Data Assimilation:

A message passing perspective

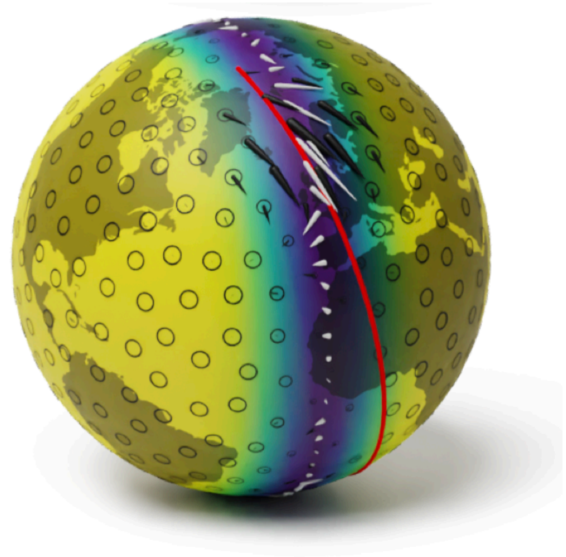
So Takao

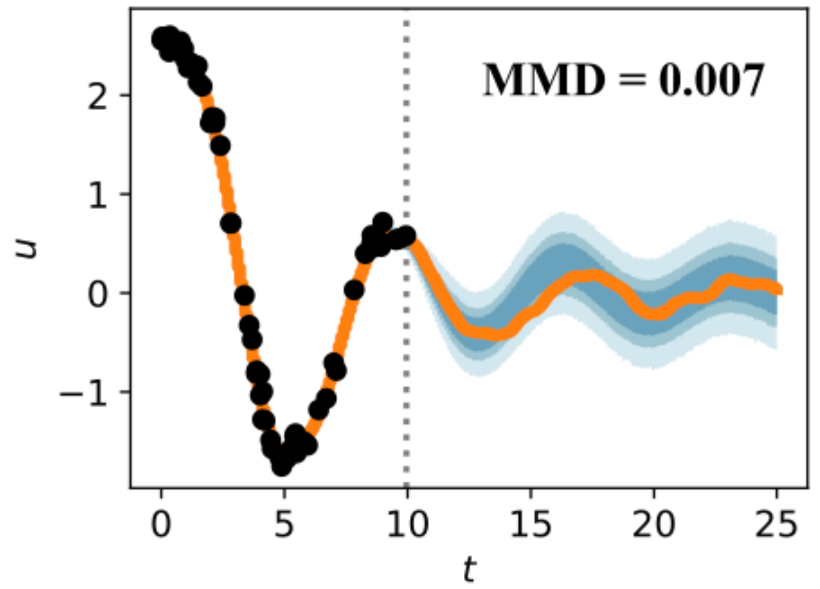
About Me

- → Imperial College London (PhD with Prof. Darryl Holm. 2016-2020)

  → University College London (Postdoc with Prof. Marc Deisenroth. 2021-2023)

  → Caltech (Postdoc with Prof. Andrew Stuart. 2024-)





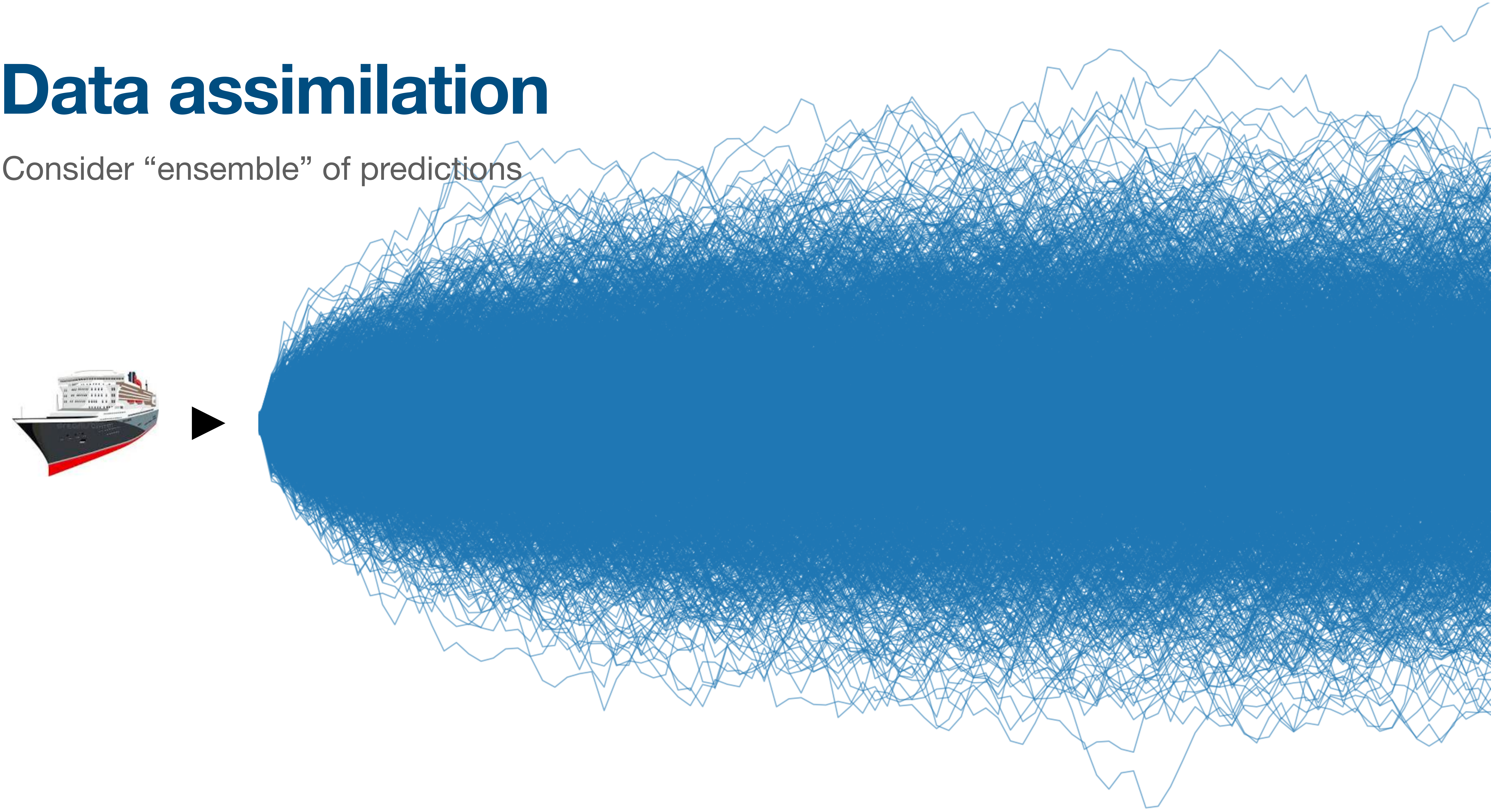
- Current research focus is in:
  - ML for data assimilation and uncertainty quantification
  - Probabilistic models with geometric / topological inductive biases
- Outside of research, I am:
  - A serious jazz nerd / jazz saxophonist
  - Amateur climber
  - Lived in Japan, Singapore, London, Los Angeles

Data assimilation

Suppose we're given a model for e.g. vehicle tracking

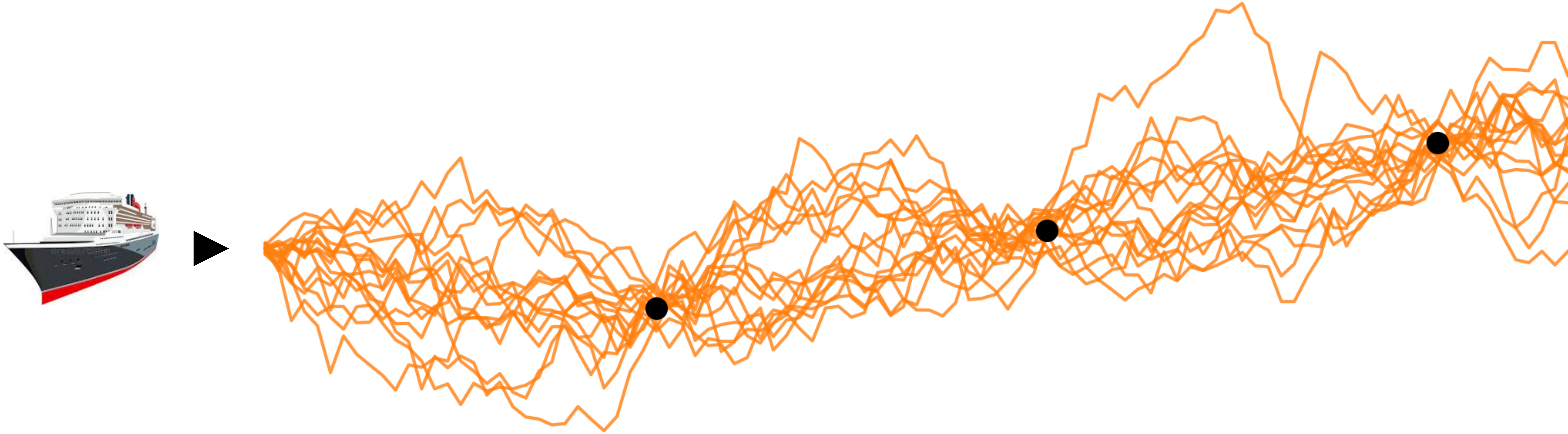
Data assimilation

Consider “ensemble” of predictions



Data assimilation

We observe some data



Data assimilation

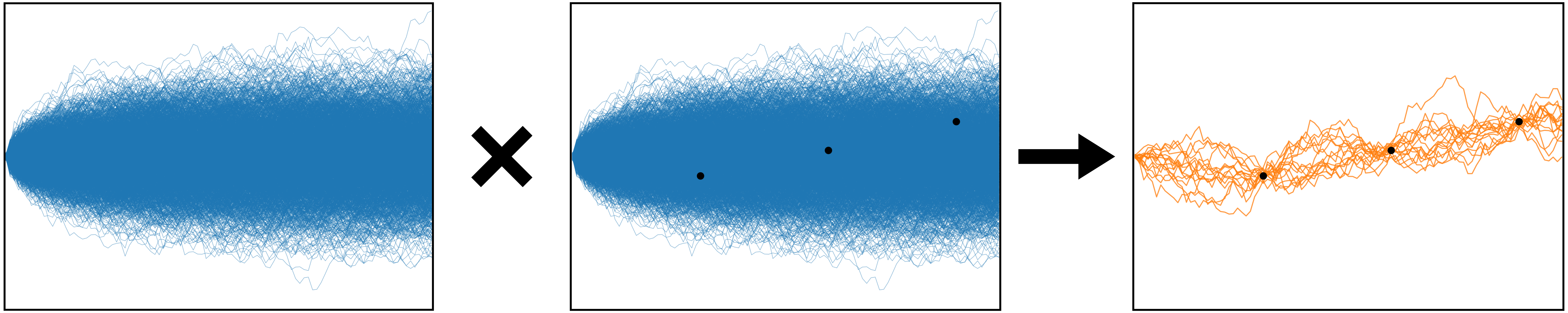
$p(z)$ - Prior belief of state

$p(y\,|\,z)$ - Observation likelihood

$$p(z\,|\,y) = \frac{p(y\,|\,z)p(z)}{\int p(y\,|\,z)p(z)\mathrm{d}z}$$

… Intractable in most situations

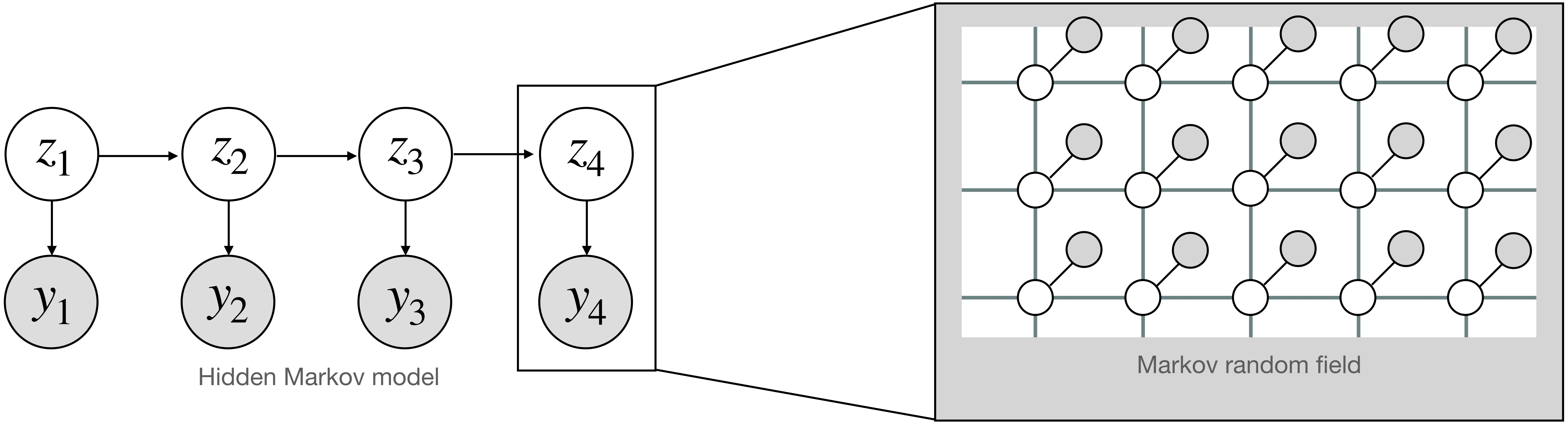
- No closed-form expression generally
- Curse of dimensionality



Prior $p(z)$ Likelihood $p(y\,|\,z)$ Posterior $p(z\,|\,y)$

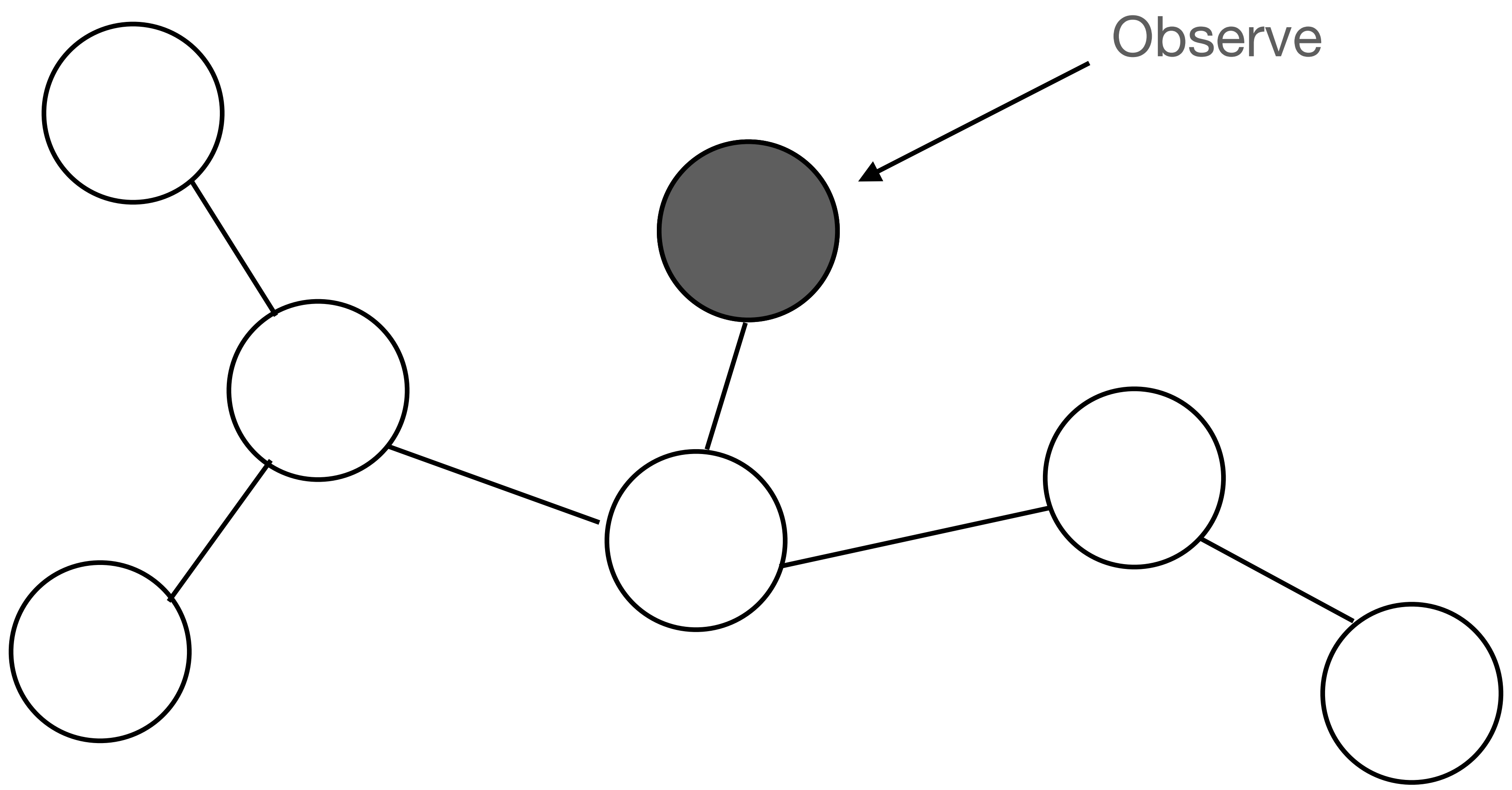
Message passing

Exploiting the “sparsity” of the underlying problem structure



Message passing

Exploiting the sparsity of the underlying graph



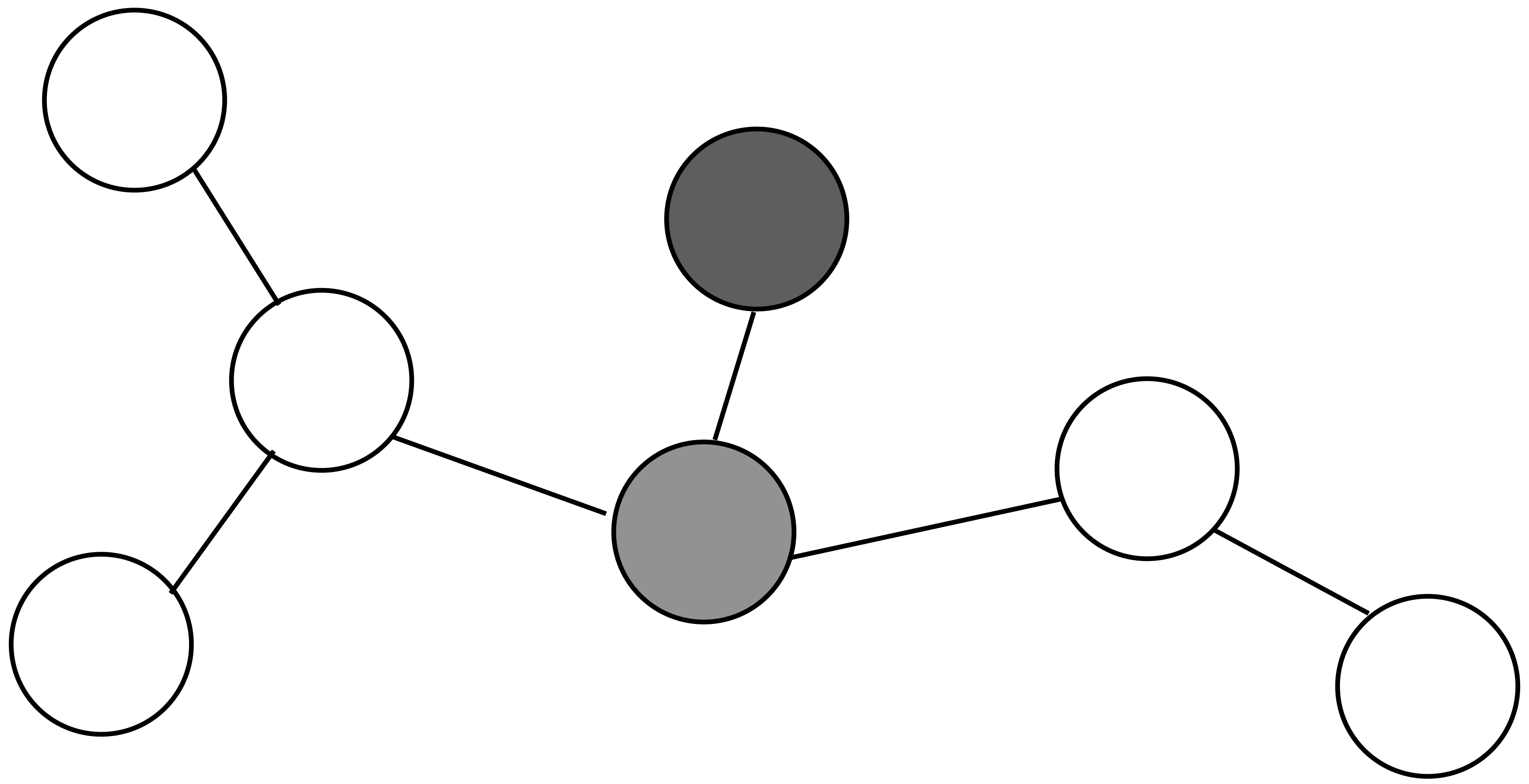
Belief Propagation (Pearl '82):

Algorithm to efficiently compute marginals $p(z_i \,|\, \boldsymbol{y})$ by passing "messages" between nodes

Pearl, Judea. *Proceedings of the Second National Conference on Artificial Intelligence,* 1982

Message passing

Exploiting the sparsity of the underlying graph



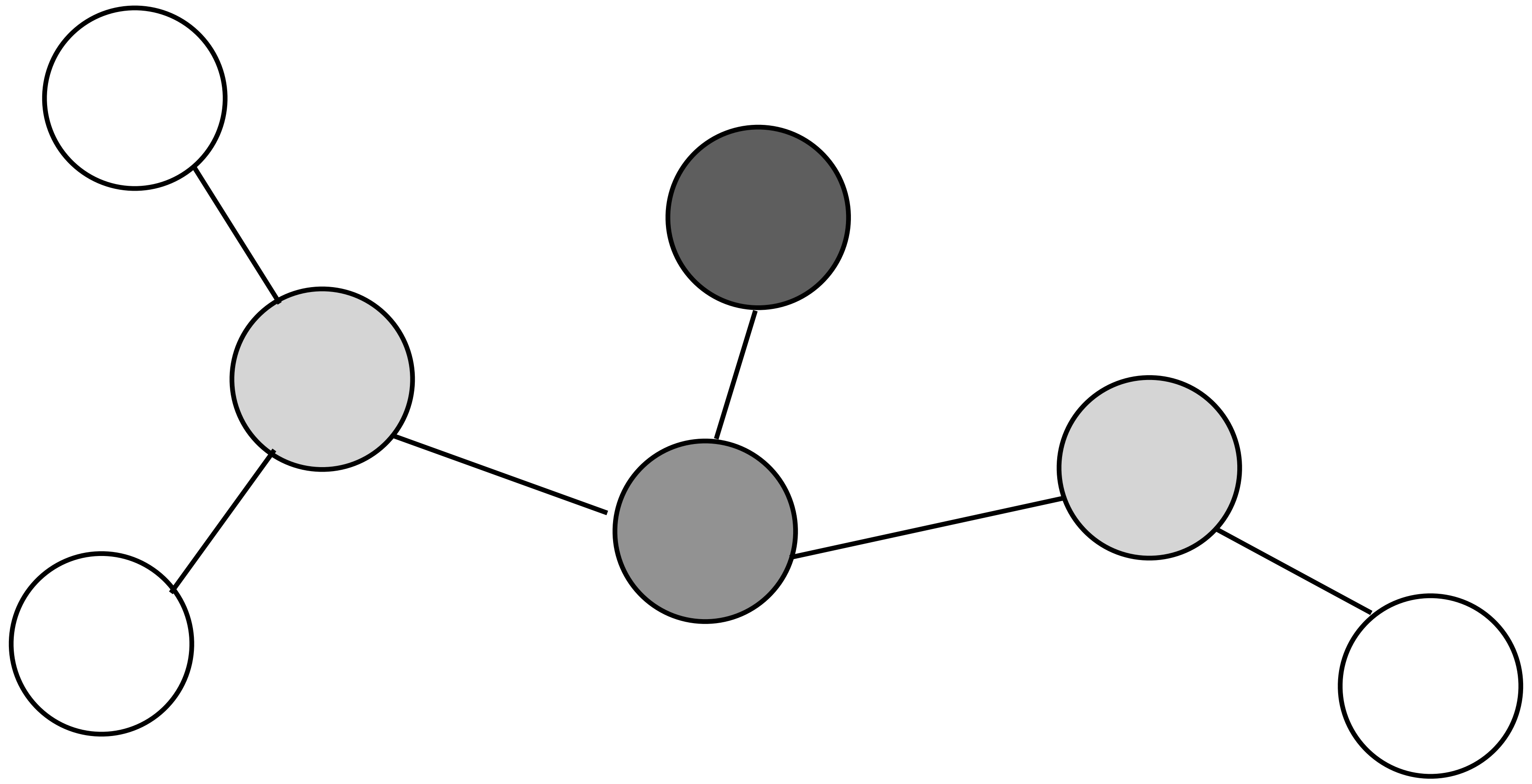
Belief Propagation (Pearl ’82):

Algorithm to efficiently compute marginals $p(z_i \mid \boldsymbol{y})$ by passing “messages” between nodes

Pearl, Judea. *Proceedings of the Second National Conference on Artificial Intelligence,* 1982

Message passing

Exploiting the sparsity of the underlying graph



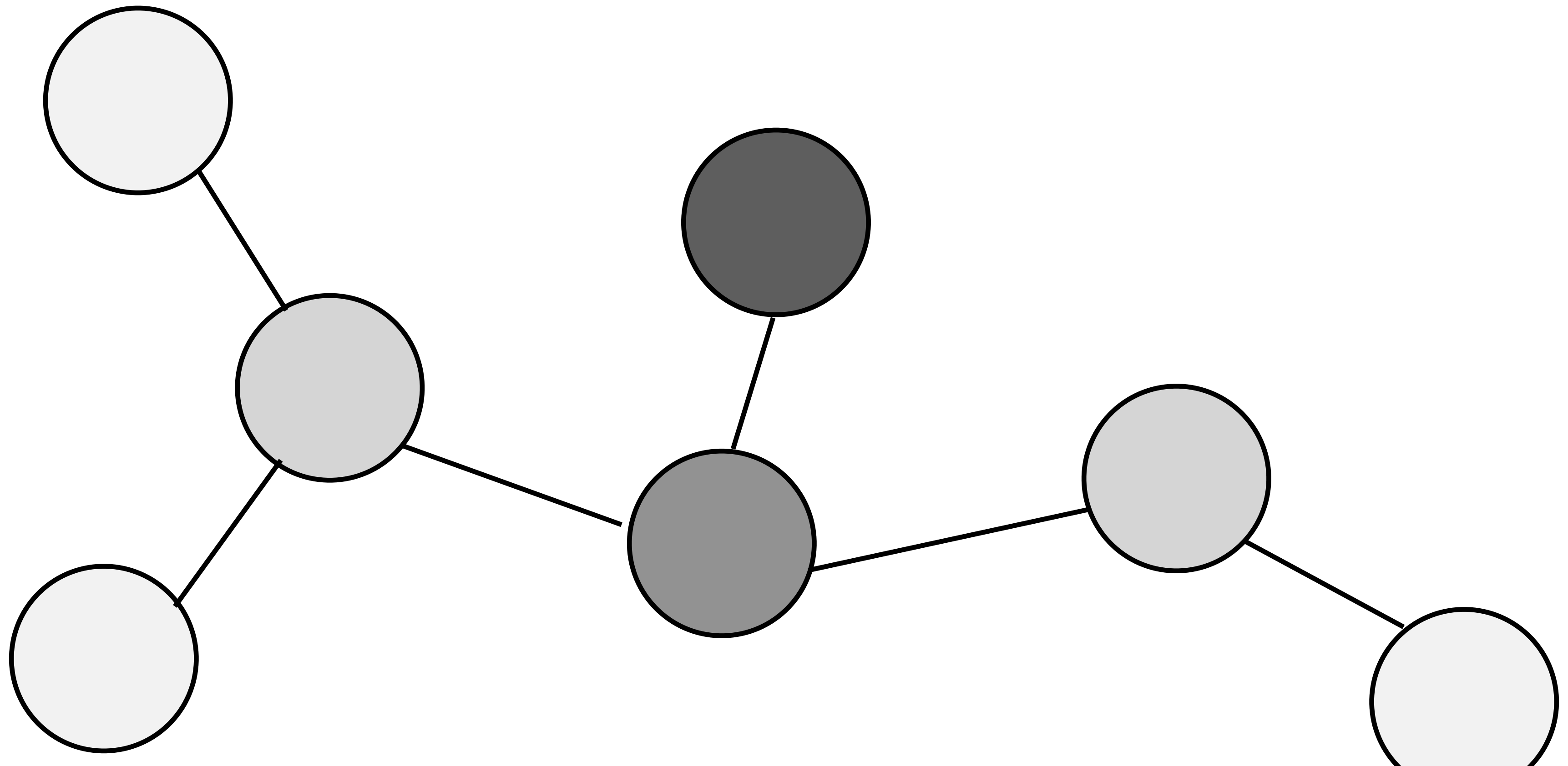
Belief Propagation (Pearl ’82):

Algorithm to efficiently compute marginals $p(z_i \mid \boldsymbol{y})$ by passing “messages” between nodes

Pearl, Judea. *Proceedings of the Second National Conference on Artificial Intelligence,* 1982

Message passing

Exploiting the sparsity of the underlying graph



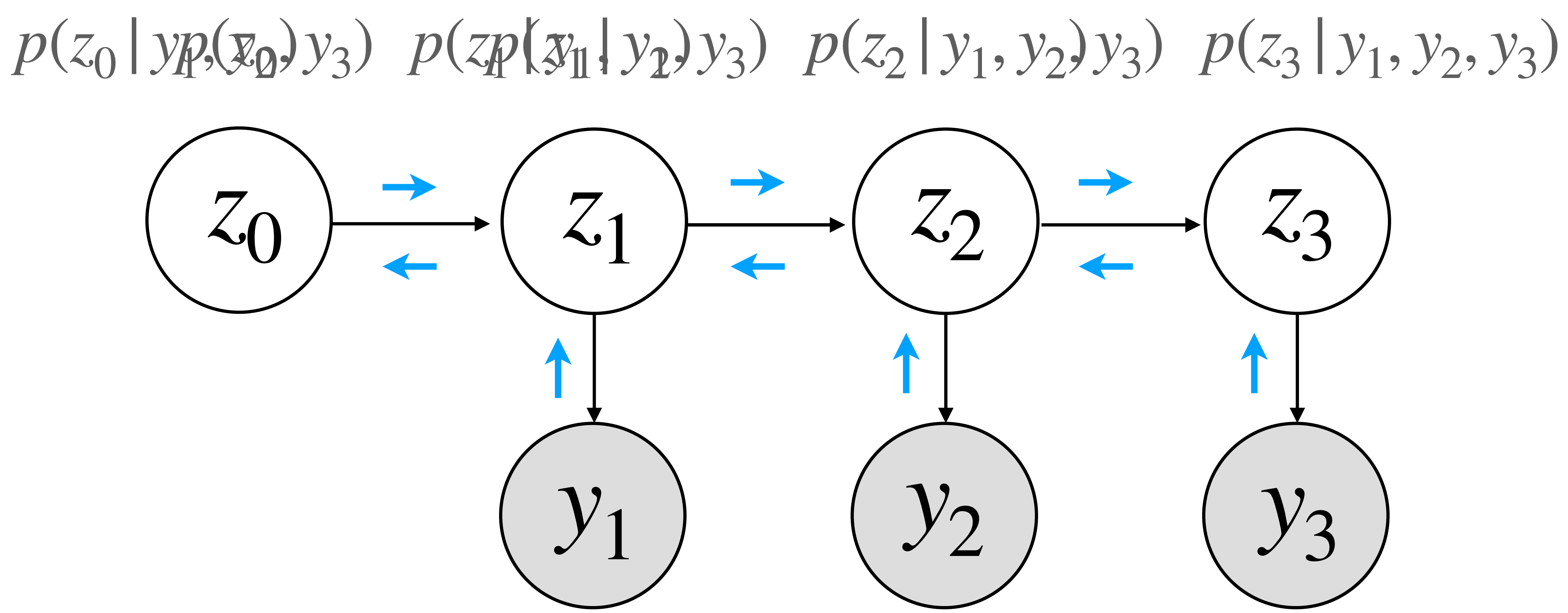
Belief Propagation (Pearl ’82):

Algorithm to efficiently compute marginals $p(z_i \mid \boldsymbol{y})$ by passing “messages” between nodes

Pearl, Judea. *Proceedings of the Second National Conference on Artificial Intelligence,* 1982

Example: Kalman filtering/smoothing

Assume: Linear dynamical system + linear observation operator



Then, the Kalman filter/smoother are special cases of belief propagation!

Iterative nonlinear state estimation using approximate EP


Sanket Kamthe*
(UCL)


So Takao*
(Caltech)


Marc Deisenroth
(Alan Turing Institute)

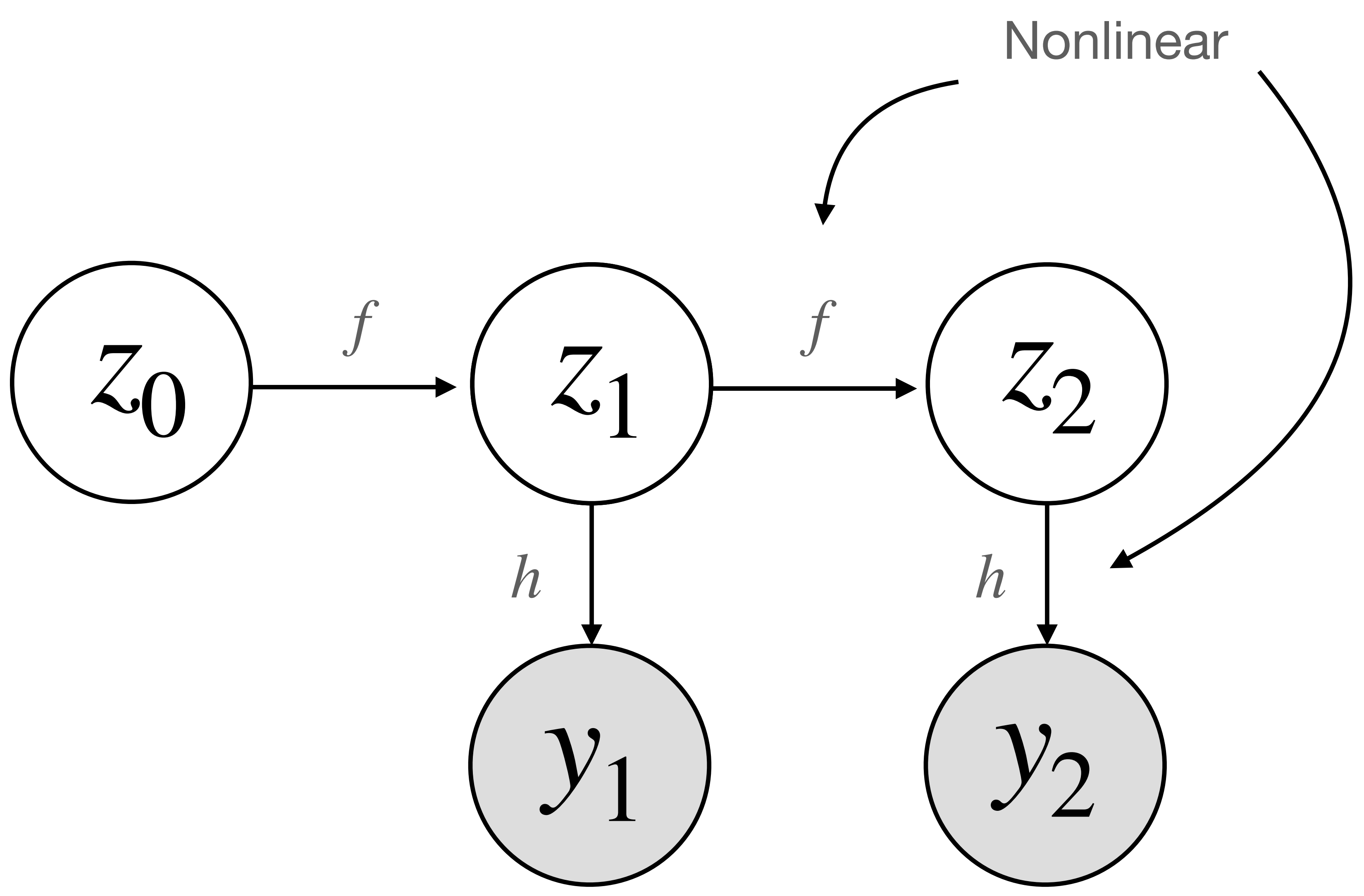

Shakir Mohamed
(Deepmind)

Kamthe, Takao, Deisenroth, Mohamed. *Transactions of Machine Learning Research, 2022*

Nonlinear data assimilation

$$z_{n+1} = f(z_n) + \varepsilon_n, \quad z_0 \sim \mathcal{N}(m, C)$$

$$y_n = h(z_n) + \eta_n, \quad n = 1,\ldots,N$$

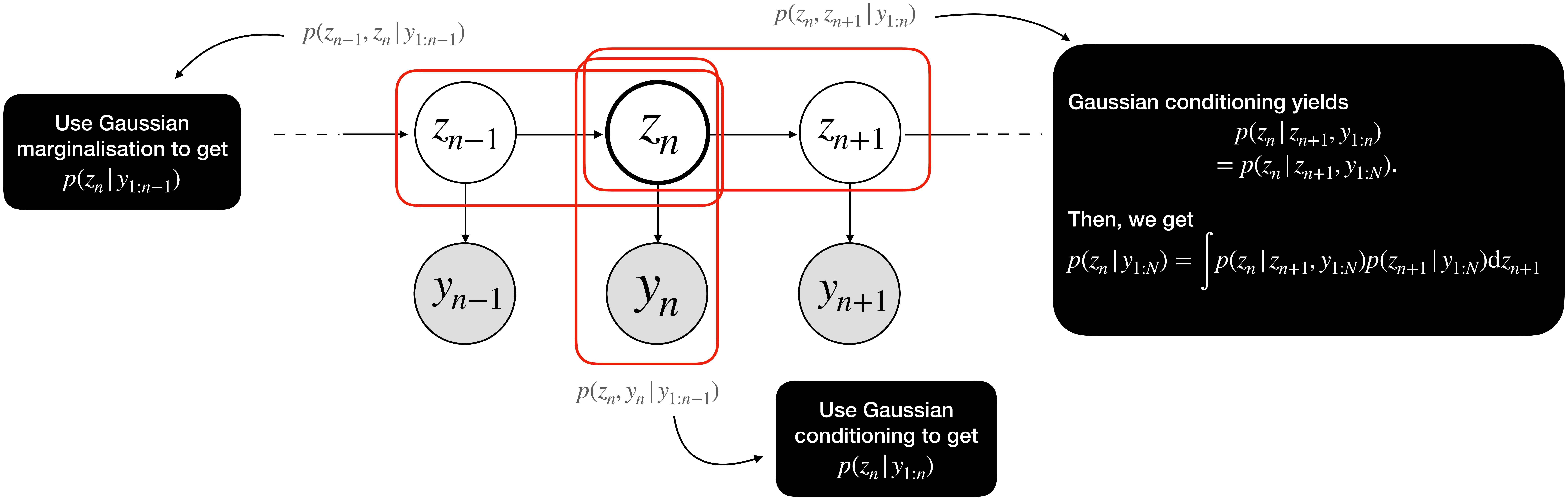


No closed-form solution exists for $p(z_n \,|\, y_{1:N})$…

However, various *Gaussian* approximations exists (e.g. ExKS, UKS, EnKS, …)

Framework for Gaussian filters/smoothers

Generally, Gaussian approximate filters/smoothers can be obtained via Gaussian approximations to $p(z_{n-1}, z_n \mid y_{1:n-1})$, $p(z_n, y_n \mid y_{1:n-1})$, $p(z_n, z_{n+1} \mid y_{1:n})$*



*Deisenroth and Ohlsson. *Proceedings of the American Control Conference, 2011*

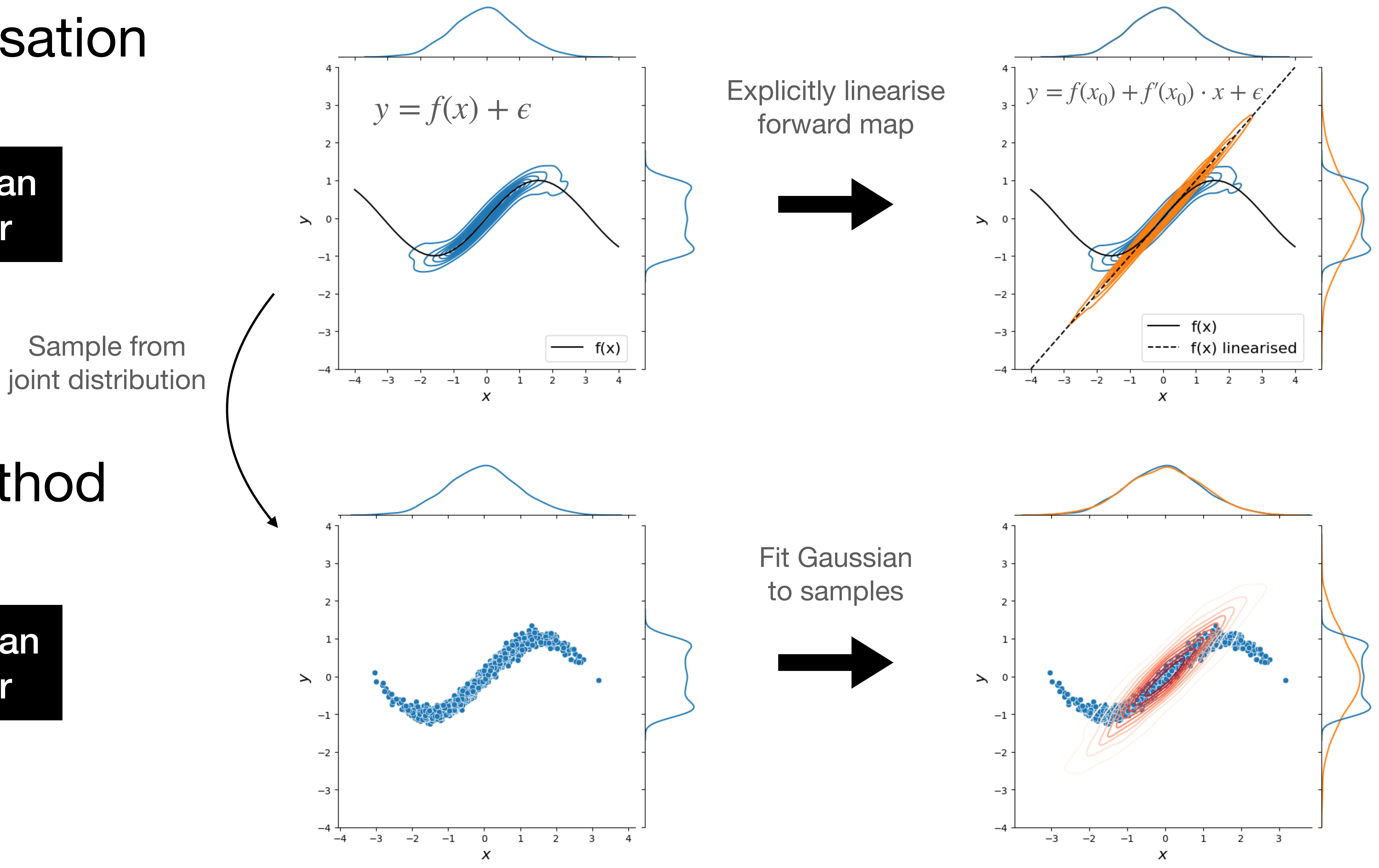
Methods for Gaussian projection

- Explicit linearisation

Extended Kalman filter/smoother

- Ensemble method

Ensemble Kalman filter/smoother



Can we find “better” Gaussian projections to improve approximations?

Variational inference

Find $q(z_n) \approx p(z_n \mid y_{1:N})$ for q in a sub-family of densities (e.g. Gaussians)

such that it minimises $KL(q \mid\mid p)$ or $KL(p \mid\mid q)$

Expectation Propagation (EP)*

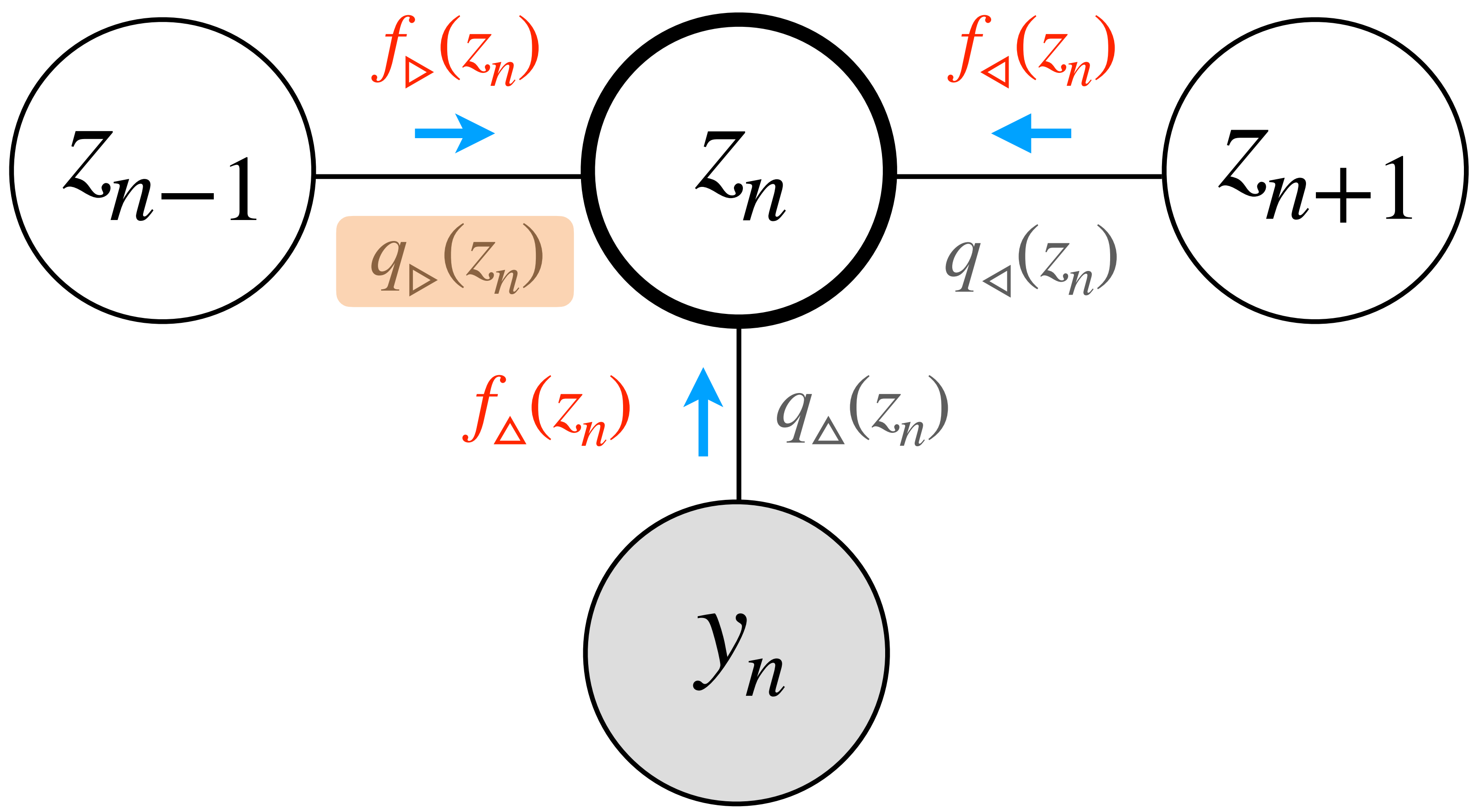
$$q^* = \arg\min_{q \in \mathfrak{G}} KL(p \mid\mid q),$$

Equivalent to *moment matching*

(i.e., mean & covariance)

*Minka, Thomas. *UAI, 2001*

Expectation Propagation (details)



Consider decomposition

$$p(z_n \,|\, y_{1:N}) \propto f_{\triangleright}(z_n)\; f_{\vartriangle}(z_n)\; f_{\triangleleft}(z_n)$$

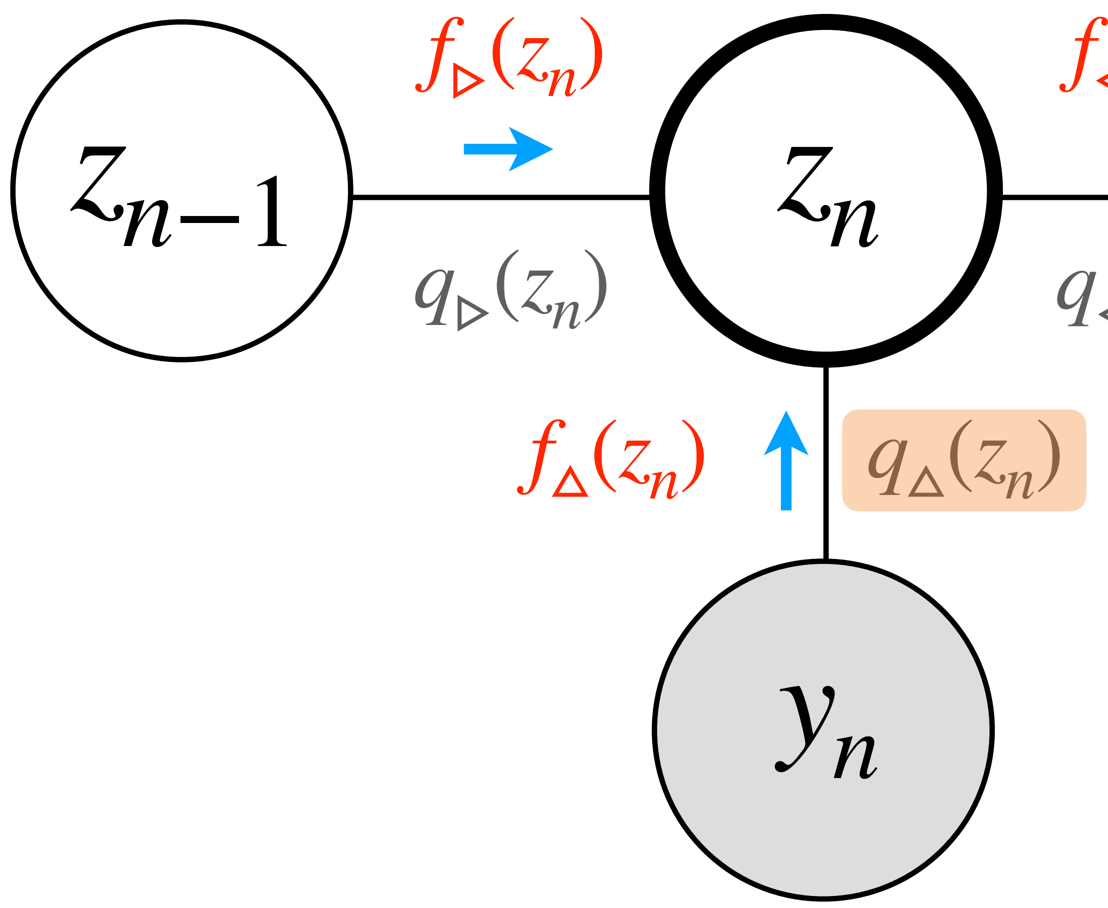
and q of the form

$$q(z_n) \propto q_{\triangleright}(z_n)\; q_{\vartriangle}(z_n)\; q_{\triangleleft}(z_n)$$

Step 1: $$q^* = \arg\min_{q \in \mathcal{G}} KL(f_{\triangleright} q_{\vartriangle} q_{\triangleleft} \,||\, q)$$

Step 2: $$q_{\triangleright}(z_n) \leftarrow \frac{q^*(z_n)}{q_{\vartriangle}(z_n) q_{\triangleleft}(z_n)}$$

Expectation Propagation (details)



Assume

$$p(z_n \mid y_{1:N}) \propto f_{\triangleright}(z_n)\, f_{\vartriangle}(z_n)\, f_{\triangleleft}(z_n)$$

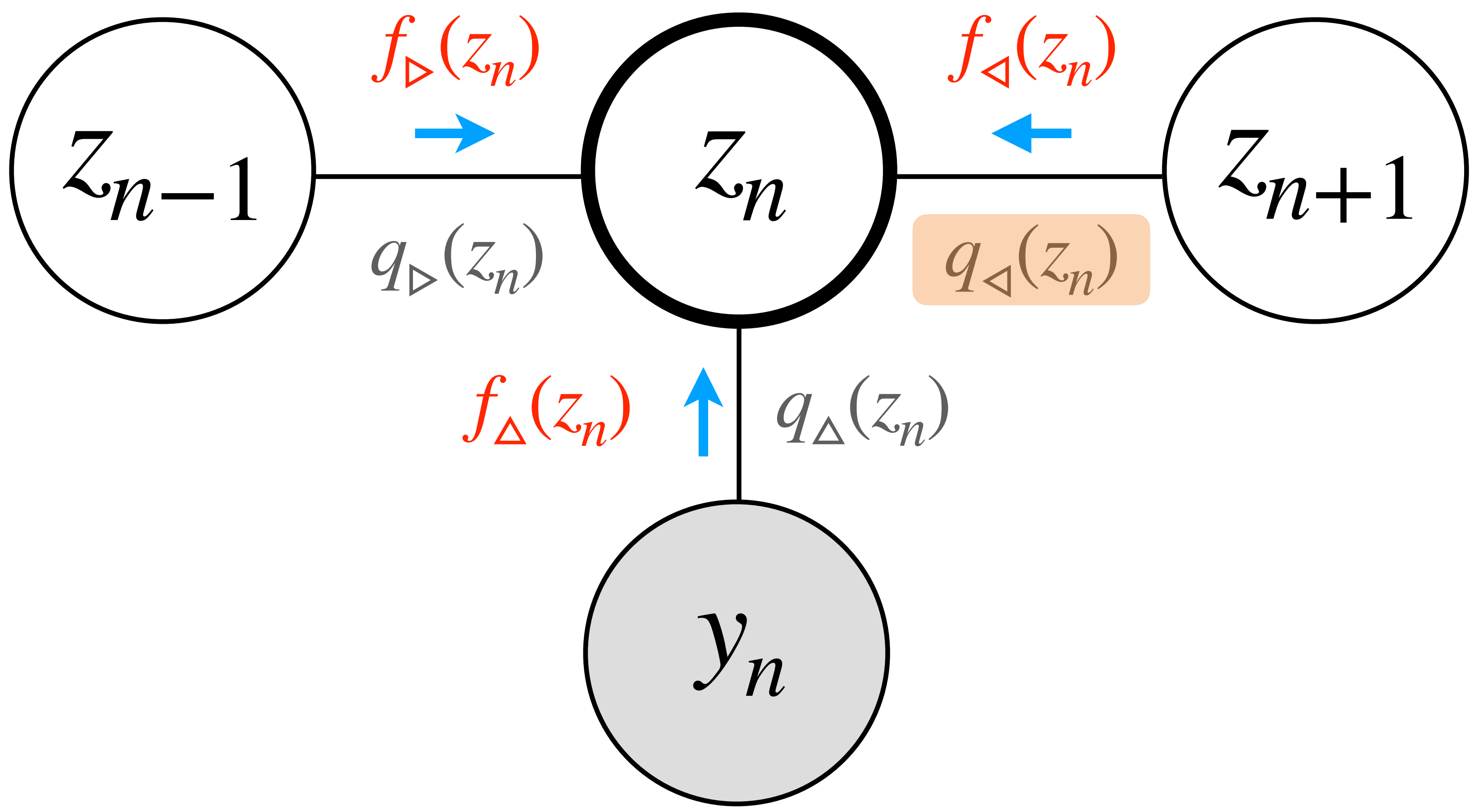
and consider q of the form

$$q(z_n) \propto q_{\triangleright}(z_n)\; q_{\vartriangle}(z_n)\; q_{\triangleleft}(z_n)$$

Step 1: $q^* = \underset{q \in \mathcal{G}}{\arg\min}\, KL(q_{\triangleright} f_{\vartriangle} q_{\triangleleft} \,||\, q)$

Step 2: $q_{\vartriangle}(z_n) \leftarrow \dfrac{q^*(z_n)}{q_{\triangleright}(z_n) q_{\triangleleft}(z_n)}$

Expectation Propagation (details)



Assume

$$p(z_n \mid y_{1:N}) \propto f_\triangleright(z_n)\ f_\vartriangle(z_n)\ f_\triangleleft(z_n)$$

and consider q of the form

$$q(z_n) \propto q_\triangleright(z_n)\ q_\vartriangle(z_n)\ q_\triangleleft(z_n)$$

Step 1: $q^* = \arg\min_{q \in \mathcal{G}} KL(q_\triangleright q_\vartriangle f_\triangleleft \,||\, q)$

Step 2: $q_\triangleleft(z_n) \leftarrow \dfrac{q^*(z_n)}{q_\triangleright(z_n) q_\vartriangle(z_n)}$

Can iterate to get better approximations!

Approximate EP algorithm

Approximate EP for State Estimation

$$p(z_n | y_{1:N}) \propto \underbrace{p(z_n | y_{1:n-1})}_{f_\triangleright(z_n) \approx q_\triangleright(z_n)} \underbrace{p(y_n | z_n)}_{f_\vartriangle(z_n) \approx q_\vartriangle(z_n)} \underbrace{p(y_{n+1:N} | z_n)}_{f_\triangleleft(z_n) \approx q_\triangleleft(z_n)}$$

Updating the forward message $q_\triangleright(z_n)$:

$$q^* = \arg\min_{q \in \mathcal{G}} KL(f_\triangleright q_\vartriangle q_\triangleleft || q)$$

Filtering distribution

$$f_\triangleright(z_n) = p(z_n | y_{1:n-1}) = \int p(z_n | z_{n-1}) p(z_{n-1} | y_{1:n-1}) \mathrm{d}z_{n-1}$$

$$\approx \int p(z_n | z_{n-1}) q_\triangleright(z_{n-1}) q_\vartriangle(z_{n-1}) \mathrm{d}z_{n-1}$$

$$\approx \int p_G(z_{n-1}, z_n) \mathrm{d}z_{n-1} = p_G(z_n)$$

Gaussian approx.

Hence, we get

$$q_\triangleright(z_n) \leftarrow \frac{q^*(z_n)}{q_\vartriangle(z_n) q_\triangleleft(z_n)}$$

$$\approx \frac{p_G(z_n)\, \cancel{q_\vartriangle(z_n)}\, \cancel{q_\triangleleft(z_n)}}{\cancel{q_\vartriangle(z_n)}\, \cancel{q_\triangleleft(z_n)}}$$

Approximate EP for State Estimation

$$p(z_n \,|\, y_{1:N}) \propto \underbrace{p(z_n \,|\, y_{1:n-1})}_{f_{\triangleright}(z_n) \approx q_{\triangleright}(z_n)} \; \underbrace{p(y_n \,|\, z_n)}_{f_{\triangle}(z_n) \approx q_{\triangle}(z_n)} \; \underbrace{p(y_{n+1:N} \,|\, z_n)}_{f_{\triangleleft}(z_n) \approx q_{\triangleleft}(z_n)}$$

Updating the upward message $q_{\triangle}(z_n)$:

$$q^* = \arg\min_{q \in \mathcal{G}} KL(q_{\triangleright} f_{\triangle} q_{\triangleleft} \,||\, q)$$

$$q_{\triangleright}(z_n) f_{\triangle}(z_n)\, q_{\triangleleft}(z_n) = q_{\triangleright}(z_n)\, p(y_n \,|\, z_n)\, q_{\triangleleft}(z_n)$$

$$Z_{\triangle} := \int q_{\triangleright}(z_n)\, p(y_n \,|\, z_n)\, q_{\triangleleft}(z_n) \mathrm{d}z_n$$

$$\approx \int p_G(y_n, z_n) \mathrm{d}z_n$$

Gaussian approx.

Trick:
Moments of $f(z)q(z)$ with Gaussian $q(z)$ can be calculated from derivatives of the log-partition function $\log Z$ w.r.t. the moments of $q(z)$.*

*Minka, Thomas. *PhD thesis, 2001*

Approximate EP for State Estimation

$$p(z_n \mid y_{1:N}) \propto \underbrace{p(z_n \mid y_{1:n-1})}_{f_{\triangleright}(z_n) \approx q_{\triangleright}(z_n)} \underbrace{p(y_n \mid z_n)}_{f_{\triangle}(z_n) \approx q_{\triangle}(z_n)} \underbrace{p(y_{n+1:N} \mid z_n)}_{f_{\triangleleft}(z_n) \approx q_{\triangleleft}(z_n)}$$

Updating the backward message $q_{\triangleleft}(z_n)$:

$$q^* = \arg\min_{q \in \mathcal{G}} KL(q_{\triangleright} q_{\triangle} f_{\triangleleft} \,||\, q)$$

$$f_{\triangleleft}(z_n) = p(y_{n+1:N} \mid z_n) = \int p(y_{n+1:N} \mid z_{n+1}) p(z_{n+1} \mid z_n) \mathrm{d}z_{n+1}$$

$$\approx \int q_{\triangle}(z_{n+1}) q_{\triangleleft}(z_{n+1}) p(z_{n+1} \mid z_n) \mathrm{d}z_{n+1}$$

$$Z_{\triangleleft} = \int q_{\triangleright}(z_n) q_{\triangle}(z_n) f_{\triangleleft}(z_n) \mathrm{d}z_n$$

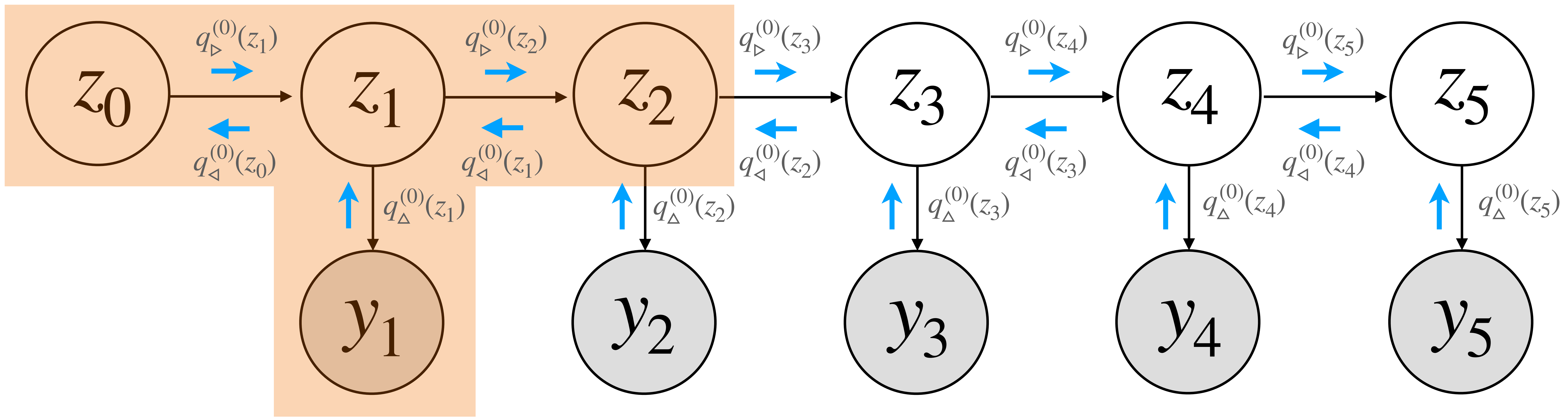
$$\approx \int q_{\triangleright}(z_n) q_{\triangle}(z_n) \left(\int q_{\triangle}(z_{n+1}) q_{\triangleleft}(z_{n+1}) p(z_{n+1} \mid z_n) \mathrm{d}z_{n+1} \right) \mathrm{d}z_n$$

$$= \int \left(\int p(z_{n+1} \mid z_n) q_{\triangleright}(z_n) q_{\triangle}(z_n) \mathrm{d}z_n \right) q_{\triangle}(z_{n+1})\, q_{\triangleleft}(z_{n+1}) \mathrm{d}z_{n+1}$$

$$\approx \int \left(\int p_G(z_n, z_{n+1}) \mathrm{d}z_n \right) q_{\triangle}(z_{n+1})\, q_{\triangleleft}(z_{n+1}) \mathrm{d}z_{n+1}$$

Gaussian approx.

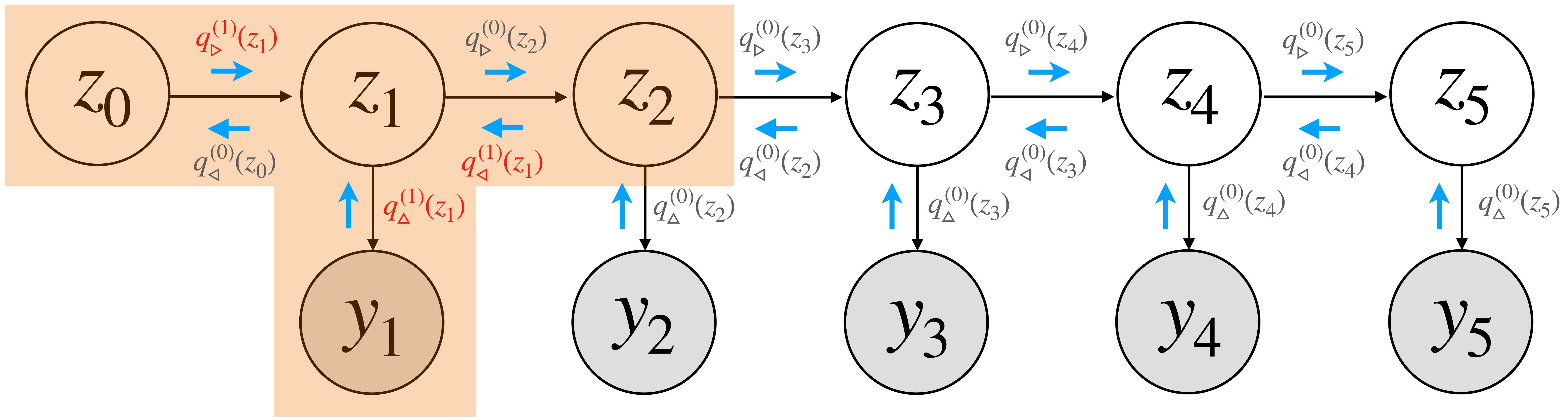
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering ≡ single forward pass of approximate EP
- Nonlinear Gaussian smoothing ≡ single forward-backward pass of approximate EP

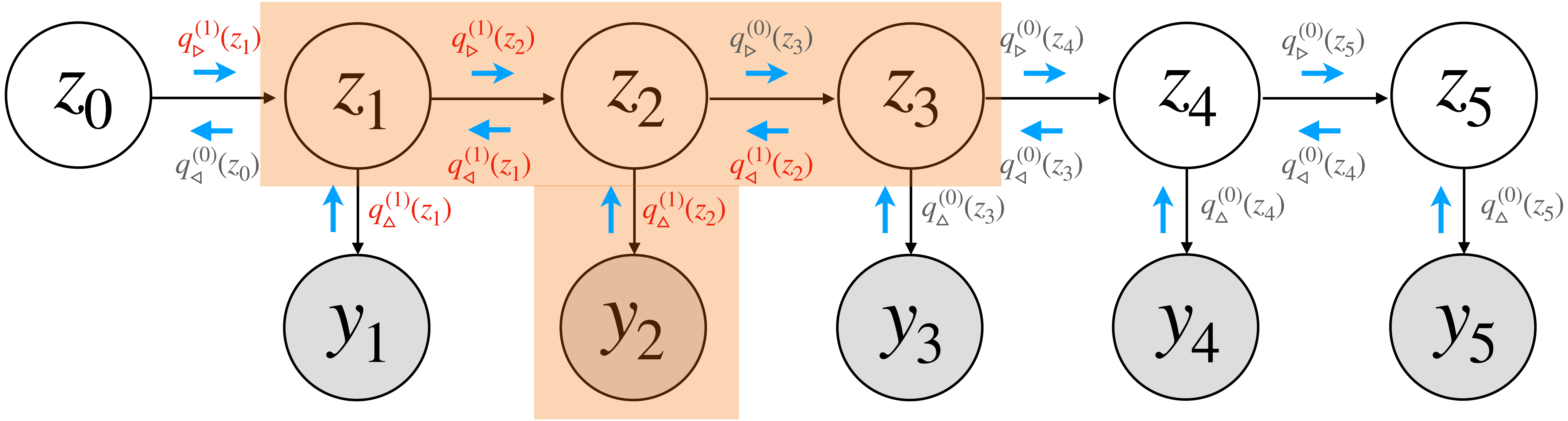
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering ≡ single forward pass of approximate EP
- Nonlinear Gaussian smoothing ≡ single forward-backward pass of approximate EP

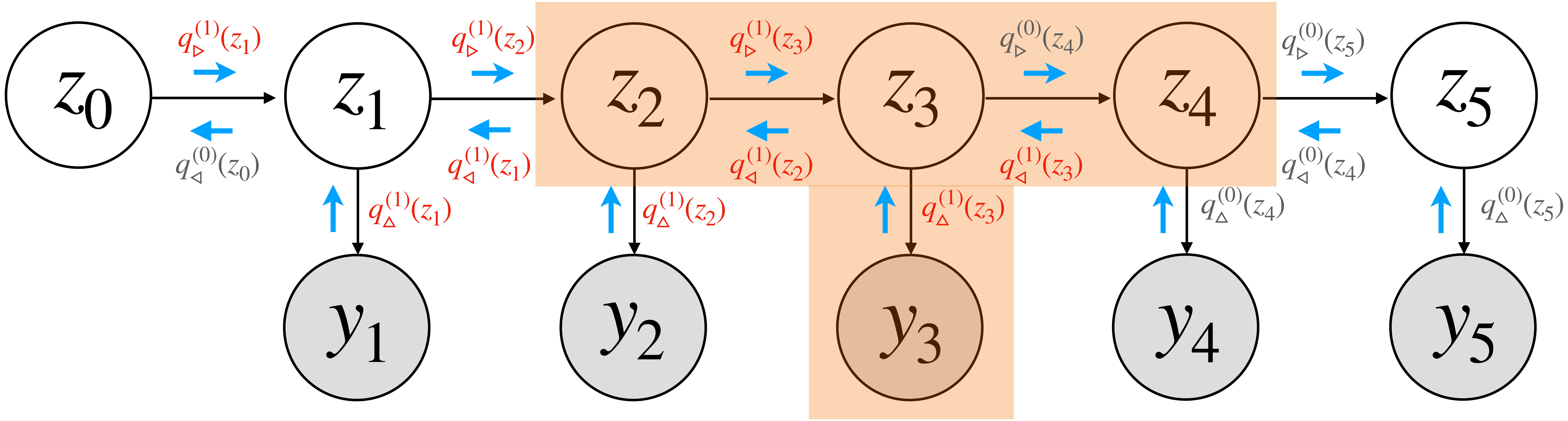
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering ≡ single forward pass of approximate EP
- Nonlinear Gaussian smoothing ≡ single forward-backward pass of approximate EP

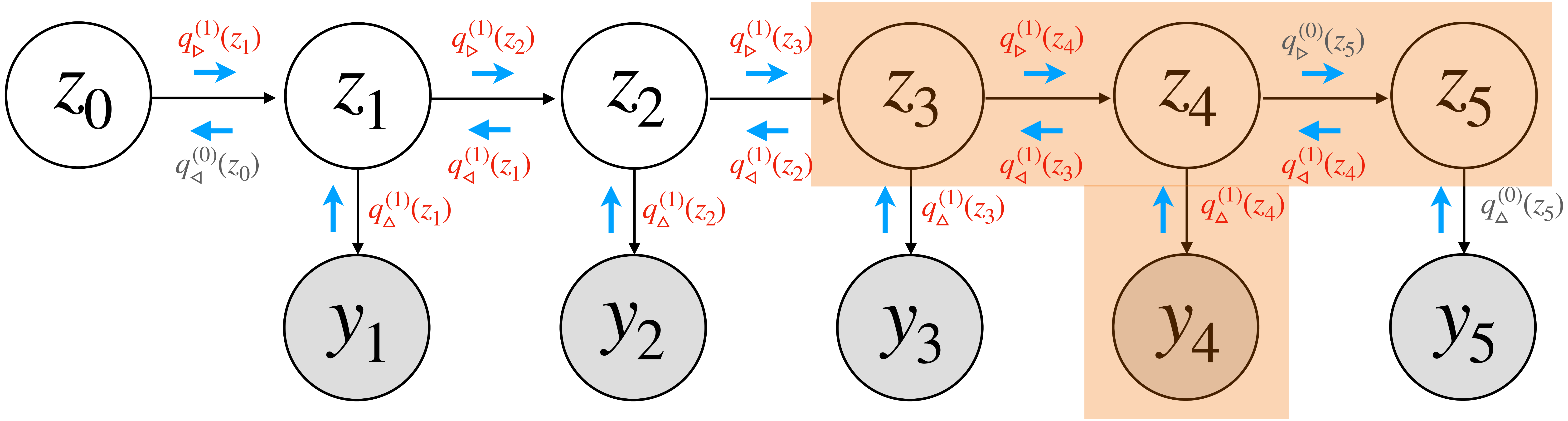
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering ≡ single forward pass of approximate EP
- Nonlinear Gaussian smoothing ≡ single forward-backward pass of approximate EP

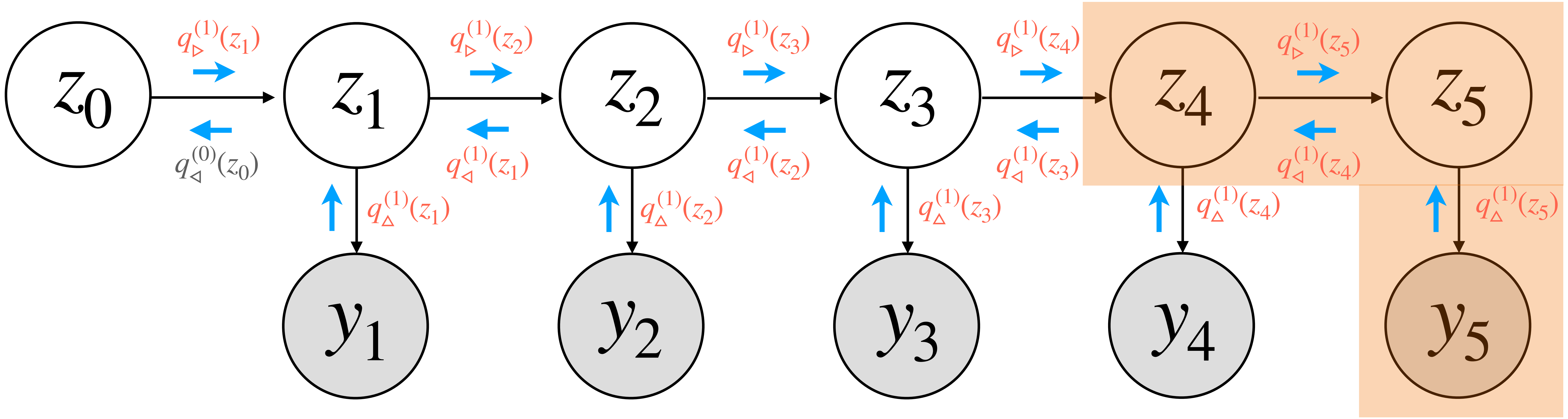
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering $\equiv$ single forward pass of approximate EP
- Nonlinear Gaussian smoothing $\equiv$ single forward-backward pass of approximate EP

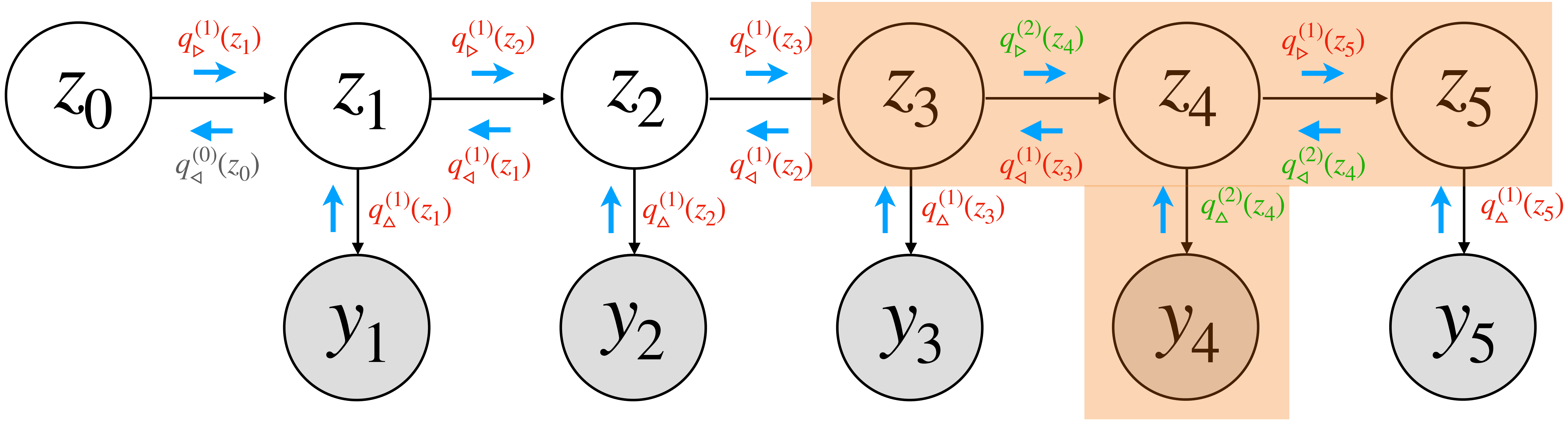
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering ≡ single forward pass of approximate EP
- Nonlinear Gaussian smoothing ≡ single forward-backward pass of approximate EP

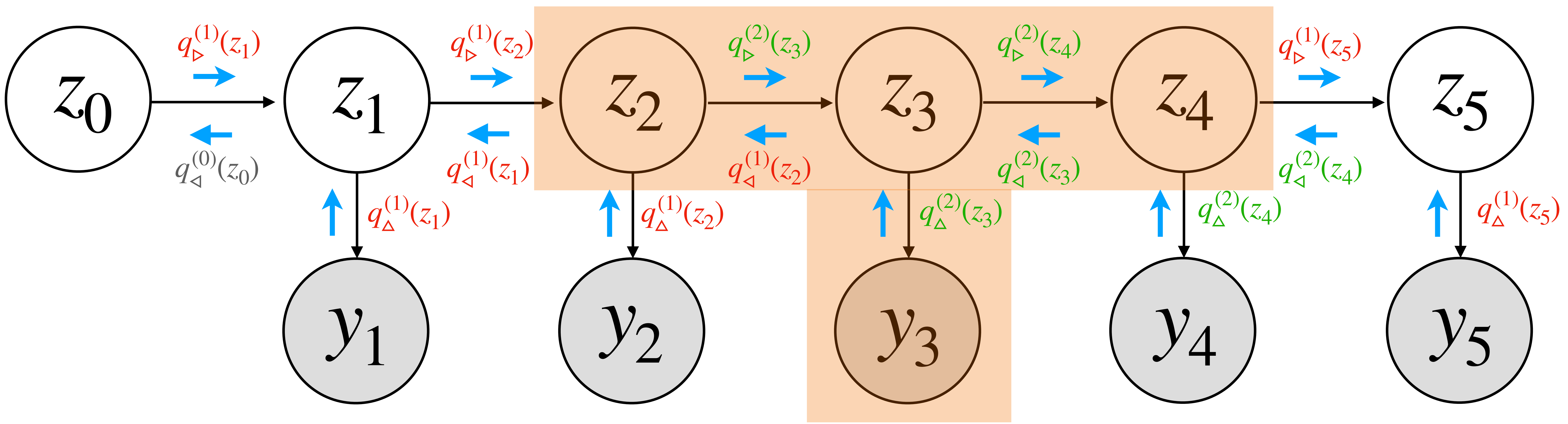
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering $\equiv$ single forward pass of approximate EP
- Nonlinear Gaussian smoothing $\equiv$ single forward-backward pass of approximate EP

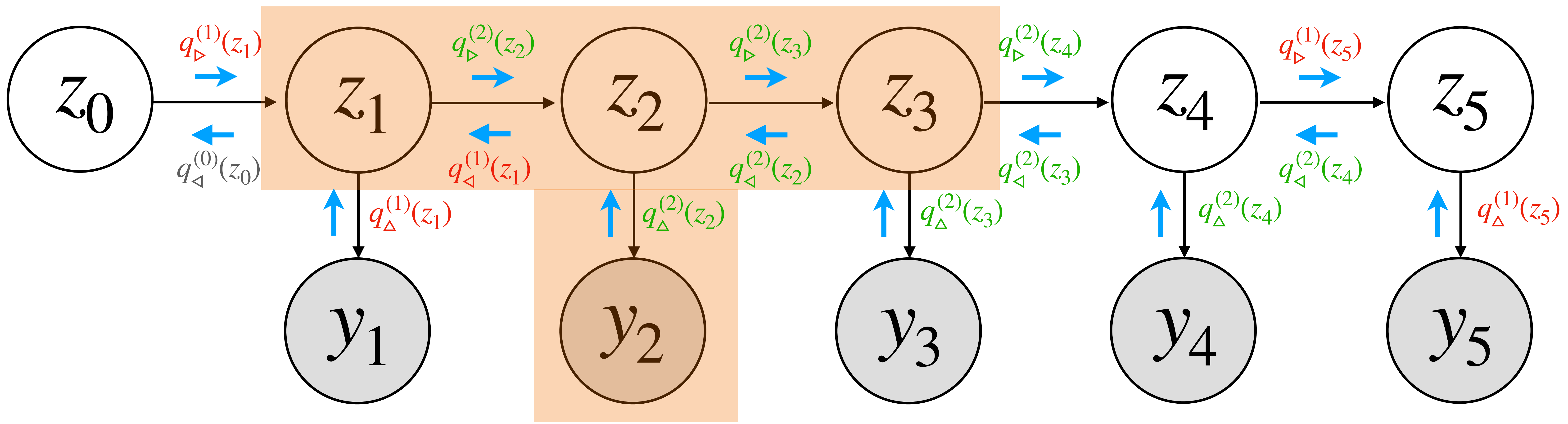
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering $\equiv$ single forward pass of approximate EP
- Nonlinear Gaussian smoothing $\equiv$ single forward-backward pass of approximate EP

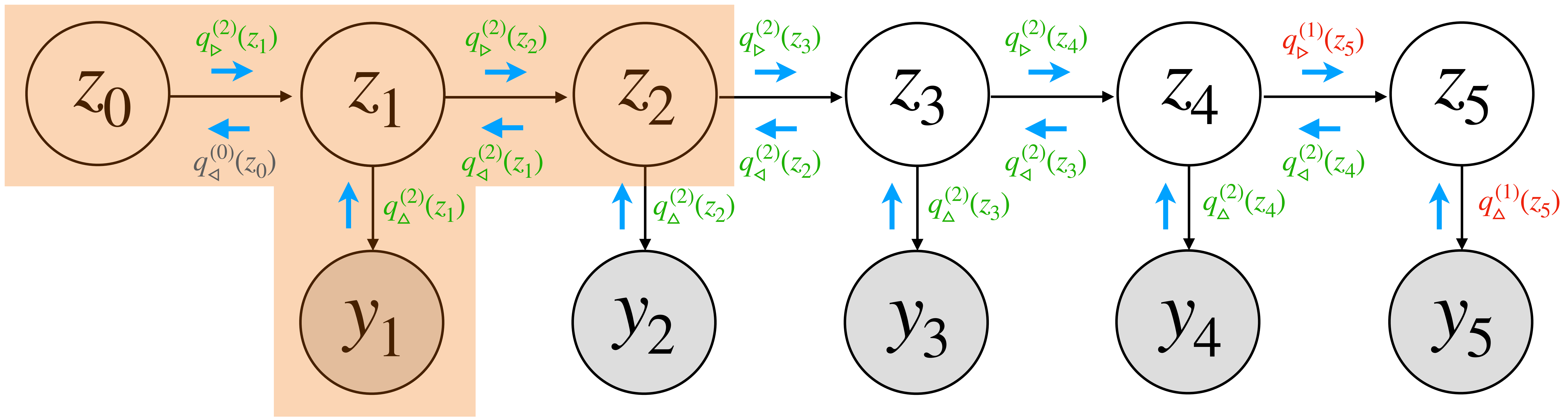
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering ≡ single forward pass of approximate EP
- Nonlinear Gaussian smoothing ≡ single forward-backward pass of approximate EP

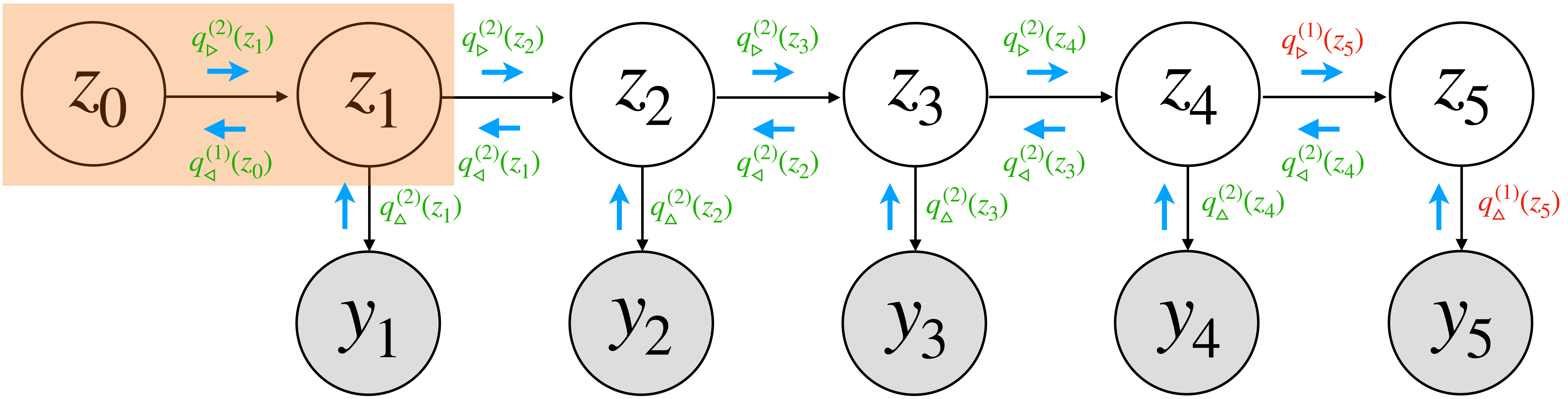
Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering $\equiv$ single forward pass of approximate EP
- Nonlinear Gaussian smoothing $\equiv$ single forward-backward pass of approximate EP

Nonlinear Gaussian filtering/smoothing is approximate EP!



Theorem

- Nonlinear Gaussian filtering ≡ single forward pass of approximate EP
- Nonlinear Gaussian smoothing ≡ single forward-backward pass of approximate EP

Path to iterated smoother

By iterating further, can we get better results?

Related works: Bell (1994), Ypma & Heskes (2003), Bocquet & Sakov (2013)

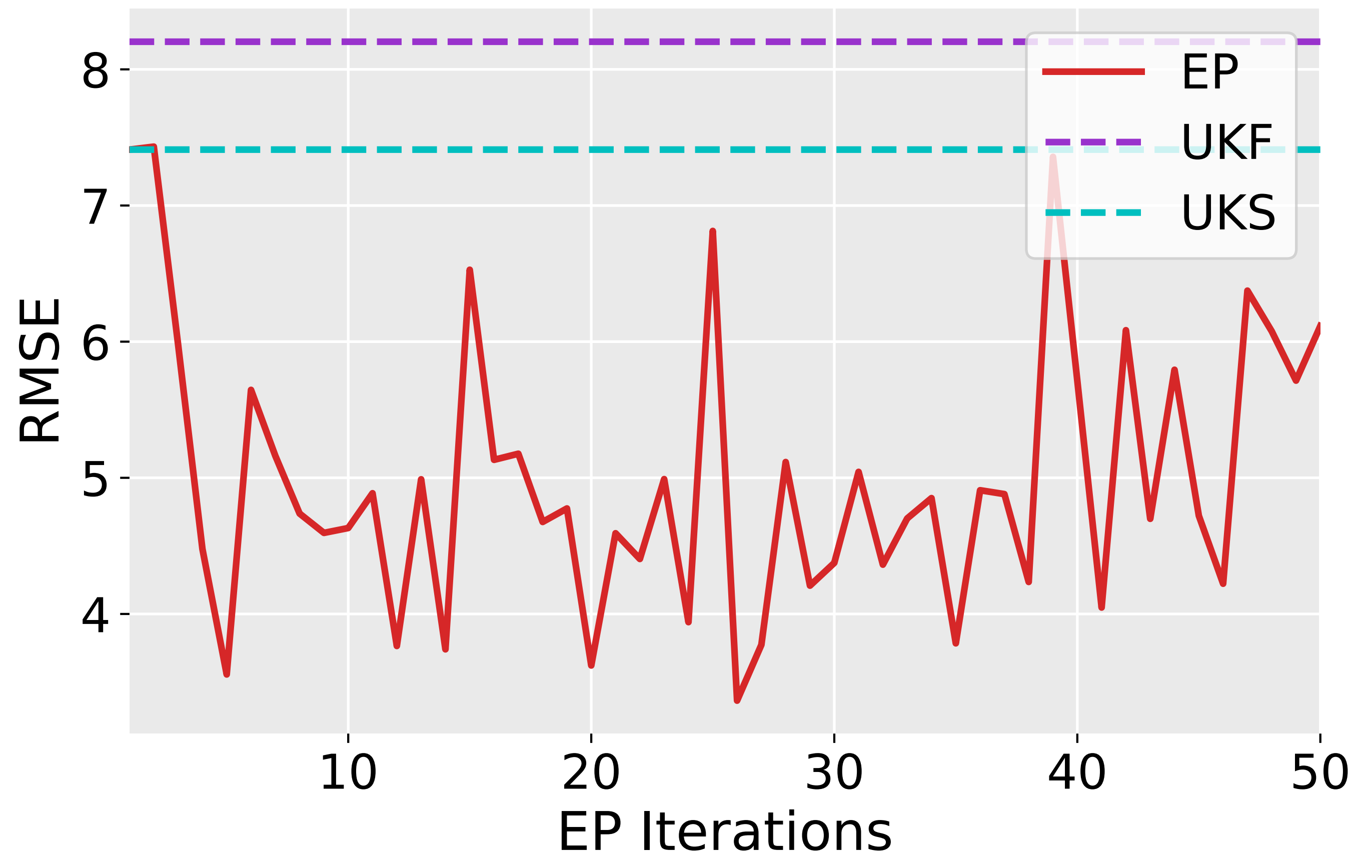
Results on a toy nonlinear problem:

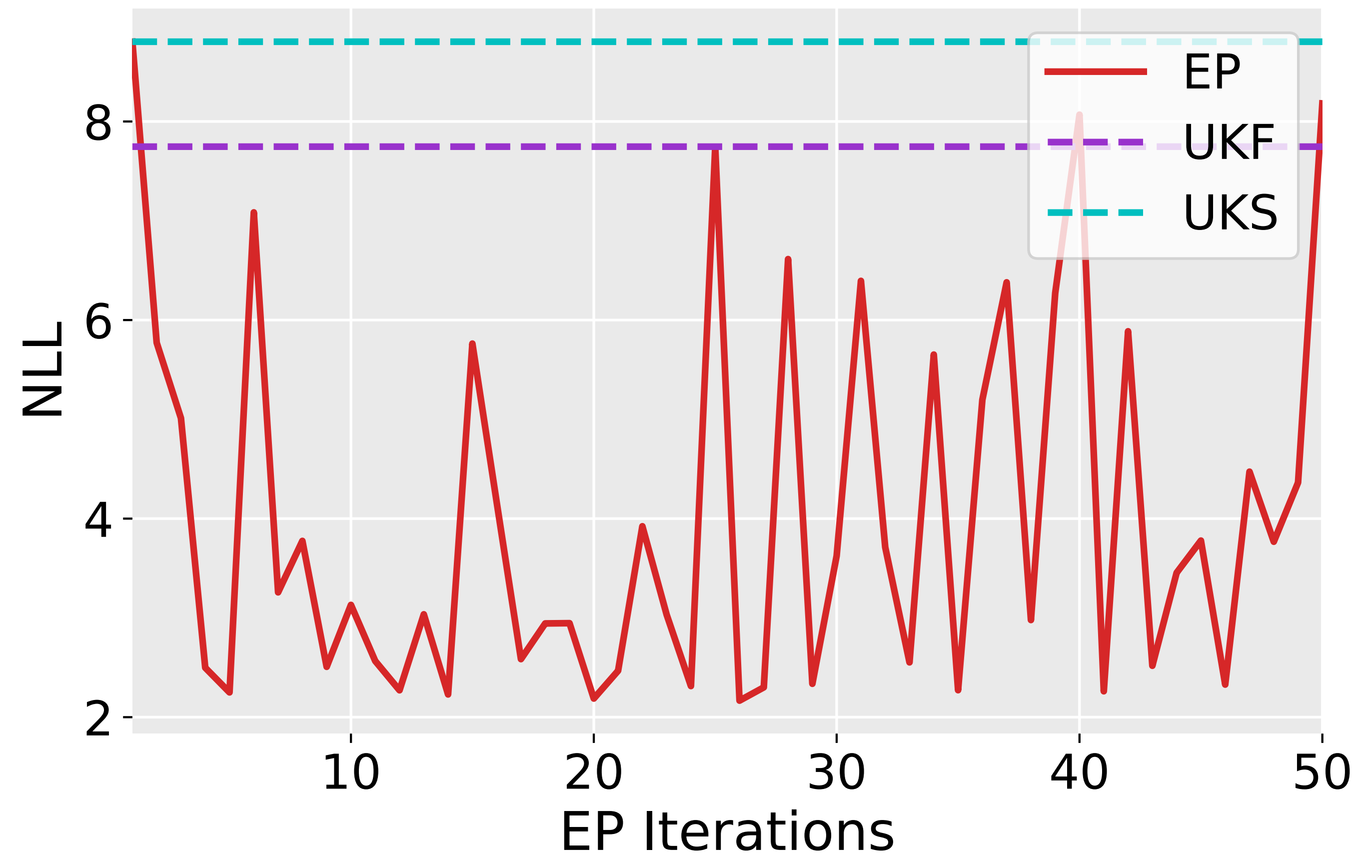
Model:

$$x_{n+1} = \frac{x_n}{2} + \frac{25x_n}{1+x_n^2} + 8\cos(1.2t_n) + \varepsilon_n$$

$$y_n = \frac{x_n^2}{20} + \eta_n$$

(Used unscented transform for Gaussian approx.)





Promising, but unfortunately quite unpredictable…

Bell. *IEEE, 1994*

Ypma and Heskes. *IEEE, 2003*

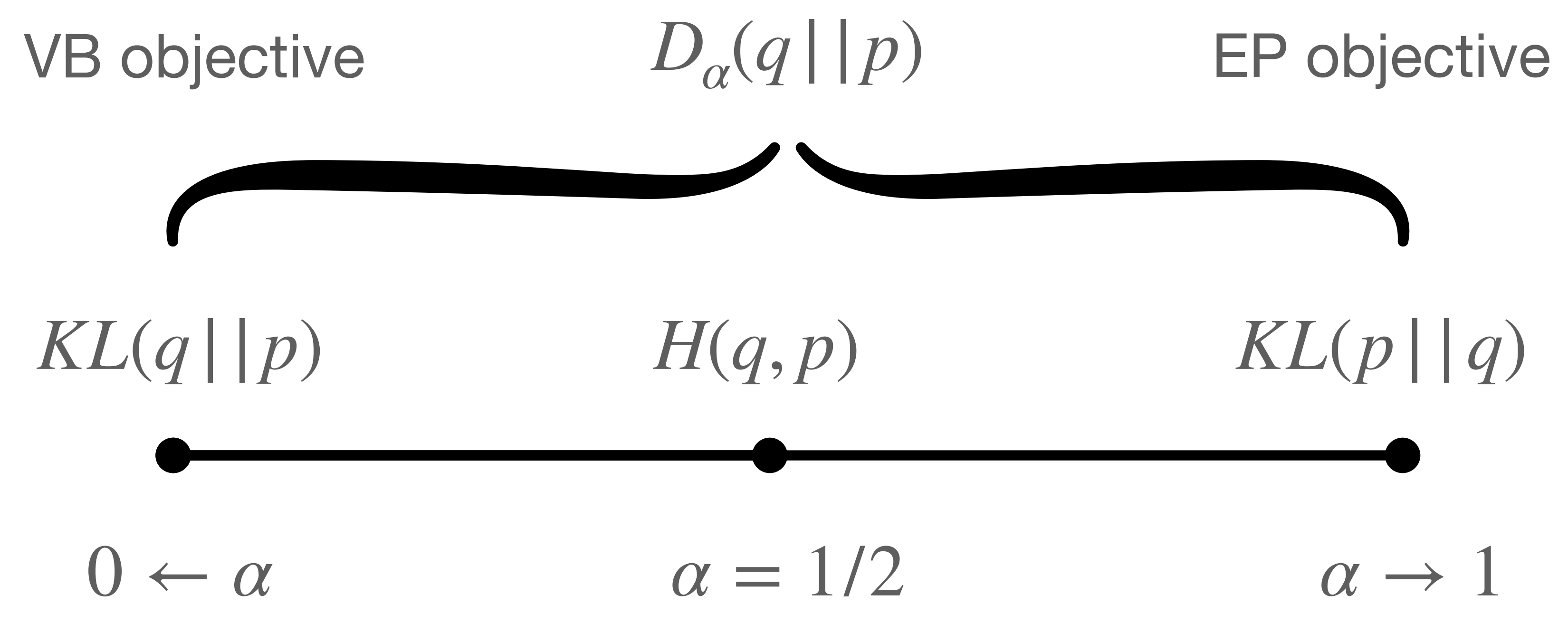
Bocquet and Sakov. *RMets, 2013*

Damped updates and Power EP

Damped updates:

$$q^{new} = \arg\min_{q \in \mathcal{G}} KL(f_i q_{\setminus i} || q), \quad i \in \{ \triangleright , \triangle , \triangleleft \}$$

$$q_i(z_n) \leftarrow \frac{q^{new}(z_n)^{\gamma} q^{old}(z_n)^{1-\gamma}}{q_{\setminus i}(z_n)}, \text{ for some } \gamma \in (0,1]$$



Power EP*:

Minimise *α-divergence*** $D_\alpha(p||q) = \frac{1}{\alpha(1-\alpha)}\left(1 - \int p(z)^{\alpha} q(z)^{1-\alpha} \mathrm{d}z\right)$ for $\alpha \in (0,1)$

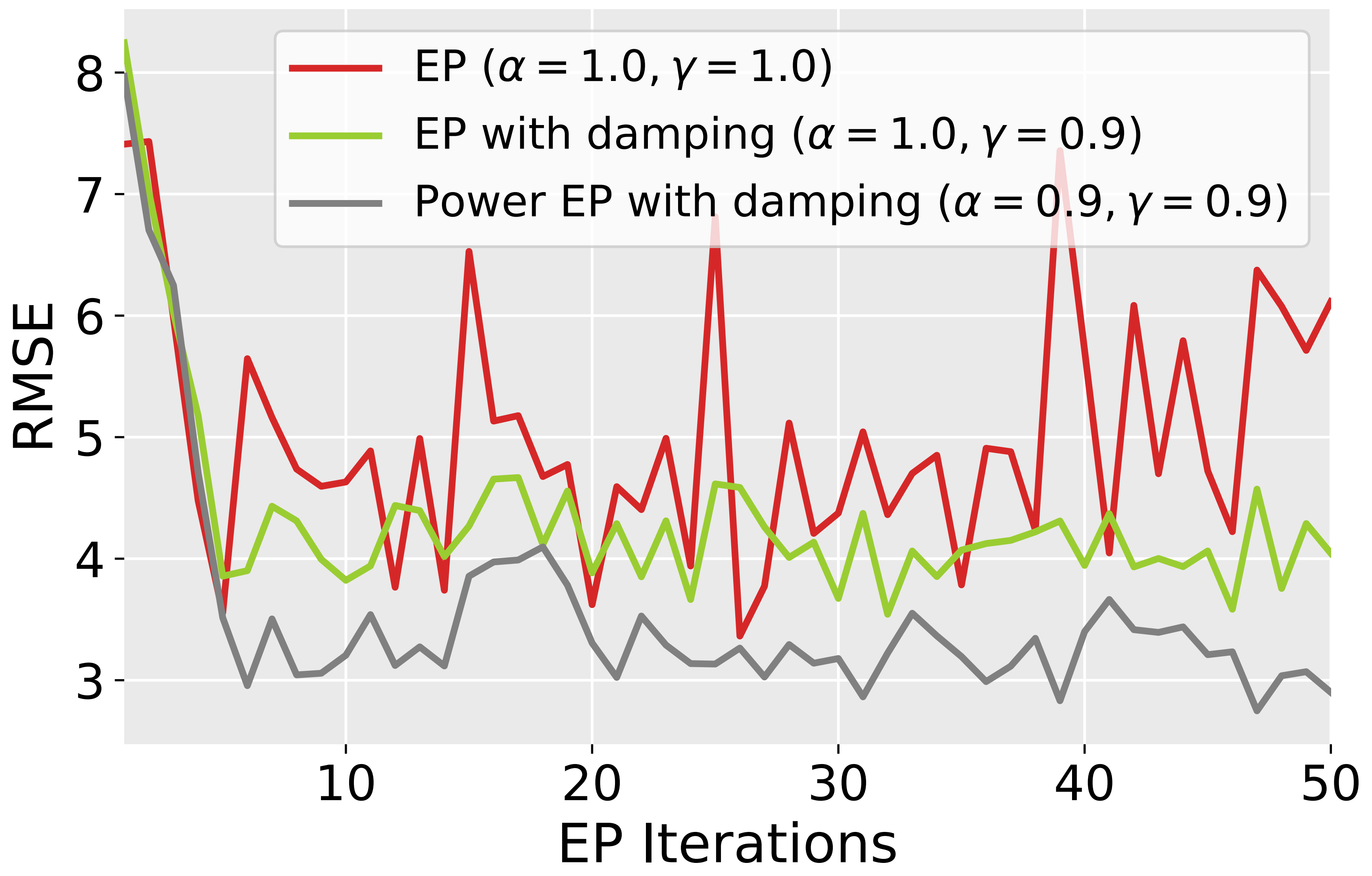
We can derive novel Kalman-type updates via *Approximate Power EP!*

*Minka, Thomas. *Microsoft Technical Research Report, 2004*

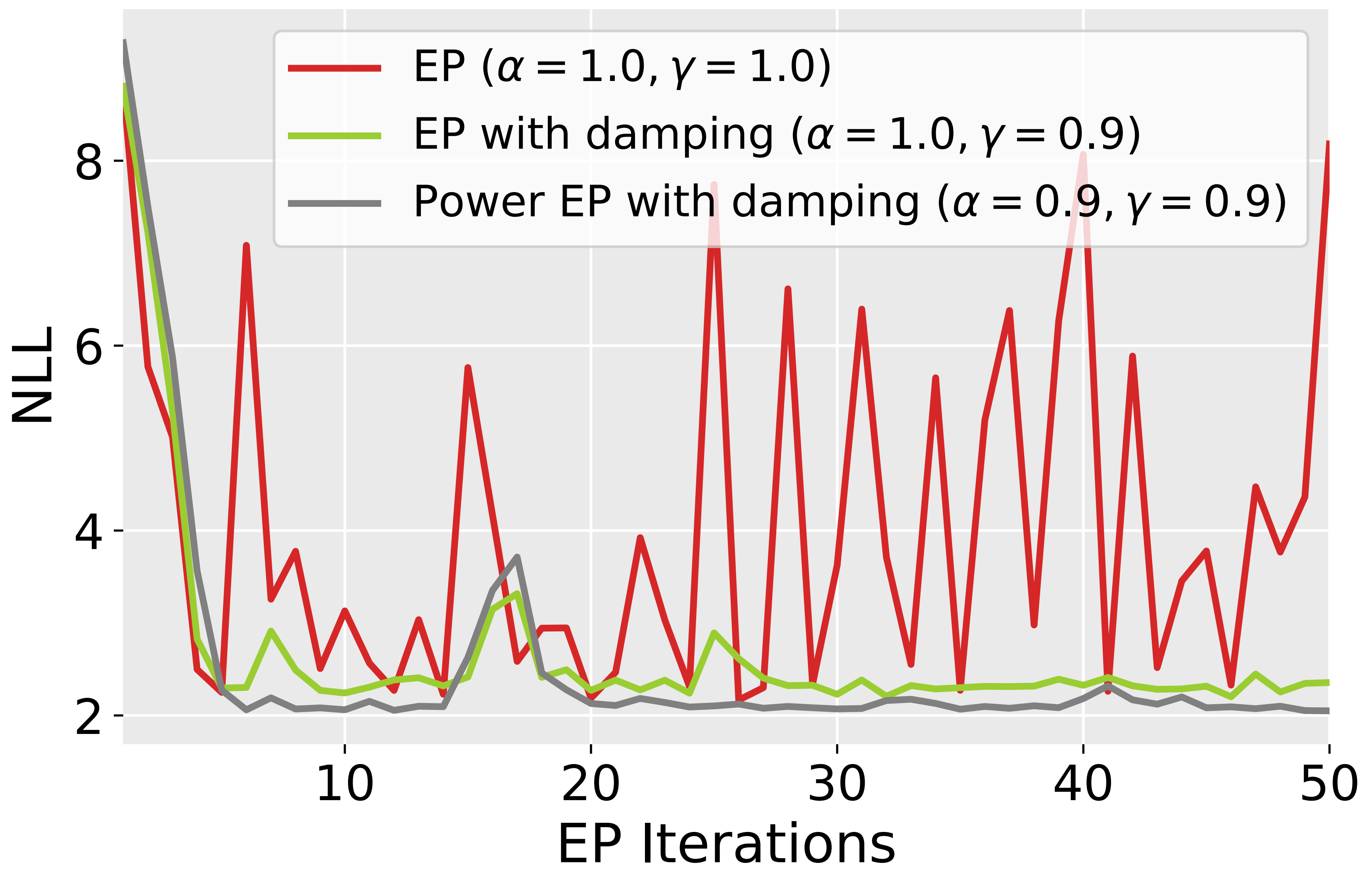
**Zhu and Rower. *Technical Report, 1995*

Results

Results



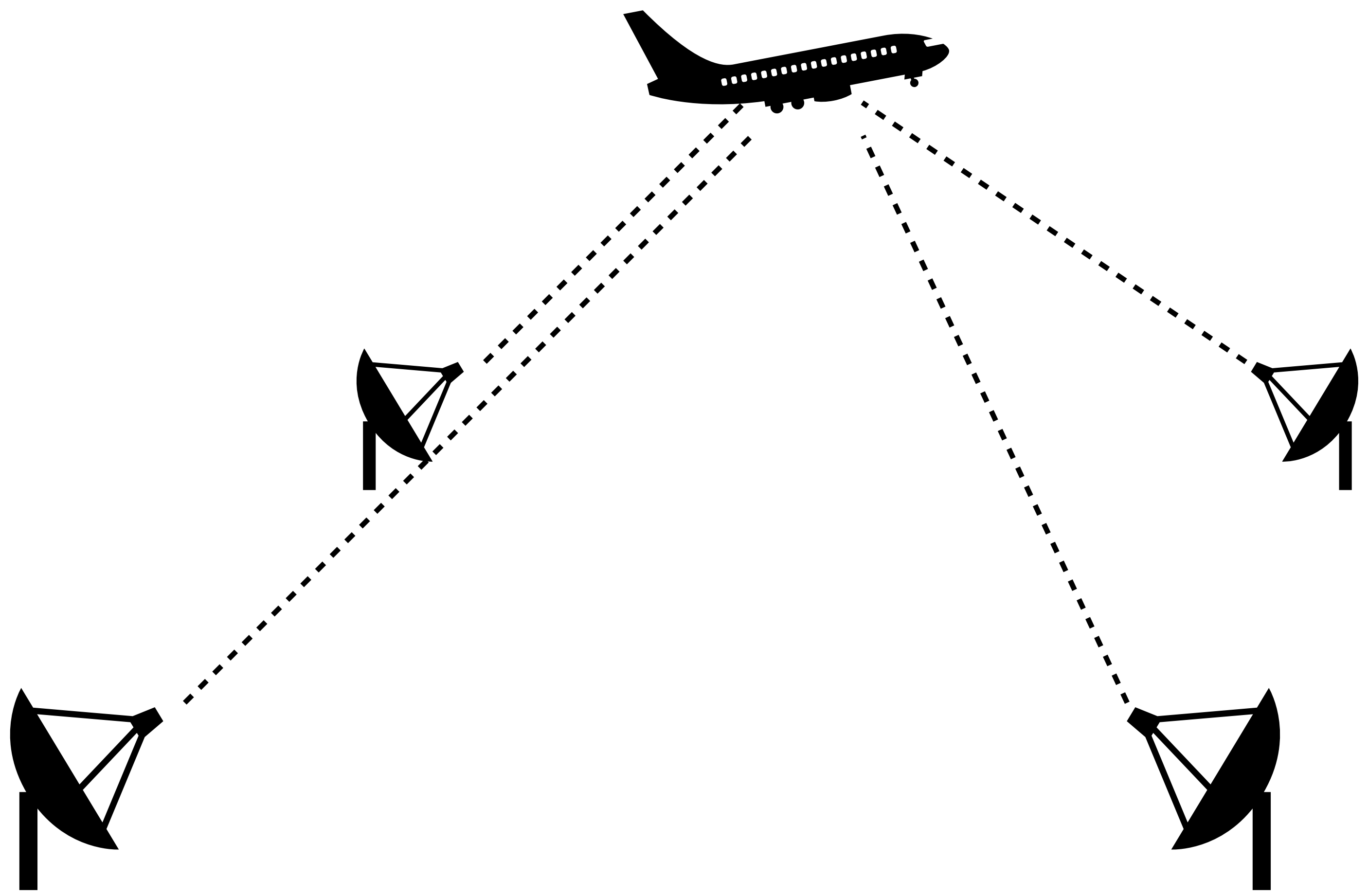
EP ($\alpha = 1.0, \gamma = 1.0$)
EP with damping ($\alpha = 1.0, \gamma = 0.9$)
Power EP with damping ($\alpha = 0.9, \gamma = 0.9$)
RMSE
8
7
6
5
4
3
10
20
30
40
50
EP Iterations



EP ($\alpha = 1.0, \gamma = 1.0$)
EP with damping ($\alpha = 1.0, \gamma = 0.9$)
Power EP with damping ($\alpha = 0.9, \gamma = 0.9$)
NLL
8
6
4
2
10
20
30
40
50
EP Iterations

Bearings-only tracking

New Kalman smoother based on the α-divergence can get better results!



	Position ($\boldsymbol{x}$)		Velocity ($\dot{\boldsymbol{x}}$)		Angular velocity (ω)	
	RMSE	MNLL	RMSE	MNLL	RMSE	MNLL
EKS	43.1 ± 5.5	11.0 ± 1.2	11.1 ± 2.4	$\mathbf{11.9 \pm 2.3}$	1.44 ± 0.35	-2.80 ± 0.27
EP	43.1 ± 5.5	11.0 ± 1.2	11.1 ± 2.4	$\mathbf{11.9 \pm 2.3}$	1.44 ± 0.35	-2.80 ± 0.27
Power EP ($\alpha = 0.8$)	36.8 ± 5.8	9.65 ± 0.84	9.82 ± 2.14	$\mathbf{11.9 \pm 3.1}$	$\mathbf{1.26 \pm 0.31}$	$\mathbf{-3.00 \pm 0.14}$
Power EP ($\alpha = 0.6$)	36.8 ± 5.9	$\mathbf{9.62 \pm 0.83}$	$\mathbf{9.80 \pm 2.13}$	12.1 ± 3.3	$\mathbf{1.26 \pm 0.33}$	-2.99 ± 0.16
Power EP ($\alpha = 0.4$)	$\mathbf{36.7 \pm 5.9}$	$\mathbf{9.61 \pm 0.83}$	$\mathbf{9.79 \pm 2.12}$	12.1 ± 3.3	$\mathbf{1.26 \pm 0.33}$	-2.99 ± 0.16
Power EP ($\alpha = 0.2$)	$\mathbf{36.7 \pm 5.9}$	$\mathbf{9.61 \pm 0.82}$	$\mathbf{9.78 \pm 2.11}$	12.2 ± 3.3	$\mathbf{1.26 \pm 0.33}$	-2.99 ± 0.16

For approximate EP / Power EP, the results are for 50 iterations

$$\begin{pmatrix} \boldsymbol{x}_{n+1} \\ \dot{\boldsymbol{x}}_{n+1} \\ \omega_{n+1} \end{pmatrix} = \mathbf{A}(\omega_n) \begin{pmatrix} \boldsymbol{x}_{n} \\ \dot{\boldsymbol{x}}_{n} \\ \omega_{n} \end{pmatrix} + \boldsymbol{\varepsilon}_n, \quad y_n^i = \arctan\left(\frac{x_n^2 - y_0^i}{x_n^1 - x_0^i} \right) + \eta_n^i$$

Lorenz ’96 (with quadratic observations)

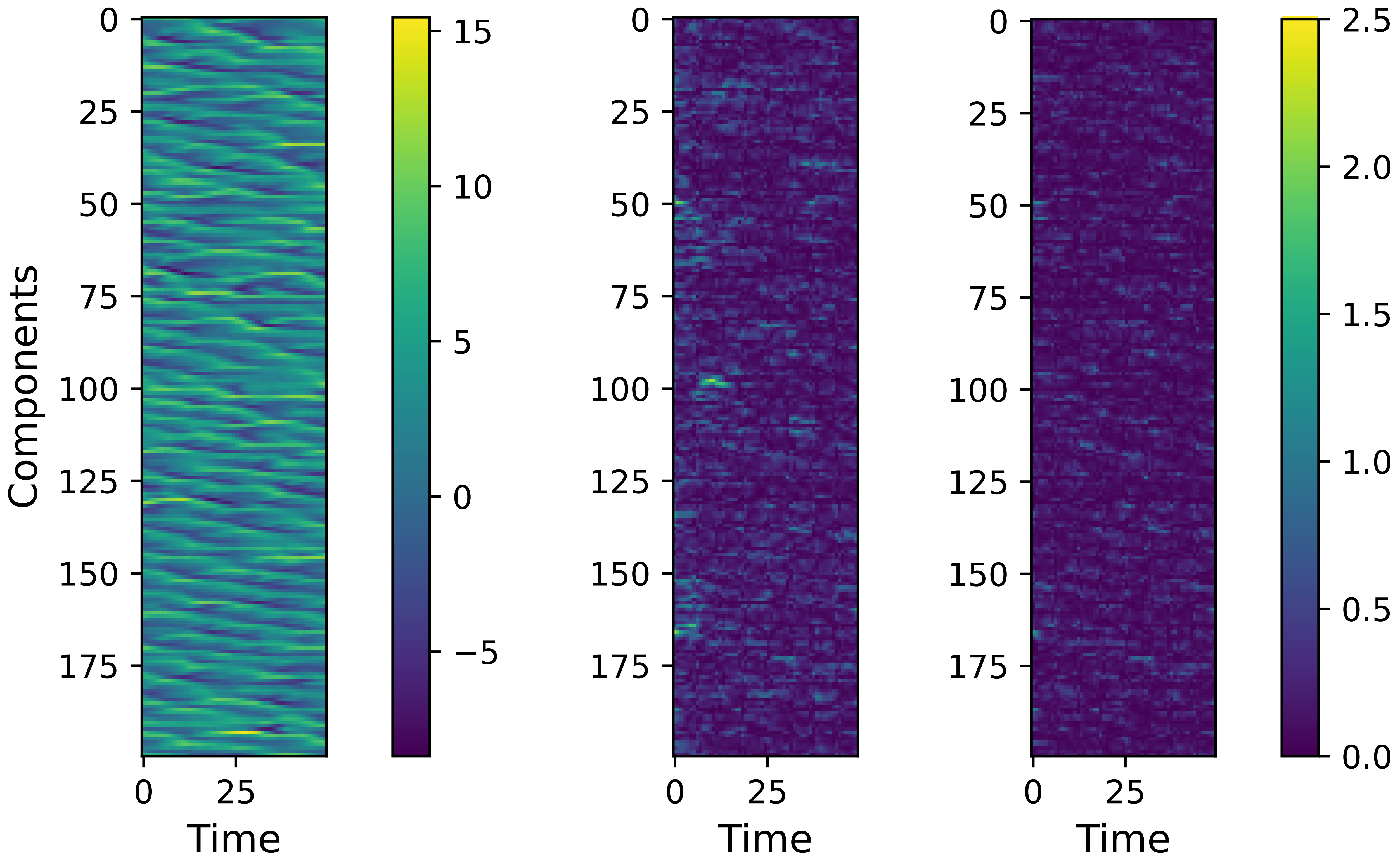
…another example where iterating can help

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \quad i = 1,\ldots,d,$$

$$y_i = x_i^2 + \eta_i$$

$d = 20$	RMSE	NLL
UKS	0.95 ± 0.09	-9.33 ± 1.19
EP	0.84 ± 0.05	-11.0 ± 0.74
Power EP ($\alpha = 0.8$)	$\mathbf{0.82} \pm \mathbf{0.04}$	$-\mathbf{11.3} \pm \mathbf{0.5}$
Power EP ($\alpha = 0.1$)	0.87 ± 0.12	-10.5 ± 2.2
$d = 200$	RMSE	NLL
UKS	4.47 ± 0.17	8.72 ± 4.13
EP	2.95 ± 0.05	-89.0 ± 1.67
Power EP ($\alpha = 0.8$)	2.80 ± 0.04	-103.0 ± 1.0
Power EP ($\alpha = 0.1$)	$\mathbf{2.61} \pm \mathbf{0.03}$	$-\mathbf{111.0} \pm \mathbf{1.0}$

Results are displayed for 10 iterations



Ground truth

UKS error

Approx. EP error (10 iters)

Summary

- We propose an approximate EP scheme for nonlinear data assimilation
- This strictly generalises nonlinear Gaussian filters/smoothers
- By storing messages, can derive iterative extensions of known smoothers
- Convergence can be aided by damping
- Also possible to develop new filters/smoothers based on α-divergence

Limitations:

- Results are mostly heuristic (e.g. no convergence guarantees)
- Does not scale to high dimensions due to the need for storing messages
- Restricted to Gaussian family. Can we go beyond this using e.g. score matching?