Neural Particle Filters

Jean-Pascal Pfister

Anna Kutschireiter

Simone Surace

Henning Sprekeler

Bernstein workshop: Neural Sampling

ETH Zürich

University of Zurich UZH


BBC

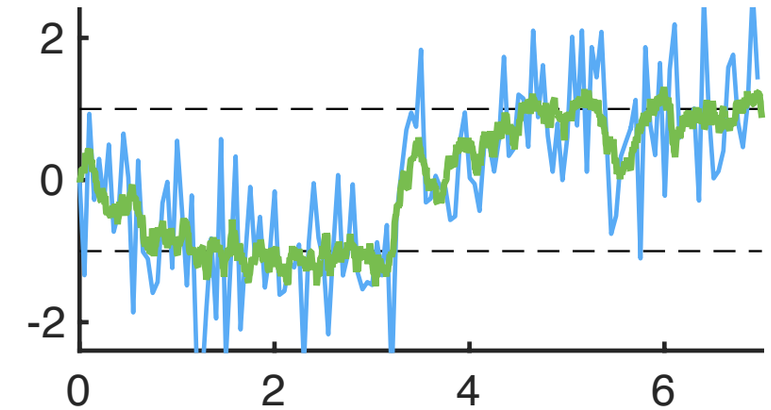
Outline

- The filtering task
  - formalisation and desiderata
- The Bootstrap Particle Filter
  - curse of dimensionality
- The Neural Particle Filter
  - inference and learning
  - Avoiding the curse of dimensionality
- The spiking Neural Particle Filter

Outline

- The filtering task
  - formalisation and desiderata
- The Bootstrap Particle Filter
  - curse of dimensionality
- The Neural Particle Filter
  - inference and learning
  - Avoiding the curse of dimensionality
- The spiking Neural Particle Filter

Mathematical formulation of the problem



Generative model

Hidden process (e.g. the position of the prey):

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)dt + \Sigma_x^{1/2} d\mathbf{w}_t$$

Observation process (e.g. visual stimulus):

$$d\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t)dt + \Sigma_y^{1/2} d\mathbf{v}_t$$

Question: $p(\mathbf{x}_t|\mathcal{Y}_t) = \ ?$

What is the posterior distribution over the hidden causes $\mathbf{x}_t$ given the past observations $\mathcal{Y}_t = \{y_s, s \leq t\}$?

Desiderata for the filter

- ➡ Can represent dynamic distributions
- ➡ Can represent arbitrary distributions
- ➡ Works with nonlinear generative models
- ➡ is scalable with dimensions
- ➡ is learnable online
- ➡ is implementable in a neural circuit
- ➡ is robust to neuron pruning

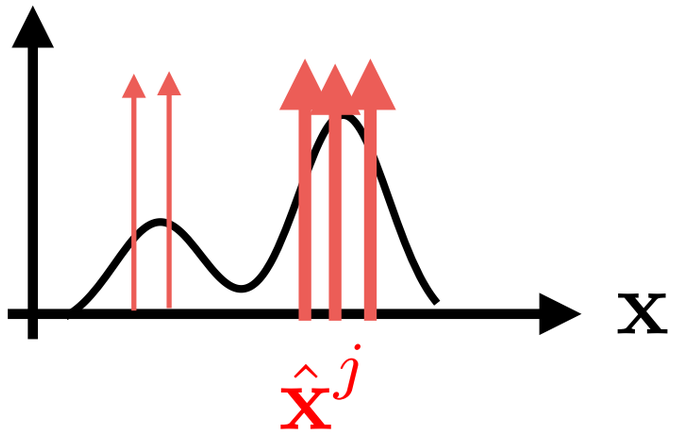
Outline

- The filtering task
  - formalisation and desiderata
- The Bootstrap Particle Filter
  - curse of dimensionality
- The Neural Particle Filter
  - inference and learning
  - Avoiding the curse of dimensionality
- The spiking Neural Particle Filter

The Bootstrap Particle Filter

Particle representation

$$p(\mathbf{x}_t|\mathbf{y}_{0\ldots t}) = \sum_{i=1}^{N} w_t^i \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t^i)$$



weight update

$$w_t^i \propto w_{t-dt}^i p(d\mathbf{y}_t|\hat{\mathbf{x}}_t^i)$$

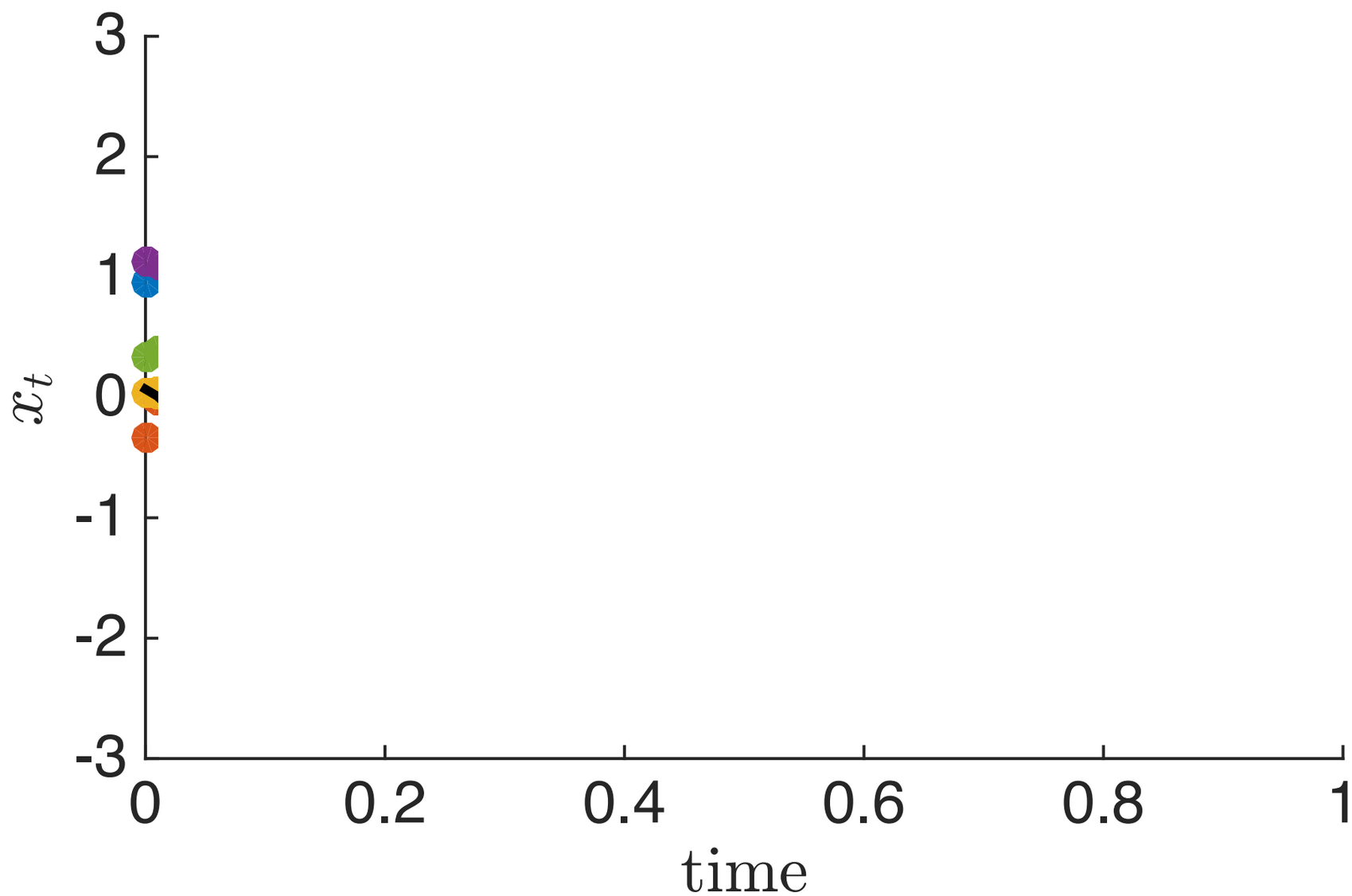
Particle dynamic

$$d\hat{\mathbf{x}}_t^i = \mathbf{f}(\hat{\mathbf{x}}_t^i)dt + \Sigma_x^{1/2} d\hat{\mathbf{w}}_t^i$$

Problem 1: Biological implementation ??

Problem 2: collapse of the # of effective particles

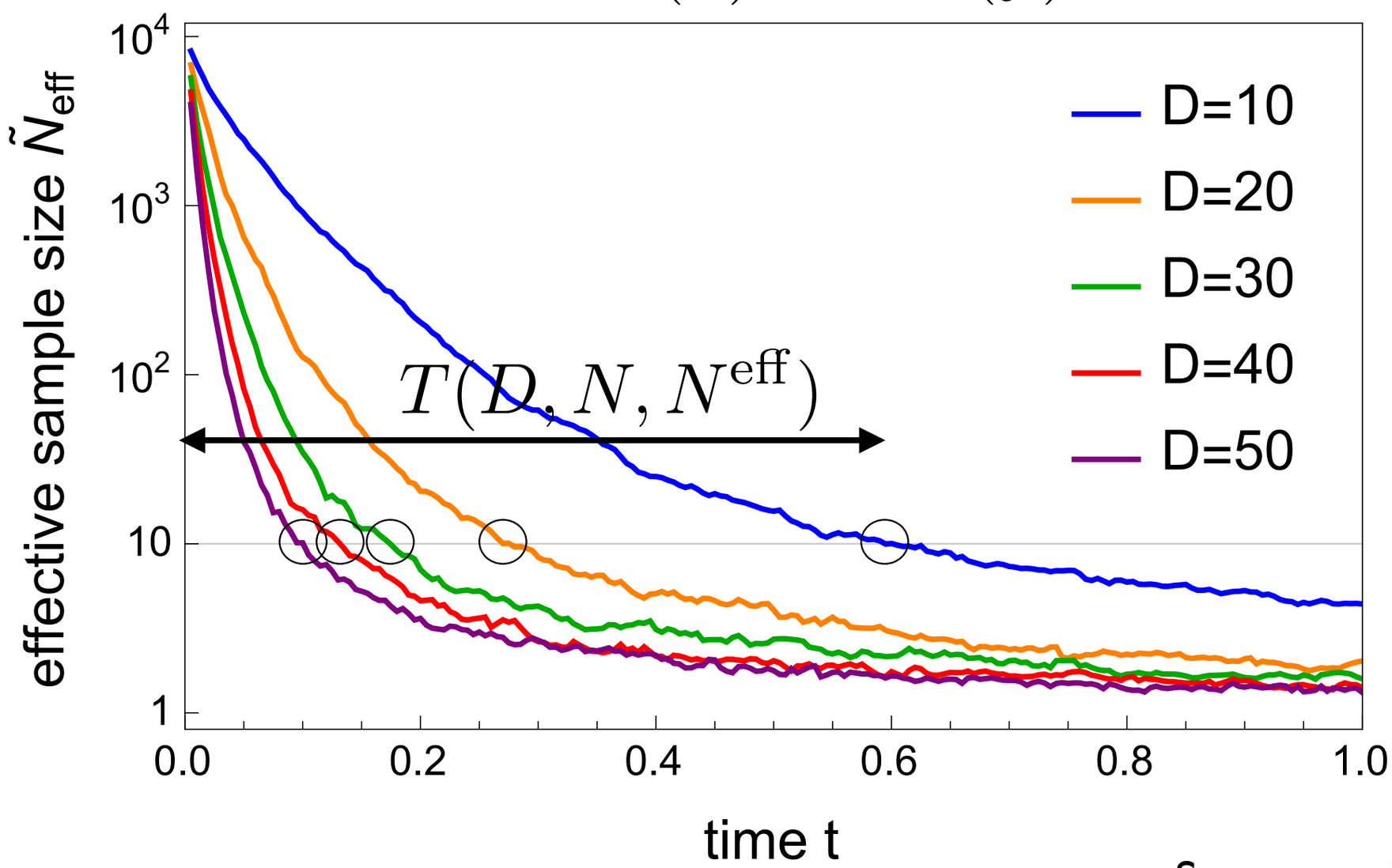
$$N_t^{\text{eff}} = \left(\sum_i \left(w_t^i \right)^2 \right)^{-1} \to 0$$



Problem 2: collapse of the # of effective particles

$$dN_t^{\text{eff}} \propto -DN_t^{\text{eff}}dt$$

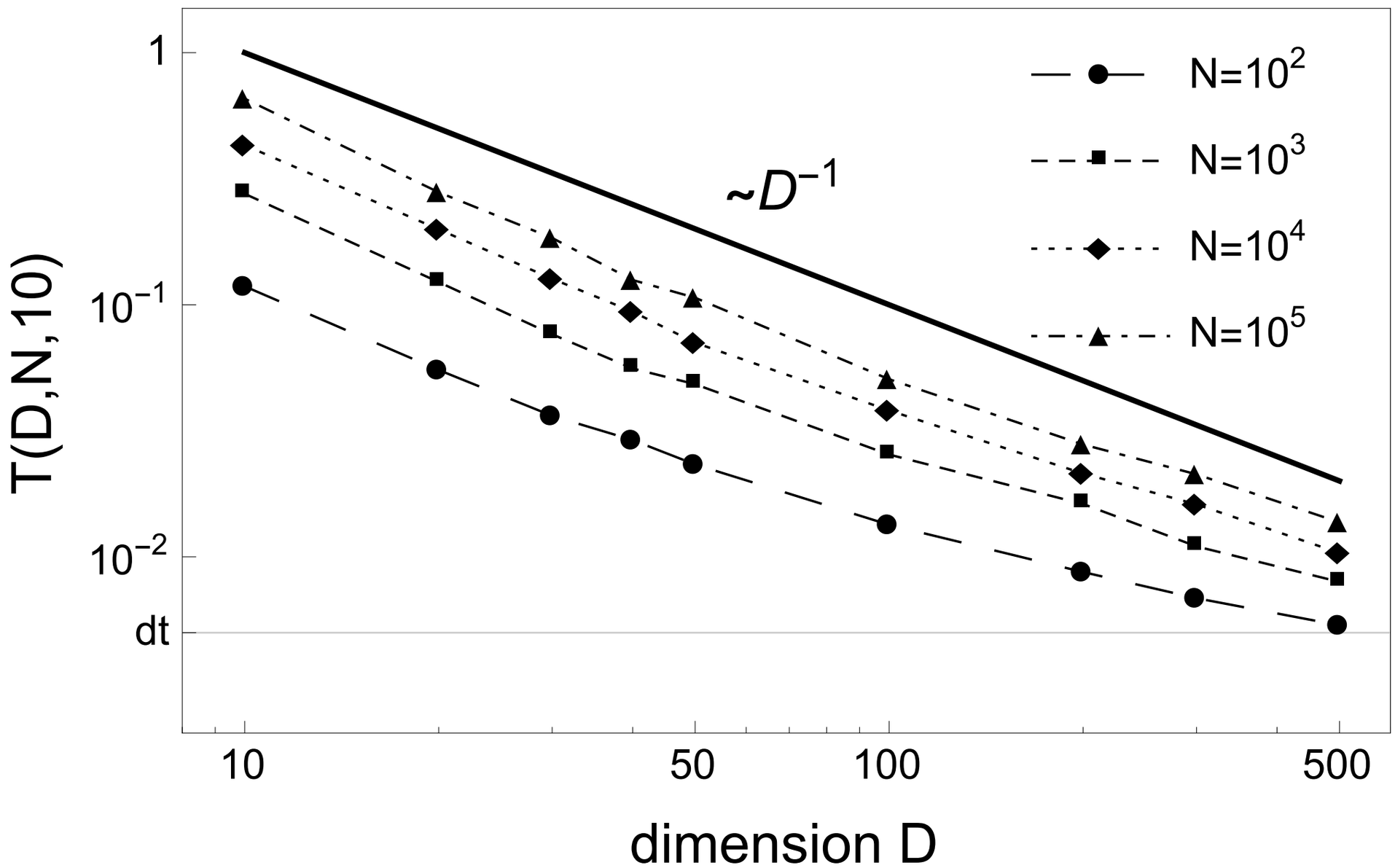
$$D = \dim(\mathbf{x}) = \dim(\mathbf{y})$$



Surace et al. *ArXiv*. 2017

Problem 2: collapse of the # of effective particles

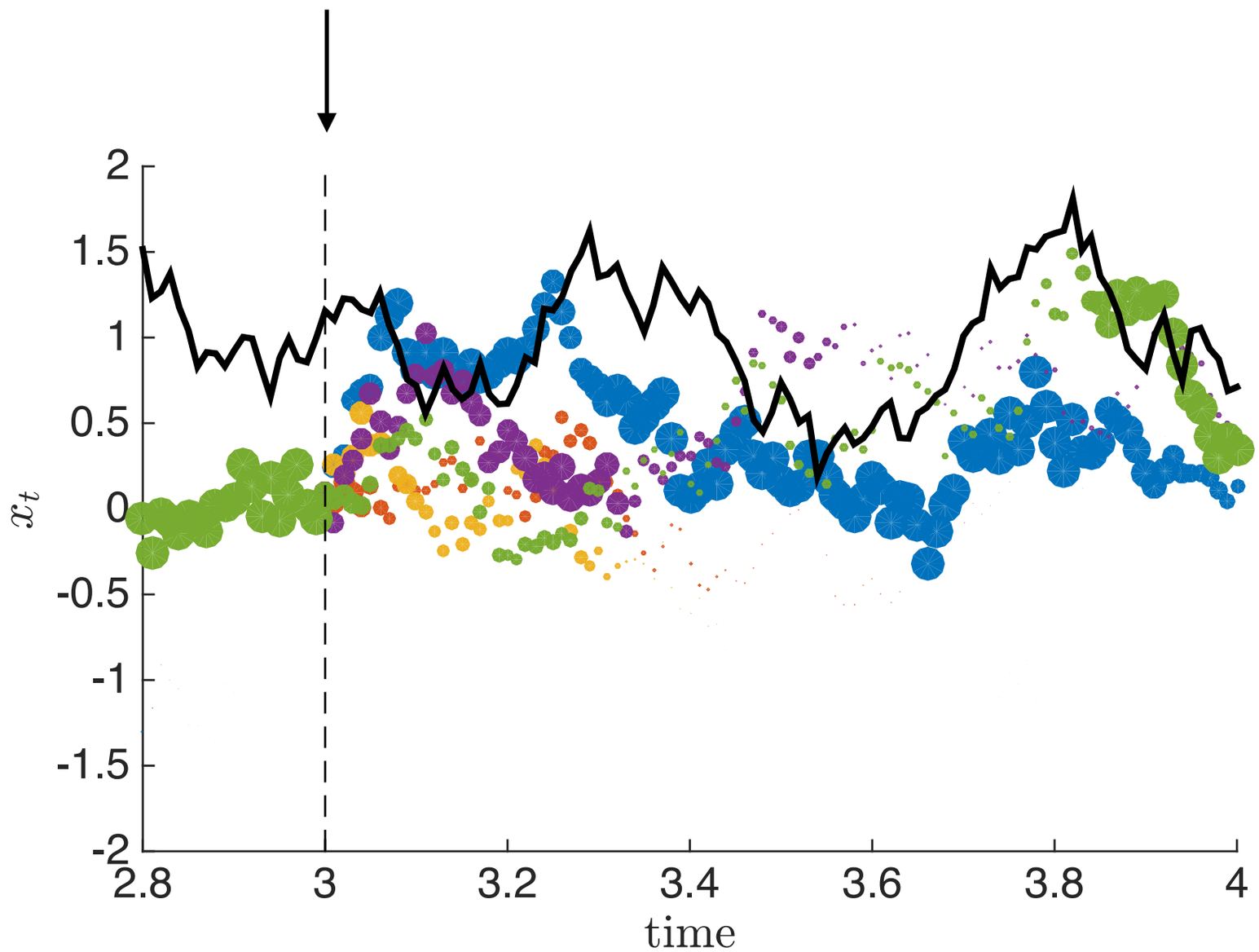
$$dN_t^{\text{eff}} \propto -DN_t^{\text{eff}}dt$$



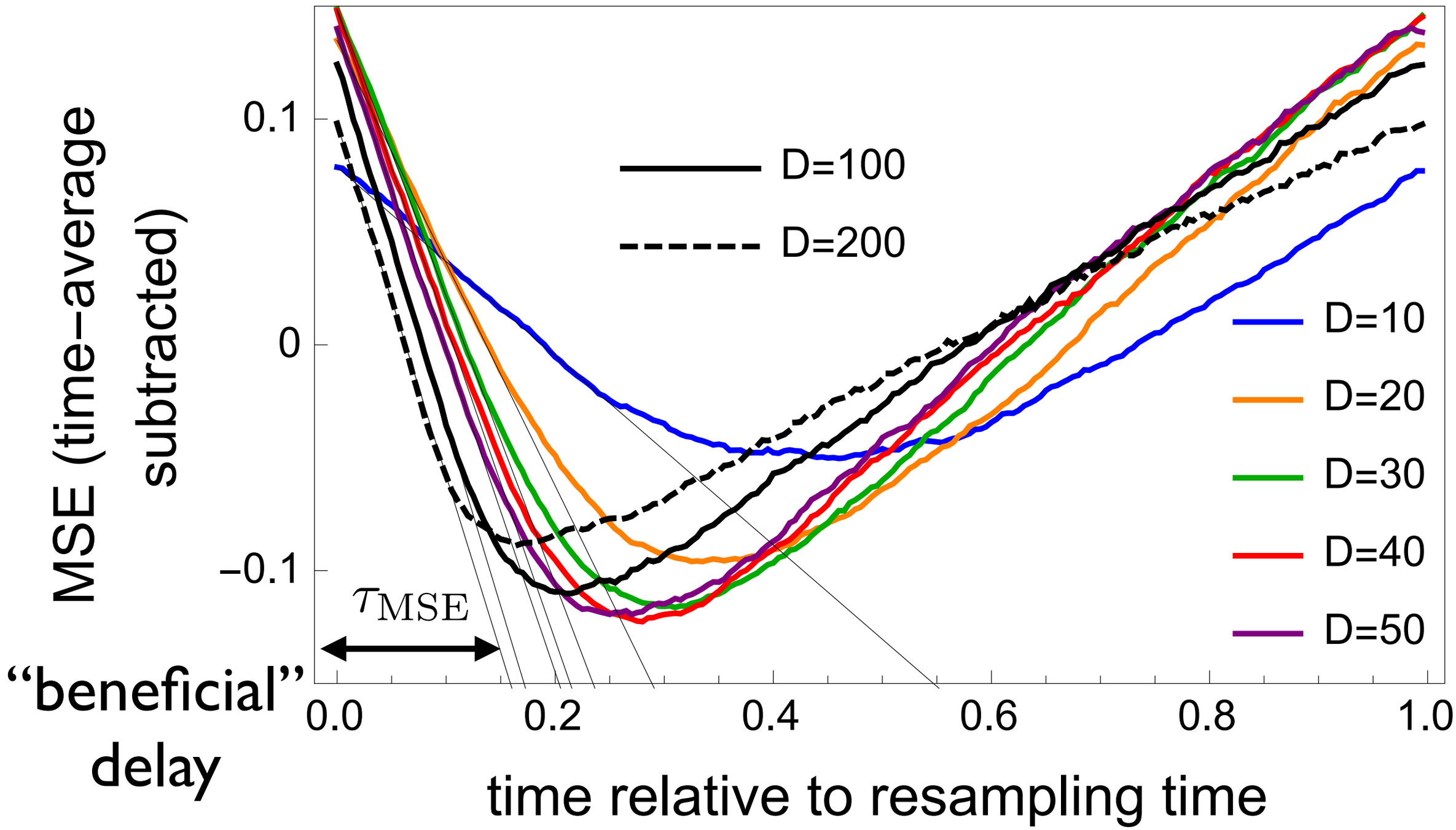
Surace et al. *ArXiv*. 2017

What about resampling ?

resampling time



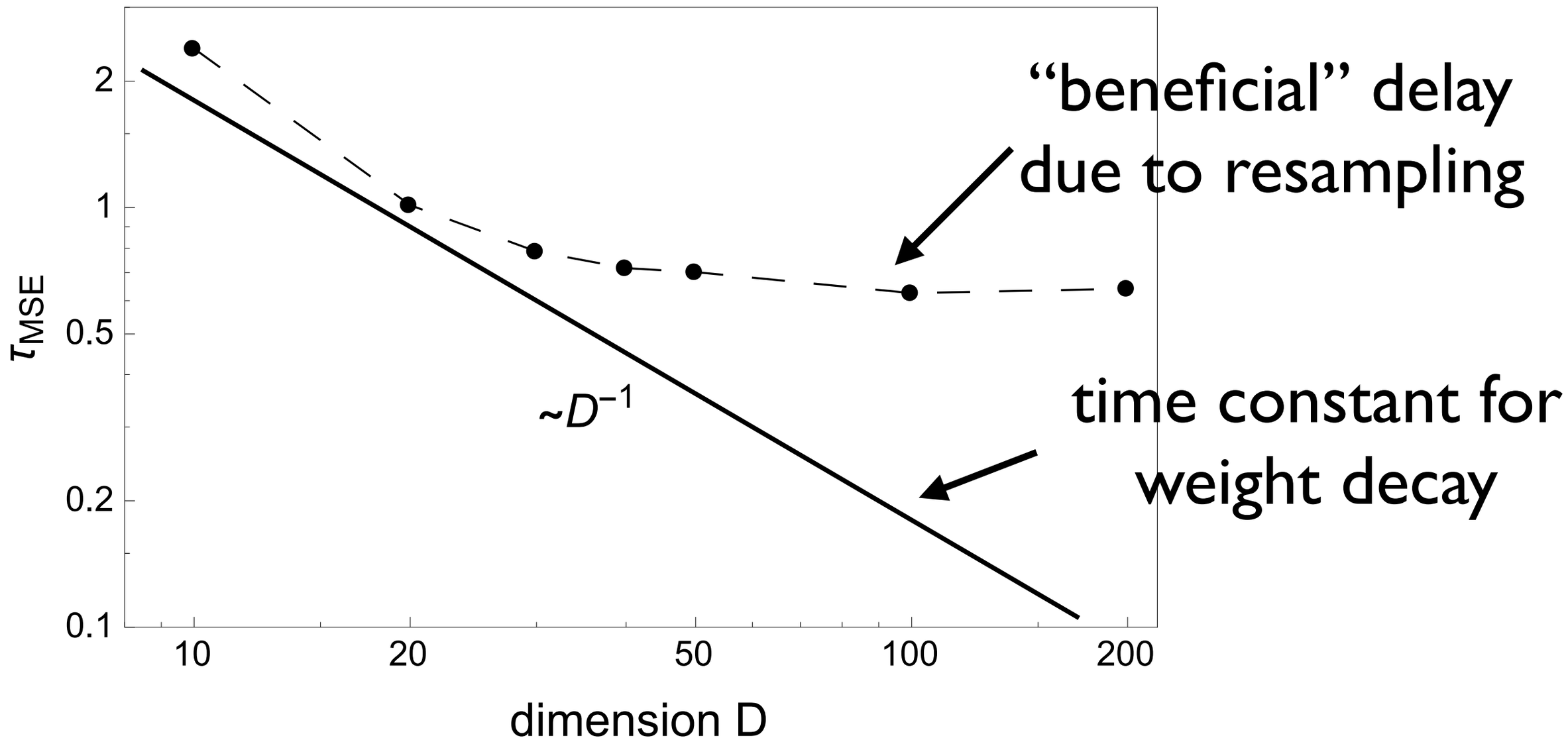
Resampling helps only after a certain delay



"beneficial" delay

Surace et al. *ArXiv*. 2017

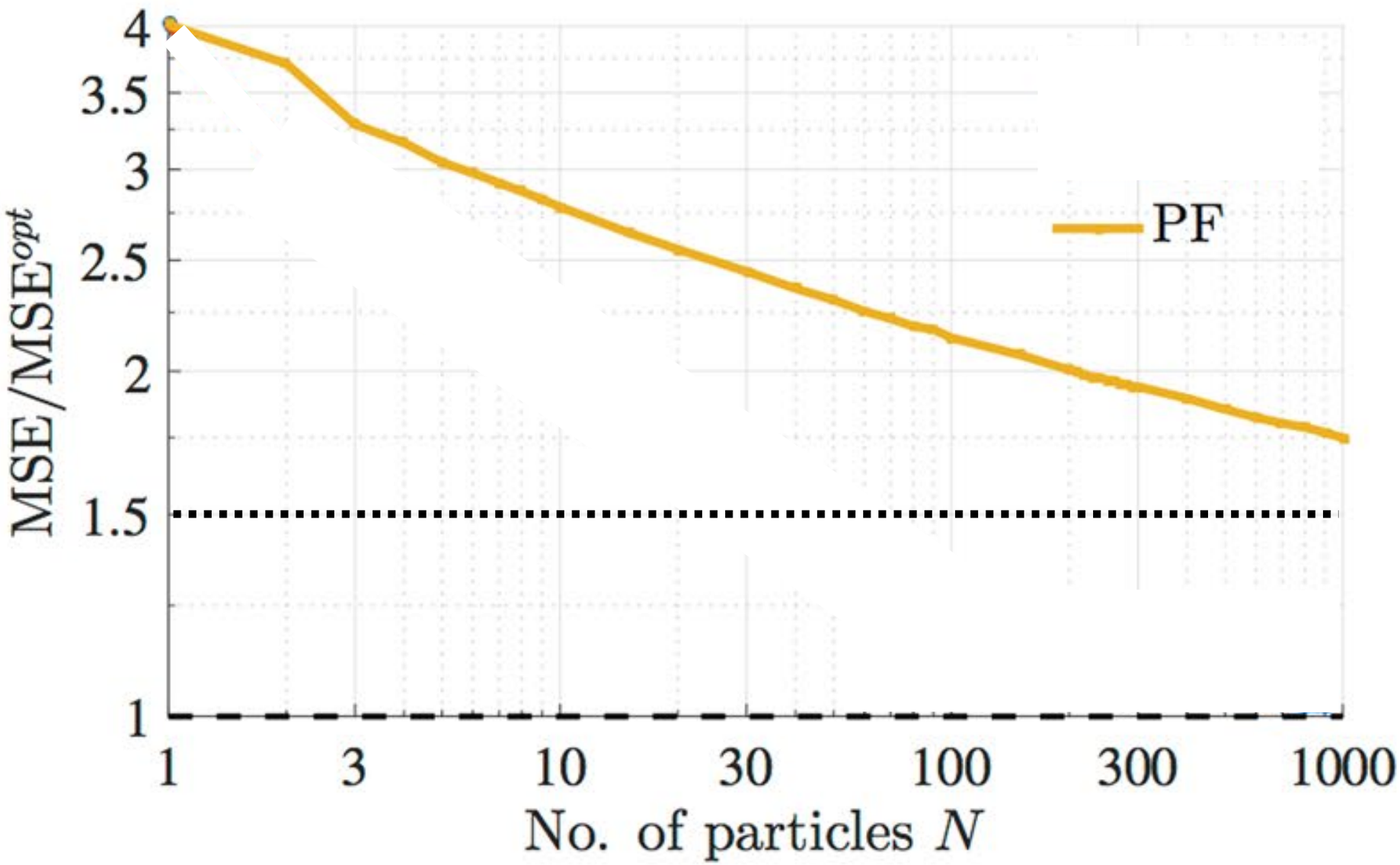
Resampling does not help



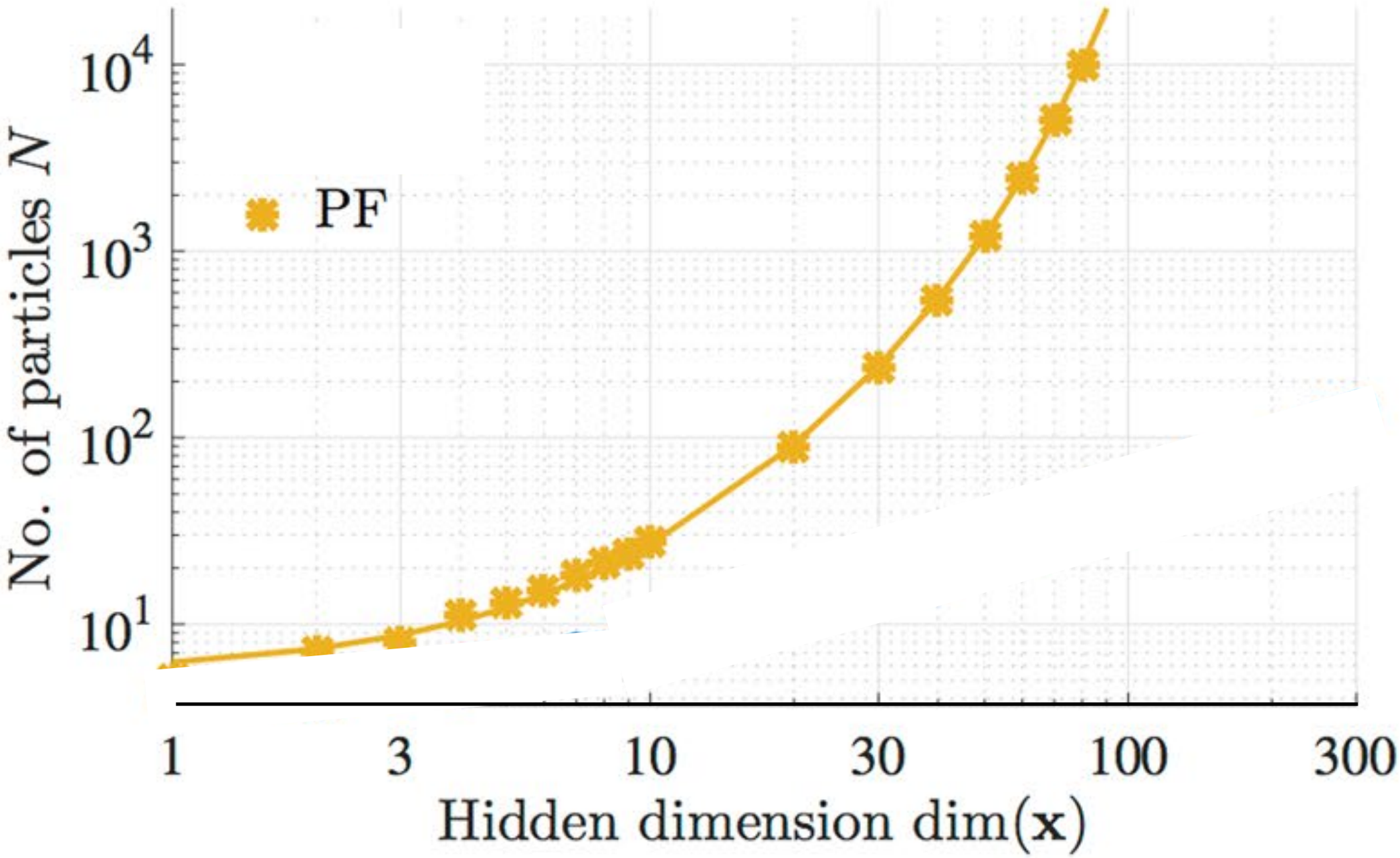
Surace et al. *ArXiv*. 2017

curse of dimensionality of the BPF

linear generative model $d = 80$



curse of dimensionality of the BPF



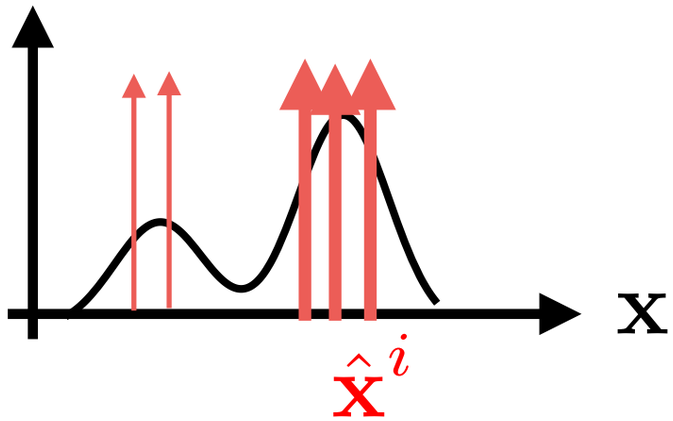
Outline

- The filtering task
  - formalisation and desiderata
- The Bootstrap Particle Filter
  - curse of dimensionality
- The Neural Particle Filter
  - inference and learning
  - Avoiding the curse of dimensionality
- The spiking Neural Particle Filter

The ~~Bootstrap~~ Neural Particle Filter

Particle representation

$$p(\mathbf{x}_t|\mathbf{y}_{0\ldots t}) = \sum_{i=1}^{N} w_t^i \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t^i)$$



weight update

$$w_t^i \propto w_{t-dt}^i p(d\mathbf{y}_t|\hat{\mathbf{x}}_t^i)$$

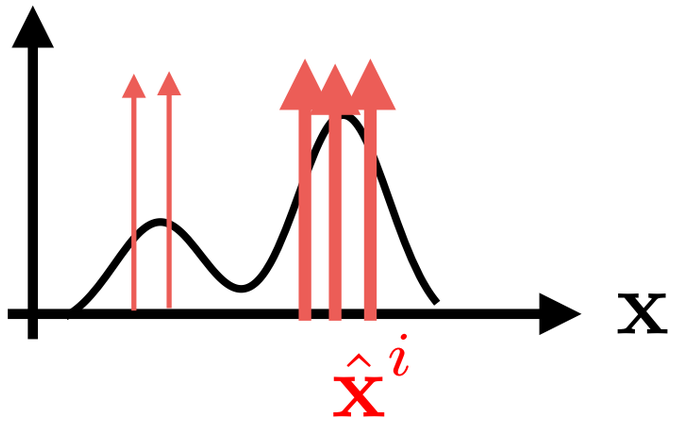
Particle dynamic

$$d\hat{\mathbf{x}}_t^i = \mathbf{f}(\hat{\mathbf{x}}_t^i)dt + \Sigma_x^{1/2} d\hat{\mathbf{w}}_t^i$$

The Neural Particle Filter

Particle representation

$$p(\mathbf{x}_t|\mathbf{y}_{0\ldots t}) = \sum_{i=1}^{N} w_t^i \delta(\mathbf{x}_t - \hat{\mathbf{x}}_t^i)$$



no weight update

$$w_t^i = \frac{1}{N}$$

Particle dynamic

$$d\hat{\mathbf{x}}_t^i = \mathbf{f}(\hat{\mathbf{x}}_t^i)dt + W_t(d\mathbf{y}_t - \mathbf{g}(\hat{\mathbf{x}}_t^i)dt) + \Sigma_x^{1/2} d\hat{\mathbf{w}}_t^i$$

"Kalman" gain: $W_t = \mathrm{cov}(\mathbf{x}_t, \mathbf{g}(\mathbf{x}_t))$

The Neural Particle Filter

Hidden process:

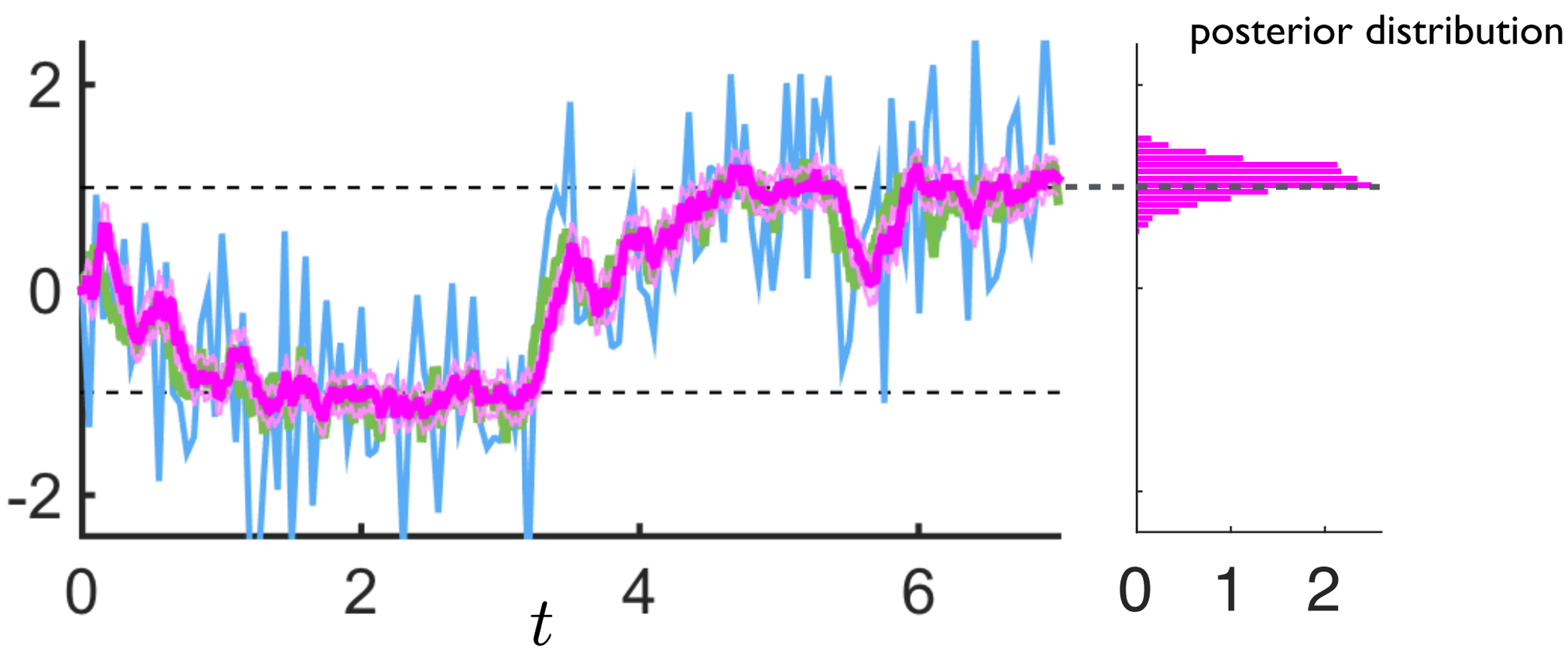
$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)dt + \Sigma_x^{1/2} d\mathbf{w}_t$$

Observation process:

$$d\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t)dt + \Sigma_y^{1/2} d\mathbf{v}_t$$

Neural Particle Filter:

$$\boxed{d\hat{\mathbf{x}}_t = \mathbf{f}(\hat{\mathbf{x}}_t)dt + W(d\mathbf{y}_t - \mathbf{g}(\hat{\mathbf{x}}_t)dt) + \Sigma_x^{1/2} d\hat{\mathbf{w}}_t}$$



Implementation of the Neural Particle Filter

Hidden process:

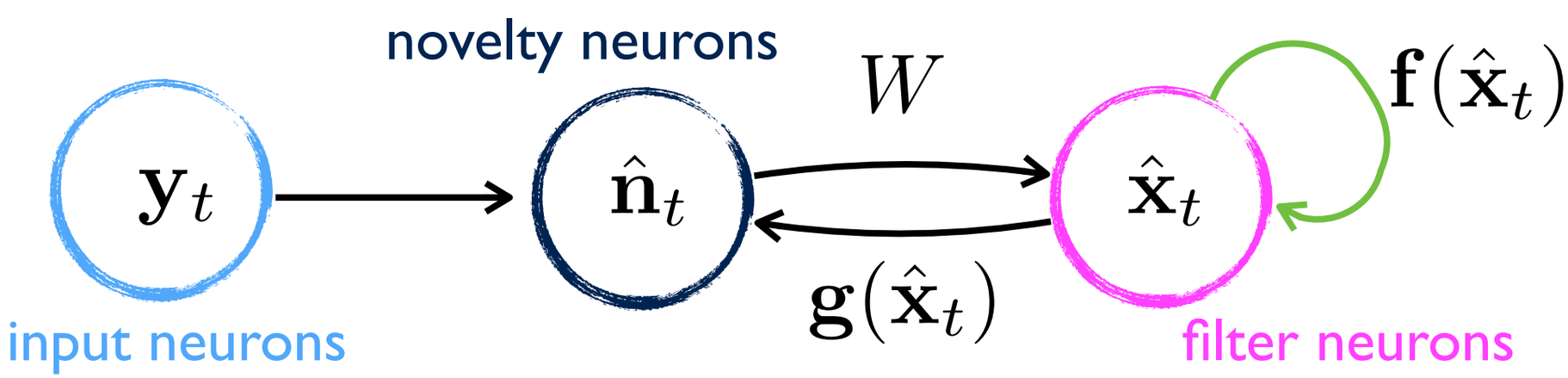
$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)dt + \Sigma_x^{1/2} d\mathbf{w}_t$$

Observation process:

$$d\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t)dt + \Sigma_y^{1/2} d\mathbf{v}_t$$

Neural Particle Filter:

$$\boxed{d\hat{\mathbf{x}}_t = \mathbf{f}(\hat{\mathbf{x}}_t)dt + \underbrace{W(d\mathbf{y}_t - \mathbf{g}(\hat{\mathbf{x}}_t)dt)}_{\mathbf{n}_t} + \Sigma_x^{1/2} d\hat{\mathbf{w}}_t}$$



Parameter learning

Innovation process: $d\mathbf{n}_t = d\mathbf{y}_t - \langle \mathbf{g}_\theta(\mathbf{x}_t) \rangle_{p_\theta(\mathbf{x_t}|\mathcal{Y}_t)} dt$

prediction error

observations

prediction

Wiener process !

$$d\mathbf{y}_t = \langle \mathbf{g}(\mathbf{x}_t) \rangle_{p_\theta(\mathbf{x}_t|\mathcal{Y}_t)} + d\mathbf{n}_t \longrightarrow \mathbb{Q}_\theta(\mathcal{Y}_t)$$

$$d\mathbf{y}_t = d\mathbf{w}_t \longrightarrow \mathbb{P}(\mathcal{Y}_t)$$

$$\Lambda_\theta(\mathcal{Y}_t) = \frac{d\mathbb{Q}_\theta(\mathcal{Y}_t)}{d\mathbb{P}(\mathcal{Y}t)} = \exp\left(\int_0^t \langle \mathbf{g}_\theta(\mathbf{x}_s) \rangle^T d\mathbf{y}_s - \frac{1}{2} \langle \mathbf{g}_\theta(\mathbf{x}_s) \rangle^T \langle \mathbf{g}_\theta(\mathbf{x}_s) \rangle ds \right)$$

Radon-Nikodym derivative

Girsanov theorem

$\langle \cdot \rangle_{p_\theta(\mathbf{x}_s|\mathcal{Y}_s)}$

Bain and Crisan, 2009 Surace and Pfister, 2017

Parameter learning

Informal: same as minimize the prediction error:

$$\dot{\theta} = \frac{\eta}{2dt} \nabla_\theta \| d\mathbf{y}_t - \langle \mathbf{g}_\theta(\mathbf{x}_t) \rangle_{p_\theta(\mathbf{x}_t | \mathcal{Y}_t)} dt \|^2$$

Online Learning rule: by gradient ascent on $d \log \Lambda_\theta(\mathcal{Y}_t)$

$$\boxed{\dot{\theta} = \eta (\nabla_\theta \langle \mathbf{g}_\theta(\mathbf{x}_t) \rangle^T)(d\mathbf{y}_t - \langle \mathbf{g}_\theta(\mathbf{x}_t) \rangle dt)}$$

$$\Lambda_\theta(\mathcal{Y}_t) = \frac{d\mathbb{Q}_\theta(\mathcal{Y}_t)}{d\mathbb{P}(\mathcal{Y}t)} \quad = \quad \exp\left(\int_0^t \langle \mathbf{g}_\theta(\mathbf{x}_s) \rangle^T d\mathbf{y}_s - \frac{1}{2} \langle \mathbf{g}_\theta(\mathbf{x}_s) \rangle^T \langle \mathbf{g}_\theta(\mathbf{x}_s) \rangle ds \right)$$

Surace and Pfister, 2017

Dynamic multi-sensory perception

Position of the prey:

$$dx_t = 3x_t\left(1 - x_t^2\right) dt + d\omega_t,$$

Visual input:

$$dv_t = x_t\, dt + \sigma_v d\beta_t,$$

Auditory input:

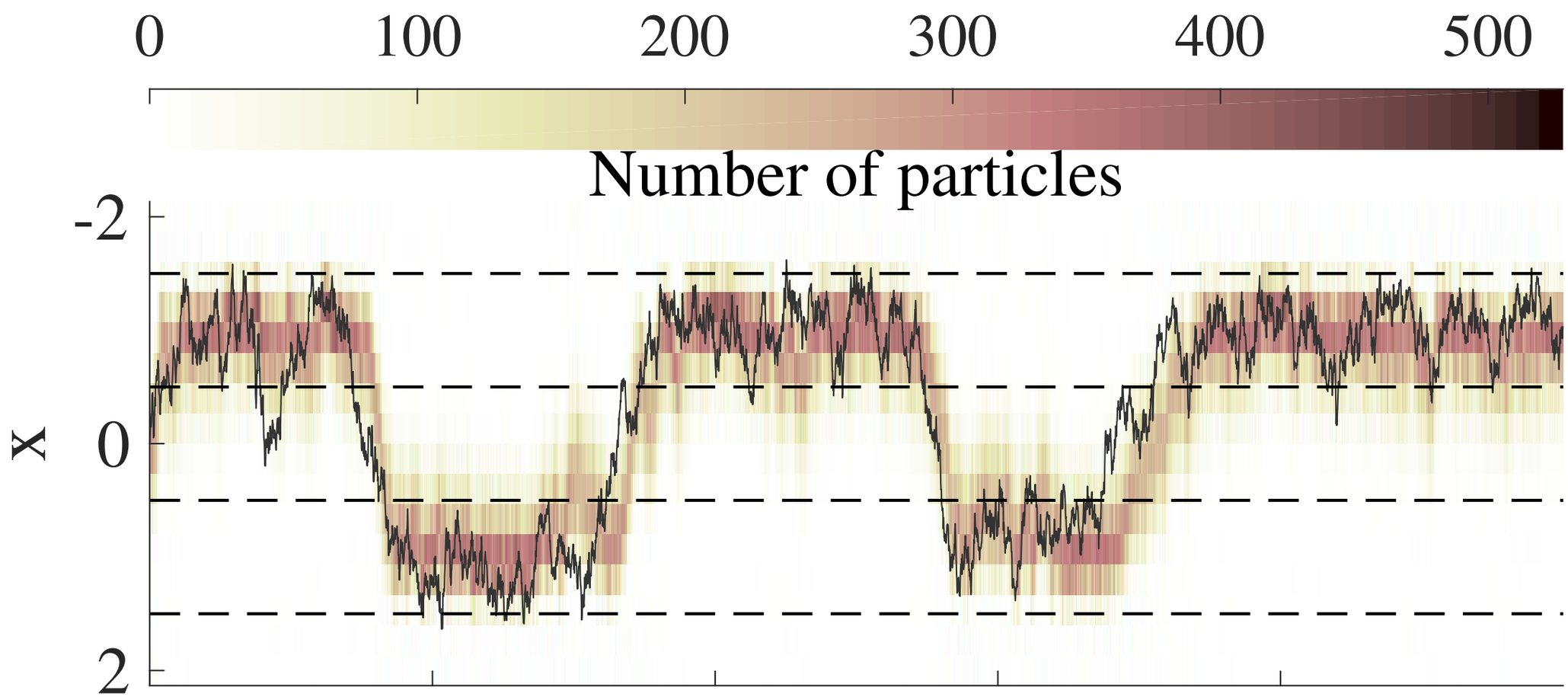
$$da_t = \tanh(2x_t)\, dt + \sigma_a d\gamma_t$$

Neural Particle Filter:

$$d\hat{x}_t = 3\hat{x}_t(1 - \hat{x}_t^2)dt + d\omega_t$$

$$+W_t^{(v)}(dv_t - \hat{x}_t dt) + W_t^{(a)}(da_t - \tanh(2\hat{x}_t)dt)$$

Position estimation

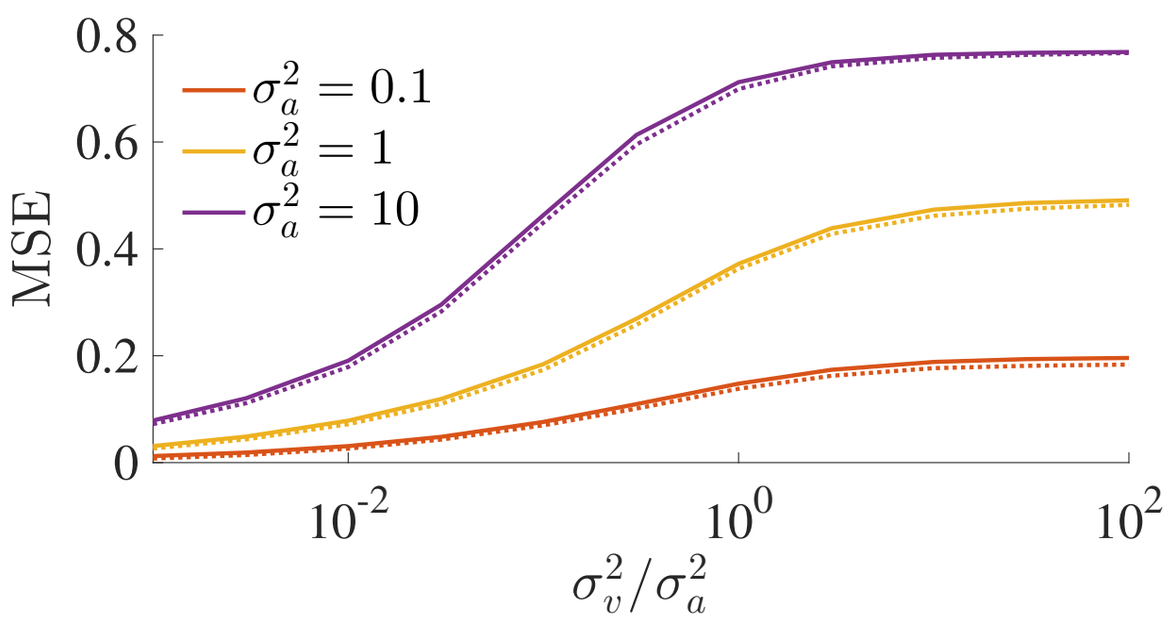


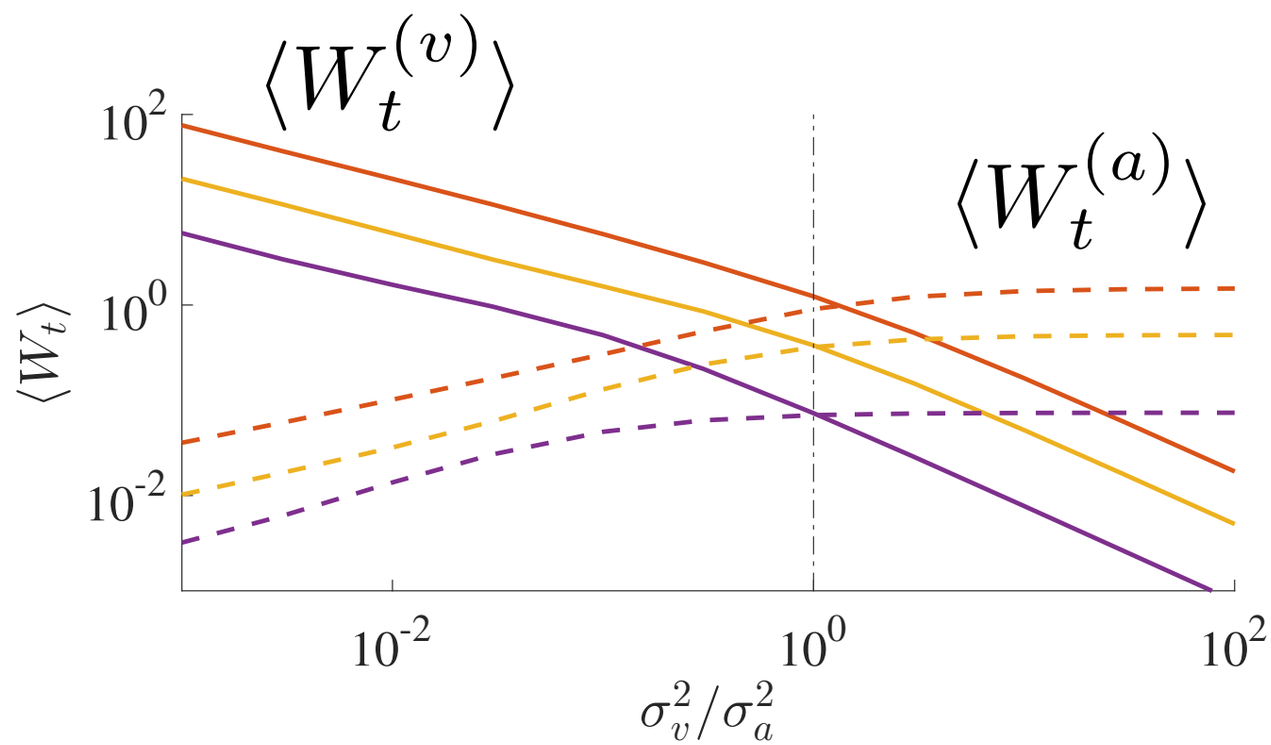
Kutschireiter et al. *Nat. Sc. Rep.* 2017

Performance of the Neural Particle Filter

— Neural Particle Filter

··· Particle Filter





Kutschireiter et al. *Nat. Sc. Rep.* 2017

ML learning vs Hebbian learning

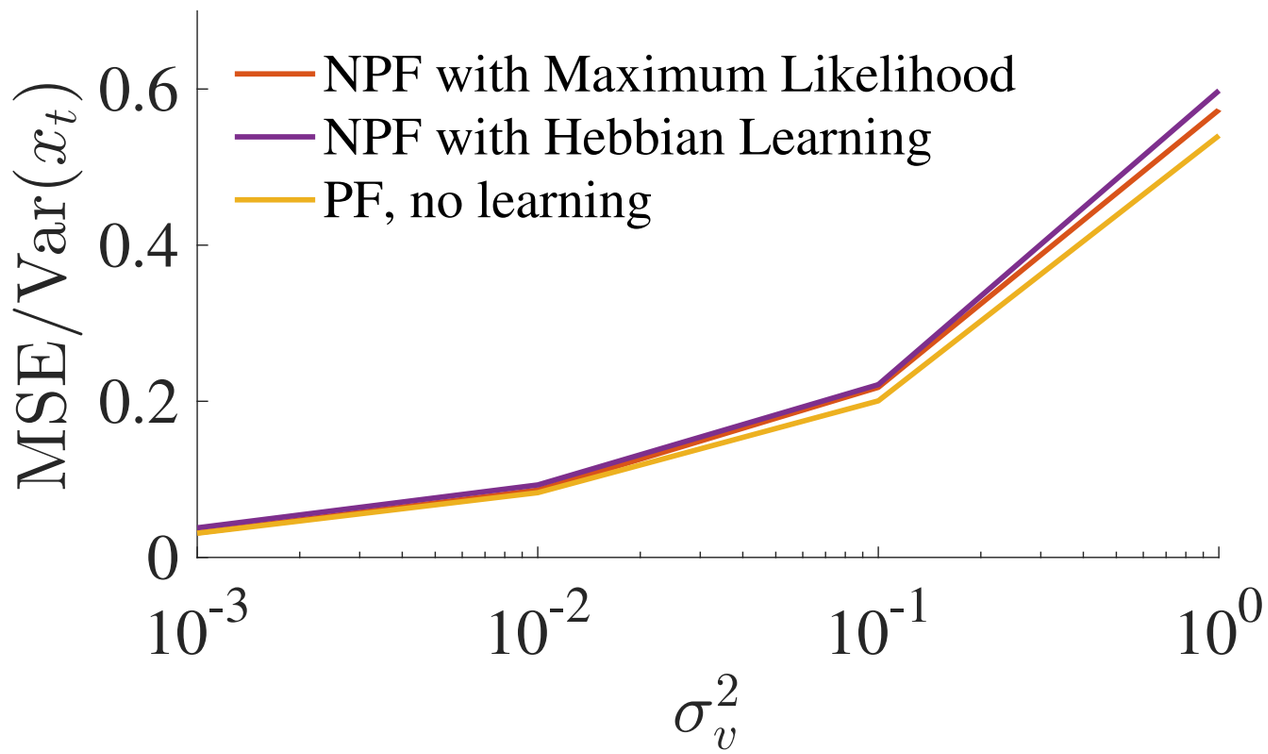
$$\dot{\theta} = \eta(\nabla_\theta \langle \mathbf{g}_\theta(\mathbf{x_t})\rangle^T)(d\mathbf{y}_t - \langle \mathbf{g}_\theta(\mathbf{x}_t)\rangle dt)$$

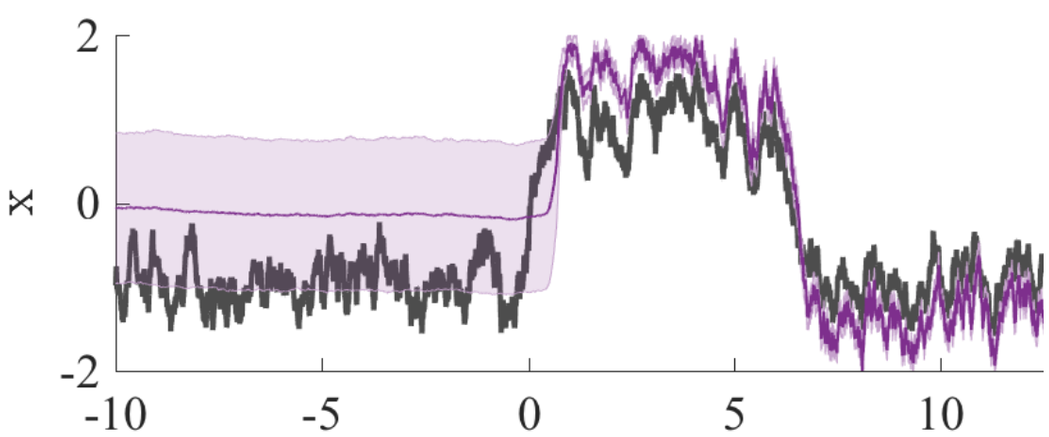
$$\mathbf{g}_\theta(\mathbf{x}) = J\mathbf{x}$$

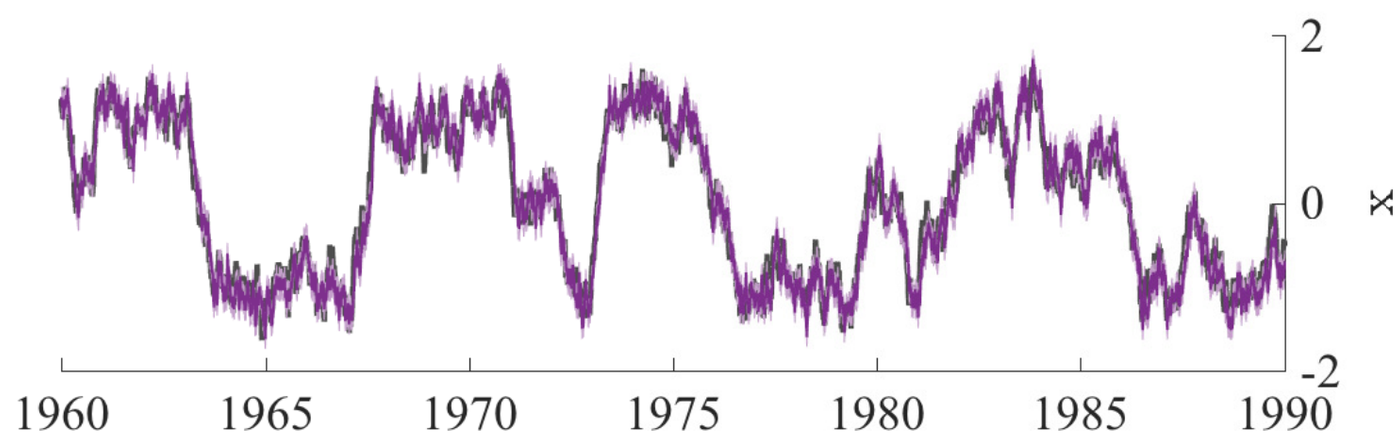
small Σ_y

$$\dot{J} \approx \eta(d\mathbf{y}_t - J\mathbf{x}_t dt)\mathbf{x}_t^T$$

Hebbian learning rule

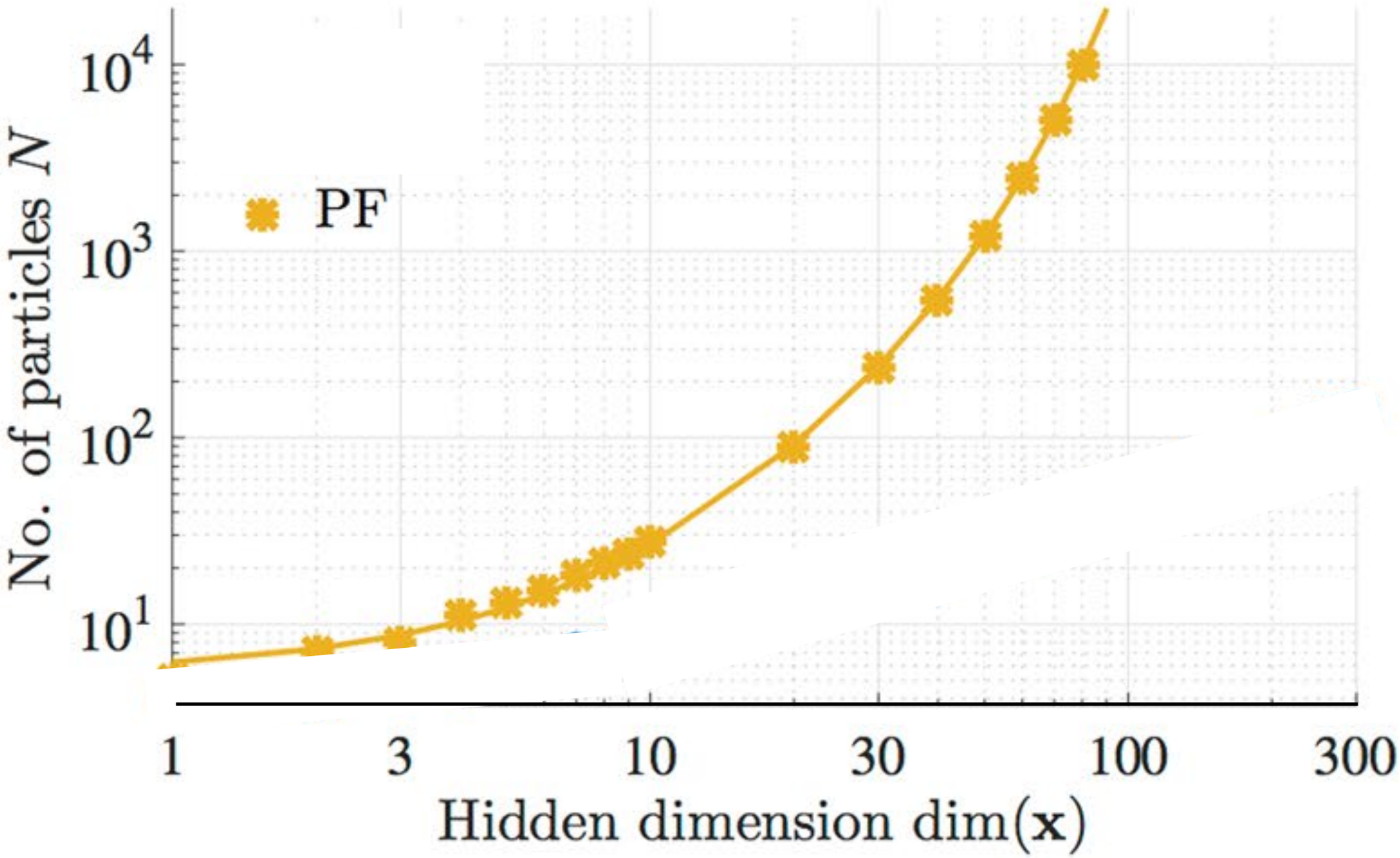






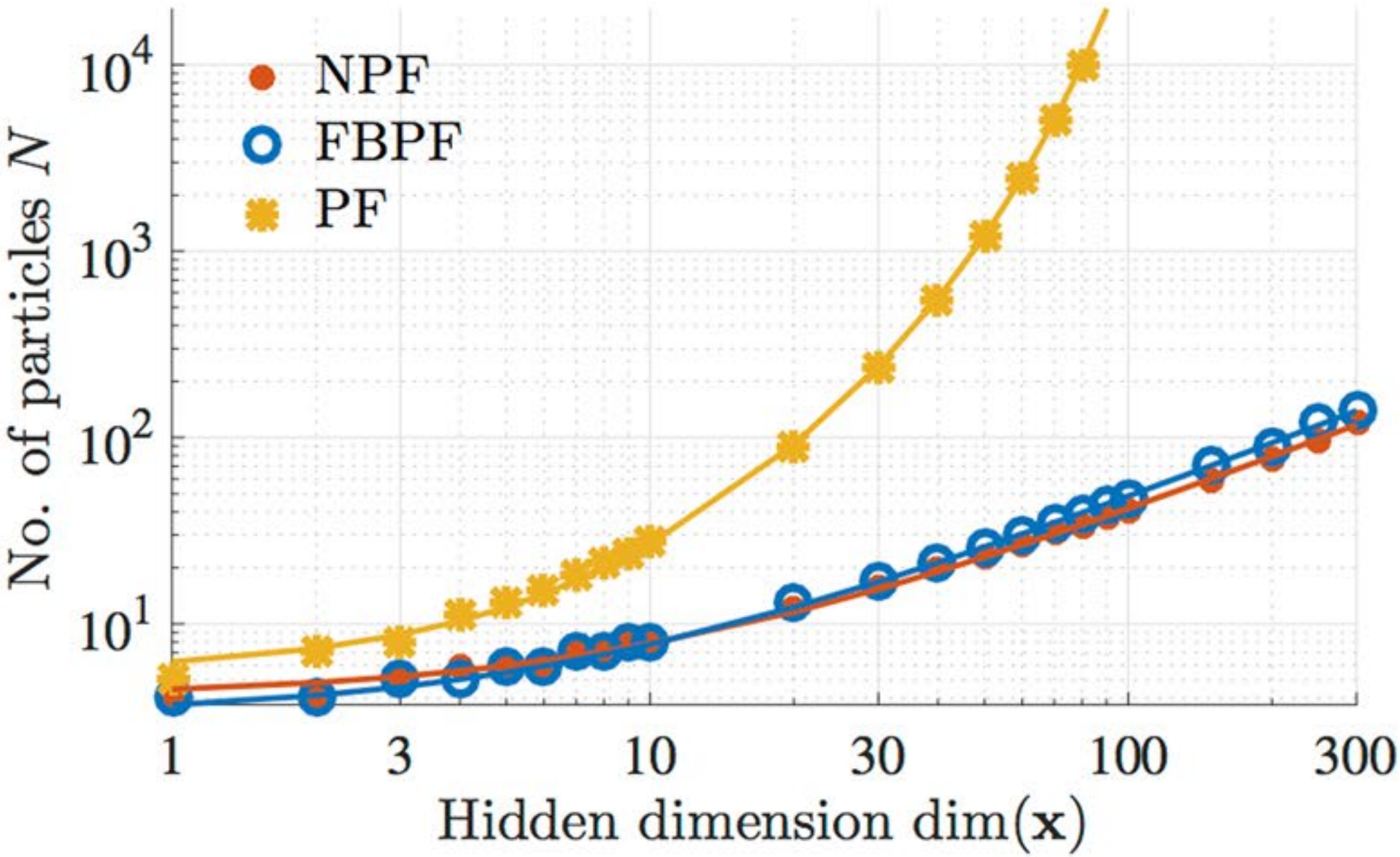
Kutschireiter et al. *Nat. Sc. Rep.* 2017

NPF avoids the curse of dimensionality



Kutschireiter et al. *Nat. Sc. Rep.* 2017

NPF avoids the curse of dimensionality



Kutschireiter et al. *Nat. Sc. Rep.* 2017

Outline

- The filtering task
  - formalisation and desiderata
- The Bootstrap Particle Filter
  - curse of dimensionality
- The Neural Particle Filter
  - inference and learning
  - Avoiding the curse of dimensionality
- The spiking Neural Particle Filter

Spiking Neural Particle Filter (sNPF)

Generative model

Hidden process:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t)dt + \Sigma_x^{1/2} d\mathbf{w}_t$$

Observation process:

$$d\mathbf{N} \sim \mathrm{Poisson}(\mathbf{g}(\mathbf{x})dt)$$

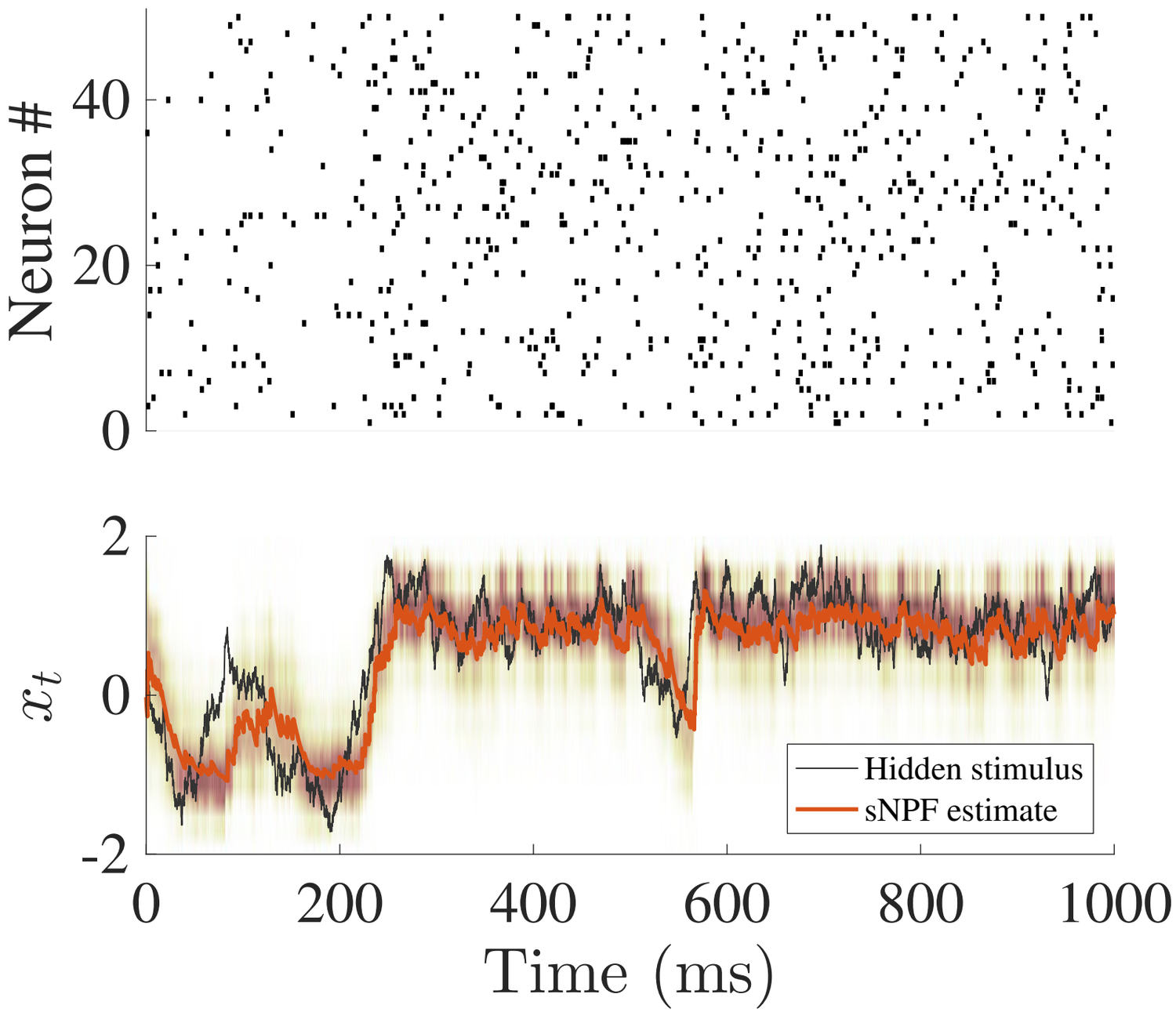
Spiking Neural Particle Filter

$$d\hat{\mathbf{x}}_t = \mathbf{f}(\hat{\mathbf{x}}_t)dt + W(d\mathbf{N}_t - \mathbf{g}(\hat{\mathbf{x}}_t)dt) + \Sigma_x^{1/2} d\hat{\mathbf{w}}_t$$

"Kalman" gain: $$W_t = \frac{\mathrm{cov}(\mathbf{g}(\mathbf{x}), \mathbf{x})}{\langle \mathbf{g}(\mathbf{x}) \rangle}$$

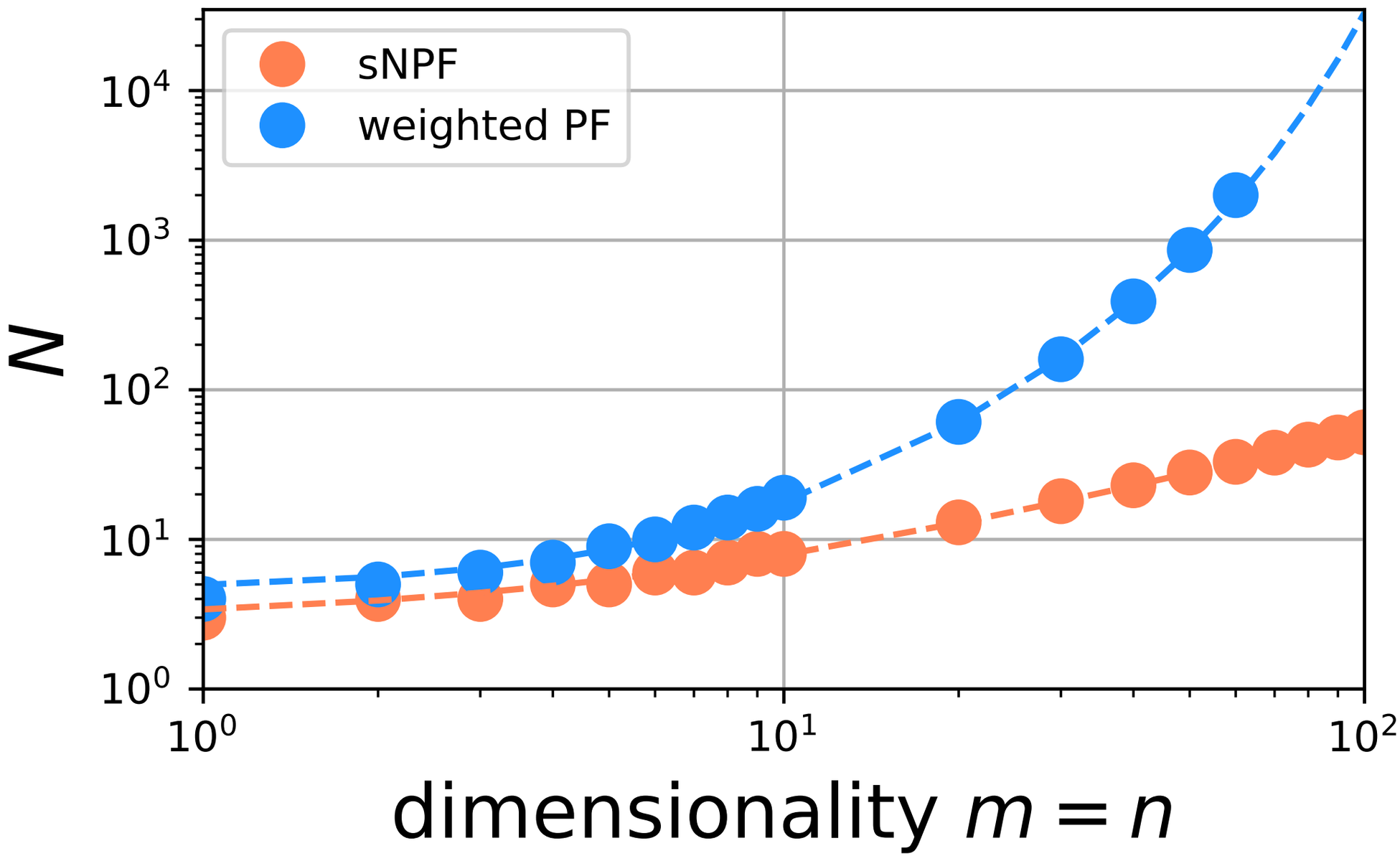
Kutschireiter et al. *Bernstein conference poster*

The sNPF nicely tracks the stimulus



The Spiking Neural Particle Filter avoids the COD

Number of particles required to reach MSE = 0.8 var(prior)



Summary: comparison of filters

	KF	BPF	NPF
➡ Can represent dynamic distributions	✓	✓	✓
➡ Can represent arbitrary distributions	✗	✓	✓
➡ Works with nonlinear gen. models	✗	✓	✓
➡ is scalable with dimensions	✓	✗	✓
➡ is learnable online	✓	✓	✓
➡ is implementable in a neural circuit	?	✗	(✓)
➡ is robust to neuron pruning	✗	✓	✓

Acknowledgements and bibliography

Anna Kutschireiter

Surace, Kutschireiter and Pfister
How to avoid the curse of dimensionality: scalability of particle filters with and without importance weights,
ArXiv, 2017

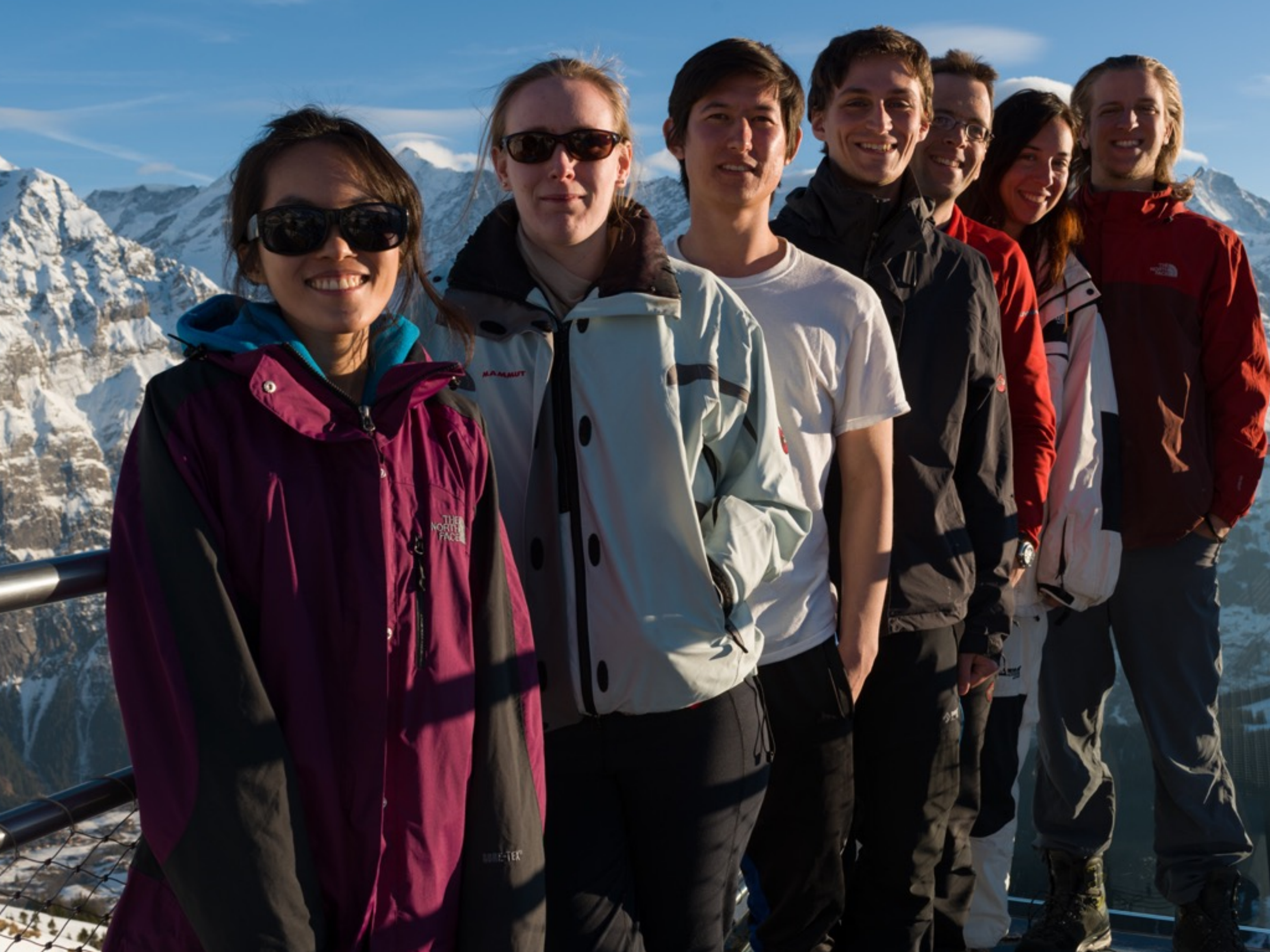
Henning Sprekeler

Kutschireiter, Surace, Sprekeler and Pfister,
Nonlinear Bayesian filtering and learning: a neuronal dynamics for perception.
Nature Scientific Report, 7: 8722, 2017

Simone Surace

Surace and Pfister,
Online Maximum Likelihood Estimation of the Parameters of Partially Observed Diffusion Processes,
NIPS Times series workshop, 2016.

FNSNF
Swiss National Science Foundation


THE NORTH FACE
MAMMUT