Sequential inference using low-dimensional couplings

Youssef Marzouk
joint work with Alessio Spantini and Daniele Bigoni

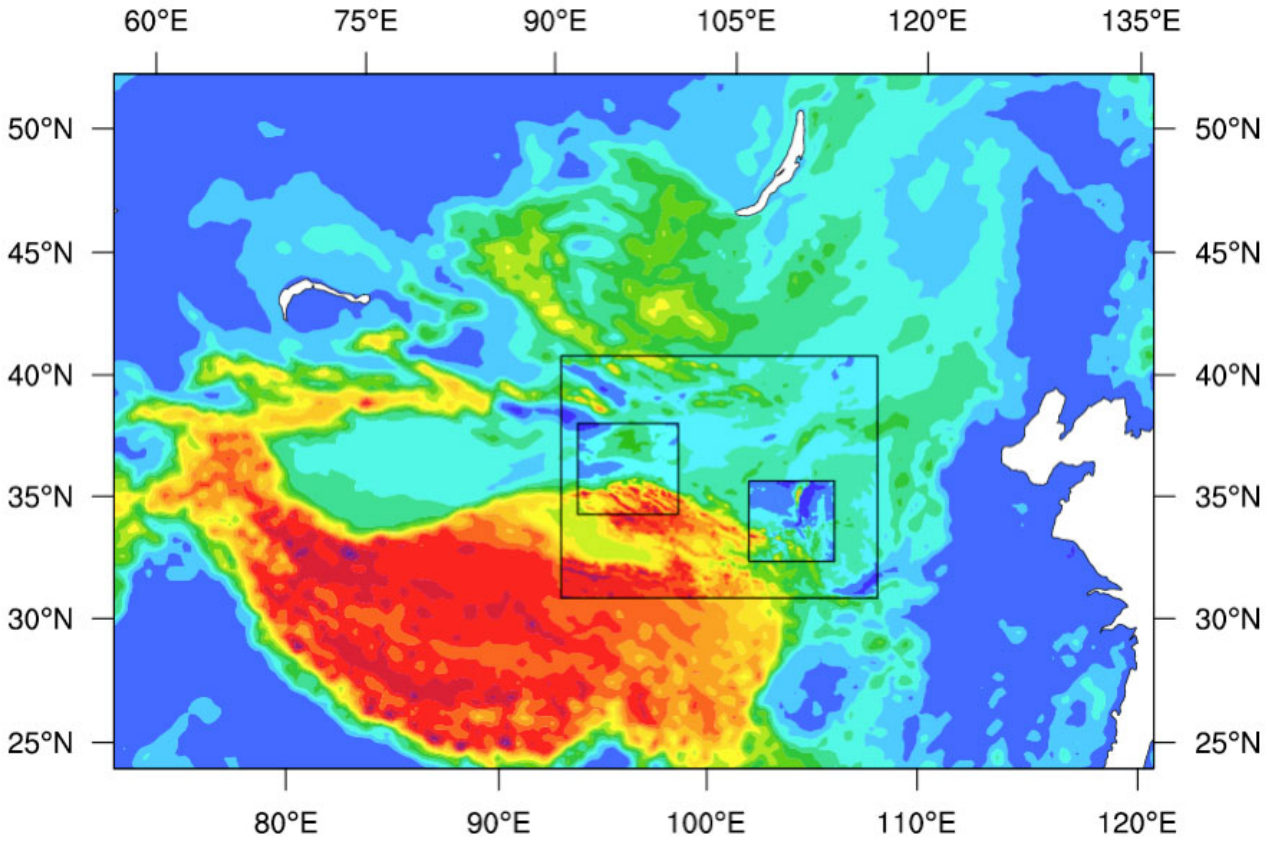
Department of Aeronautics and Astronautics
Center for Computational Engineering
Statistics and Data Science Center

Massachusetts Institute of Technology
http://uqgroup.mit.edu

Support from DOE ASCR, NSF, DARPA

12 November 2017

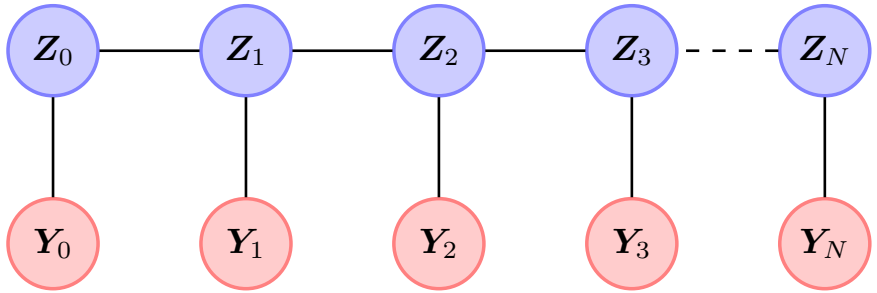
Sequential Bayesian inference



- State estimation (e.g., *filtering* and *smoothing*) or *joint state and parameter estimation*, in a Bayesian setting
  - Need **recursive, online** algorithms for characterizing the posterior distribution

Bayesian filtering and smoothing

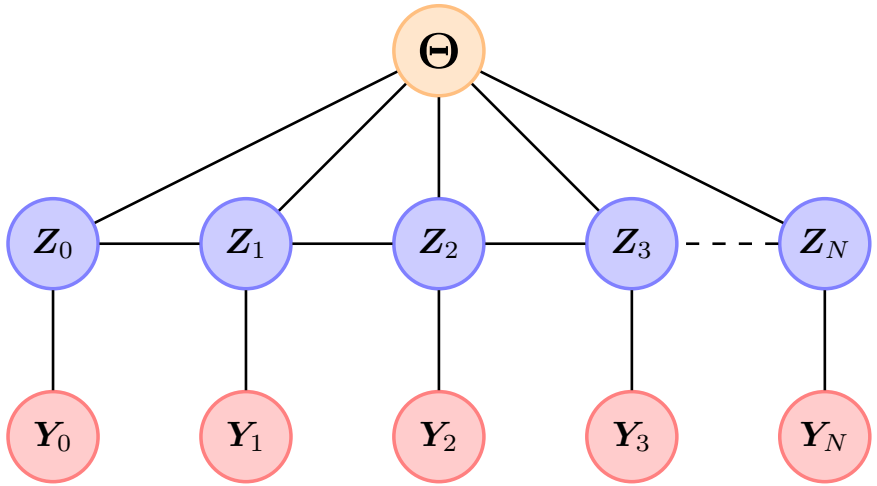
- **Nonlinear/non-Gaussian** state-space model:
  - Transition density $\pi_{\mathbf{Z}_k|\mathbf{Z}_{k-1}}$
  - Observation density (likelihood) $\pi_{\mathbf{Y}_k|\mathbf{Z}_k}$



- Interested in **recursively** updating the full Bayesian solution:
  $\pi_{\mathbf{Z}_{0:k}\,|\,\mathbf{y}_{0:k}} \rightarrow \pi_{\mathbf{Z}_{0:k+1}\,|\,\mathbf{y}_{0:k+1}}$ (smoothing)
- Or focus on approximating the filtering distribution:
  $\pi_{\mathbf{Z}_k\,|\,\mathbf{y}_{0:k}} \rightarrow \pi_{\mathbf{Z}_{k+1}\,|\,\mathbf{y}_{0:k+1}}$ (marginals of the full Bayesian solution)

Joint state and parameter inference

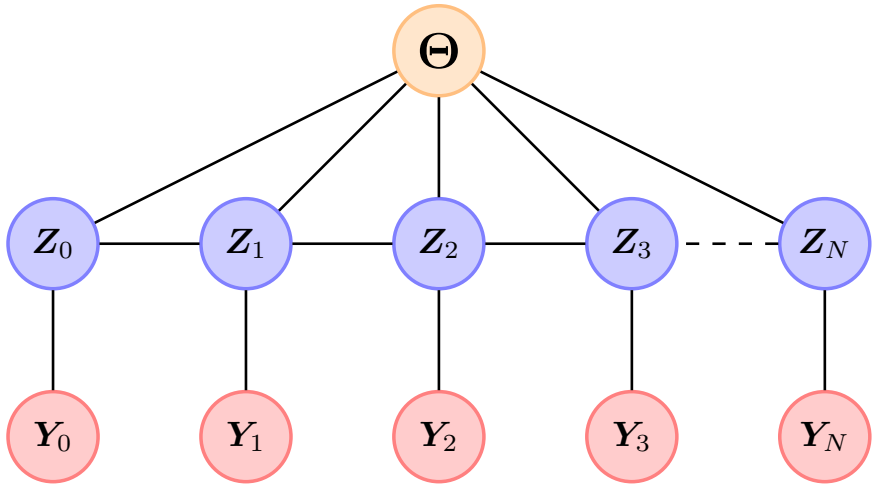
- **Nonlinear/non-Gaussian** state-space model with static params Θ:
  - Transition density $\pi_{\mathbf{Z}_k|\mathbf{Z}_{k-1},\Theta}$
  - Observation density (likelihood) $\pi_{\mathbf{Y}_k|\mathbf{Z}_k}$



- Again, interested in recursively updating the full Bayesian solution:
  $\pi_{\mathbf{Z}_{0:k},\Theta\,|\,\mathbf{y}_{0:k}} \rightarrow \pi_{\mathbf{Z}_{0:k+1},\Theta\,|\,\mathbf{y}_{0:k+1}}$
  (smoothing + *sequential parameter inference*)

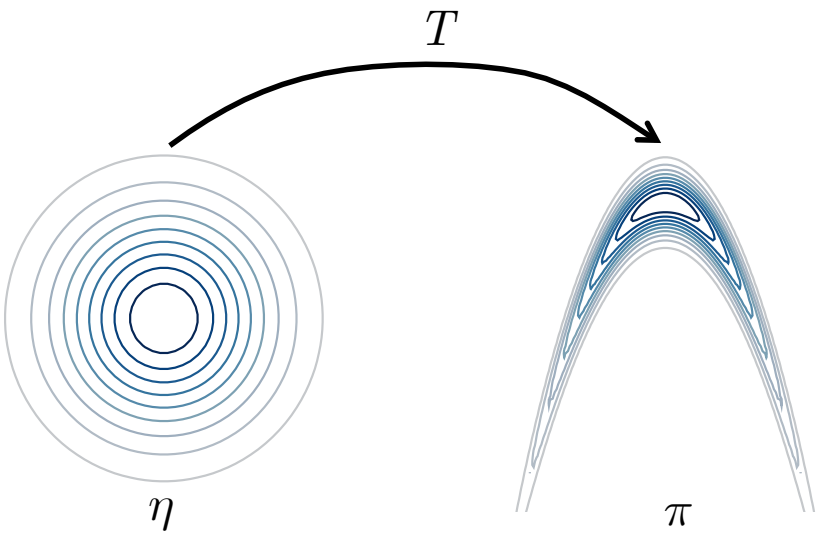
Joint state and parameter inference

- **Nonlinear/non-Gaussian** state-space model with static params Θ:
  - Transition density $\pi_{\mathbf{Z}_k|\mathbf{Z}_{k-1},\Theta}$
  - Observation density (likelihood) $\pi_{\mathbf{Y}_k|\mathbf{Z}_k}$



- Again, interested in recursively updating the full Bayesian solution:
  $\pi_{\mathbf{Z}_{0:k},\Theta\,|\,\mathbf{y}_{0:k}} \rightarrow \pi_{\mathbf{Z}_{0:k+1},\Theta\,|\,\mathbf{y}_{0:k+1}}$
  (smoothing + *sequential parameter inference*)
- Relate these goals to notions of **coupling and transport...**

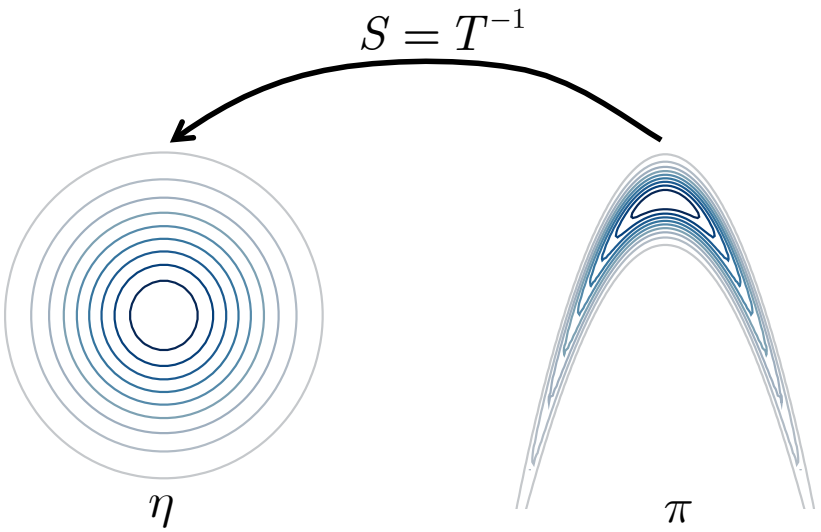
Deterministic couplings of probability measures



Core idea

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \eta = \pi$

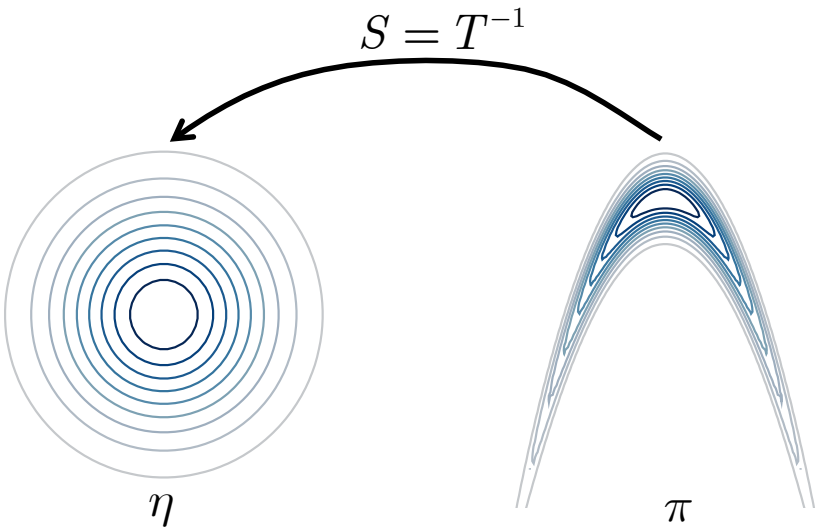
Deterministic couplings of probability measures



Core idea

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \eta = \pi$
- Equivalently, find $S = T^{-1}$ such that $S_\sharp \pi = \eta$

Deterministic couplings of probability measures



Core idea

- Choose a *reference distribution* η (e.g., standard Gaussian)
- Seek a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ such that $T_\sharp \eta = \pi$
- Equivalently, find $S = T^{-1}$ such that $S_\sharp \pi = \eta$
- Might be content satisfying these conditions only *approximately*...

Topics for this talk

1. Variational approaches for joint state–parameter inference
2. A class of nonlinear filters induced by local couplings

Choice of transport map

A useful building block is the **Knothe-Rosenblatt rearrangement:**

$$T(x) = \begin{bmatrix} T^1(x_1) \\ T^2(x_1, x_2) \\ \vdots \\ T^n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

- Exists and is unique (up to ordering) under mild conditions on η, π
- Jacobian determinant easy to evaluate
- Inverse map $S = T^{-1}$ also lower triangular
- "Exposes" marginals, will enable conditional sampling...

Choice of transport map

A useful building block is the **Knothe-Rosenblatt rearrangement:**

$$T(x) = \begin{bmatrix} T^1(x_1) \\ T^2(x_1, x_2) \\ \vdots \\ T^n(x_1, x_2, \ldots, x_n) \end{bmatrix}$$

- Exists and is unique (up to ordering) under mild conditions on η, π
- Jacobian determinant easy to evaluate
- Inverse map $S = T^{-1}$ also lower triangular
- "Exposes" marginals, will enable conditional sampling...
- Numerical approximations can employ a *monotone parameterization*, guaranteeing $\partial_{x_k} T^k > 0$ for arbitrary a_k, b_k

$$T^k(x_1, \ldots, x_k) = a_k(x_1, \ldots, x_{k-1}) + \int_0^{x_k} \exp(b_k(x_1, \ldots, x_{k-1}, w))\, dw$$

Part 1: variational inference

Variational characterization of the direct map T [Moselhy & M 2012]:

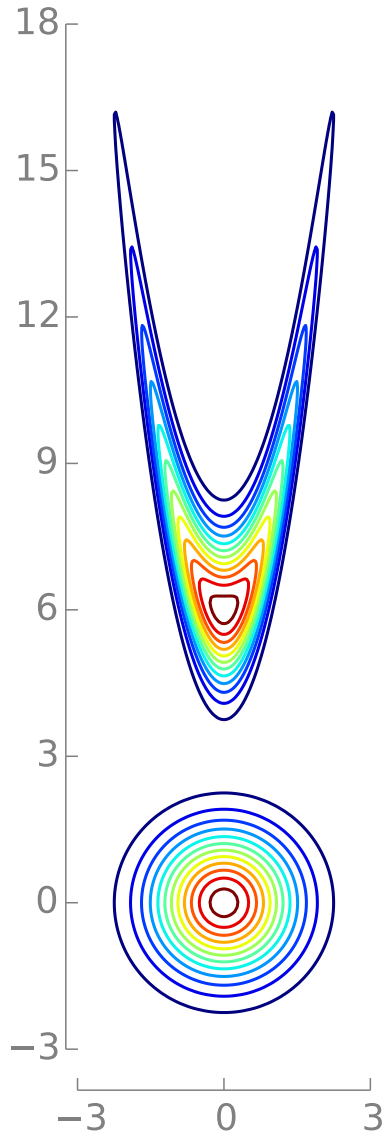
$$\min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(T_\sharp \eta \,||\, \pi) = \min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(\eta \,||\, T_\sharp^{-1} \pi)$$

- $\mathcal{T}_\triangle$ is the set of monotone lower **triangular** maps
  - Contains the *Knothe-Rosenblatt* rearrangement
- Expectation is with respect to the *reference* measure η
  - Compute via, e.g., Monte Carlo, QMC, quadrature
- Use unnormalized evaluations of π and its gradients
- No MCMC or importance sampling

Simple example

$$\min_{T} \mathbb{E}_{\eta}[-\log \pi \circ T - \sum_{k} \log \partial_{x_k} T^k]$$

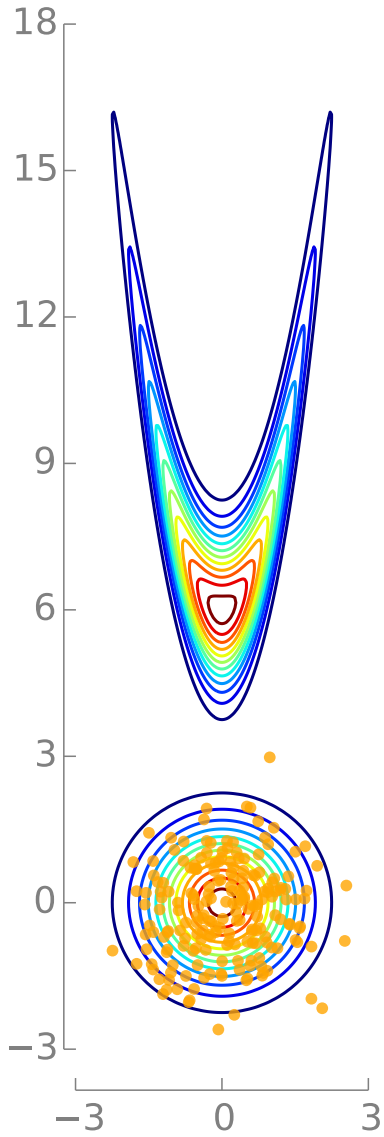
- Parameterized map $T \in \mathcal{T}_{\triangle}^{h} \subset \mathcal{T}_{\triangle}$
- Optimize over coefficients of functions $(a_k)_{k=1}^{n}$, $(b_k)_{k=1}^{n}$
- Use gradient-based optimization
- Approximate $\mathbb{E}_{\eta}[g] \approx \sum_i w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference



Simple example

$$\min_{T} \mathbb{E}_{\eta}[-\log \pi \circ T - \sum_{k} \log \partial_{x_k} T^k]$$

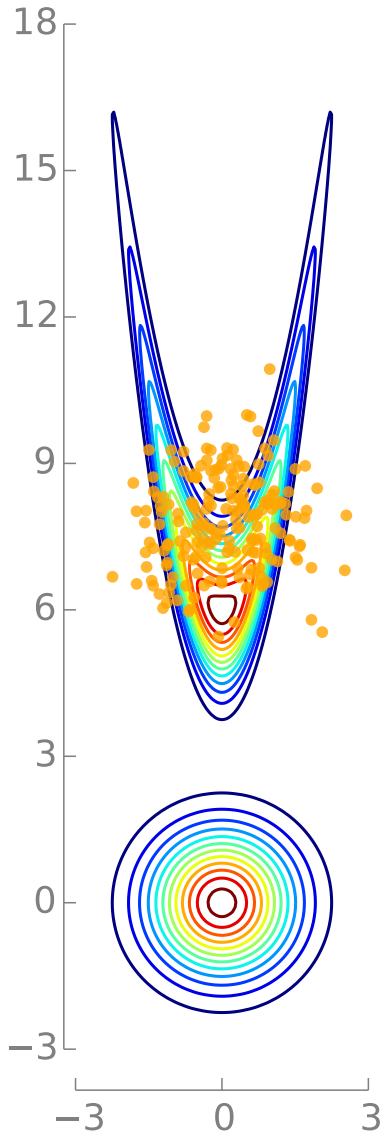
- Parameterized map $T \in \mathcal{T}_{\triangle}^{h} \subset \mathcal{T}_{\triangle}$
- Optimize over coefficients of functions $(a_k)_{k=1}^{n}$, $(b_k)_{k=1}^{n}$
- Use gradient-based optimization
- Approximate $\mathbb{E}_{\eta}[g] \approx \sum_i w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference



Simple example

$$\min_{T} \mathbb{E}_{\eta}[-\log \pi \circ T - \sum_{k} \log \partial_{x_k} T^k]$$

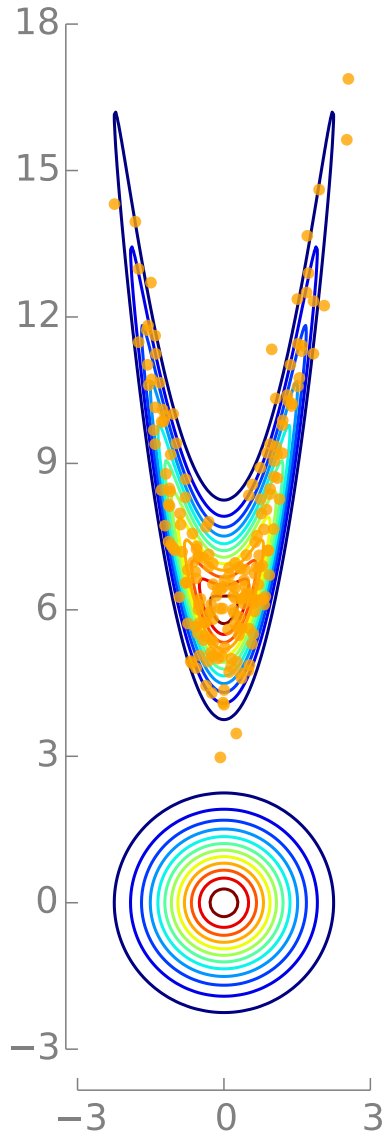
- Parameterized map $T \in \mathcal{T}_{\triangle}^{h} \subset \mathcal{T}_{\triangle}$
- Optimize over coefficients of functions $(a_k)_{k=1}^{n}$, $(b_k)_{k=1}^{n}$
- Use gradient-based optimization
- Approximate $\mathbb{E}_{\eta}[g] \approx \sum_i w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference



Simple example

$$\min_{T} \mathbb{E}_{\eta}[-\log \pi \circ T - \sum_{k} \log \partial_{x_k} T^k]$$

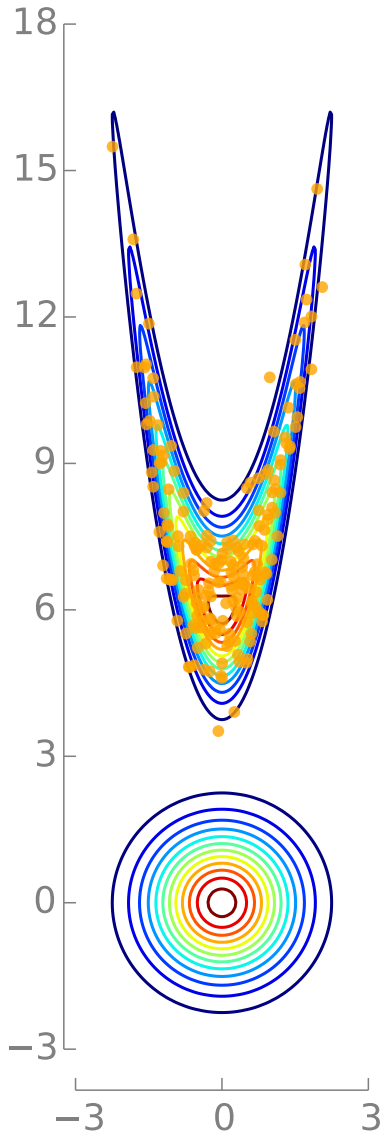
- Parameterized map $T \in \mathcal{T}_{\triangle}^{h} \subset \mathcal{T}_{\triangle}$
- Optimize over coefficients of functions $(a_k)_{k=1}^{n}$, $(b_k)_{k=1}^{n}$
- Use gradient-based optimization
- Approximate $\mathbb{E}_{\eta}[g] \approx \sum_i w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference



Simple example

$$\min_{T} \mathbb{E}_{\eta}[-\log \pi \circ T - \sum_{k} \log \partial_{x_k} T^k]$$

- Parameterized map $T \in \mathcal{T}_{\triangle}^{h} \subset \mathcal{T}_{\triangle}$
- Optimize over coefficients of functions $(a_k)_{k=1}^{n}$, $(b_k)_{k=1}^{n}$
- Use gradient-based optimization
- Approximate $\mathbb{E}_{\eta}[g] \approx \sum_i w_i \, g(\mathbf{x}_i)$
- The posterior is in the tail of the reference



Useful features

- **Move** samples; don't just reweigh them
- *Independent* and *cheap* samples: $x_i \sim \eta \Rightarrow T(x_i)$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(\, T_\sharp \eta \,||\, \pi \,) \approx \frac{1}{2} \mathbb{V}\text{ar}_\eta \left[\log \frac{\eta}{T_\sharp^{-1} \bar{\pi}} \right]$$

Useful features

- **Move** samples; don't just reweigh them
- *Independent* and *cheap* samples: $x_i \sim \eta \Rightarrow T(x_i)$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(\, T_\sharp \eta \,||\, \pi \,) \approx \frac{1}{2} \mathbb{V}\mathrm{ar}_\eta \left[\log \frac{\eta}{T_\sharp^{-1} \bar{\pi}} \right]$$

- Can either accept bias or reduce it by:
  - Increasing the complexity of the map T
  - Sampling the pullback $T_\sharp^{-1}\pi$ using MCMC or importance sampling

Useful features

- **Move** samples; don't just reweigh them
- *Independent* and *cheap* samples: $x_i \sim \eta \Rightarrow T(x_i)$
- Clear convergence criterion, even with unnormalized target density:

$$\mathcal{D}_{KL}(\, T_\sharp \eta \,||\, \pi \,) \approx \frac{1}{2} \mathbb{V}\mathrm{ar}_\eta \left[\log \frac{\eta}{T_\sharp^{-1} \bar{\pi}} \right]$$

- Can either accept bias or reduce it by:
  - Increasing the complexity of the map T
  - Sampling the pullback $T_\sharp^{-1}\pi$ using MCMC or importance sampling
- The KR parameterization is just **one of many** variational constructions for continuous transport (cf. Stein variational gradient descent [Liu & Wang 2016], normalizing flows [Rezende & Mohamed 2015], Gibbs flows [Heng *et al.* 2015], etc.)

Low-dimensional structure

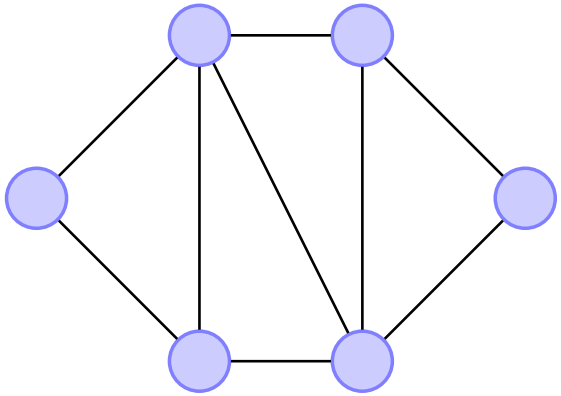
- **Key challenge:** maps in high dimensions
  - Major bottleneck: **representation** of the map, e.g., cardinality of the map basis
- How to make the construction/representation of **high-dimensional** transports tractable?

Low-dimensional structure

- **Key challenge:** maps in high dimensions
  - Major bottleneck: **representation** of the map, e.g., cardinality of the map basis
- How to make the construction/representation of **high-dimensional** transports tractable?
- Main idea: exploit **Markov structure** of the target distribution

Markov random fields

- Let $Z_1, \ldots, Z_n$ be random variables with joint density $\pi > 0$



$$(i,j) \notin \mathcal{E} \quad \text{iff} \quad Z_i \perp\!\!\!\perp Z_j \,|\, \mathbf{Z}_{\mathcal{V}\setminus\{i,j\}}$$

- $\mathcal{G}$ encodes **conditional independence** (an I-map for π)
- Choice of the probabilistic model $\Longrightarrow$ graphical structure

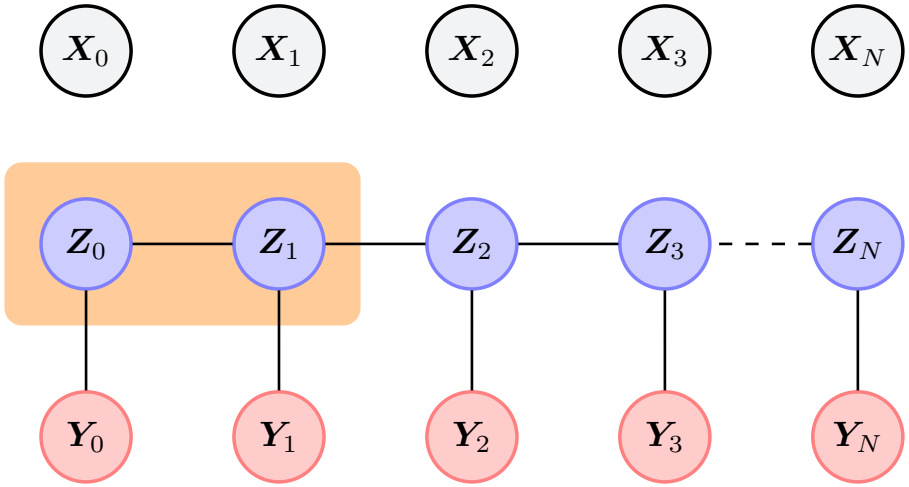
Decomposable transport maps

- **Definition:** a decomposable transport is a map $T = T_1 \circ \cdots \circ T_k$ that factorizes as the composition of finitely many maps of low effective dimension that are triangular (up to a permutation), e.g.,

$$T(\mathbf{x}) = \underbrace{\begin{bmatrix} A_1(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \\ B_1(\mathbf{x}_2, \mathbf{x}_3) \\ C_1(\mathbf{x}_3) \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \mathbf{x}_6 \end{bmatrix}}_{T_1} \circ \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ A_2(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5) \\ B_2(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5) \\ C_2(\mathbf{x}_4, \mathbf{x}_5) \\ D_2(\mathbf{x}_5) \\ \mathbf{x}_6 \end{bmatrix}}_{T_2} \circ \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ A_3(\mathbf{x}_4) \\ B_3(\mathbf{x}_4, \mathbf{x}_5) \\ C_3(\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6) \end{bmatrix}}_{T_3}$$

- **Theorem:** [Spantini et al. (2017)] Decomposable graphical models for π lead to decomposable direct maps T, provided that $\eta(\mathbf{x}) = \prod_i \eta(x_i)$

Application to state-space models (chain graph)



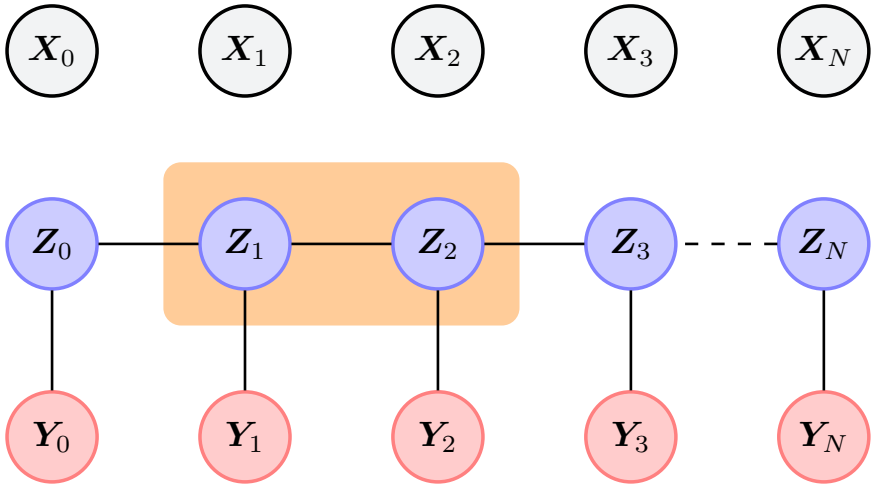
- Compute $\mathfrak{M}_0 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.

$$\mathfrak{M}_0(\mathbf{x}_0, \mathbf{x}_1) = \begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \end{bmatrix}$$

- Reference: $\eta_{\mathbf{X}_0} \eta_{\mathbf{X}_1}$
- Target: $\pi_{\mathbf{Z}_0}\, \pi_{\mathbf{Z}_1|\mathbf{Z}_0}\, \pi_{\mathbf{Y}_0|\mathbf{Z}_0}\, \pi_{\mathbf{Y}_1|\mathbf{Z}_1}$
- $\dim(\mathfrak{M}_0) \simeq 2 \times \dim(\mathbf{Z}_0)$

$$T_0(\mathbf{x}) = \begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

Second step: compute another 2-D map



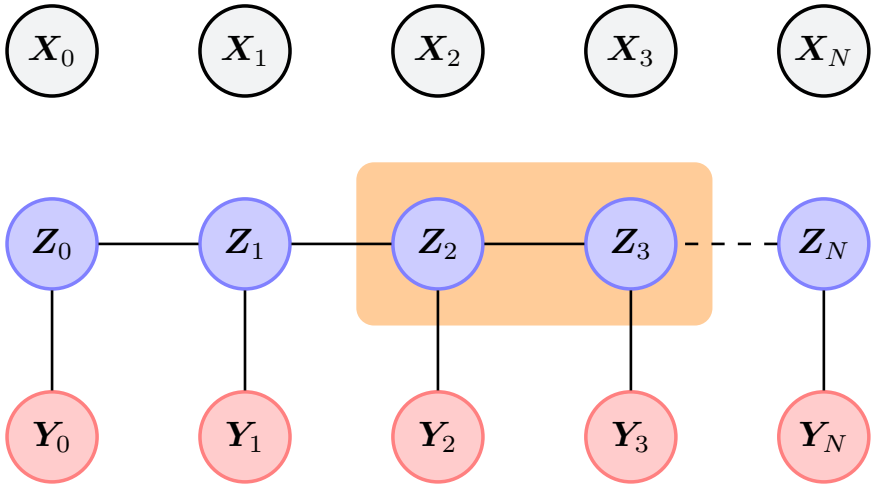
- Compute $\mathfrak{M}_1 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.

$$\mathfrak{M}_1(\mathbf{x}_1, \mathbf{x}_2) = \begin{bmatrix} A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \end{bmatrix}$$

- Reference: $\eta_{\mathbf{X}_1} \eta_{\mathbf{X}_2}$
- Target: $\eta_{\mathbf{X}_1}\, \pi_{\mathbf{Y}_2|\mathbf{Z}_2}\, \pi_{\mathbf{Z}_2|\mathbf{Z}_1}(\cdot|\, B_0\,(\cdot)\,)$
- Uses only one component of $\mathfrak{M}_0$

$$T_1(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_0 \\ A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

Proceed recursively forward in time



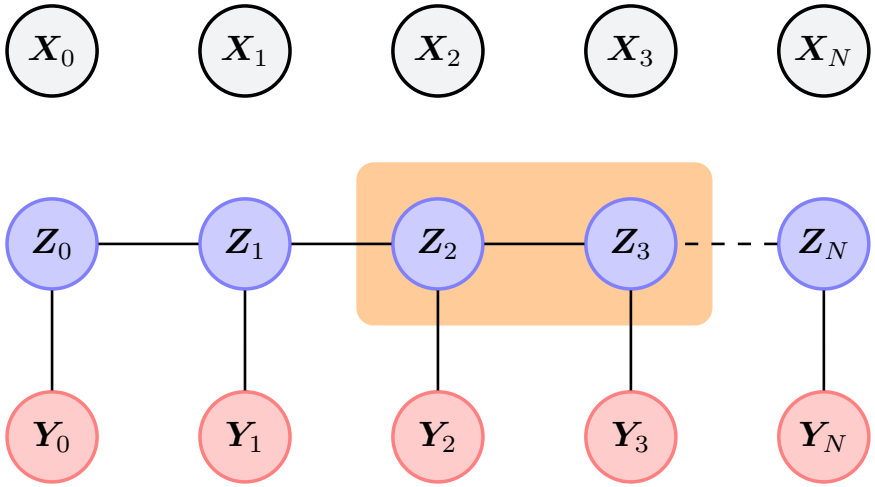
- Compute $\mathfrak{M}_2 : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ s.t.

$$\mathfrak{M}_2(\mathbf{x}_2, \mathbf{x}_3) = \begin{bmatrix} A_2(\mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_3) \end{bmatrix}$$

- Reference: $\eta_{\mathbf{X}_2} \eta_{\mathbf{X}_3}$
- Target: $\eta_{\mathbf{X}_2}\, \pi_{\mathbf{Y}_3|\mathbf{Z}_3}\, \pi_{\mathbf{Z}_3|\mathbf{Z}_2}(\cdot|\, B_1\,(\cdot)\,)$
- Uses only one component of $\mathfrak{M}_1$

$$T_2(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ A_2(\mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_3) \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

A decomposition theorem for chains



Theorem.

1. $(B_k)_\sharp \, \eta \mathsf{X}_{k+1} = \pi_{\mathsf{Z}_{k+1} \,|\, \mathsf{Y}_{0:k+1}}$ (*filtering*)
2. $(\mathfrak{M}_k)_\sharp \, \eta \mathsf{X}_{k:k+1} \simeq \pi_{\mathsf{Z}_k, \mathsf{Z}_{k+1} \,|\, \mathsf{Y}_{0:k+1}}$ (*lag-1 smoothing*)
3. $(T_1 \circ \cdots \circ T_k)_\sharp \, \eta \mathsf{X}_{0:k+1} = \pi_{\mathsf{Z}_{0:k+1} | \, \mathsf{Y}_{0:k+1}}$ (*full Bayesian solution*)

A nested decomposable map

- $\mathfrak{T}_k = T_0 \circ T_1 \circ \cdots \circ T_k$ characterizes the joint dist $\pi_{\mathbf{Z}_{0:k+1}|\,\mathbf{Y}_{0:k+1}}$

$$\mathfrak{T}_k \quad (\mathbf{x}) = \underbrace{\begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_0} \circ \underbrace{\begin{bmatrix} \mathbf{x}_0 \\ A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_1} \circ$$

- Trivial to go from $\mathfrak{T}_k$ to $\mathfrak{T}_{k+1}$: just append a new map T_{k+1}
- No need to recompute $T_0, \ldots, T_k$ (nested transports)
- $\mathfrak{T}_k$ is dense and high-dimensional but **decomposable**

A nested decomposable map

- $\mathfrak{T}_k = T_0 \circ T_1 \circ \cdots \circ T_k$ characterizes the joint dist $\pi_{\mathbf{Z}_{0:k+1}|\,\mathbf{Y}_{0:k+1}}$

$$\mathfrak{T}_{k+1}(\mathbf{x}) = \underbrace{\begin{bmatrix} A_0(\mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_0} \circ \underbrace{\begin{bmatrix} \mathbf{x}_0 \\ A_1(\mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_1} \circ \underbrace{\begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ A_2(\mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_3) \\ \mathbf{x}_4 \\ \mathbf{x}_5 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_2} \circ \cdots$$

- Trivial to go from $\mathfrak{T}_k$ to $\mathfrak{T}_{k+1}$: just append a new map T_{k+1}
- No need to recompute $T_0, \ldots, T_k$ (nested transports)
- $\mathfrak{T}_k$ is dense and high-dimensional but **decomposable**

A single-pass algorithm on the model

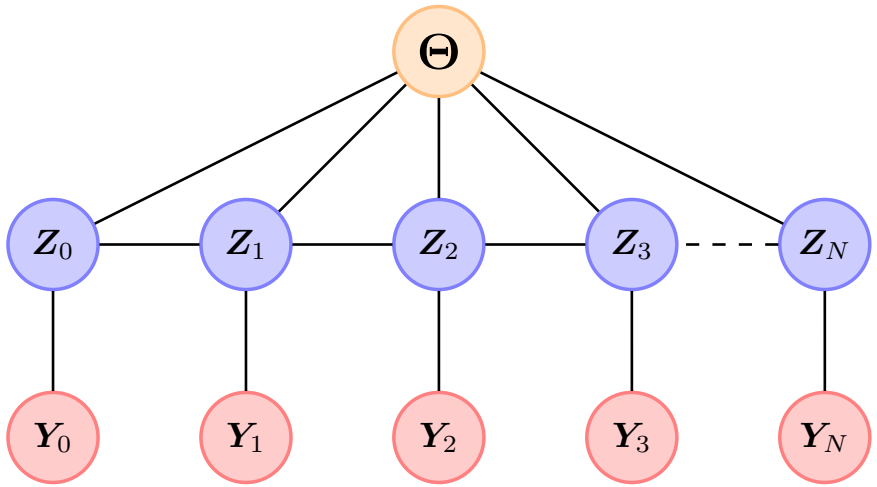
- **Meta-algorithm:**
  1. Compute the maps $\mathfrak{M}_0, \mathfrak{M}_1, \ldots$, each of dimension $2 \times \dim(\mathbf{Z}_0)$
  2. Embed each $\mathfrak{M}_j$ into an identity map to form T_j
  3. Evaluate $T_0 \circ \cdots \circ T_k$ for the full Bayesian solution
- **Remarks**:
  - A single pass on the state-space model
  - **Non-Gaussian** generalization of the Rauch-Tung-Striebel smoother
  - Bias is *only* due to the numerical approximation of each map $\mathfrak{M}_i$
  - Can either accept the bias or reduce it by:
    - Increasing the complexity of each map $\mathfrak{M}_i$, or
    - Computing **weights** given by the proposal density

$$(T_0 \circ T_1 \circ \cdots \circ T_k)_\sharp \, \eta_{\mathbf{X}_{0:k+1}}$$

    - The cost of evaluating smoothing weights grows linearly with time

Joint parameter/state estimation

- Generalize to sequential **joint parameter/state estimation**

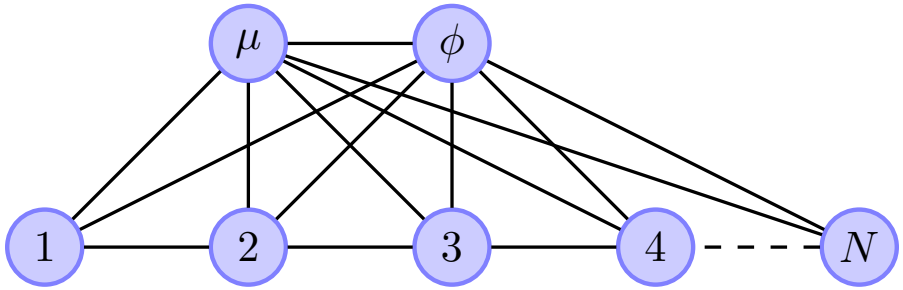


- $(T_0 \circ \cdots \circ T_k)_\sharp \, \eta_\Theta \, \eta_{\mathbf{X}_{0:k+1}} = \pi_{\Theta, \mathbf{Z}_{0:k+1} \mid \mathbf{Y}_{0:k+1}}$ *(full Bayesian solution)*
- Now $\dim(\mathfrak{M}_j) = 2 \times \dim(\mathbf{Z}_j) + \dim(\Theta)$
- **Remarks**:
  - No artificial dynamic for the static parameters
  - No a priori fixed-lag smoothing approximation

Example: stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = \mathbf{Id}$$

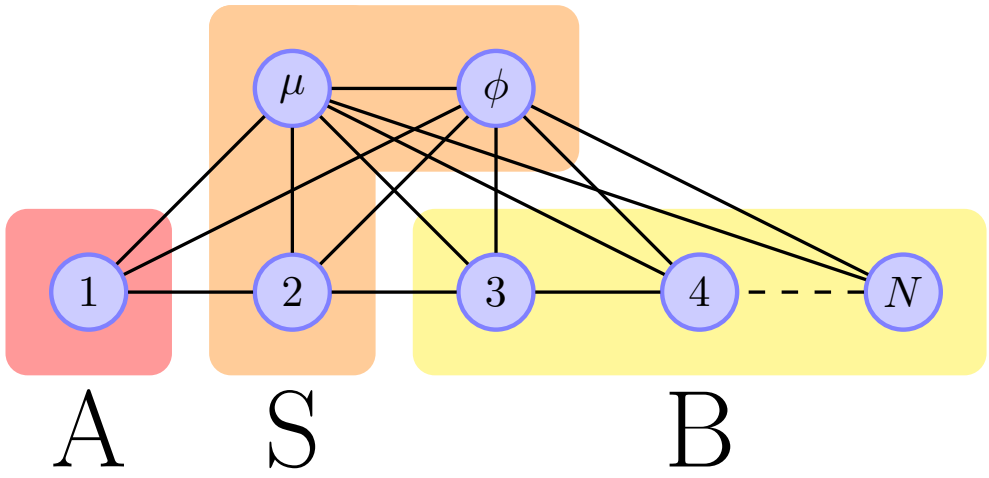


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Start with the identity map

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = \mathbf{Id}$$

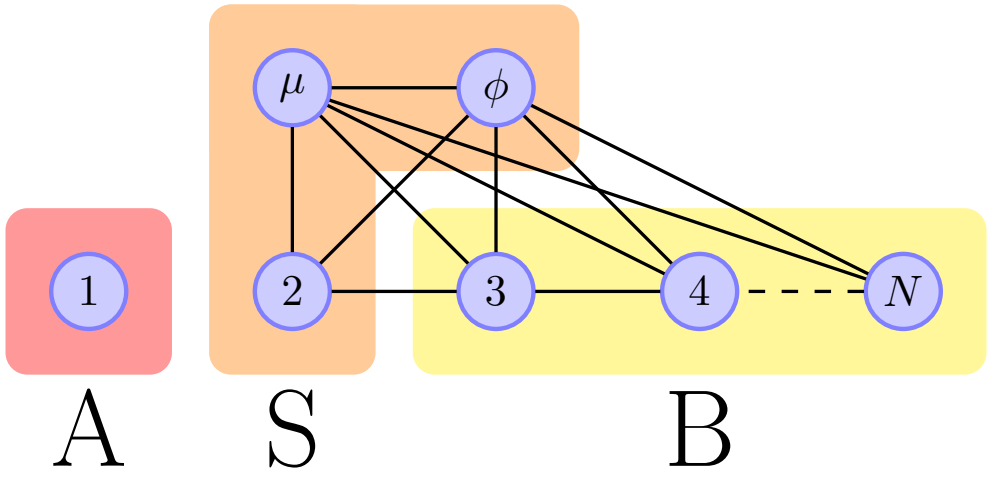


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Find a good first decomposition of $\mathcal{G}$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0$$

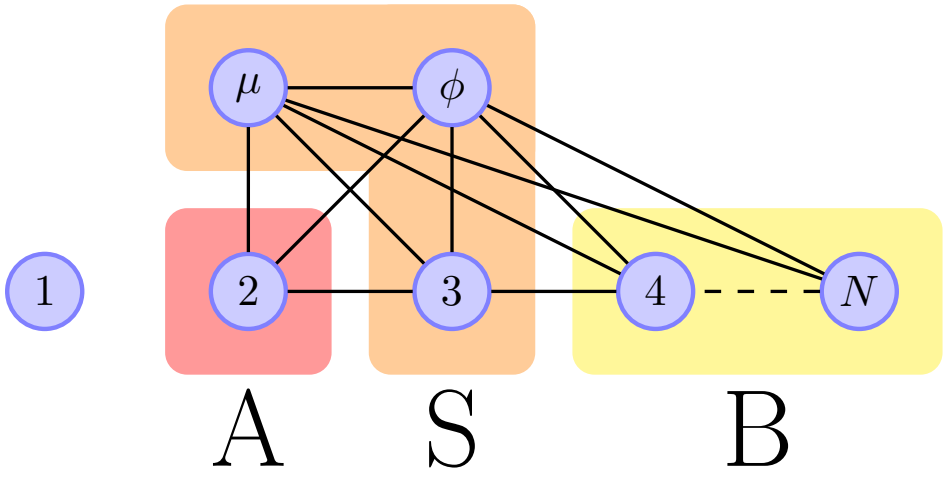


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Compute an (essentially) 4-D T_0 and pull back π
- Underlying approximation of $\mu, \phi, \mathbf{Z}_1|\mathbf{Y}_1$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0$$

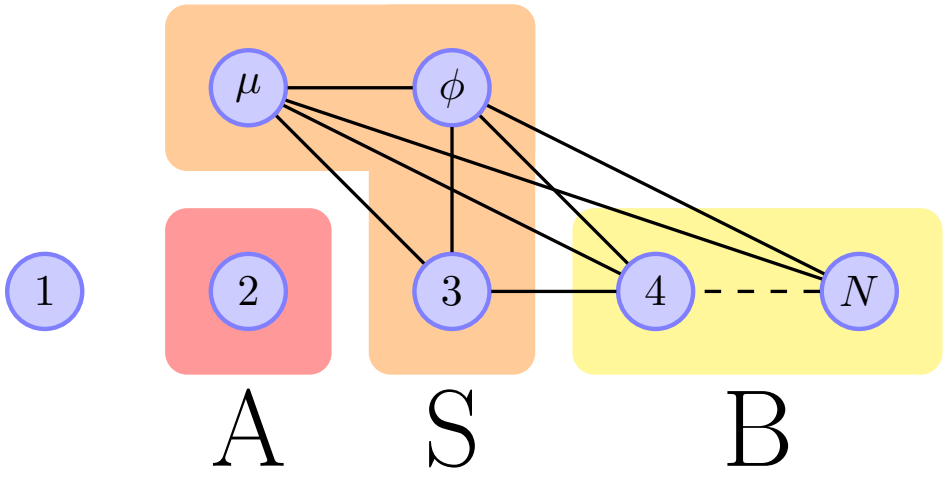


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Find a new decomposition
- Underlying approximation of $\mu, \phi, \mathbf{Z}_1|\mathbf{Y}_1$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1$$

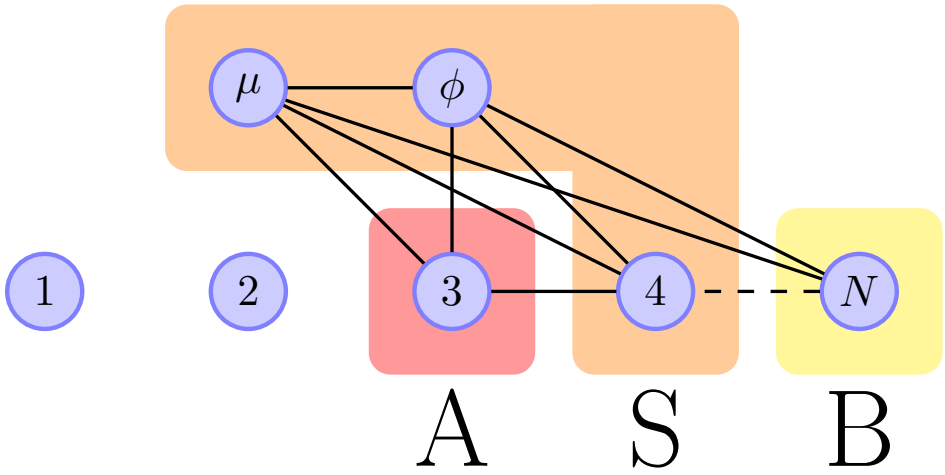


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Compute an (essentially) 4-D T_1 and pull back π
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:2} | \mathbf{Y}_{1:2}$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1$$

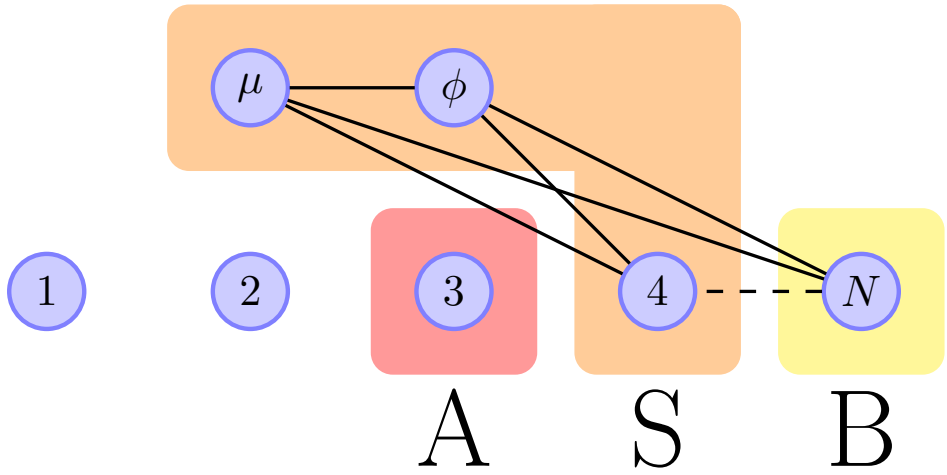


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Continue the recursion until no edges are left...
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:2} | \mathbf{Y}_{1:2}$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1 \circ T_2$$

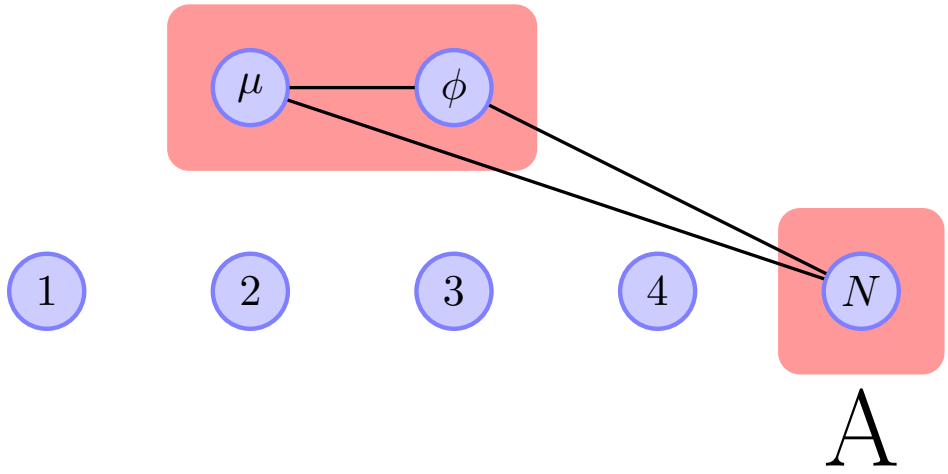


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Continue the recursion until no edges are left...
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:3}|\mathbf{Y}_{1:3}$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1 \circ T_2 \circ \cdots \circ T_{N-3}$$

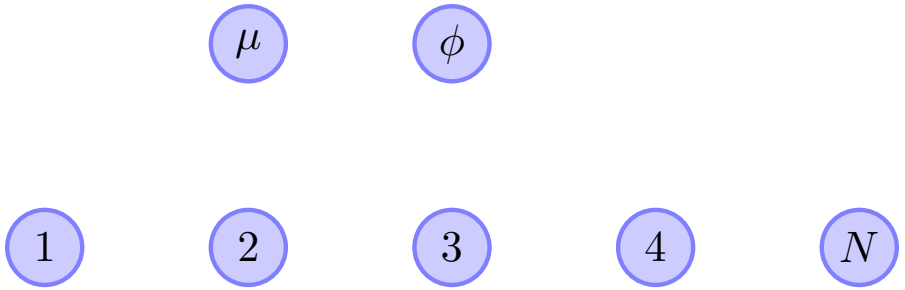


- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Continue the recursion until no edges are left...
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:N-1} | \mathbf{Y}_{1:N-1}$

Stochastic volatility model

- Build the decomposition recursively

$$\mathfrak{T} = T_0 \circ T_1 \circ T_2 \circ \cdots \circ T_{N-3} \circ T_{N-2}$$



- **Figure:** Markov structure for the pullback of π through $\mathfrak{T}$
- Each map T_k is essentially 4-D regardless of N
- Underlying approximation of $\mu, \phi, \mathbf{Z}_{1:N}|\mathbf{Y}_{1:N}$

Another decomposable map

$$\mathfrak{T}_{k+1}(\mathbf{x}) = \underbrace{\begin{bmatrix} P_0(x_\theta) \\ A_0(\mathbf{x}_\theta, \mathbf{x}_0, \mathbf{x}_1) \\ B_0(\mathbf{x}_\theta, \mathbf{x}_1) \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_0} \circ \underbrace{\begin{bmatrix} P_1(x_\theta) \\ \mathbf{x}_0 \\ A_1(\mathbf{x}_\theta, \mathbf{x}_1, \mathbf{x}_2) \\ B_1(\mathbf{x}_\theta, \mathbf{x}_2) \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_1} \circ \underbrace{\begin{bmatrix} P_2(x_\theta) \\ \mathbf{x}_0 \\ \mathbf{x}_1 \\ A_2(\mathbf{x}_\theta, \mathbf{x}_2, \mathbf{x}_3) \\ B_2(\mathbf{x}_\theta, \mathbf{x}_3) \\ \mathbf{x}_4 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{T_2} \circ \cdots$$

- $(P_0 \circ \cdots \circ P_k)_\sharp \, \eta_\Theta = \pi_{\Theta \,|\, \mathbf{Y}_{0:k+1}}$ *(parameter inference)*
- If $\mathfrak{P}_k = P_0 \circ \cdots \circ P_k$, then $\mathfrak{P}_k$ can be computed recursively as

$$\mathfrak{P}_k = \mathfrak{P}_{k-1} \circ P_k$$

$\implies$ cost of evaluating $\mathfrak{P}_k$ does not grow with k

Example: stochastic volatility model

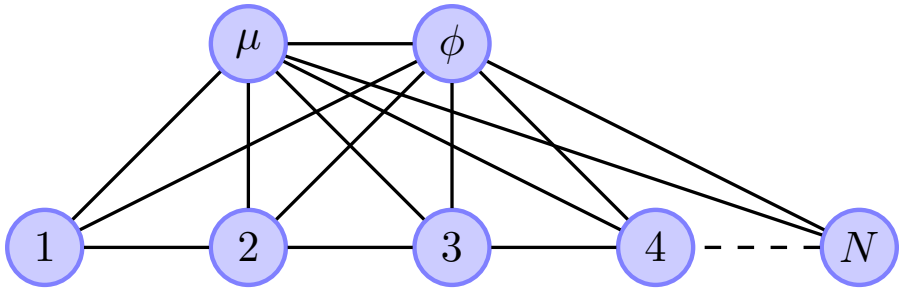
- **Stochastic volatility model:** Latent log-volatilities take the form of an AR(1) process for $t = 1, \ldots, N$:

$$Z_{t+1} = \mu + \phi\,(Z_t - \mu) + \eta_t, \quad \eta_t \sim \mathcal{N}(0,1), \quad Z_1 \sim \mathcal{N}(0, 1/1-\phi^2)$$

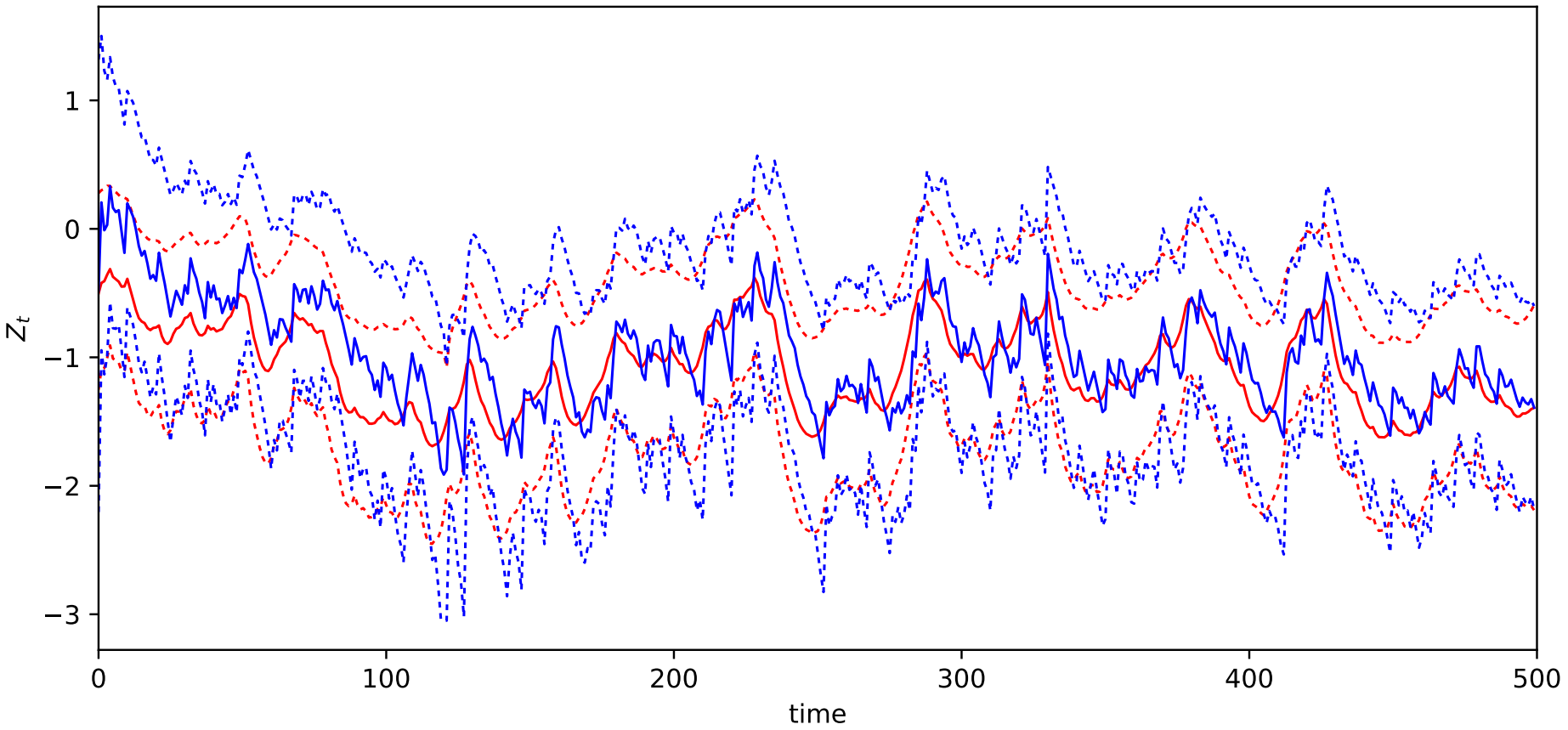
  - Observe the mean return for holding an asset at time t

$$Y_t = \varepsilon_t \, \exp(\, 0.5\, Z_t\,), \quad \varepsilon_t \sim \mathcal{N}(0,1), \quad t = 1, \ldots, N$$

  - Markov structure for $\pi \sim \mu, \phi, \mathbf{Z}_{1:N} | \mathbf{Y}_{1:N}$ is given by:

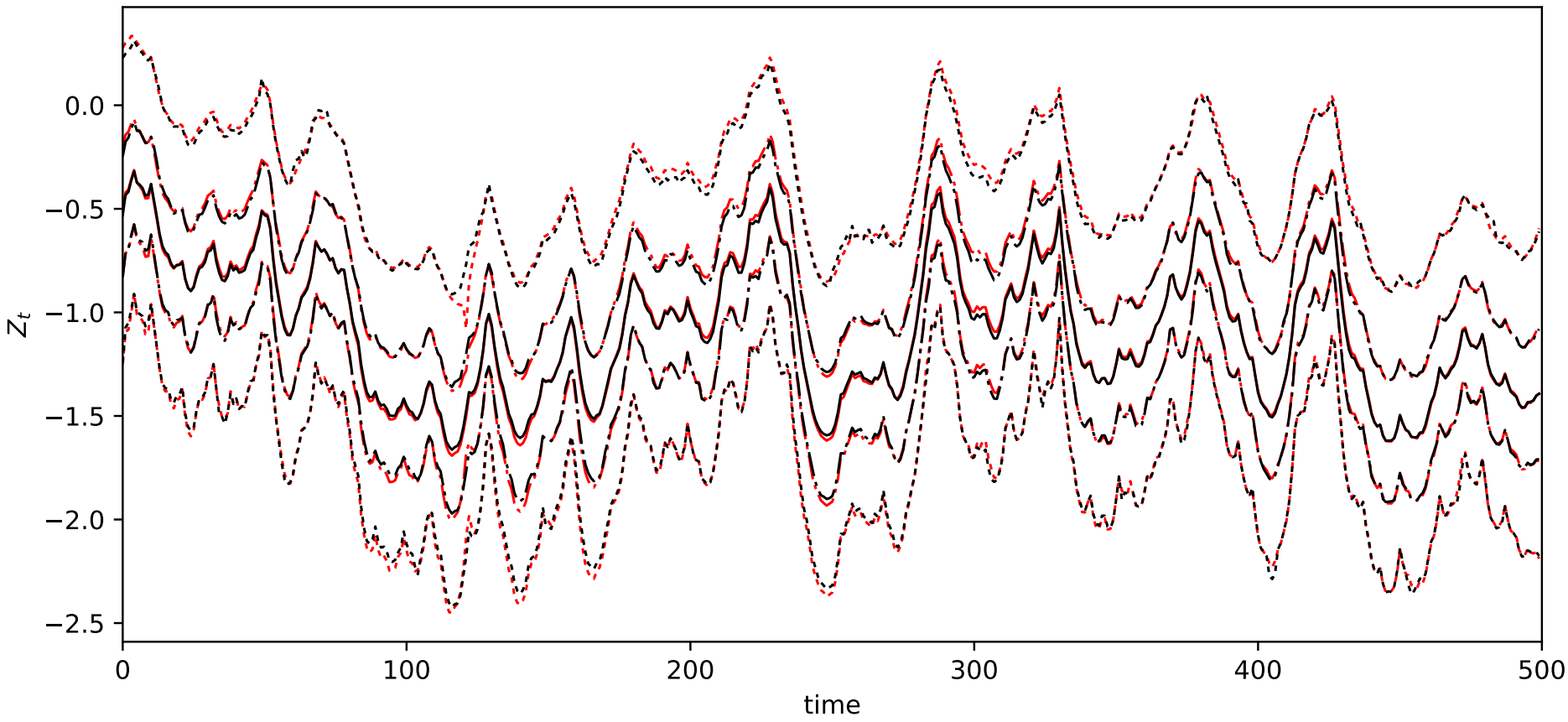


Stochastic volatility example



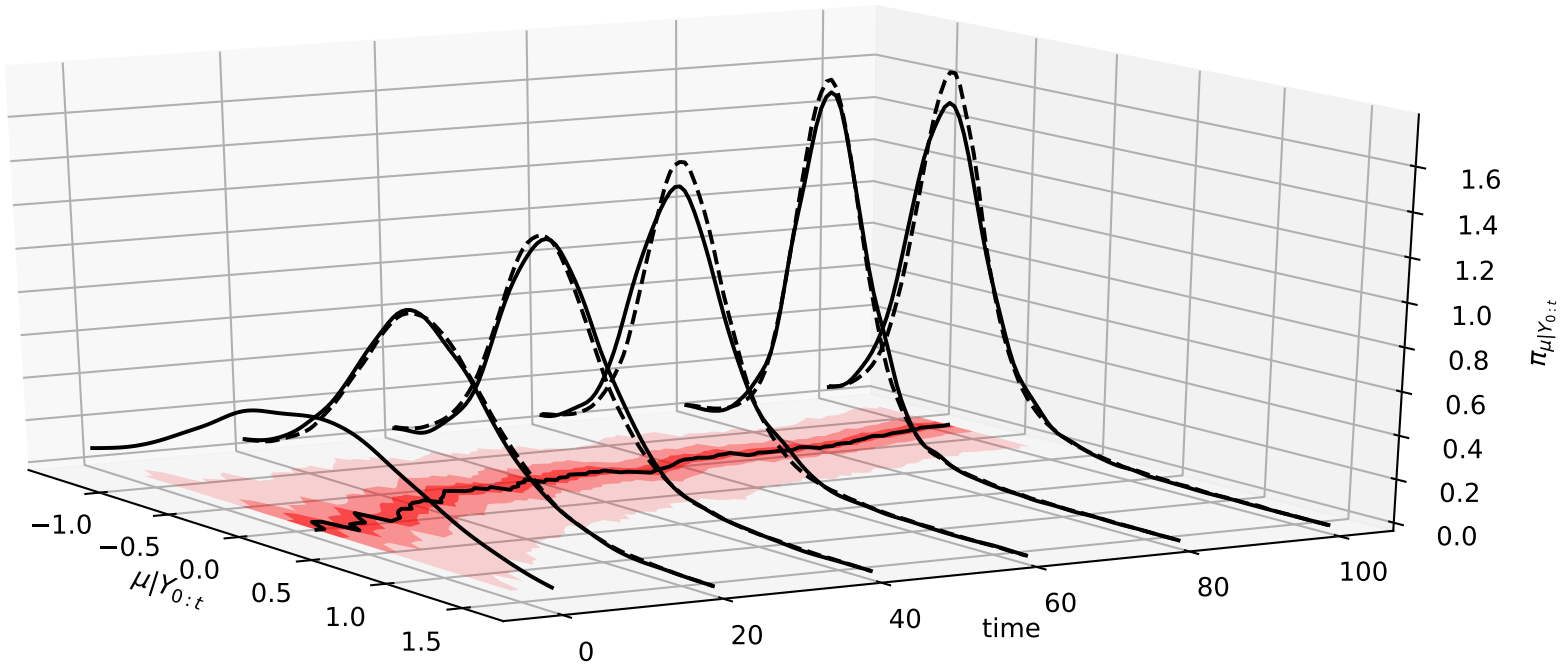
- Filtering (blue) versus smoothing (red) marginals

Stochastic volatility example



- Quantiles of smoothing marginals (red) compared to MCMC (black)

Stochastic volatility example



- **Sequential parameter estimation:** marginals of hyperparameter μ, conditioning on successively more observations $\mathbf{y}_{0:k}$

Stochastic volatility example

- Variance diagnostic $\mathbb{V}\text{ar}_\eta[\log(\eta/T_\sharp^{-1}\bar{\pi})]$ values, for a 502-dimensional target π :
  - Laplace map = 5.65; degree–1 maps = 1.49; *degree–3 maps = 0.731*
- **Important open question:** how does error in the approximation of the parameter posterior evolve over time?

Part 2: large-scale approximate filtering

- Consider the filtering of state-space models with:
  1. High-dimensional states
  2. Challenging nonlinear dynamics (e.g., chaotic systems)
  3. Intractable transition kernels: can only obtain *forecast* samples, i.e., draws from $\pi_{\mathbf{Z}_{k+1} \,|\, \mathbf{y}_{0:k}}$
  4. Limited model evaluations, e.g., small ensemble sizes
  5. Sparse and local observations in space/time
- These constraints reflect typical challenges faced in numerical weather prediction, geophysical data assimilation
- State-of-the-art results (in terms of tracking error) are *currently* obtained with localized versions of the **ensemble Kalman filter**

Algorithms for filtering

Ensemble Kalman filter (EnKF):

- More successful than particle filters in large-scale data assimilation applications
- Idealized scheme: transform the *forecast ensemble* (samples from $\pi_{\mathbf{Z}_{k+1} \,|\, \mathbf{y}_{0:k}}$) to the *analysis ensemble* (samples from $\pi_{\mathbf{Z}_{k+1} \,|\, \mathbf{y}_{0:k+1}}$) using a linear map
- Construction of a linear map proceeds from Gaussian approximation and regularizing assumptions (localization/tapering, inflation)
- *Move* samples; no weights or resampling!

Limitations of the EnKF:

- **Biased;** does not converge to the true Bayesian solution
  - Yet perhaps we should trade some variance for bias!
- No guarantees on UQ [Law & Stuart 2012]

Algorithms for filtering

Limitations of the EnKF:

- **Biased;** does not converge to the true Bayesian solution
  - Yet perhaps we should trade some variance for bias!
- No guarantees on UQ [Law & Stuart 2012]

Some questions:

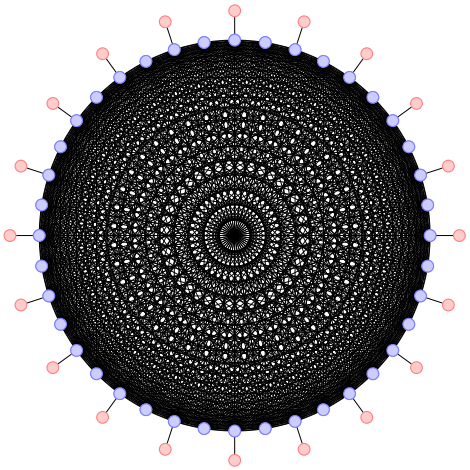
- For a given ensemble size N, are we doing the best we can?
- EnKF is not guaranteed to perform better as N increases, and in some situations performs worse! Can this be mitigated?
- Can we get closer to the Bayesian solution, while preserving robustness of EnKF approaches?

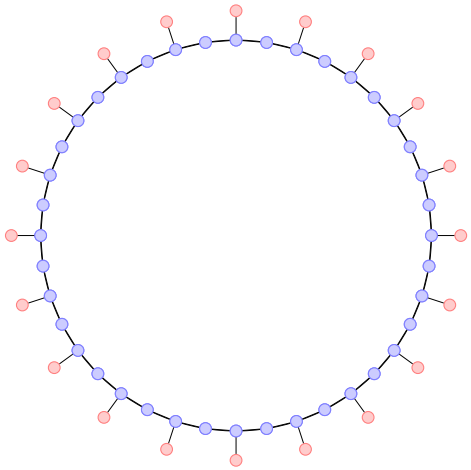
Nonlinear filters induced by local couplings

Main idea

Transform the forecast ensemble into approximate samples from the filtering distribution via local, low-dimensional, nonlinear couplings

- Effectively approximate the **projection** of the filtering distribution onto a manifold of sparse non-Gaussian Markov random fields
- Introduce local nonlinear features in a **stable** fashion
  - Recover the EnKF as transformations become linear and other regularizing assumptions relaxed





Ingredient #1: sparse triangular maps

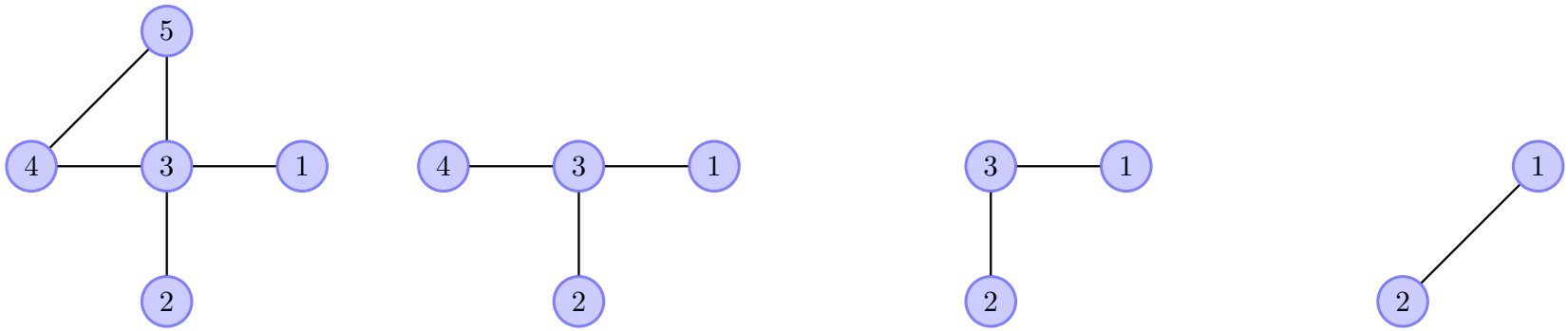
- Given a reference η and a target π, focus on the sparsity of the *inverse* Knothe-Rosenblatt (KR) rearrangement, i.e., $S_\sharp \pi = \eta$

$$S(\mathbf{x}) = \begin{bmatrix} S^1(x_1) \\ S^2(x_1, x_2) \\ S^3(x_1, x_2, x_3) \\ \vdots \\ S^n(x_1, x_2, \ldots, x_n) \end{bmatrix} \Longrightarrow \begin{bmatrix} S^1(x_1) \\ S^2(x_1, x_2) \\ S^3(\quad x_2, x_3) \\ \vdots \\ S^n(\qquad\qquad x_{n-1}, x_n) \end{bmatrix}$$

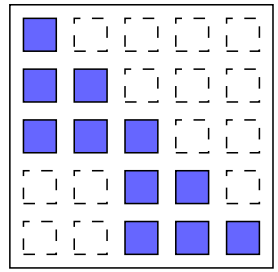
- **Theorem:** [Spantini et al. (2017)] The KR rearrangement (a **nonlinear** function) inherits the same sparsity pattern as the Cholesky factor of the incidence matrix (properly scaled) of a graphical model for π, provided that

$$\eta(\mathbf{x}) = \prod_i \eta(x_i)\,.$$

How to compute the sparsity pattern



- **Compute marginal graphs:** $\mathcal{G}^{i-1}$ is obtained from $\mathcal{G}^i$ by removing node i and by turning its neighborhood into a clique (like *variable elimination*)
- **Sparsity of inverse transport:** the i-th component of S can depend, at most, on the variables in a neighborhood of node i in $\mathcal{G}^i$
- Sparsity depends on the ordering of the variables (same heuristics as *sparse Cholesky*)



$$\mathbf{P}_{kj} = \partial_{x_j} S^k$$

Ingredient #2: conditional simulation from triangular maps

- Begin with a map $S_\sharp \pi = \eta$, with $\mathbf{Z}_{1:n} \sim \pi$ and $\mathbf{X}_{1:n} \sim \eta$
- Let $\eta = \prod_{i=1}^{n} \eta_i$
- Consider the triangular map $S_{\boldsymbol{\xi}_{1:k}}$ given by

$$\mathbf{z}_{k+1}, \ldots, \mathbf{z}_n \mapsto \begin{bmatrix} S^{k+1}(\boldsymbol{\xi}_{1:k}, \mathbf{z}_{k+1}) \\ \vdots \\ S^{n}(\boldsymbol{\xi}_{1:k}, \mathbf{z}_{k+1}, \ldots, \mathbf{z}_n) \end{bmatrix}$$

- $S_{\boldsymbol{\xi}_{1:k}}$ **pushes forward** the conditional $\pi_{\mathbf{Z}_{k+1:n}|\mathbf{Z}_{1:k}=\boldsymbol{\xi}_{1:k}}$ to the reference marginal $\eta_{k+1:n}$
- **Sample** from $\pi_{\mathbf{Z}_{k+1:n}|\mathbf{Z}_{1:k}=\boldsymbol{\xi}_{1:k}}$ by drawing $\mathbf{x}_{k+1:n}$ from $\eta_{k+1:n}$ and inverting $S_{\boldsymbol{\xi}_{1:k}}$
  - Solve the triangular system $S_{\boldsymbol{\xi}_{1:k}}(\mathbf{z}_{k+1:n}) = \mathbf{x}_{k+1:n}$

Ingredient #3: constructing transport maps

Two types of calculations:

1. Maps from unnormalized densities (discussed earlier)

$$\min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(T_\sharp \eta \,||\, \pi) = \min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(\eta \,||\, T_\sharp^{-1} \pi)$$

   - Use evaluations of π (and its gradients) directly

Ingredient #3: constructing transport maps

Two types of calculations:

1. Maps from unnormalized densities (discussed earlier)

$$\min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(T_\sharp \eta \,||\, \pi) = \min_{T \in \mathcal{T}_\triangle} \mathcal{D}_{KL}(\eta \,||\, T_\sharp^{-1} \pi)$$

   - Use evaluations of π (and its gradients) directly

2. Maps from samples:

$$\min_{S \in \mathcal{S}_\triangle} \mathcal{D}_{KL}(\pi \,||\, S_\sharp^{-1} \eta)$$

   - $\mathcal{S}_\triangle$ is a set of **(sparse)** monotone triangular maps
   - For Gaussian η, this problem is **convex** and **separable**
   - Specifically, given samples $\{\mathbf{z}^i\}_{i=1}^M$, find component S^k via:

$$\begin{aligned} \min_{S^k} \quad & \frac{1}{M} \sum_{i=1}^{M} \left[\frac{1}{2} (S^k(\mathbf{z}^i))^2 - \log \partial_k S^k(\mathbf{z}^i) \right] \\ \text{s.t.} \quad & \partial_k S^k > 0 \ \text{ on } \mathbb{R}^k \end{aligned}$$

Back to the filtering algorithm

Main idea

1. Assimilation: transform the forecast ensemble into approximate samples from the filtering distribution via local, low-dimensional, and nonlinear couplings
2. Evolution: apply the dynamics to obtain the next forecast ensemble

Key steps of the assimilation algorithm:

1 Approximate the *forecast distribution* on a manifold of sparse non-Gaussian Markov random fields

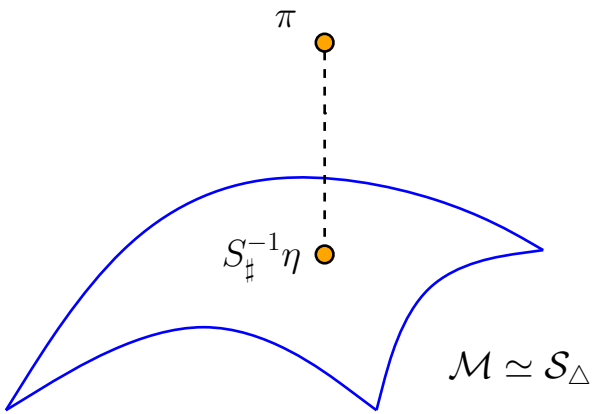
2a Local *assimilation* of the observations

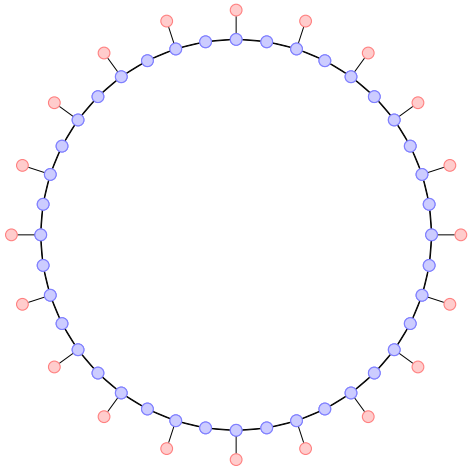
2b *Propagation* of information across the state

Abstraction of the assimilation problem:

- We have *samples* from the prior & can evaluate the likelihood

Projection onto a manifold of sparse MRFs



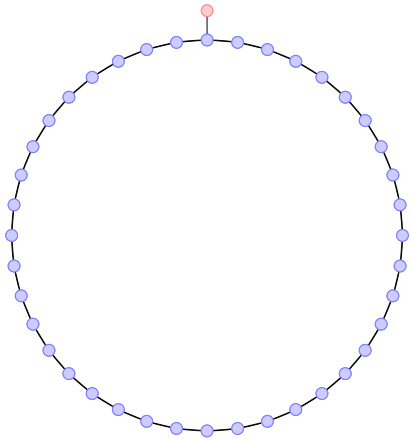


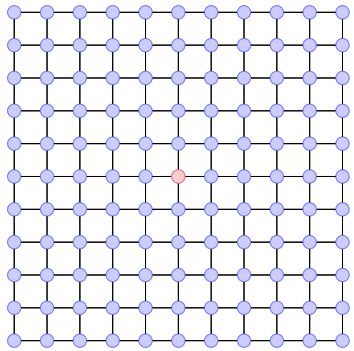
- Choose the approximation space $\mathcal{S}_\triangle$ (finite space of **sparse** lower triangular maps) to enforce a desired **Markov structure**
- Given samples $(\mathbf{z}_i)_{i=1}^M$ from the forecast distribution, we approximate $\pi_{\mathbf{Z}_{k+1}\,|\,\mathbf{y}_{0:k}}$ on $\mathcal{M}$
- **Approach:** learn an inverse map $S : \mathbb{R}^n \to \mathbb{R}^n$ *from samples*

$$\min_{S \in \mathcal{S}_\triangle} \mathcal{D}_{KL}(\pi \,||\, S_\sharp^{-1}\, \eta \,)$$

- Compute each component $S^k : \Omega \to \mathbb{R}$ via **convex** optimization
  - Choose any parameterization of S^k that departs (if desired) from linearity by adding local **nonlinear** terms (e.g., polynomials, RBFs)

Assimilation and propagation



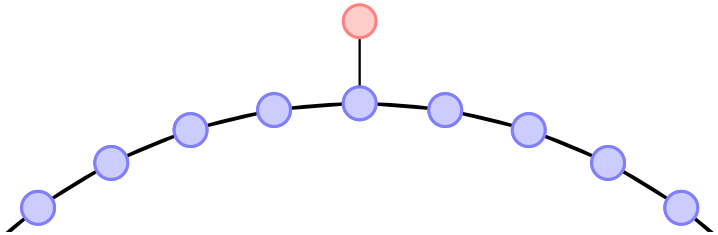


- For simplicity, consider assimilating one observation at a time ...
- Notice that $\pi(\mathbf{z}|y) = \pi(\mathbf{z}_1|y)\pi(\mathbf{z}_{\sim 1}|\mathbf{z}_1)$
- **Local assimilation:** simulate from $\pi(\mathbf{z}_1|y)$
  - First map component S^1 *pushes forward* the prior $\pi_{\mathbf{z}_1}$ to η_1; yields an approximation $(S^1)^{-1}_\sharp \eta_1$ of the prior
  - Seek a direct map T^1 with target density

$$\pi(\mathbf{z}_1|y) \propto \pi(y|\mathbf{z}_1)\, \eta_1\big(S^1(\mathbf{z}_1)\big)\partial_{\mathbf{z}_1} S^1(\mathbf{z}_1)$$

  - Then $T^1 \circ S^1$ transforms forecast samples of $\mathbf{z}_1$ to analysis/posterior samples of $\mathbf{z}_1$

Assimilation and propagation

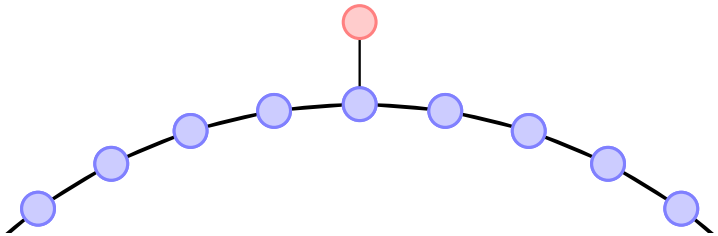


- **Propagation:** Sample from the conditional $\pi(\mathbf{z}_{\sim 1}|\mathbf{z}_1)$ given samples from the marginal $\pi(\mathbf{z}_1|y)$
- Given the inverse map S, use the conditional simulation strategy described earlier:
  - The map $S_{\boldsymbol{\xi}} : \mathbb{R}^{n-1} \to \mathbb{R}^{n-1}$,

$$\mathbf{z}_2, \ldots, \mathbf{z}_n \mapsto \begin{bmatrix} S^2(\boldsymbol{\xi}, \mathbf{z}_2) \\ \vdots \\ S^n(\boldsymbol{\xi}, \mathbf{z}_2, \ldots, \mathbf{z}_n) \end{bmatrix},$$

    pushes forward $\pi_{\mathbf{Z}_{2:n}|\mathbf{z}_1=\boldsymbol{\xi}}$ to $\eta_{2:n}$
  - Sparse Markov structure yields further simplifications in S

Assimilation and propagation



Local assimilation + propagation:

- Then the **combined** map:

$$\mathcal{T}(\mathbf{z}) = \begin{bmatrix} T^1(\mathbf{z}_1) \\ S^{-1}_{T^1(\mathbf{z}_1)}(\mathbf{z}_2, \ldots, \mathbf{z}_n) \end{bmatrix} \circ S(\mathbf{z})$$

  transforms the forecast ensemble to the analysis ensemble!

- Can *iterate* this construction to assimilate each additional observation, or generalize to multiple/batch observations

Lorenz 96 system (40-dimensional state)

- A "hard" test-case configuration [Bengtsson 2003, Lei & Bickel 2010]

$$\begin{aligned} \frac{\mathrm{d}\mathbf{Z}_j}{\mathrm{d}t} &= (\mathbf{Z}_{j+1} - \mathbf{Z}_{j-2})\mathbf{Z}_{j-1} - \mathbf{Z}_j + F, \qquad j = 1, \ldots, 40 \\ \mathbf{Y}_j &= \mathbf{Z}_j + \varepsilon_j, \qquad j = 1, 3, 5 \ldots, 39 \end{aligned}$$

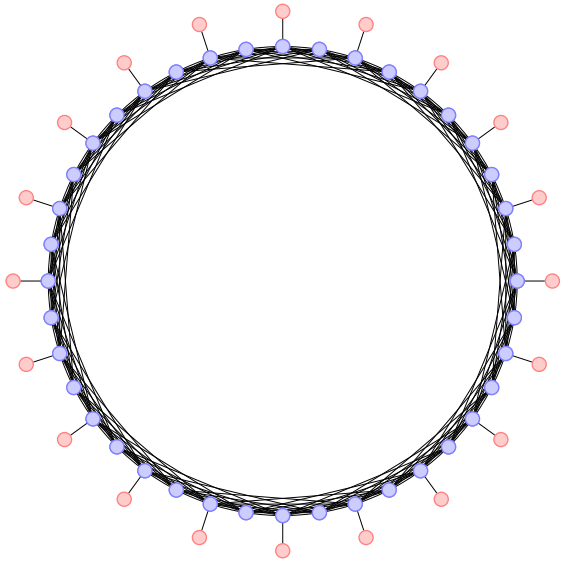
  - $F = 8$ (chaotic regime) and $\varepsilon_j \sim \mathcal{N}(0, 0.5)$
  - Time between observations: $\Delta_{\text{obs}} = 0.4$ (**large**)
  - Results averaged over 2000 assimilation cycles

	#particles: 400	
	EnKF*	**LocNLF**
med RMSE	0.88	**0.64**
avg RMSE	0.97	**0.74**
var RMSE	0.12	**0.06**

#particles: 200	
LocLF	**LocNLF**
0.91	**0.66**
1.02	**0.79**
0.1	**0.09**

- The nonlinear filter is $\approx 25\%$ more accurate in RMSE than EnKF

Lorenz 96: details on the filtering approximation

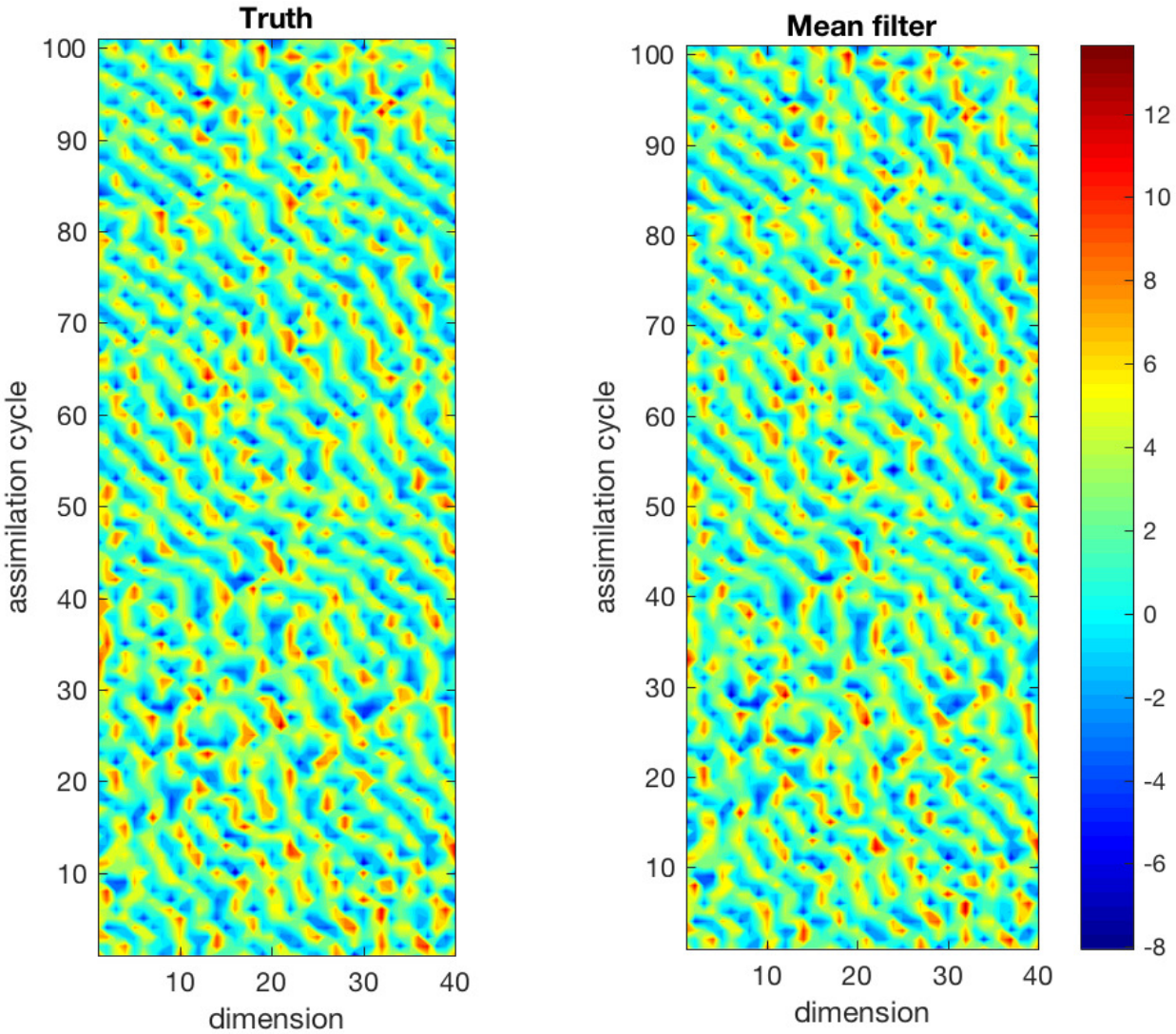


- Observations were assimilated one at a time
- **Approximate** Markov structure: 5-way interactions
- Each conditional $\pi(x_k|x_{j_1},\ldots,x_{j_p})$ was learned via a **separable** map

$$S^k(x_{j_1},\ldots,x_{j_p},\,x_k) = \psi(x_{j_1}) + \ldots + \psi(x_{j_p}) + \psi(x_k),$$

  where $\psi(x) = a_0 + a_1 \cdot x + \sum_{i>1} a_i \exp(-(x-c_i)^2/\sigma)$.
- Much **more general** parameterizations are of course possible!

Lorenz 96: tracking performance of the filter



- Introducing simple **localized nonlinearities** can make a difference!

Discussion

Nonlinear filters induced by local couplings:

- Construct *nonlinear transformations* to simulate conditioning on data
- In philosophy, like the EnKF: no weights or importance sampling, move the ensemble members, accept some bias
- Nonlinear features in the map capture *non-Gaussianity* of the prior/posterior distributions
- Choice of map, and of **sparse Markov structure** (conditional independence), provide necessary **regularization** to the problem
- Due to sparsity, operations can be made **local**: essential to successful filtering in high dimensions with small ensembles!

Conclusions

- Bayesian inference through the construction of deterministic couplings
- Computation of transport maps in high dimensions, leveraging the **Markov structure** of the posterior:
  1. Decomposability of direct transports
     - Variational approaches for *Bayesian filtering, smoothing, and sequential parameter inference*
  2. Sparsity of triangular transports
     - *A class of nonlinear filters induced by local couplings*
- **Ongoing work:**
  - Error analysis of approximate filtering schemes
  - *Structure learning* for continuous non-Gaussian Markov random fields [Morrison, Baptista, & M NIPS 2017]
  - Mapping sparse quadrature or QMC schemes
  - Nonparametric transports and *gradient flows*

References

- A. Spantini, D. Bigoni, Y. Marzouk. "Inference via low-dimensional couplings." arXiv:1703.06131
- R. Morrison, R. Baptista, Y. Marzouk. "Beyond normality: learning sparse probabilistic graphical models in the non-Gaussian setting." NIPS 2017.
- Y. Marzouk, T. Moselhy, M. Parno, A. Spantini, "An introduction to sampling via measure transport." *Handbook of Uncertainty Quantification*, R. Ghanem, D. Higdon, H. Owhadi, eds. Springer (2016). arXiv:1602.05023.
  (broad introduction to transport maps)

- M. Parno, T. Moselhy, Y. Marzouk, "A multiscale strategy for Bayesian inference using transport maps." *SIAM JUQ*, 4: 1160–1190 (2016).
- M. Parno, Y. Marzouk, "Transport map accelerated Markov chain Monte Carlo." arXiv:1412.5492.
- T. Moselhy, Y. Marzouk, "Bayesian inference with optimal maps." *J. Comp. Phys.*, 231: 7815–7850 (2012).
- **Python code at** `http://transportmaps.mit.edu`