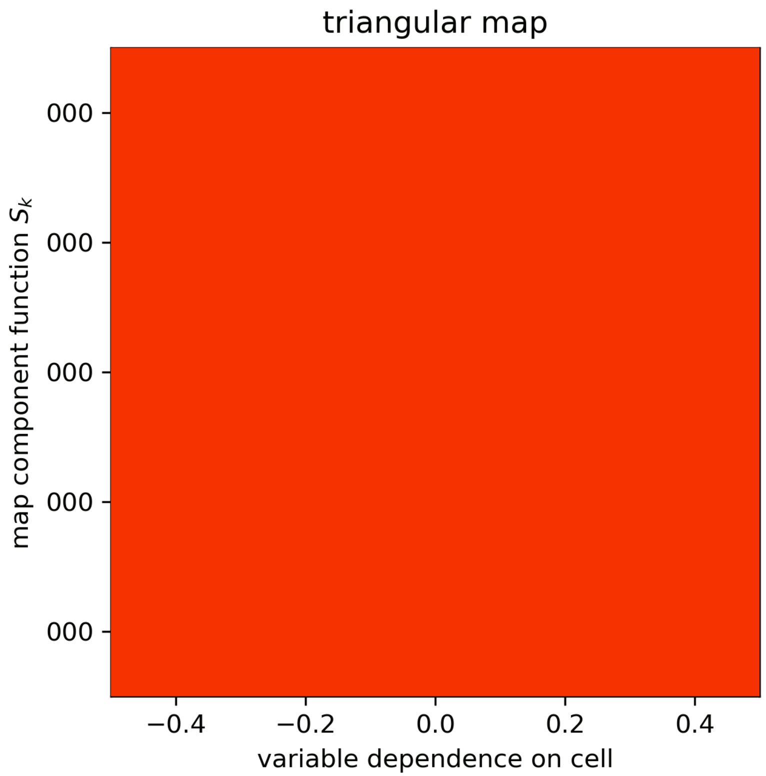
Schäfer, F., Katzfuss, M., & Owhadi, H. (2021). Sparse Cholesky Factorization by Kullback--Leibler Minimization. *SIAM Journal on scientific computing*, *43*(3).

Scaling triangular measure transport

Finding **conditional independence** with the Schäfer algorithm:

1. Initiate an empty set Ω
2. Select the state x_i farthest from the Ω or the boundary
3. Set the minimum distance as a neighbourhood radius r
4. This state depends on previous states $x_j \in \Omega$ with radius $2r$.
5. Add x_i to Ω. Go to step 2.

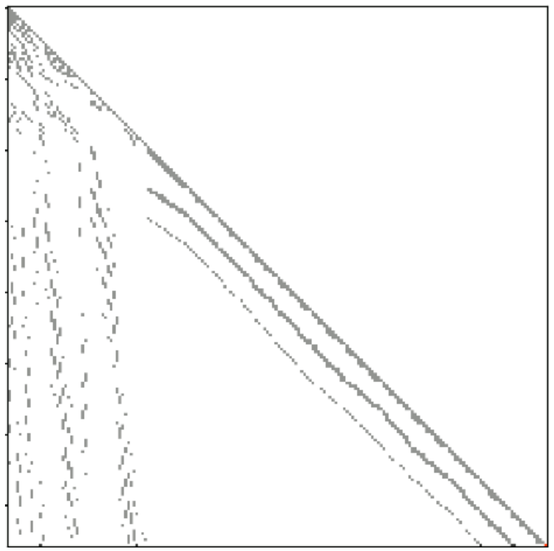




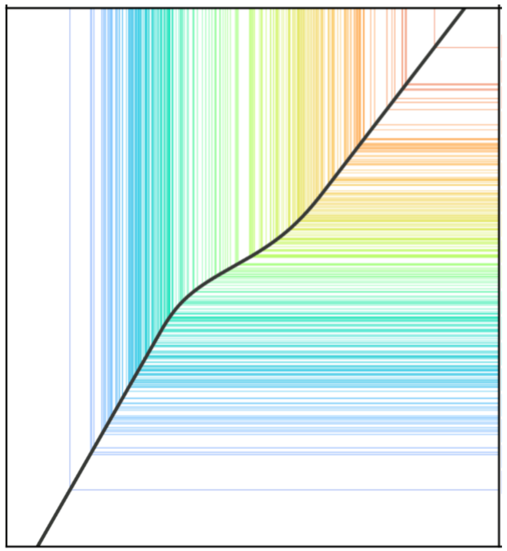
Schäfer, F., Katzfuss, M., & Owhadi, H. (2021). Sparse Cholesky Factorization by Kullback--Leibler Minimization. *SIAM Journal on scientific computing*, *43*(3).

Making triangular maps useful

Applying triangular maps to real systems requires two things:



Sparsity
numerical efficiency
in high-dimensional systems



Parsimony
identifying the
optimal degree of nonlinearity

Draw from (well-known) GAM framework

A **smoothness** and **identifiability** argument leads to optimizing:

$$S_k(\boldsymbol{x}_{1:k}) = \underbrace{\arg\min_{S} \mathcal{J}_k(\boldsymbol{x}_{1:k}, S_k)}_{\text{map objective}} + \underbrace{\lambda \int S_k''(\boldsymbol{x}_{1:k}) d\boldsymbol{x}_{1:k}}_{\text{smoothness penalty}}$$

A **variational** argument (for simplified cases) leads to map components

$$S_k(\boldsymbol{x}_{1:k}; \boldsymbol{c}_k) = \sum_{i=1}^{m} c_{k,i} b_{k,i}(\boldsymbol{x}_{1:k})$$

A **statistical** argument simplifies this by imposing **additivity** and **fewer knots**:

$$S_k(\boldsymbol{x}_{1:k}; \boldsymbol{c}_k) = \boldsymbol{c}_{k,1}\boldsymbol{b}_{k,1}(x_1) + \cdots + \boldsymbol{c}_{k,k}\boldsymbol{b}_{k,k}(x_k)$$

Draw from (well-known) GAM framework

An **information theoretic** argument yields an outer objective:

$$\mathbb{E}_{\pi}\mathbb{E}_{\hat{\pi}}\left[\mathcal{J}_k(\boldsymbol{x}_{1:k},\boldsymbol{\lambda})\right] = \underbrace{\mathbb{E}_{\hat{\pi}}\left[\mathcal{J}_k(\boldsymbol{x}_{1:k},\hat{\boldsymbol{c}}_k(\boldsymbol{\lambda}))\right]}_{\text{negative log likelihood}} + \underbrace{n^{-1}\,\mathrm{trace}\left[\left(\nabla^2_{\boldsymbol{c}_k} J_{\text{pen.}}\right)^{-1}\nabla^2_{\boldsymbol{c}_k} J_{\text{unpen.}}\right]}_{\text{effective degrees of freedom}}$$

where

$$\begin{array}{rl} \hat{\boldsymbol{c}}_k(\boldsymbol{\lambda}) & \text{optimal coefficients } \boldsymbol{k} \text{ for this } \boldsymbol{\lambda} \\ \left(\nabla^2_{\boldsymbol{c}_k} J_{\text{pen.}}\right)^{-1} & \text{inverse of Hessian of penalized objective} \\ \nabla^2_{\boldsymbol{c}_k} J_{\text{unpen.}} & \text{Hessian of unpenalized objective} \end{array}$$

Computation using automatic differentiation and IFT

Remember that $\boldsymbol{c}_k$ and $\boldsymbol{\lambda}$ are entangled by the inner objective

$$\hat{\boldsymbol{c}}_k(\boldsymbol{\lambda}) = \arg\min J_{\text{pen.}}(\boldsymbol{x}_{1:k}; \boldsymbol{c}_k, \boldsymbol{\lambda}).$$

Define the outer objective (not considering entanglement)

$$h\left(\boldsymbol{c}_k, \boldsymbol{\lambda}\right) = J_{\text{unpen.}}(\boldsymbol{x}_{1:k}; \boldsymbol{c}_k) + n^{-1}\,\text{trace}\left[\left(\nabla^2_{\boldsymbol{c}_k} J_{\text{pen.}}\right)^{-1} \nabla^2_{\boldsymbol{c}_k} J_{\text{unpen.}}\right]$$

IFT gives the gradient of the outer objective wrt $\boldsymbol{\lambda}$

$$\frac{d}{d\boldsymbol{\lambda}} \mathbb{E}_{\pi}\mathbb{E}_{\hat{\pi}}\left[\mathcal{J}_k(\boldsymbol{x}_{1:k}, \boldsymbol{c}_k)\right] = \nabla_{\boldsymbol{\lambda}} h - \nabla_{\boldsymbol{\lambda}} h^{\intercal} \left(\nabla_{\boldsymbol{\lambda}\boldsymbol{\lambda}} J(\hat{\boldsymbol{c}}_k, \boldsymbol{\lambda})\right)^{-1} \nabla_{\boldsymbol{c}_k \boldsymbol{\lambda}} J(\hat{\boldsymbol{c}}_k, \boldsymbol{\lambda})$$

Griewank, Andreas and Andrea Walther (2008). *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM.
Kristensen, Kasper et al. (2016). *TMB: Automatic differentiation and Laplace approximation.* In: Journal of Statistical Software 70.i05.

Adapting triangular measure transport

One option for parameterizing a map component function S_k is a **separable formulation**:

$$S_3(x_1, x_2, x_3) = \underbrace{\boldsymbol{c}_{3,1}^\intercal \boldsymbol{b}_{3,1}(x_1) + \boldsymbol{c}_{3,2}^\intercal \boldsymbol{b}_{3,2}(x_2)}_{\text{nonmonotone part}} + \underbrace{\tilde{\boldsymbol{c}}_{3,3}^\intercal \boldsymbol{b}_{3,3}(x_3)}_{\text{monotone part}}$$

$\boldsymbol{b}_{k,i}(x_i)$	basis function evaluations	vector
$\boldsymbol{c}_{k,i}, i < m$	coefficients	vector
$\tilde{\boldsymbol{c}}_{k,m}$	(positive) coefficients	vector

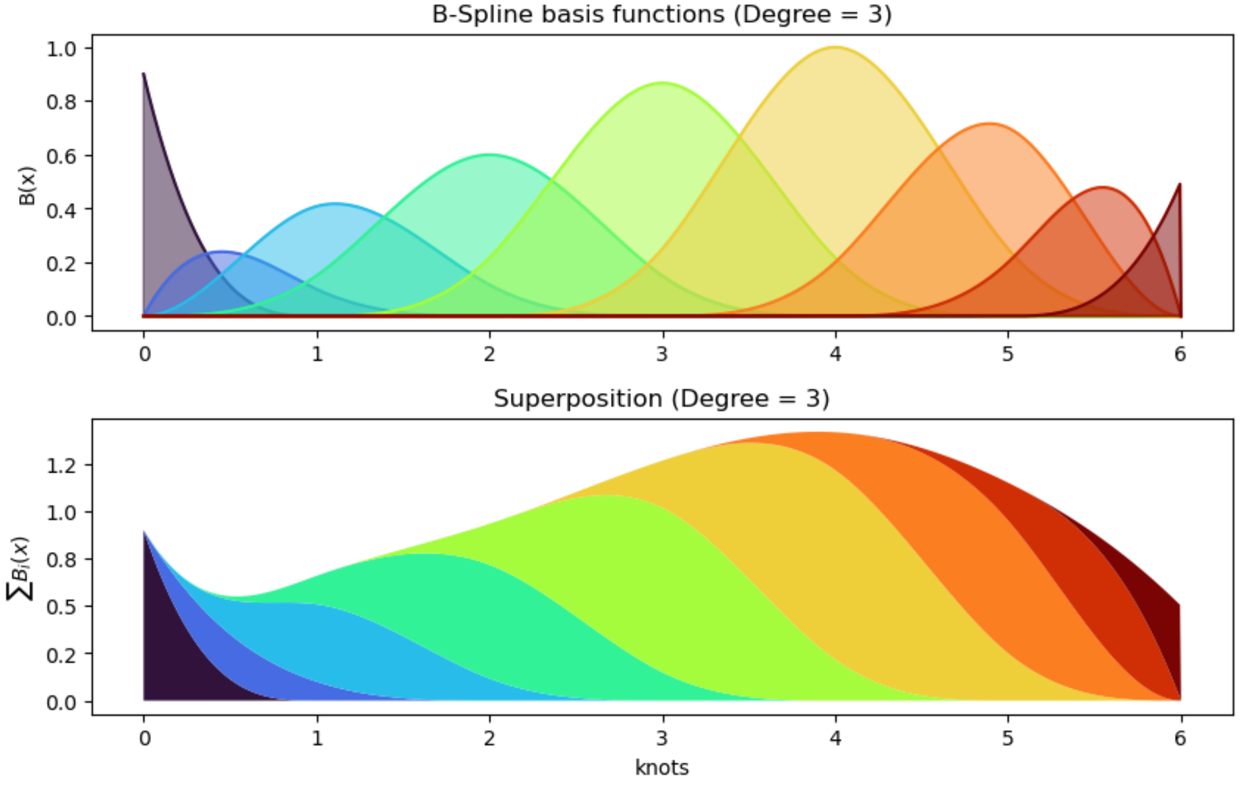
In this presentation, we choose **P-Splines** as basis functions $\boldsymbol{b}_{k,i}$.

Adapting triangular measure transport

A **P-Spline** is a B-Spline (basis spline) that penalizes differences between the coefficients at neighbouring knots.

- This promotes **smoothness**
- This controls the **degrees of freedom**

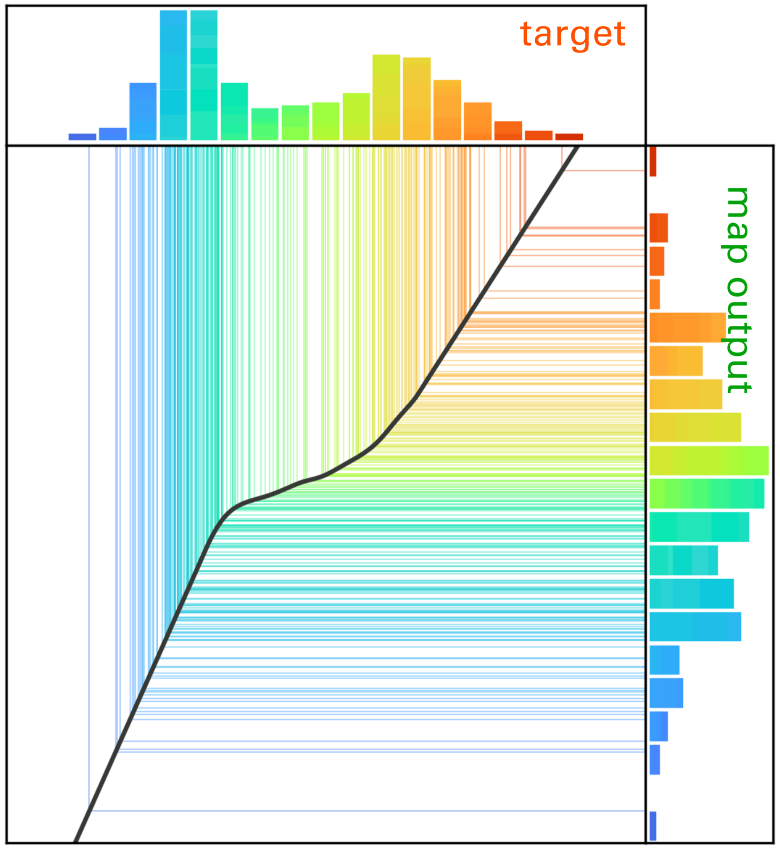
Optimizing the smoothing penalty hyper-parameter for the Akaike Information Criterion (AIC) finds the **optimal trade-off** between nonlinearity and simplicity.

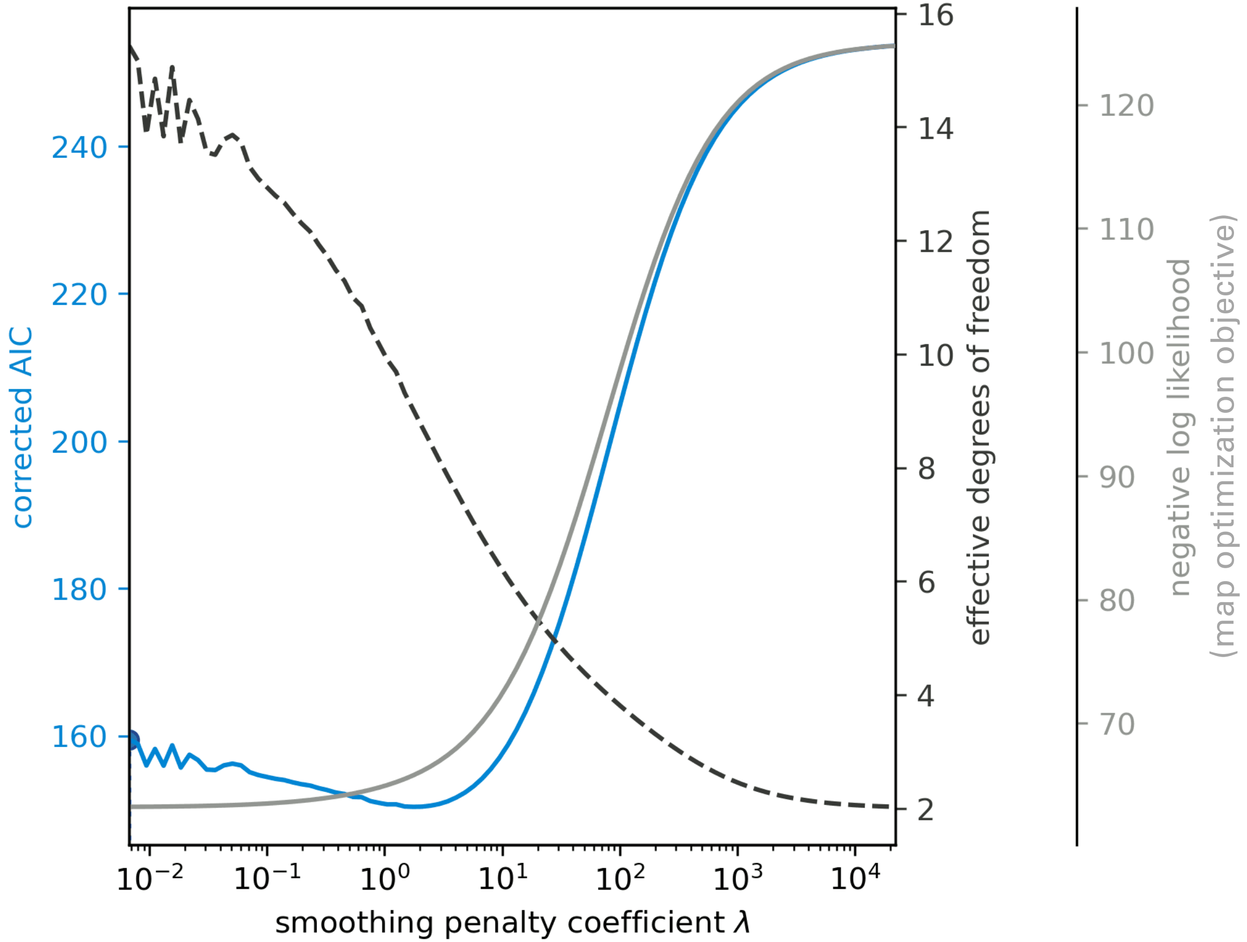


Eilers, P. H., & Marx, B. D. (2021). *Practical smoothing: The joys of P-splines*. Cambridge University Press.

Adapting triangular measure transport

By adjusting the smoothing coefficient, the outer optimization controls the **complexity** and **degree of nonlinearity** of the map component function.



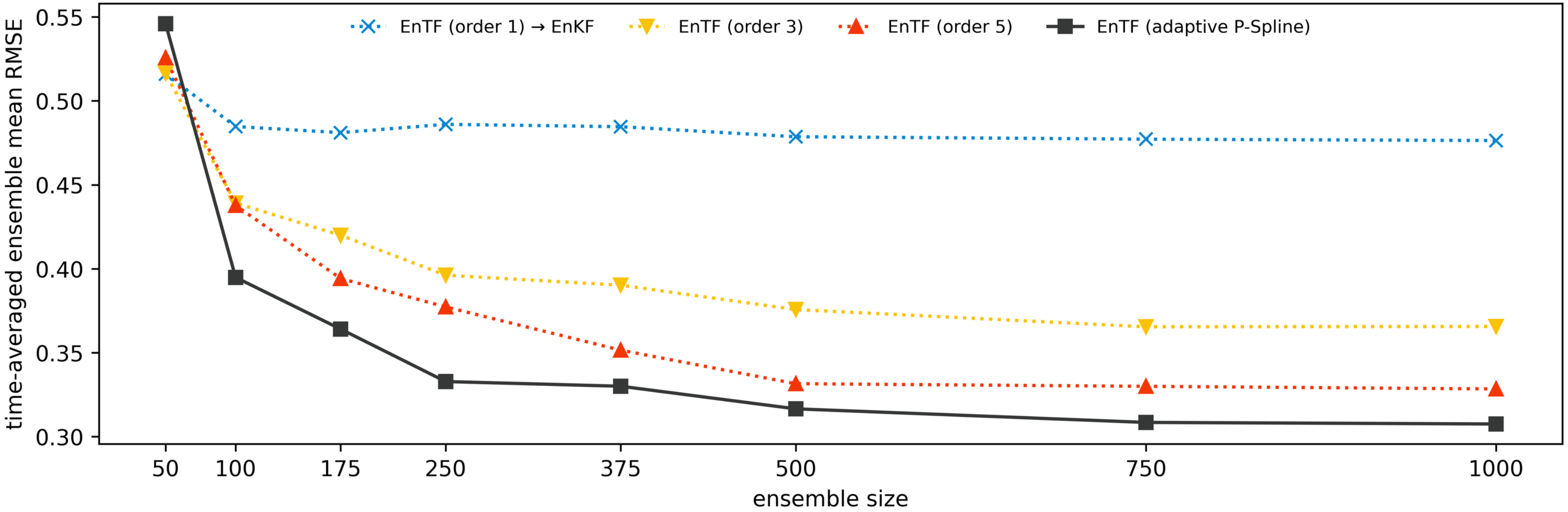


Numerical results: Lorenz-63

Test case: Lorenz-63

timesteps:	1000
model error std:	5 / ensemble size
obs error std:	2
time step length:	0.1

Numerical results: Lorenz-63



Non-adaptive EnTF simulation

- Increasing levels of map complexity
- Use of L2-regularization and inflation
- Best combination per ensemble size
- Averaged across 10 random seeds

Adaptive P-Spline EnTF

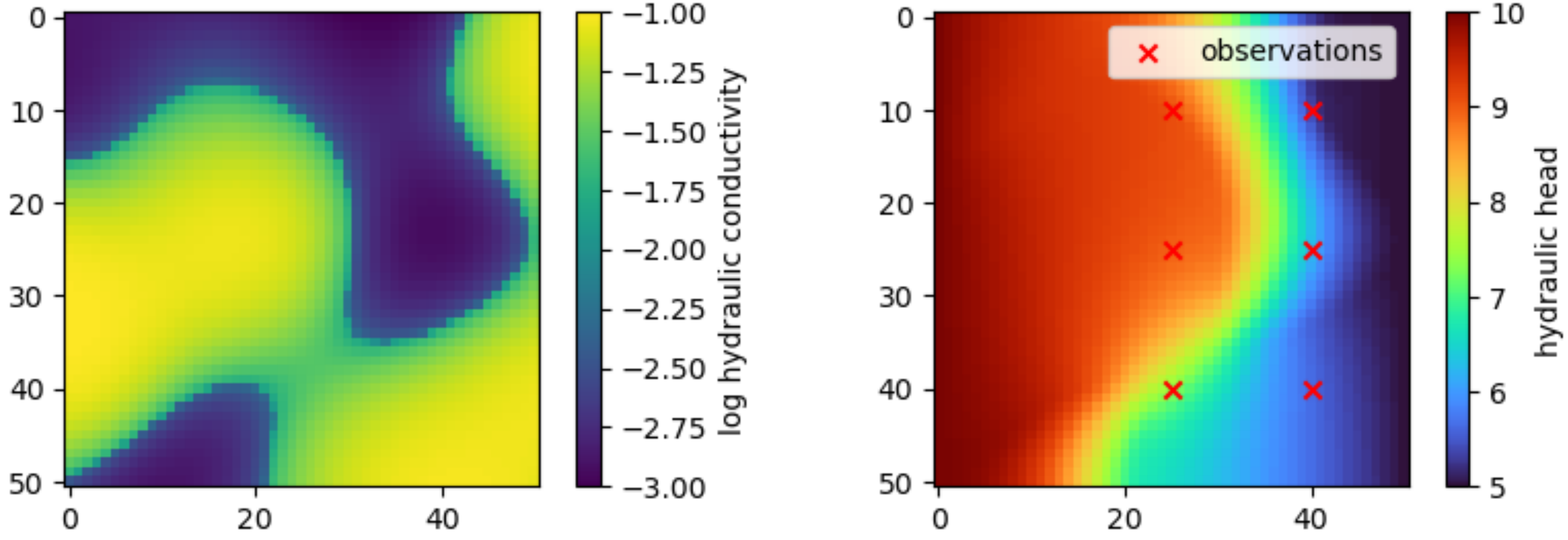
- Number of basis functions set to $N^{1/3}$
- Outer optimization seeks optimal lambda over 10 time steps, then uses median

Ramgraber, M., Baptista, R., McLaughlin, D., & Marzouk, Y. (2023). Ensemble transport smoothing. Part II: Nonlinear updates. *Journal of Computational Physics: X*, *17*, 100133.

Numerical results: groundwater model

Test case: Groundwater model

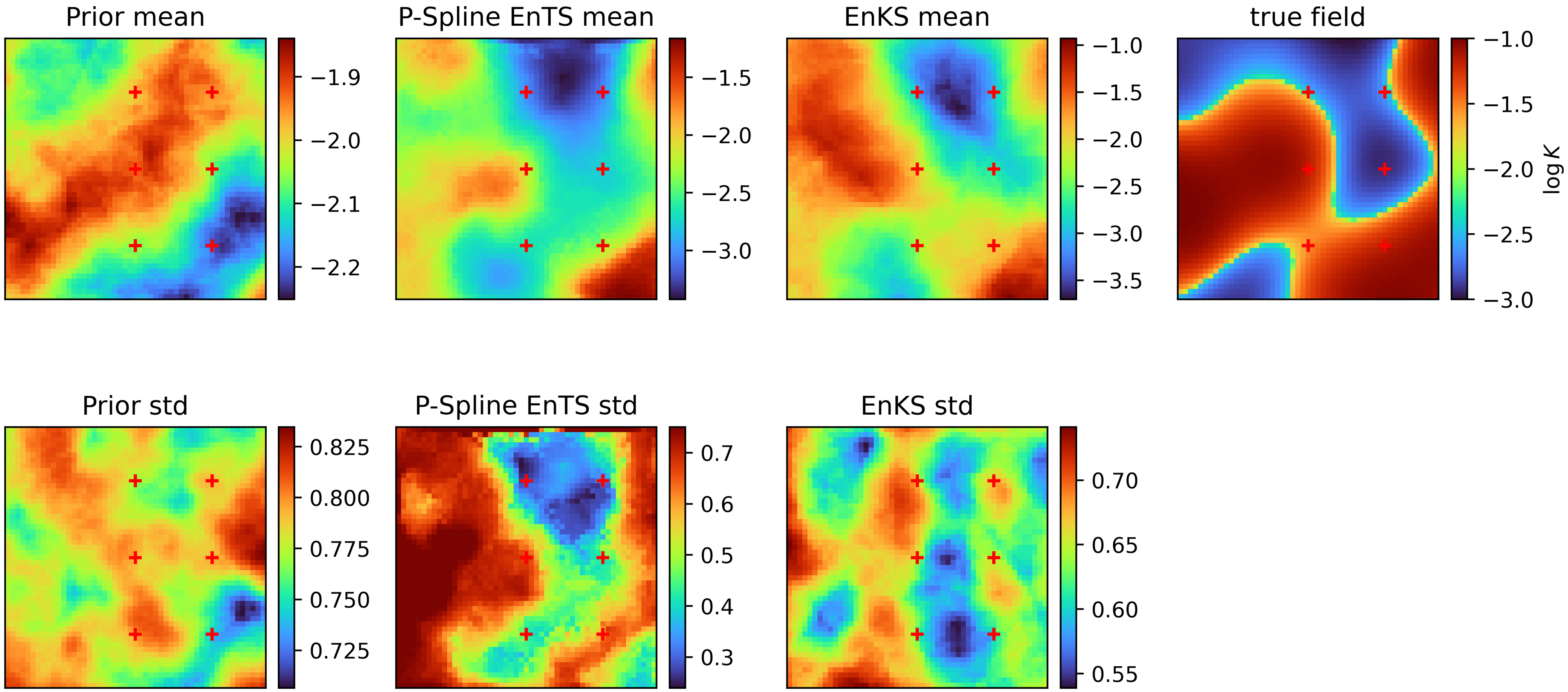
grid size:	51 by 51	**obs error std**:	0.01
ensemble size:	100	**prior**:	strongly bimodal K
dynamics:	steady-state	**boundaries**:	fixed head (left: 10 / right: 5)



The true conductivity field and the resulting hydraulic head distribution

Numerical results: groundwater model

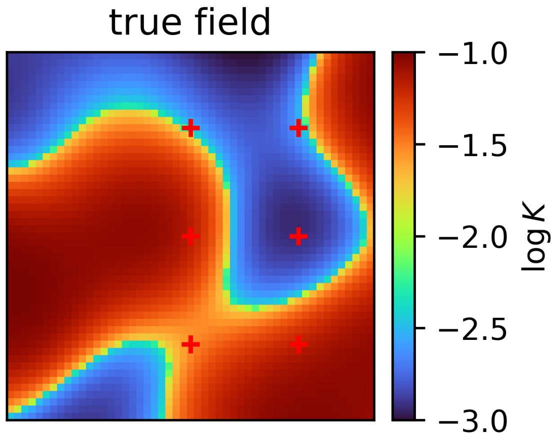
We have also tried using this approach to infer **hydraulic conductivities** based on observed pressure values. In this model, hydraulic conductivities are **strongly bimodal**.

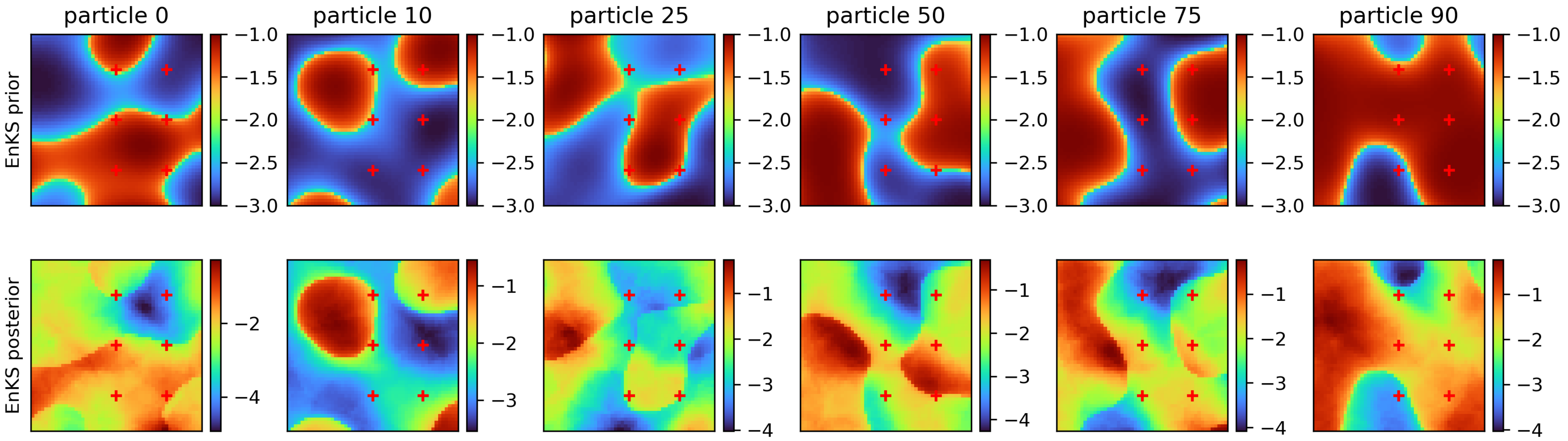


Numerical results: groundwater model

Analyzing posterior ensemble realizations reveals that:

- the **EnKS** blurs out the geological features

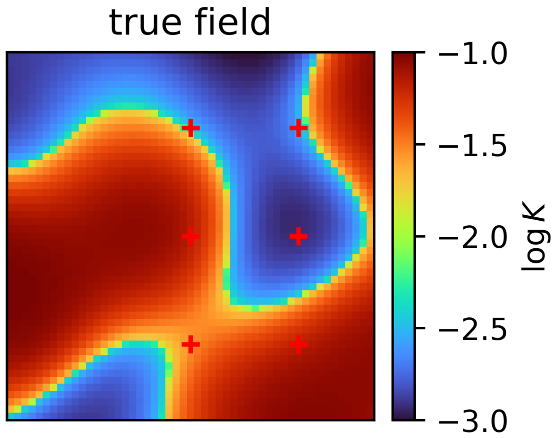


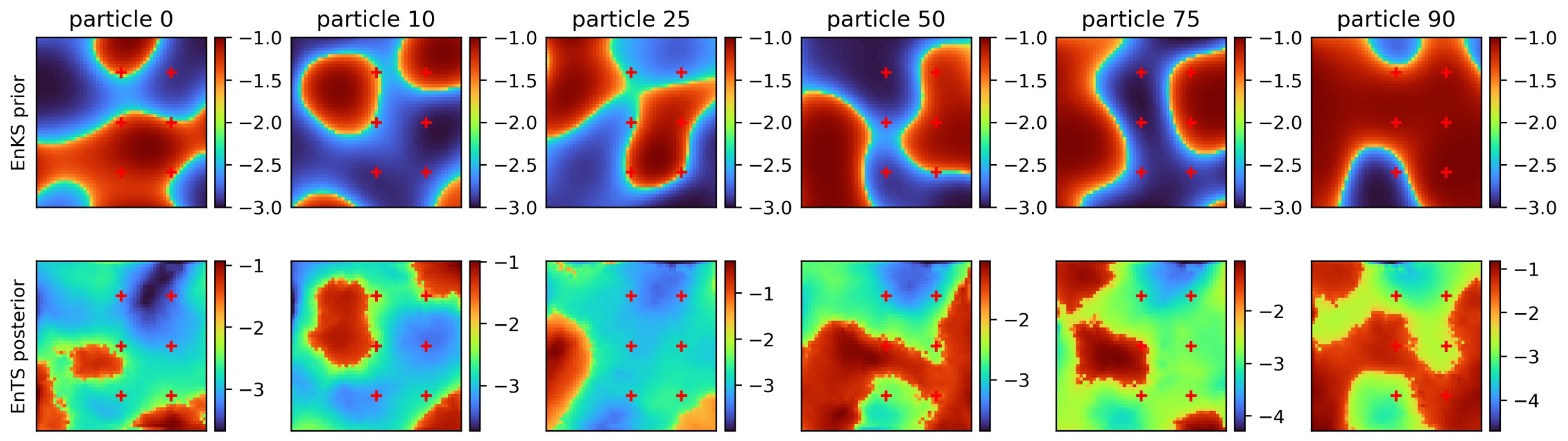


Numerical results: groundwater model

Analyzing posterior ensemble realizations reveals that:

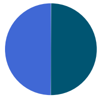
- the **EnKS** blurs out the geological features
- the **P-spline EnTS** preserves the bimodality



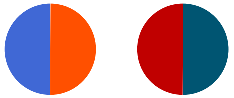


Numerical results: groundwater model

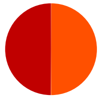
We can also see this effect if we compare log K values in different cells of the grid. Note that we have **four clusters** of hydraulic conductivity combinations:



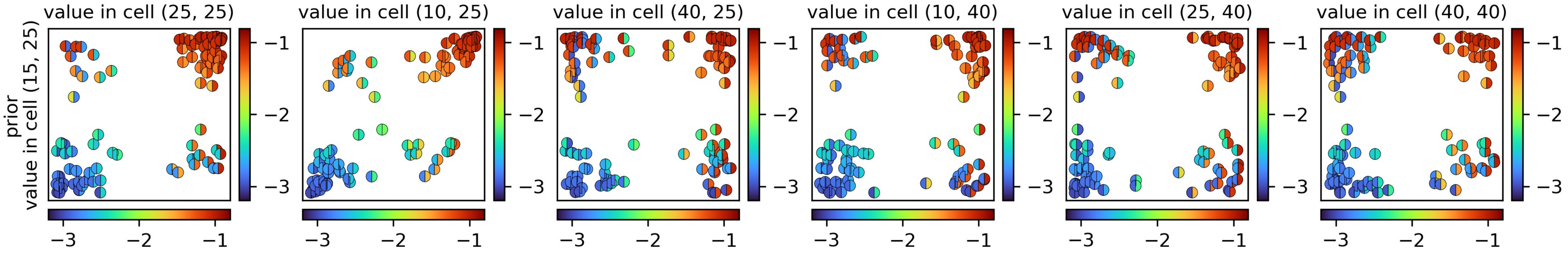
Both cells:
low K



Mixed:
low K and high K



Both cells:
high K



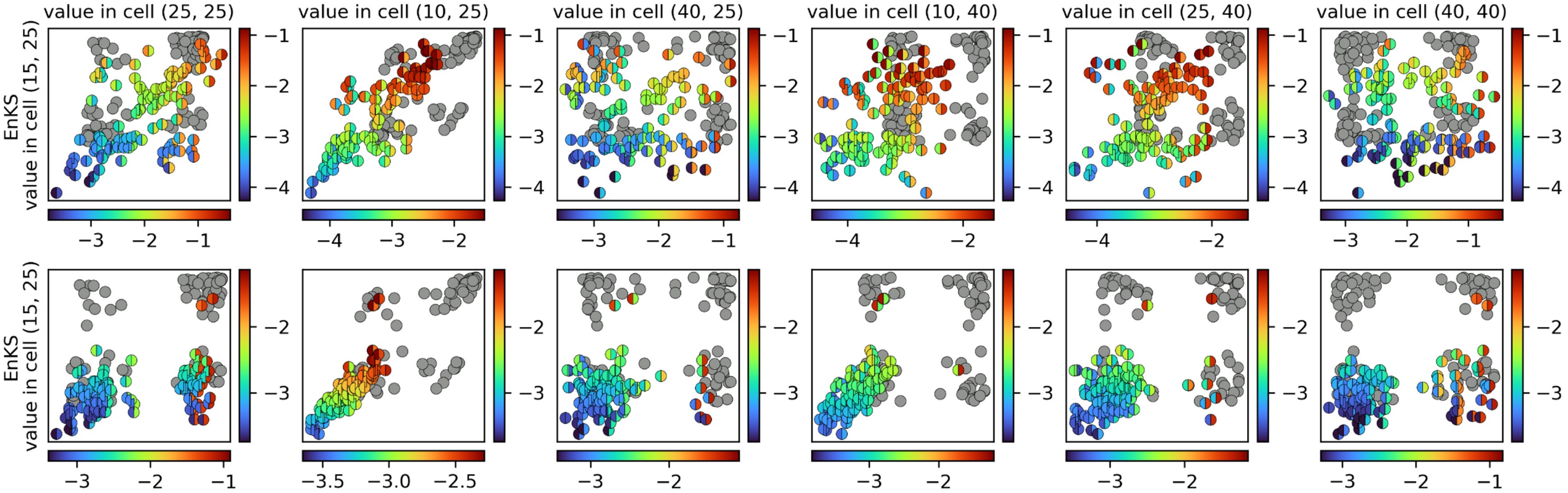
Numerical results: groundwater model

We can also see this effect if we compare log K values in different cells of the grid. Note that we have **four clusters** of hydraulic conductivity combinations:

Both cells: low K

Mixed: low K and high K

Both cells: high K



Outlook

Preliminary results are very promising:

- The Schäfer Cholesky algorithm identifies a suitable **variable ordering** and sufficient **conditional independence**
- We demonstrated that this allows us to **scale triangular transport** methods for application in larger, grid-based systems
- The resulting algorithm **preserves non-Gaussian features** significantly better than conventional linear methods, leading to better inferences.

Next steps:

- Increase the numerical efficiency (parallelization, faster optimization)
- Further develop the adaptation algorithm
- Explore alternative orderings and update strategies

Related:

Berent's Ensemble Information Filter paper

https://arxiv.org/abs/2501.09016

Thank you

for your attention!

Tutorial:

A friendly introduction to triangular transport

https://arxiv.org/abs/2503.21673

Acknowledgements:

The research leading to these results has received funding from the Dutch Research Council NWO under Talent Programme grant VI.Veni.232.140, and Equinor's DaTeS project.

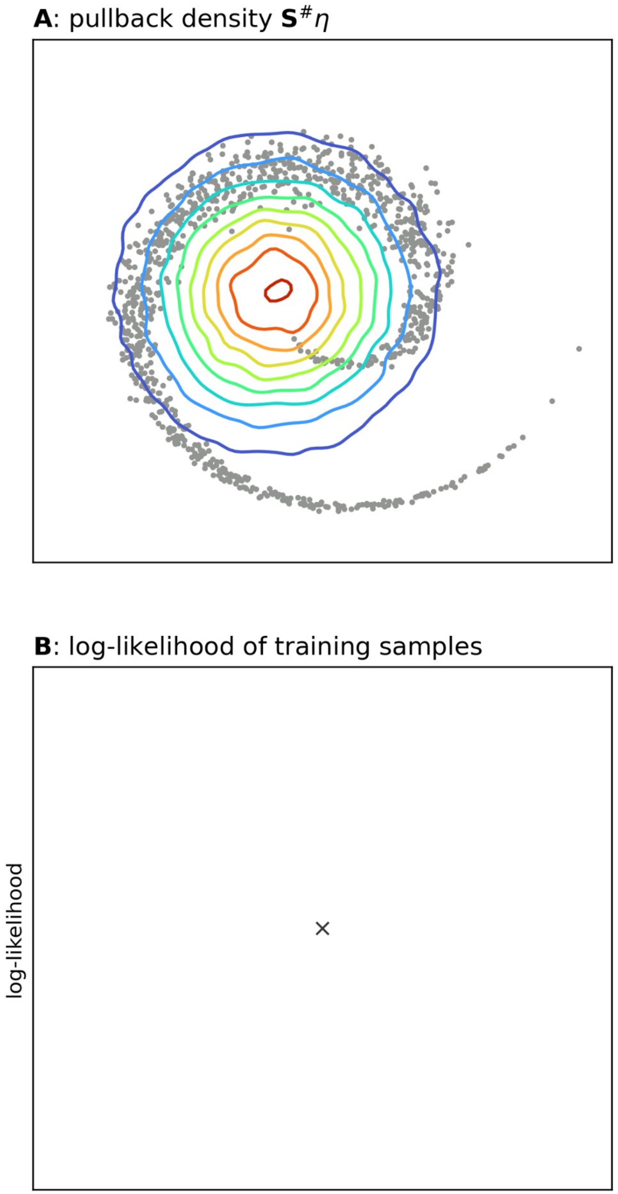
References:

Spantini, A., Baptista, R., & Marzouk, Y. (2022). Coupling techniques for nonlinear ensemble filtering. SIAM Review, 64(4), 921-953.

Ramgraber, M., Baptista, R., McLaughlin, D., & Marzouk, Y. (2023). Ensemble transport smoothing. Part II: Nonlinear updates. *Journal of Computational Physics: X*, *17*, 100133.

Eilers, P. H., & Marx, B. D. (2021). *Practical smoothing: The joys of P-splines*. Cambridge University Press.

Triangular map optimization



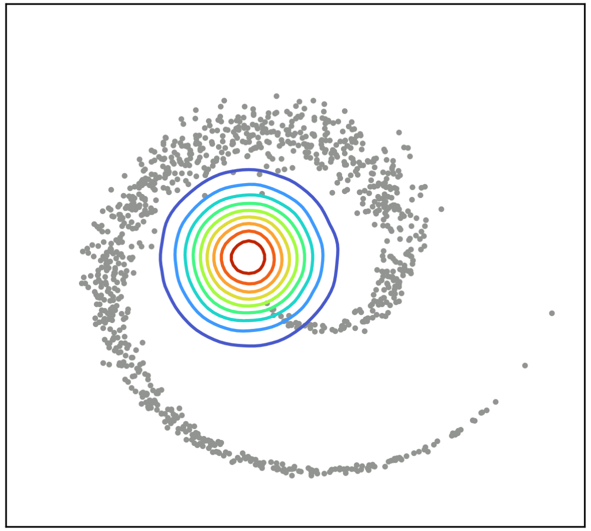
Maps from samples seek to maximize the log-likelihood of the target samples $\mathbf{x}$ over the map's pullback density $\mathbf{S}^{\sharp}\eta$:

$$\mathbf{S} \in \arg\max_{\mathbf{S}\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \log \mathbf{S}^{\#}\eta\left(\mathbf{x}^{i}\right)$$

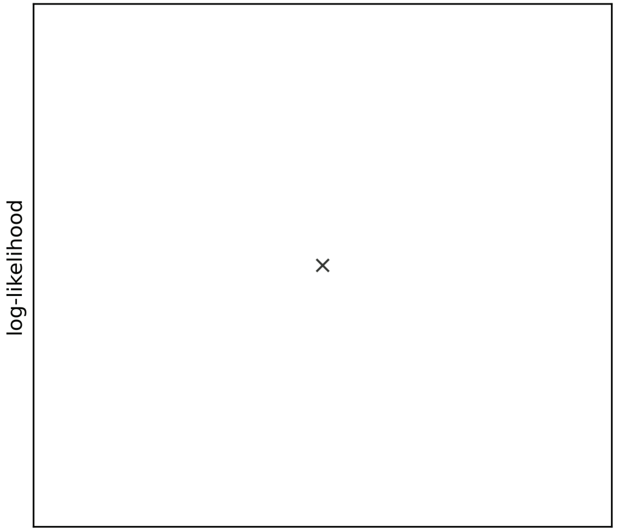
Spantini, A., Baptista, R., & Marzouk, Y. (2022). Coupling techniques for nonlinear ensemble filtering. SIAM Review, 64(4), 921-953.

Triangular map optimization

A: pullback density $\mathbf{S}^{\#}\eta$



B: log-likelihood of training samples



Maps from samples seek to maximize the log-likelihood of the target samples $\mathbf{x}$ over the map's pullback density $\mathbf{S}^{\sharp}\eta$:

$$\mathbf{S} \in \arg\max_{\mathbf{S}\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \log \mathbf{S}^{\#}\eta\left(\mathbf{x}^{i}\right)$$

This can be further decomposed into **individual optimization objectives** S_k for the map component functions:

$$\mathcal{J}\left(\boldsymbol{c}_k\right) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}\left(S_k\left(\boldsymbol{c}_k, \boldsymbol{x}_{1:k}\right)\right)^2 - \log\frac{\partial S_k\left(\boldsymbol{c}_k, \boldsymbol{x}_{1:k}\right)}{\partial x_k}\right]$$

Spantini, A., Baptista, R., & Marzouk, Y. (2022). Coupling techniques for nonlinear ensemble filtering. SIAM Review, 64(4), 921-953.