Batch simulations and uncertainty quantification in Gaussian process surrogate-based ABC

Marko Järvenpää
Aki Vehtari Pekka Marttinen

Helsinki Institute for Information Technology HIIT, Department of Computer Science, Aalto University, Espoo, Finland

AABI 2019, December 2019

The research problem & our contributions

- **Approximate Bayesian Computation (ABC)** is used for Bayesian inference when likelihood function intractable but forward simulations feasible
- We consider the case where **simulations are expensive**
- **Our contributions:** We reformulate earlier Gaussian process surrogate-based ABC methods[1][2] into a coherent '***Bayesian* ABC' framework and provide
  - an approximate method for **uncertainty quantification** of the ABC posterior moments and marginals due to the limited simulations
  - principled **batch acquisition functions** based on Bayesian decision theory
  - some new insights on connections between ***Bayesian* ABC**, ***Bayesian* quadrature** and ***Bayesian* optimisation**

[1] M. U. Gutmann and J. Corander. "Bayesian optimization for likelihood-free inference of simulator-based statistical models". In: *Journal of Machine Learning Research* 17.125 (2016), pp. 1–47.

[2] M. Järvenpää et al. "Efficient Acquisition Rules for Model-Based Approximate Bayesian Computation". In: *Bayesian Analysis* 14.2 (2019), pp. 595–622.

Some results with batch acquisition functions

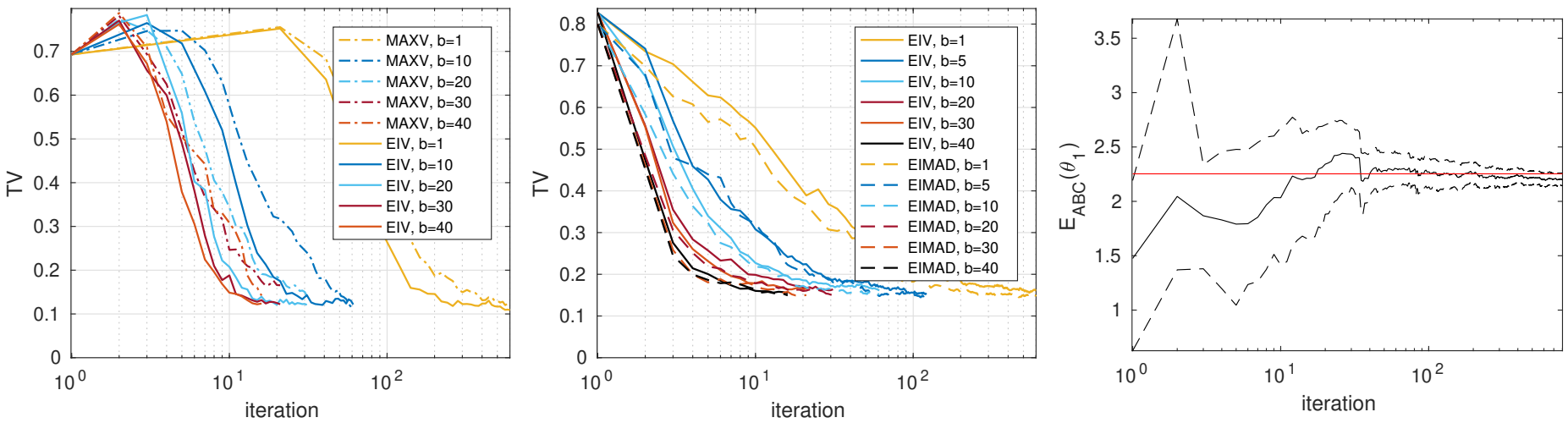


Figure: Left: Bacterial infections model (3D), middle: Lorenz model (2D), right: illustration of uncertainty quantification of ABC posterior expectation (Lorenz model).

For full details:

- Come to our poster
- Check out our working paper in arxiv[3]

[3] M. Järvenpää, A. Vehtari, and P. Marttinen. *Batch simulations and uncertainty quantification in Gaussian process surrogate-based approximate Bayesian computation*. Available at https://arxiv.org/abs/1910.06121. 2019.

Global Approximate Inference via Local Linearisation for Temporal Gaussian Processes

William Wilkinson, Paul Chang, Michael Riis Andersen, Arno Solin

Aalto University, Technical University of Denmark

AABI Symposium - 8 December 2019

Approximate Inference in temporal GPs

There exists a dual kernel / SDE form for most popular Gaussian process (GP) models

$$f(t) \sim \mathcal{GP}\big(0,\, K_\theta(t, t')\big), \qquad \mathbf{f}_k = \mathbf{A}_{\theta,k}\mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$
$$y_k \sim p(y_k \mid f(t_k)) \qquad y_k = h(\mathbf{f}_k, \mathbf{r}_k), \qquad \mathbf{r}_k \sim \mathrm{N}(0, \mathbf{R}_k)$$

Approximate Inference in temporal GPs

There exists a dual kernel / SDE form for most popular Gaussian process (GP) models

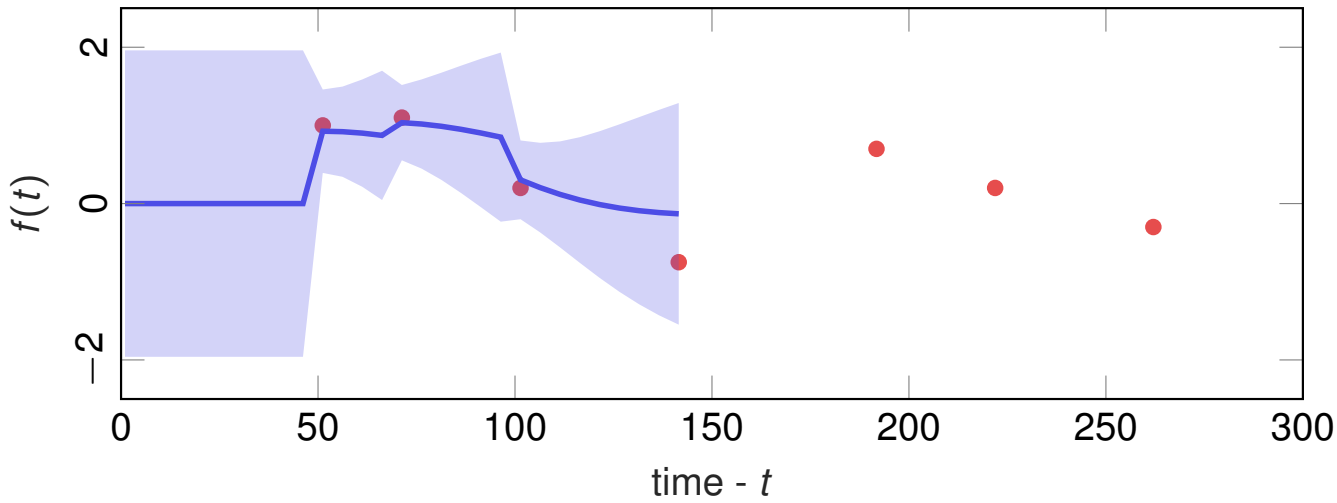
$$f(t) \sim \mathcal{GP}\big(0,\, K_\theta(t, t')\big), \qquad \mathbf{f}_k = \mathbf{A}_{\theta,k}\mathbf{f}_{k-1} + \mathbf{q}_{k-1},$$
$$y_k \sim p(y_k \mid f(t_k)) \qquad y_k = h(\mathbf{f}_k, \mathbf{r}_k), \qquad \mathbf{r}_k \sim \mathrm{N}(0, \mathbf{R}_k)$$

inference in $\mathcal{O}(n)$ via Kalman filtering and smoothing

Expectation propagation (EP)

Kalman filter update step:

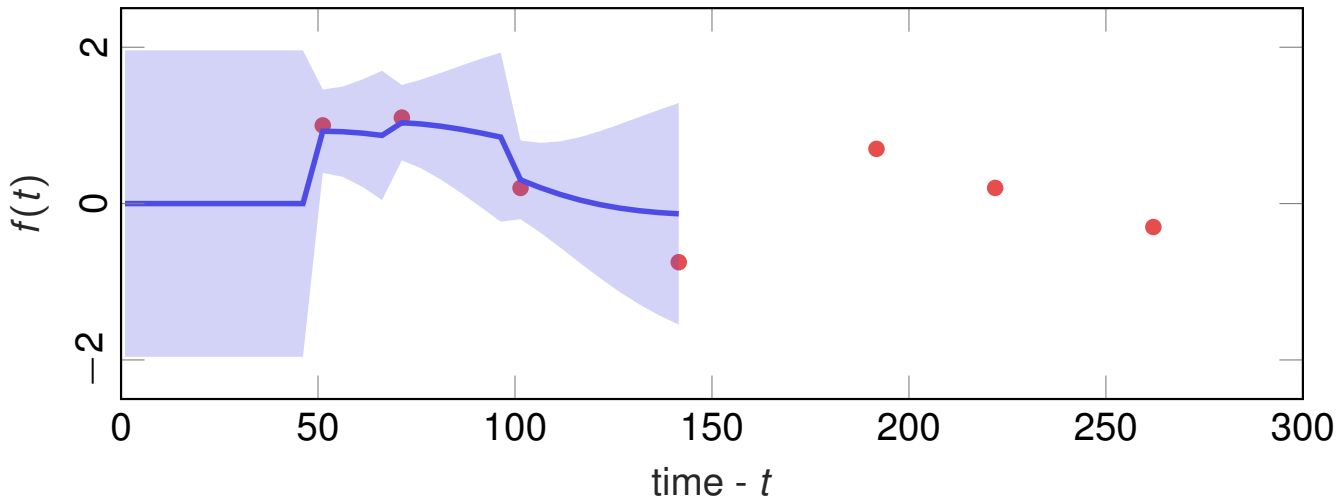
$$p(\mathbf{f}_k | y_{1:k}) \propto \mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})\, p(y_k \mid f(t_k))$$



Expectation propagation (EP)

Kalman filter update step:

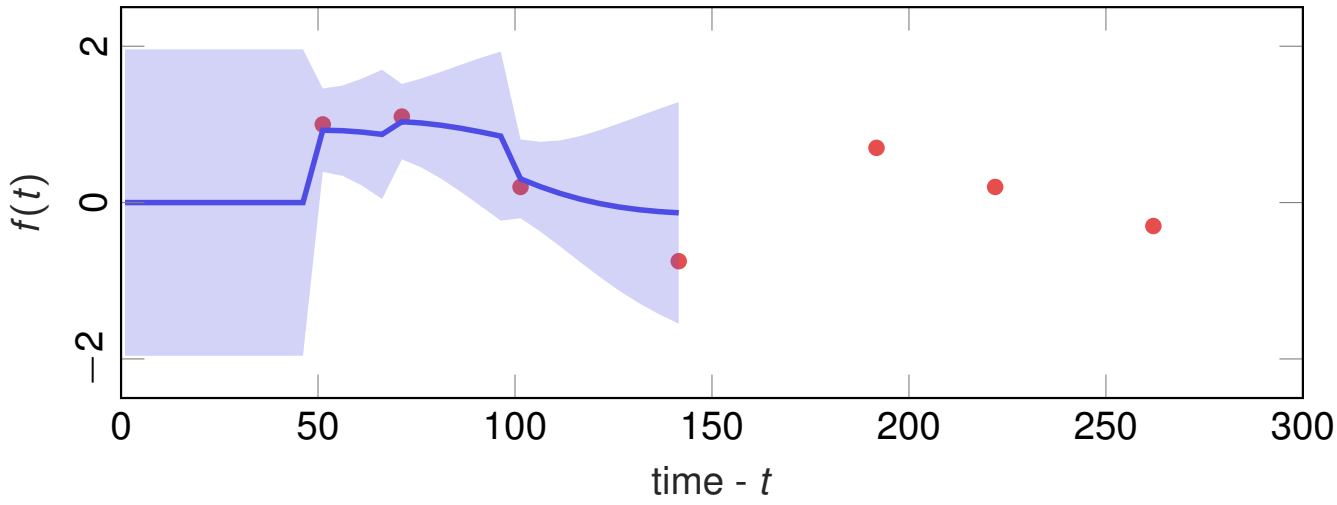
$$p(\mathbf{f}_k|y_{1:k}) \propto \underbrace{\mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})}_{\text{“cavity distribution”}} p(y_k \mid f(t_k))$$



Expectation propagation (EP)

Kalman filter update step:

$$p(\mathbf{f}_k|y_{1:k}) \propto \mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})\, p(y_k \mid f(t_k))$$
$$\approx \mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})\, \underbrace{\mathrm{N}(\mathbf{m}_k^{\text{site}}, \mathbf{P}_k^{\text{site}})}_{\text{“site”}}$$

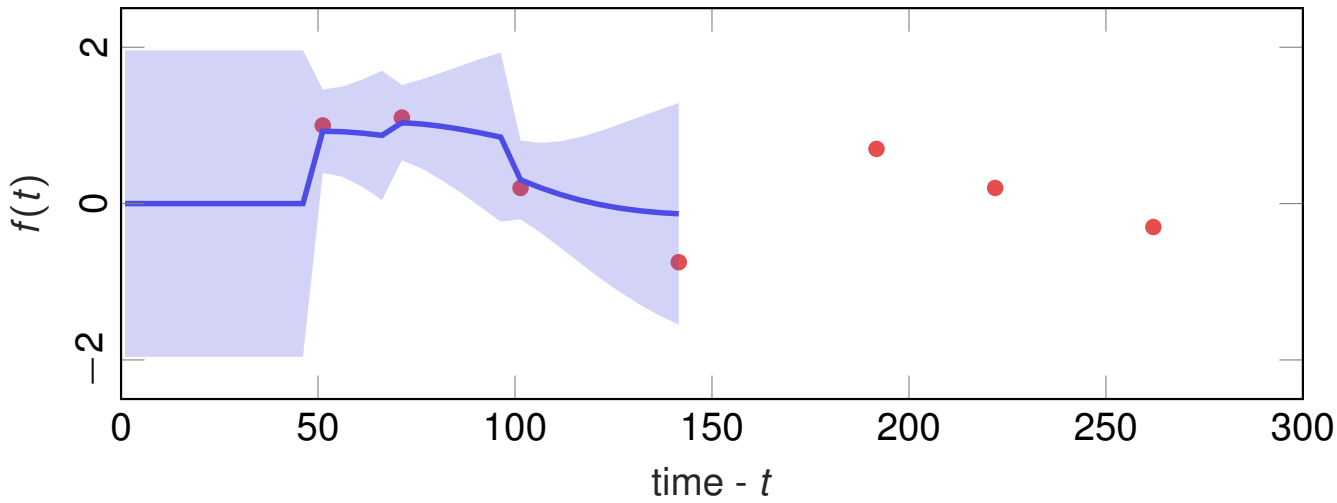


Expectation propagation (EP)

Kalman filter update step:

$$p(\mathbf{f}_k | y_{1:k}) \propto \mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})\, p(y_k \mid f(t_k))$$
$$\approx \mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})\, \mathrm{N}(\mathbf{m}_k^{\text{site}}, \mathbf{P}_k^{\text{site}})$$

EP update:
match moments

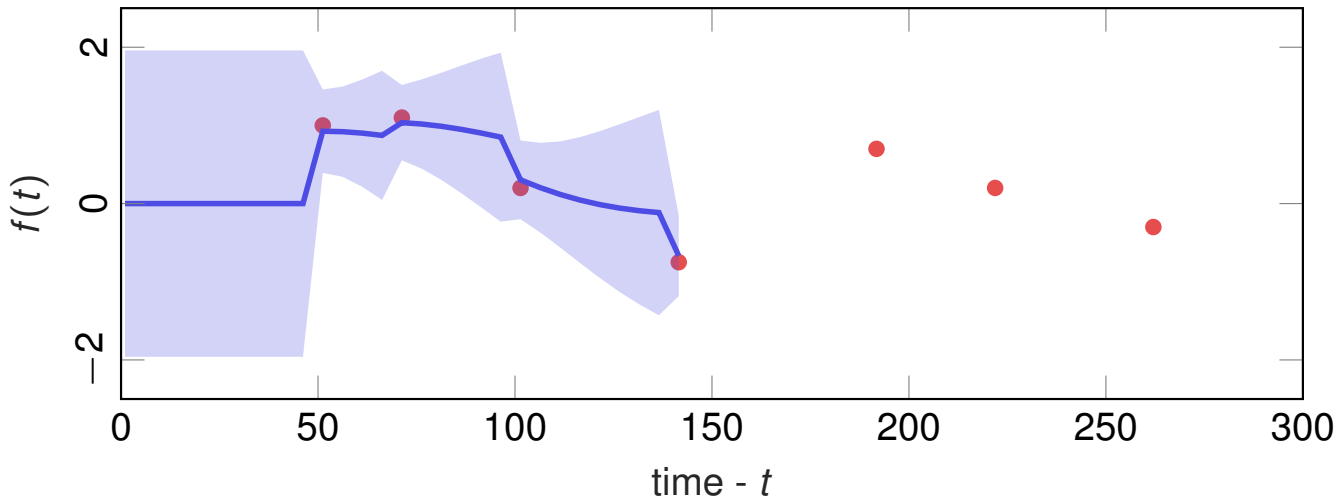


Expectation propagation (EP)

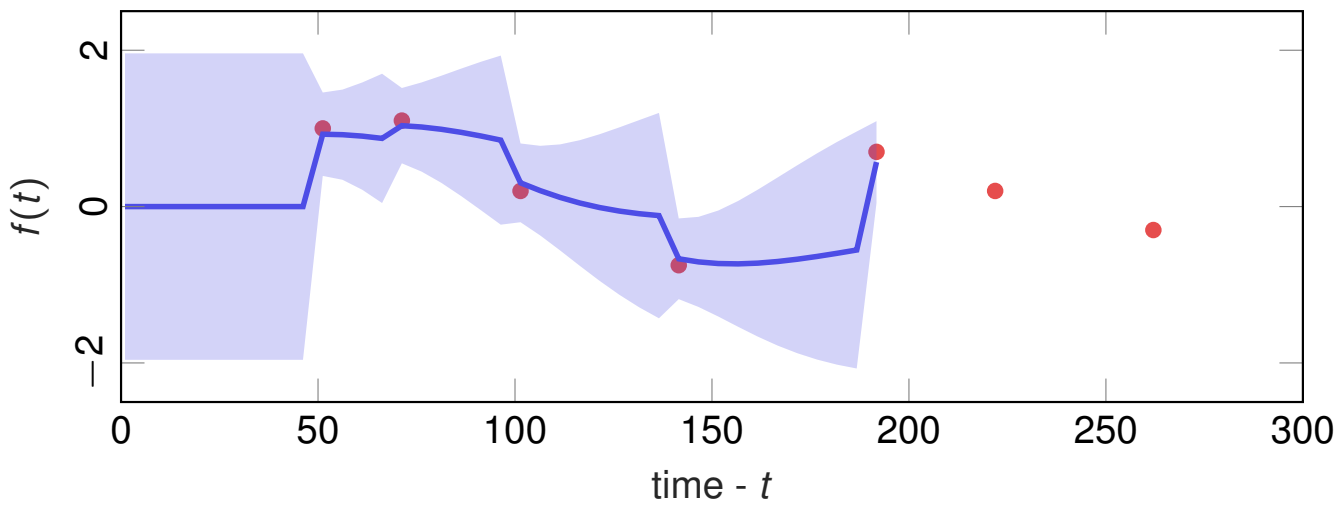
Kalman filter update step:

$$p(\mathbf{f}_k | y_{1:k}) \propto \mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})\, p(y_k \mid f(t_k))$$
$$\approx \mathrm{N}(\mathbf{m}_k^{\text{predict}}, \mathbf{P}_k^{\text{predict}})\, \mathrm{N}(\mathbf{m}_k^{\text{site}}, \mathbf{P}_k^{\text{site}})$$

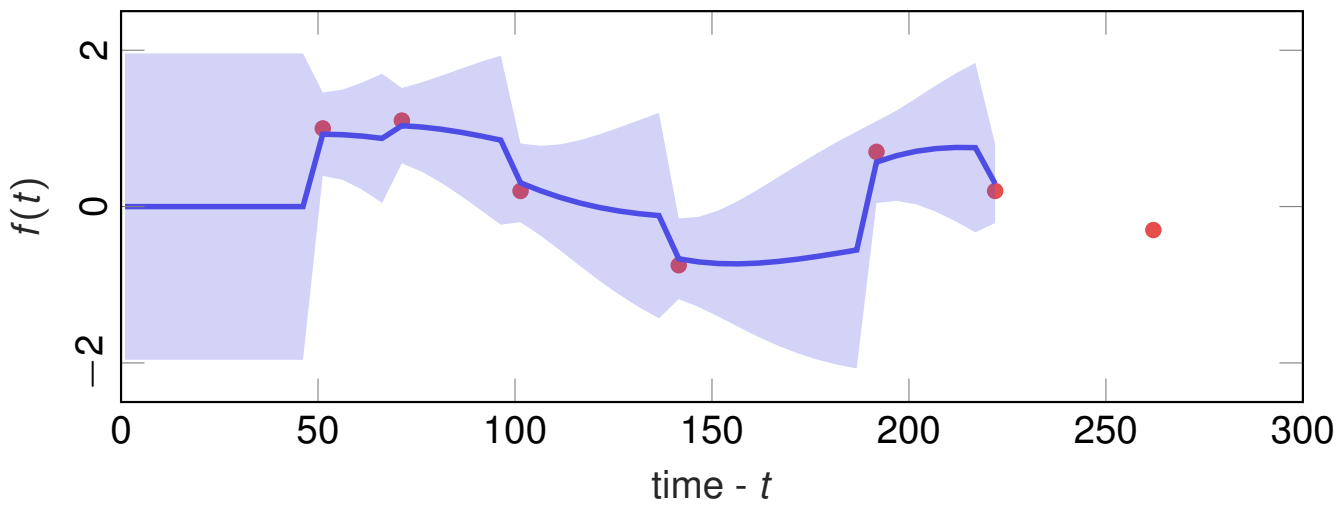
EP update:
match moments



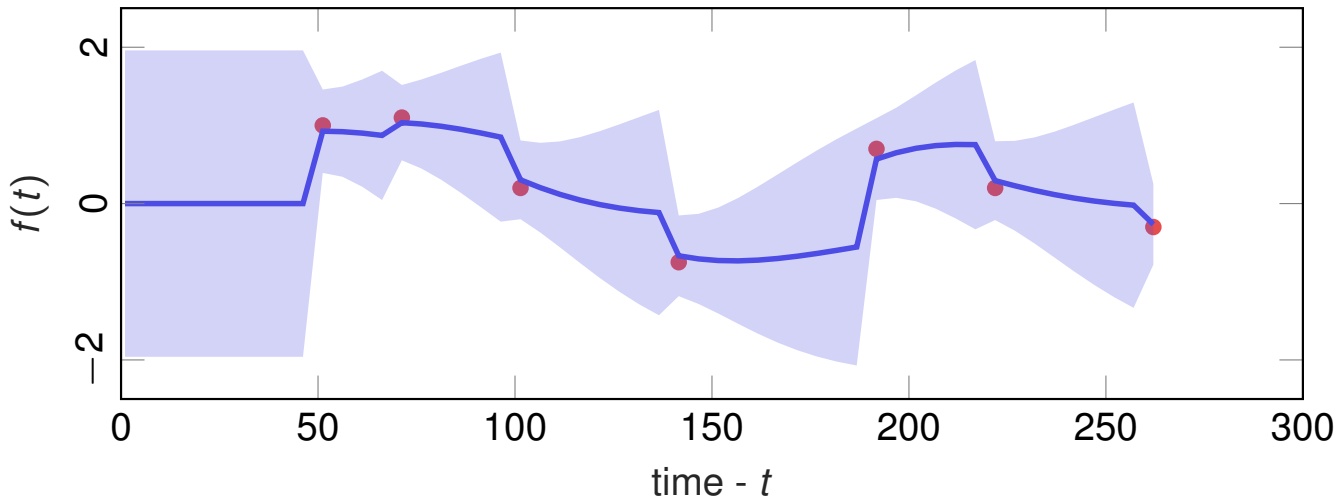
Expectation propagation (EP)



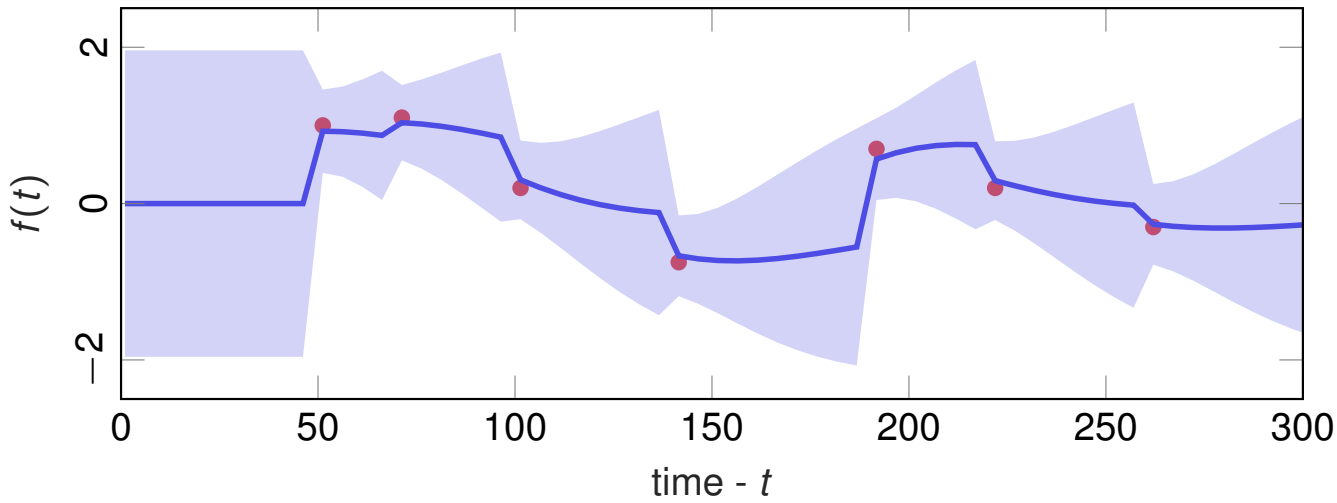
Expectation propagation (EP)



Expectation propagation (EP)



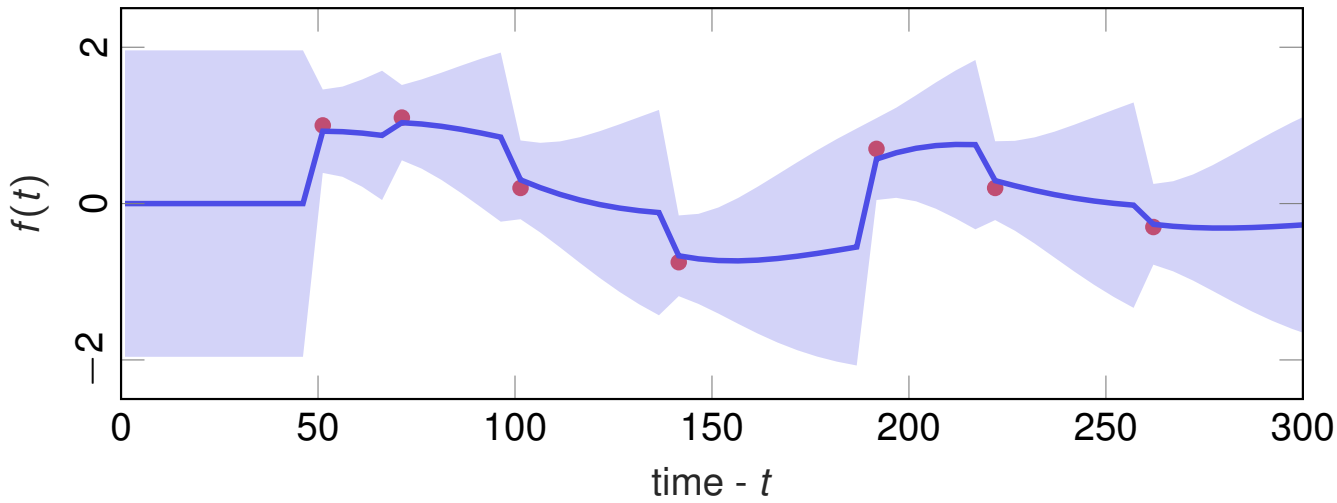
Expectation propagation (EP)



Expectation propagation (EP)

Smoothing:

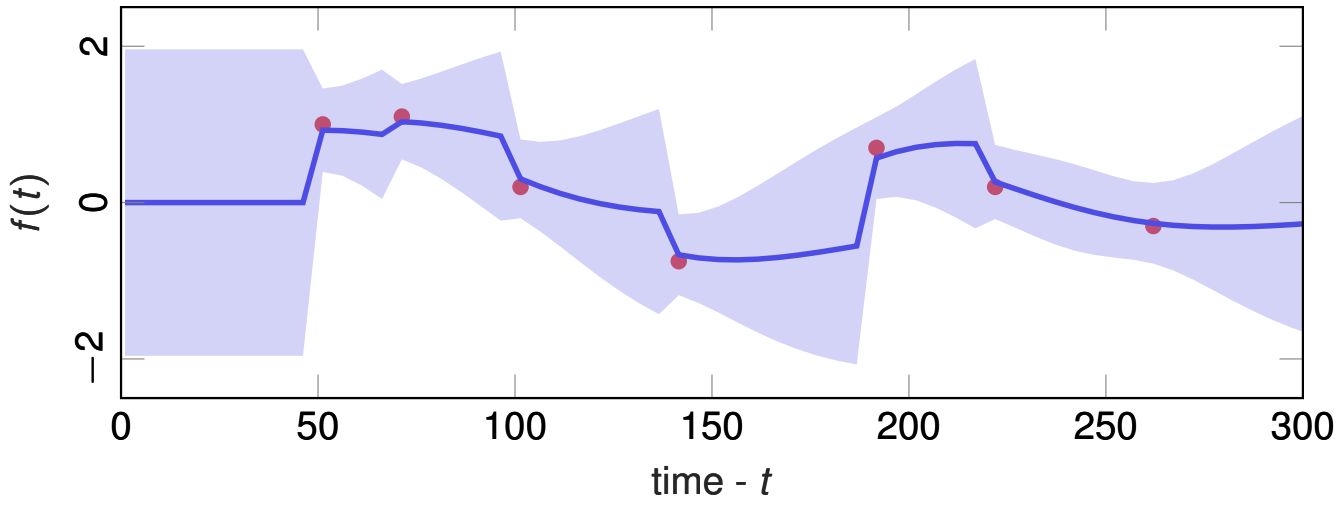
- update predictions with future observations
- update the EP sites along the way



Expectation propagation (EP)

Smoothing:

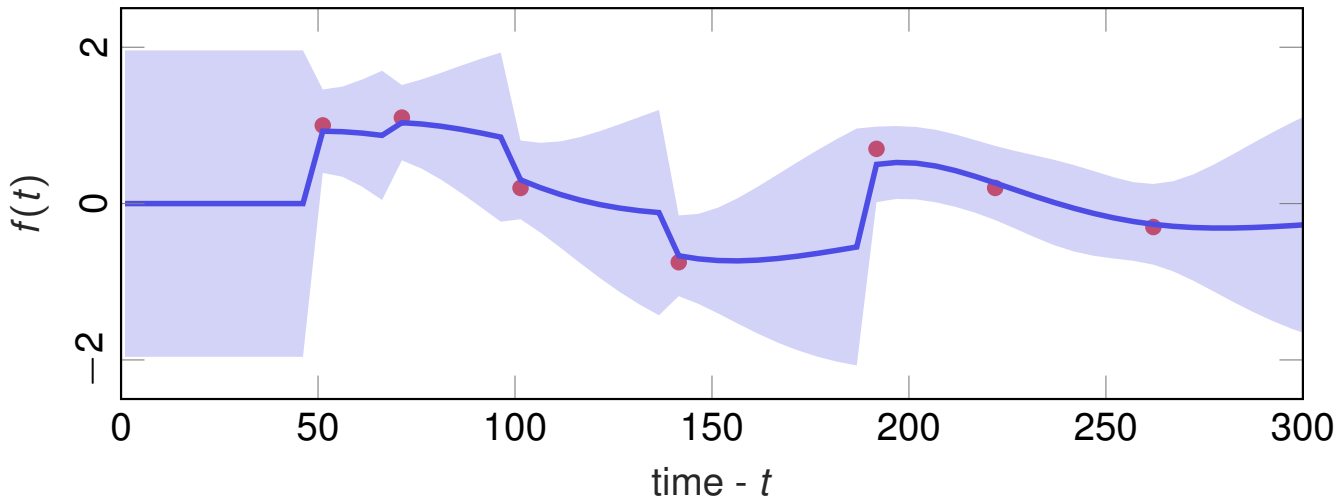
- update predictions with future observations
- update the EP sites along the way



Expectation propagation (EP)

Smoothing:

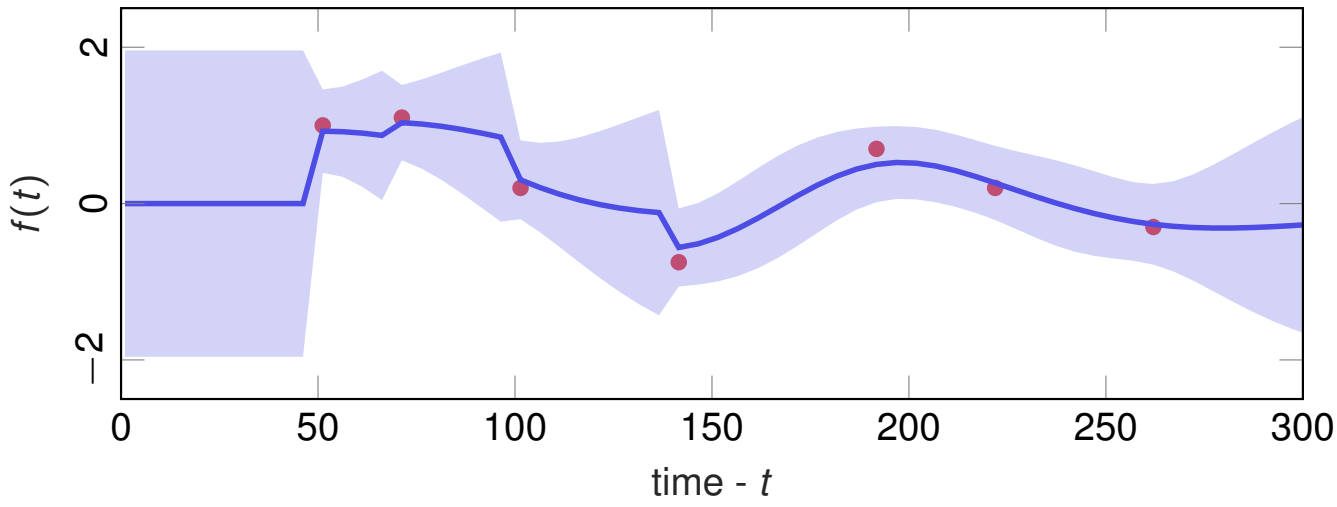
- update predictions with future observations
- update the EP sites along the way



Expectation propagation (EP)

Smoothing:

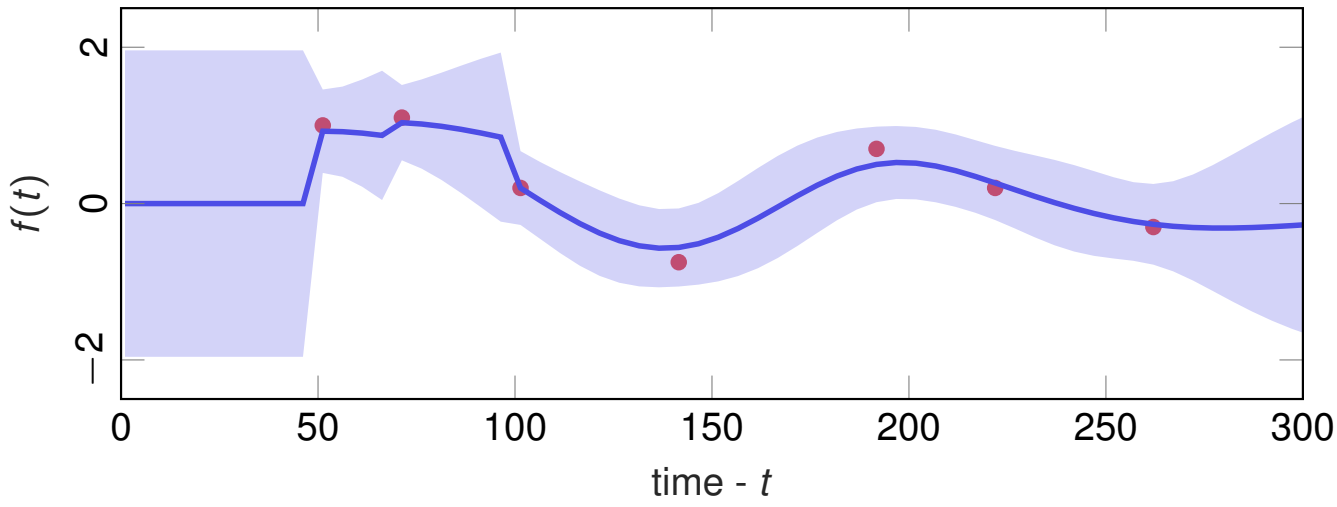
- update predictions with future observations
- update the EP sites along the way



Expectation propagation (EP)

Smoothing:

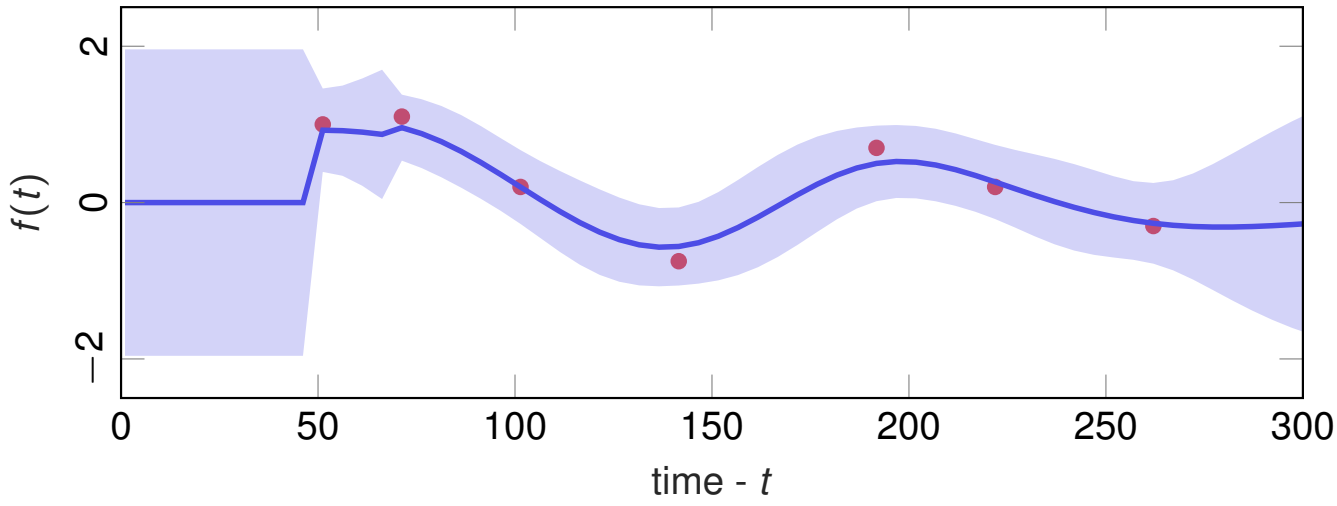
- update predictions with future observations
- update the EP sites along the way



Expectation propagation (EP)

Smoothing:

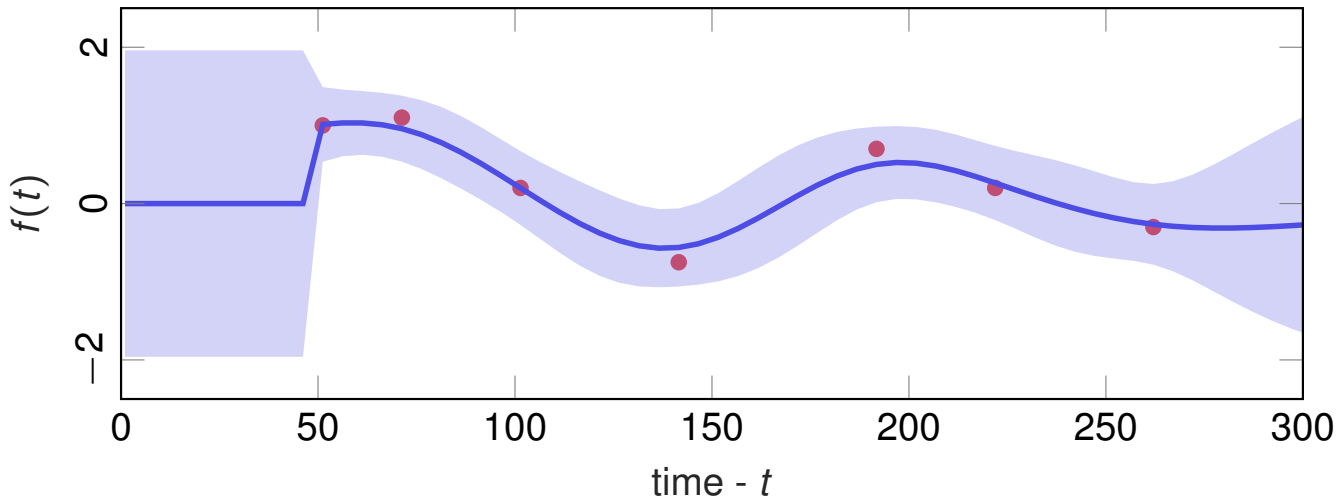
- update predictions with future observations
- update the EP sites along the way



Expectation propagation (EP)

Smoothing:

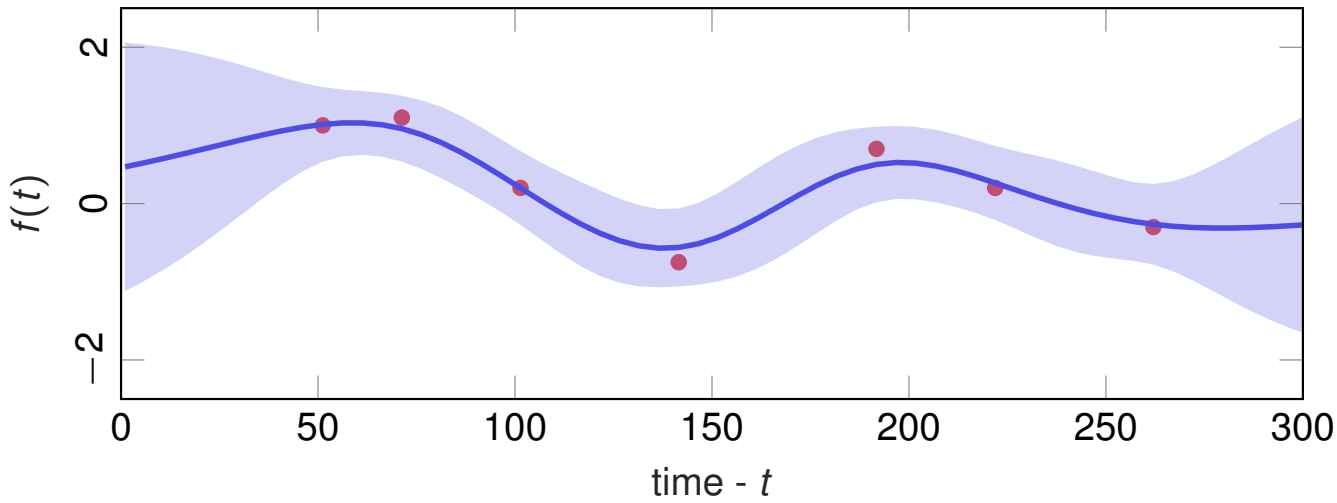
- update predictions with future observations
- update the EP sites along the way



Expectation propagation (EP)

Smoothing:

- update predictions with future observations
- update the EP sites along the way

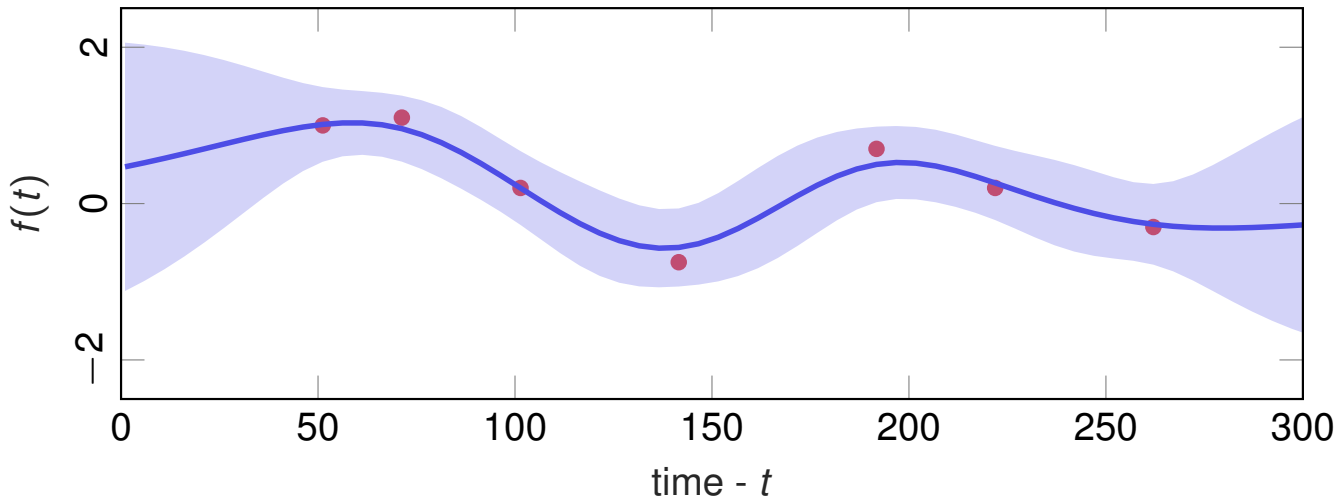


Expectation propagation (EP)

Smoothing:

- update predictions with future observations
- update the EP sites along the way

Can add in EP extras: power (α) and damping



Moment matching

Moment matching requires the intractable expectation:

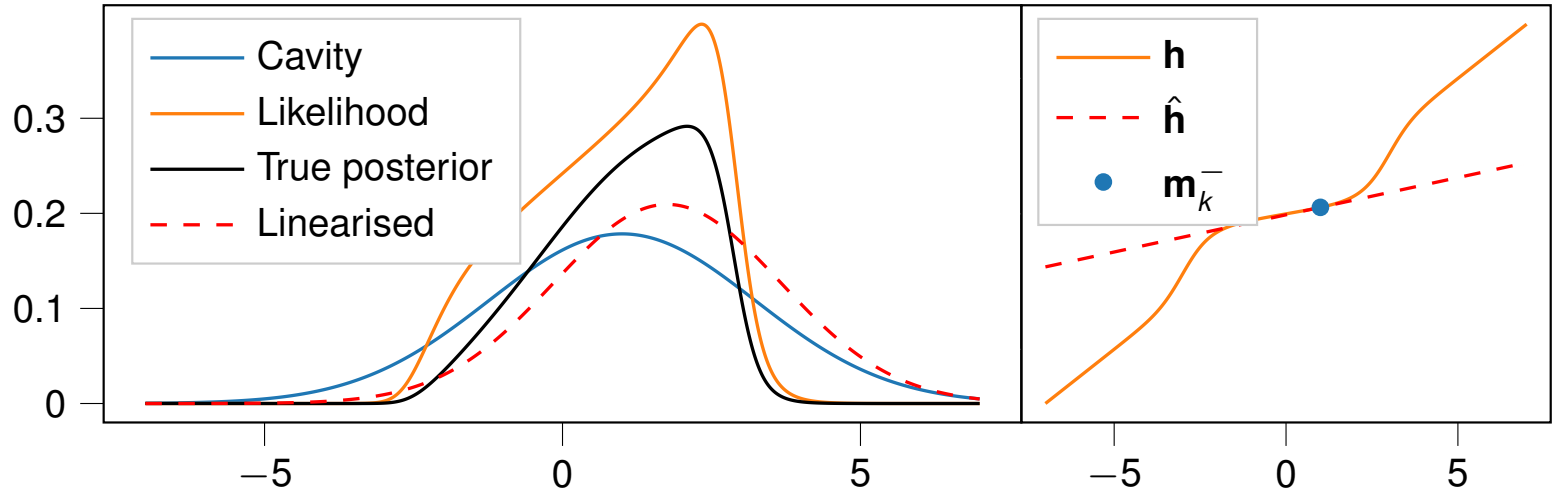
$$\mathbb{E}_{\mathsf{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}\left[p(\mathbf{y}_k \mid \mathbf{f}_k)^\alpha\right]$$

Moment matching

Moment matching requires the intractable expectation:

$$\mathbb{E}_{\mathsf{N}(\mathbf{m}_k^-, \mathbf{P}_k^-)}\left[p(\mathbf{y}_k \mid \mathbf{f}_k)^\alpha\right]$$

If we solve this by linearising the state space observation model $\mathbf{h}(\cdot)$, the EP algorithm reduces exactly to the Extended Kalman filter (for EP power $\alpha = 1$).



Unifying EP and the EKF

- For sequential data, the EKF is equivalent to single-sweep EP where the moment matching integral is solved via linearisation.

Unifying EP and the EKF

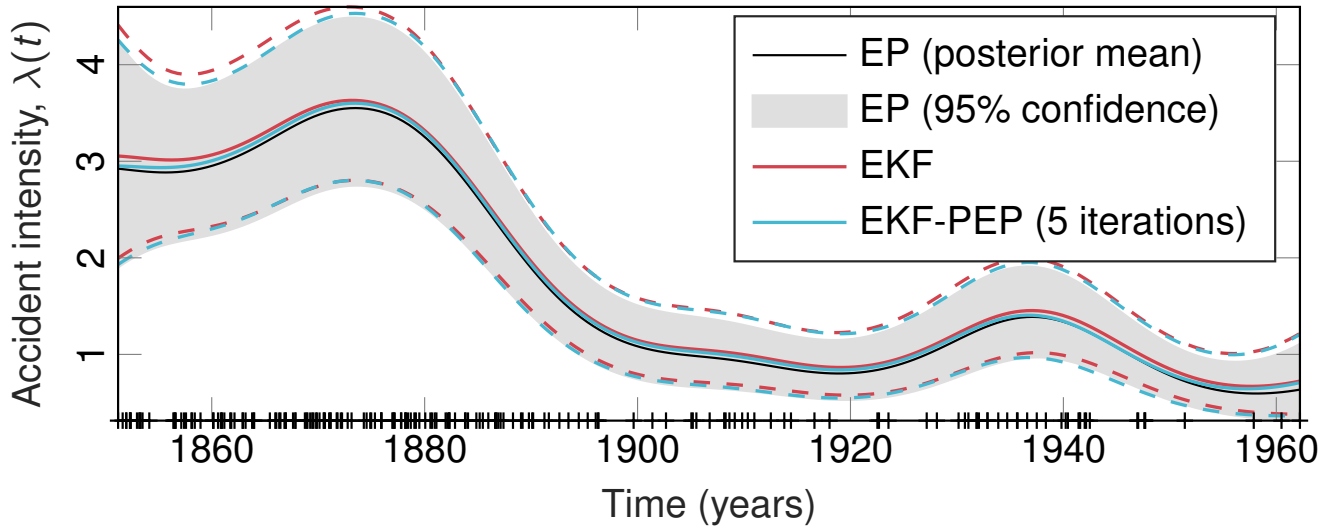
- For sequential data, the EKF is equivalent to single-sweep EP where the moment matching integral is solved via linearisation.

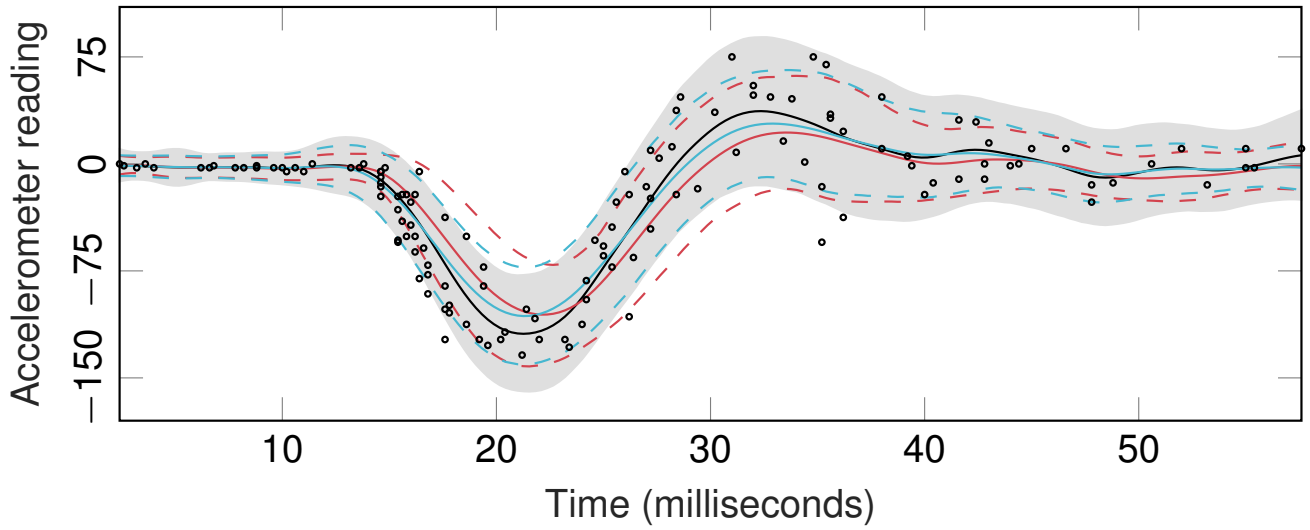
- Our algorithm iteratively refines the EKF by linearising about the cavity, rather than the filter predictions (prior).

Unifying EP and the EKF

- For sequential data, the EKF is equivalent to single-sweep EP where the moment matching integral is solved via linearisation.
- Our algorithm iteratively refines the EKF by linearising about the cavity, rather than the filter predictions (prior).
- More connections to be found (*e.g.,* UKF, PLF)

Examples





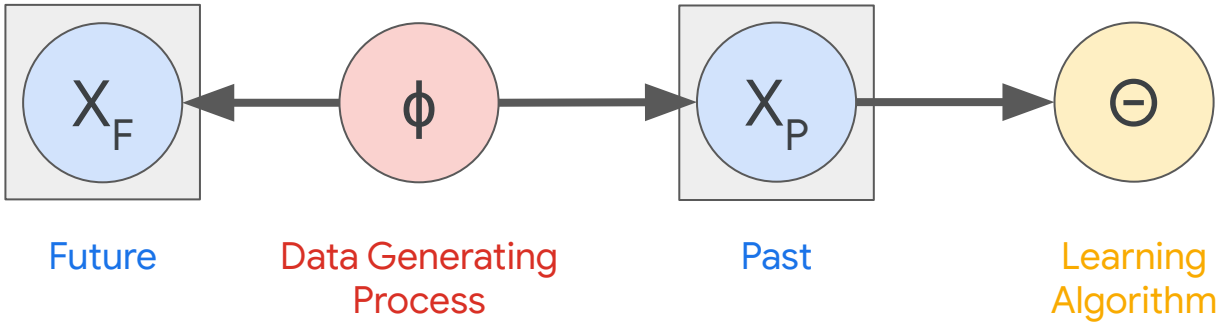
Thanks for listening

contact: **william.wilkinson@aalto.fi**

Variational Predictive Information Bottleneck

Alexander A. Alemi alemi@google.com
Google Research

arXiv:1910.10831



$$\max_{p(\theta;\boldsymbol{x_P})} I(\theta;\boldsymbol{x_F}) \text{ s.t. } I(\theta;\boldsymbol{x_P}) = I_0 \implies \min_{p(\theta|\boldsymbol{x_P})} \left\langle \log \frac{p(\theta|\boldsymbol{x_P})}{q(\theta)} - \beta \sum_i \log q(x_i|\theta) \right\rangle$$

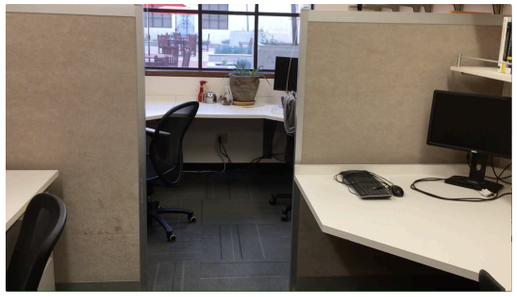
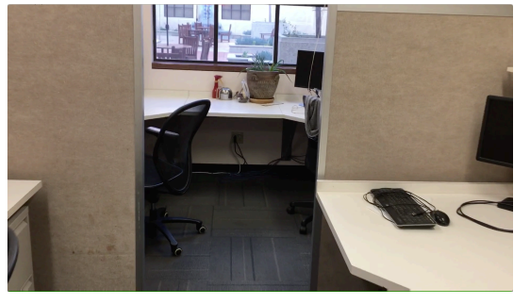
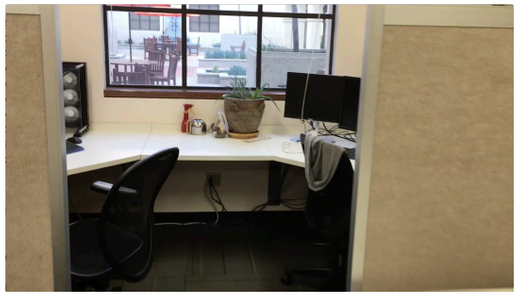
Motivates: Maximum Likelihood / Bayesian Inference / Variational Bayes / Power Likelihood / Generalized Bayesian Inference / Gibbs VI / Neural Processes / Reference Priors / Empirical Bayes / Data Augmentation / ...

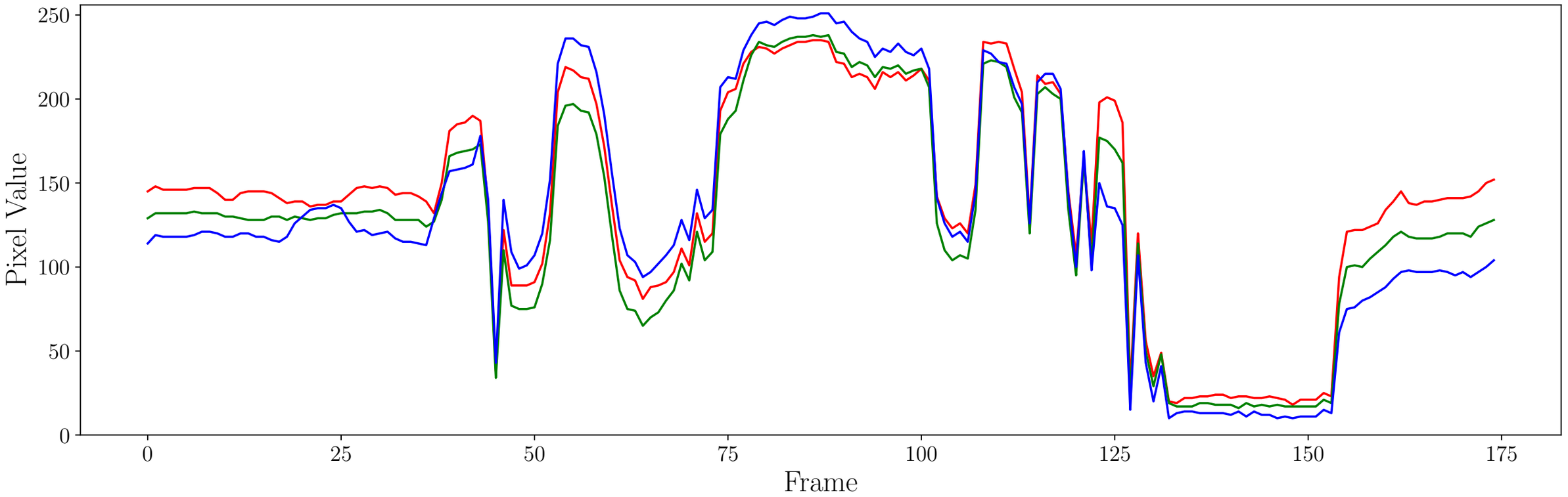
Improving *Sequential Latent Variable Models* with **Autoregressive Flows**

Joseph Marino[1], Lei Chen[2], Jiawei He[2], Stephan Mandt[3]

[1]*California Institute of Technology,*
[2]*Simon Fraser University,*
[3]*UC Irvine*

MOTIVATION

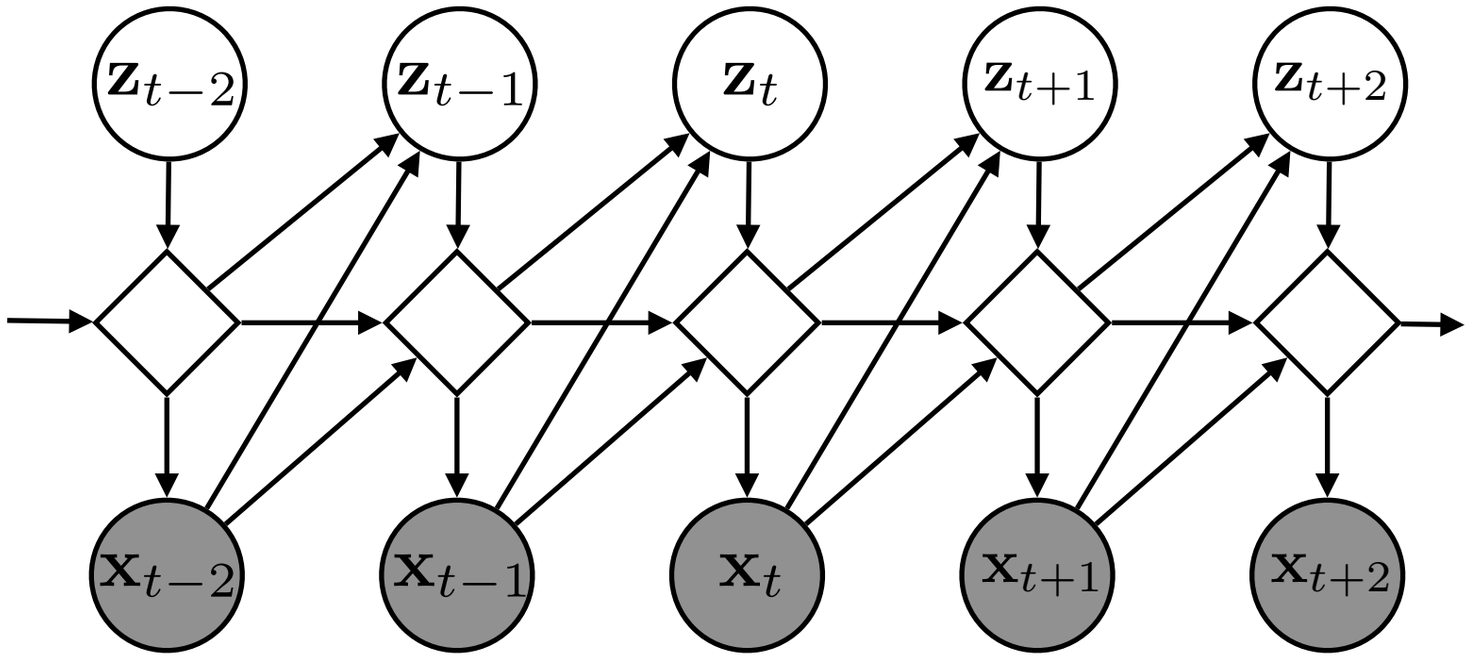




sequences in the natural world

are typically highly ***dependent in time***

MOTIVATION



sequential latent variable model

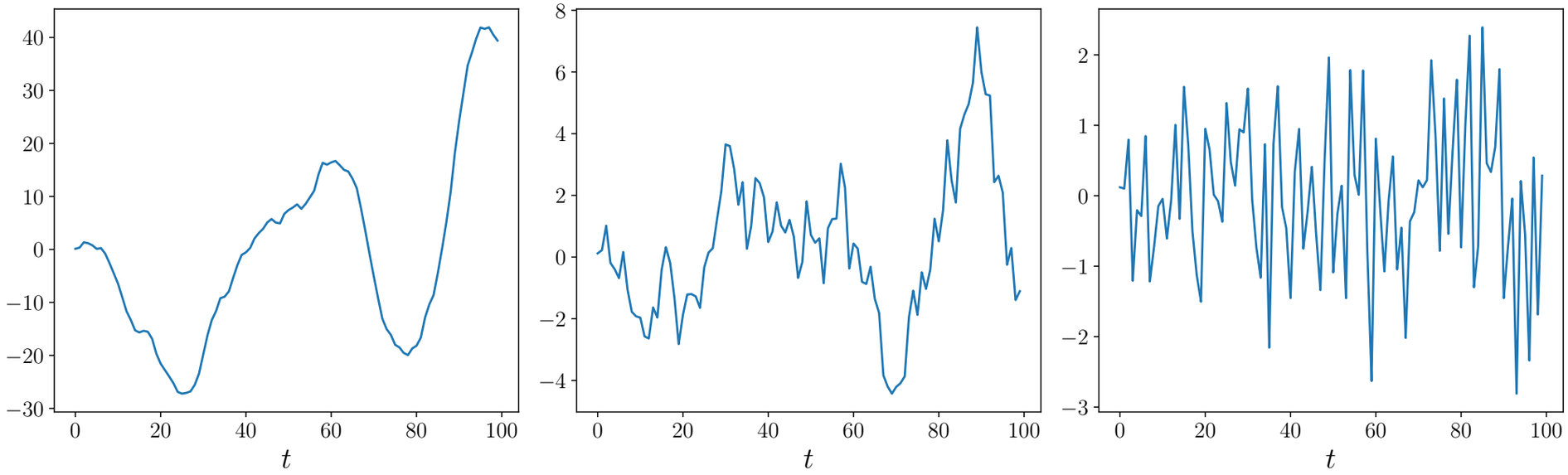
$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t}) p_\theta(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$$

how can we simplify the estimation of dynamics in $\mathbf{z}$*?*

⟶ *reduce the degree of temporal dependence*

TEMPORAL DECORRELATION

position $\mathbf{x}$ → **velocity** $\mathbf{u} = \Delta\mathbf{x}$ → **noise** $\mathbf{w} = \Delta\mathbf{u}$



high temporal dependence

no temporal dependence

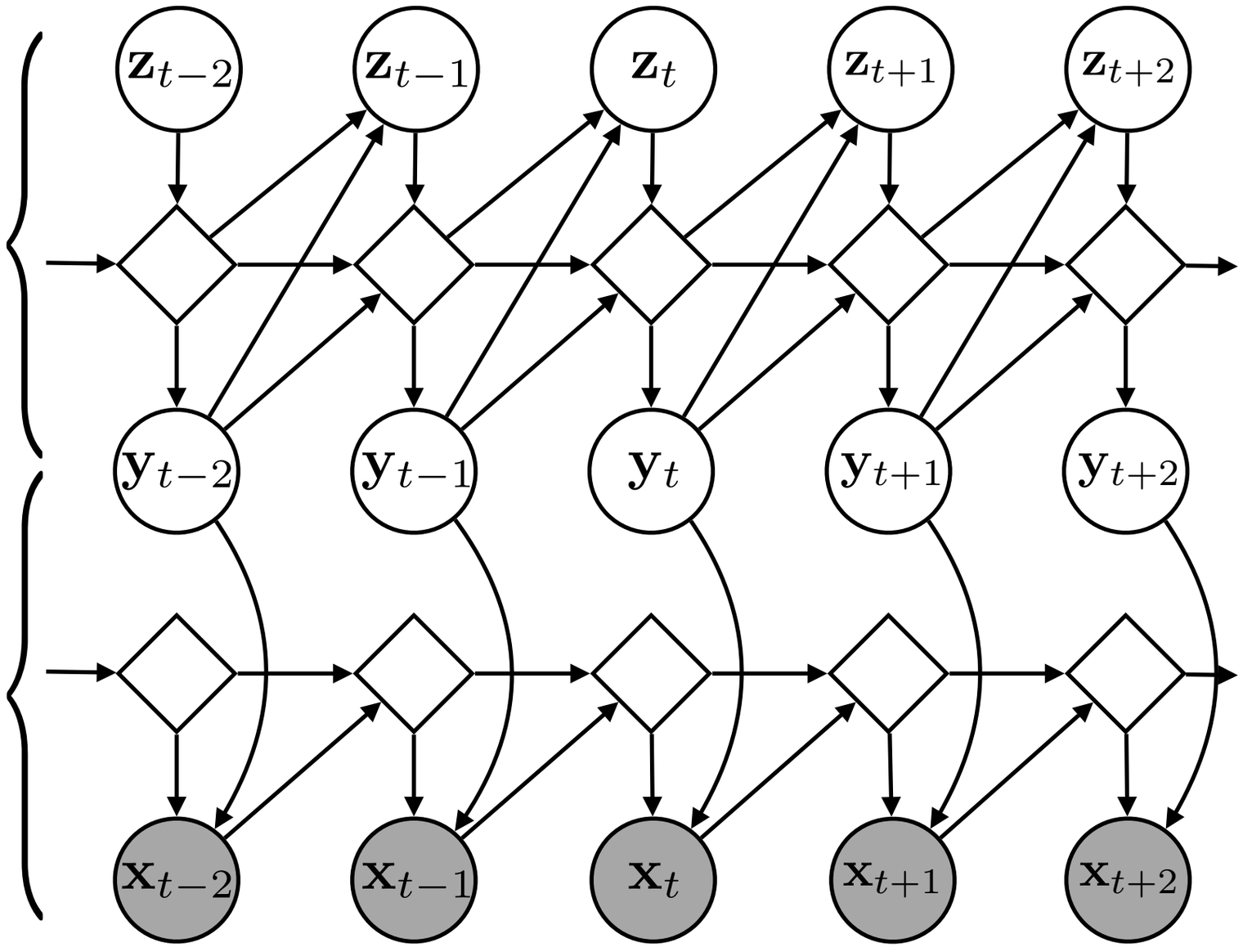
generalize temporal differences to $\mathbf{y}_t = \dfrac{\mathbf{x}_t - \boldsymbol{\mu}_\theta(\mathbf{x}_{<t})}{\boldsymbol{\sigma}_\theta(\mathbf{x}_{<t})}$

autoregressive flow across time

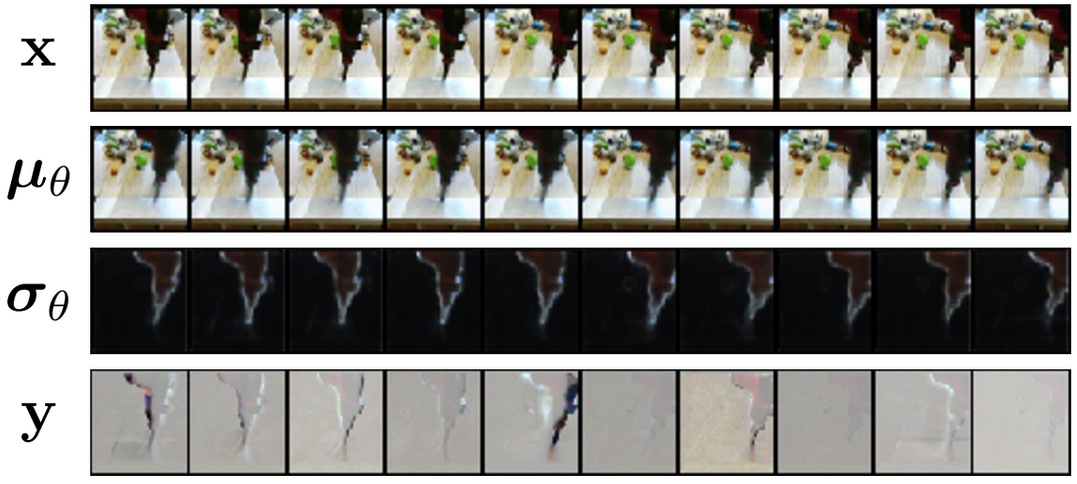
APPROACH

sequential latent variable model

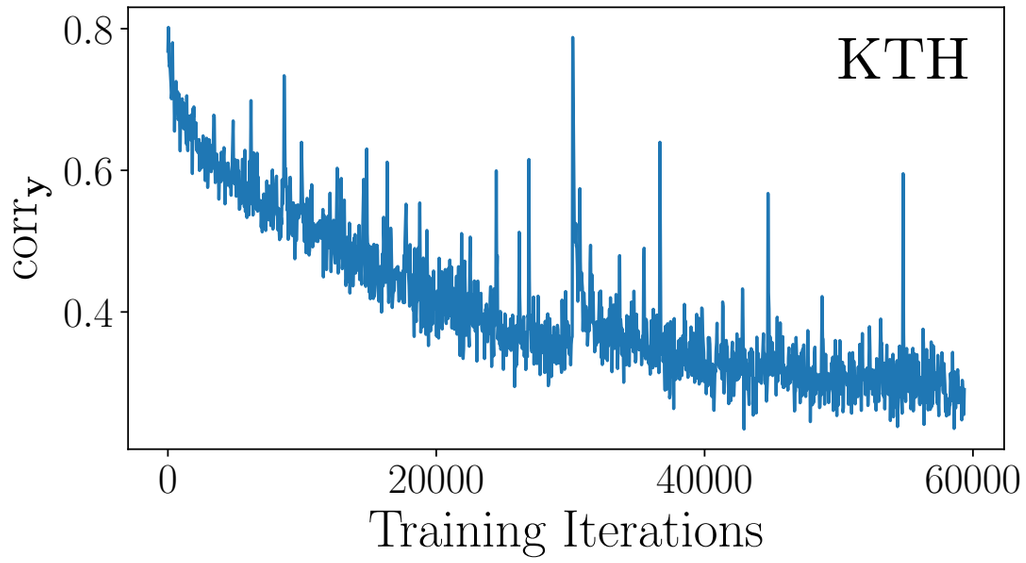
autoregressive flow



visualize flows



quantify decorrelation



performance improvements

test negative log-likelihood in *nats / dim*

	M-MNIST	BAIR	KTH
1-AF	2.15	3.05	3.34
2-AF	2.13	2.90	3.35
SLVM	$\leq$ 1.92	$\leq$ 3.57	$\leq$ 4.63
SLVM w/ 1-AF	$\leq$ **1.86**	$\leq$ **2.35**	$\leq$ **2.39**

Characterizing the High-dimensional Bias of Kernel-based Particle Inference Algorithms

Jimmy Ba[1,2], Murat A. Erdogdu[1,2], Marzyeh Ghassemi[1,2], Taiji Suzuki[3,4], Shengyang Sun[1,2], Denny Wu[1,2,4], Tianzong Zhang[5]

University of Toronto[1], Vector Institute[2], University of Tokyo[3], RIKEN AIP[4], Tsinghua University[5]

December 8, 2019

Particle-based Inference Algorithm

Goal: given access to $\nabla f(\boldsymbol{x})$, approximate $p(\boldsymbol{x}) \propto \exp(-f(\boldsymbol{x}))$ with particles $\{\boldsymbol{x}_i\}_{i=1}^n$ via iterative refinement $\boldsymbol{x}_i = \boldsymbol{x}_i + \epsilon\Delta(\boldsymbol{x}_i)$.

Stein Variational Gradient Descent [2]:

$$\Delta^{\text{SVGD}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{x}'\sim q}[\underbrace{\nabla_{\boldsymbol{x}'} \log p(\boldsymbol{x}')k(\boldsymbol{x}', \boldsymbol{x})}_{\text{driving force}}] + \mathbb{E}_{\boldsymbol{x}'\sim q}[\underbrace{\nabla_{\boldsymbol{x}'} k(\boldsymbol{x}', \boldsymbol{x})}_{\text{repulsive force}}].$$

Curse of dimensionality: marginal variance estimated by SVGD scales **inversely with dimensionality** [3] $\Rightarrow$ not suitable for high-dimensional inference problems

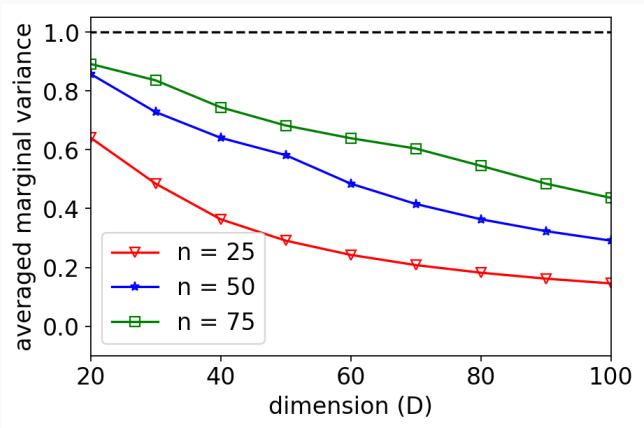
Comparing SVGD with MMD-based Algorithm

MMD-descent: update particles to decrease the MMD [1]:

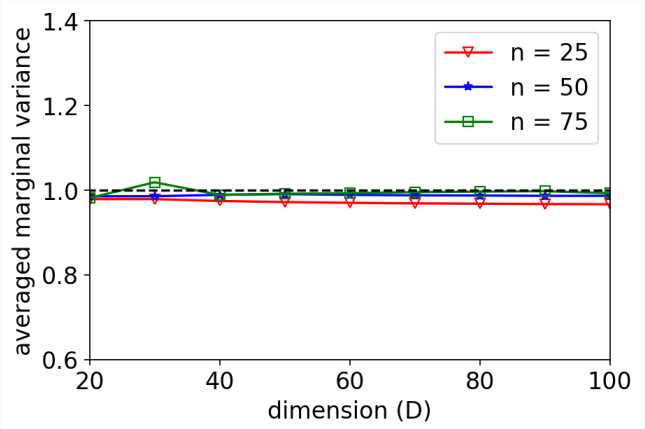
$$\Delta^{\mathrm{MMD}}(x) = \mathbb{E}_{y\sim p}[\underbrace{\nabla_x k(x,y)}_{\text{driving force}}] + \mathbb{E}_{x'\sim q}[\underbrace{-\nabla_x k(x,x')}_{\text{repulsive force}}].$$

MMD vs. SVGD:

- same repulsive force
- driving force integrated under different distributions
- MMD-descent **does not underestimate** marginal variance



(a) variance.



(b) MMD-descent.

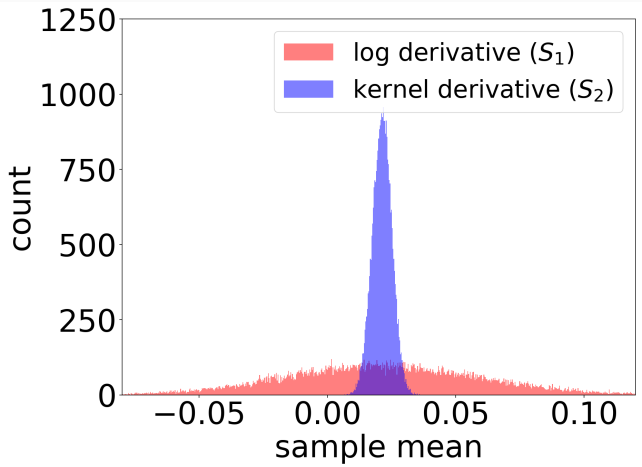
Understanding the Variance Collapse in SVGD

Peril of Integration by Parts:

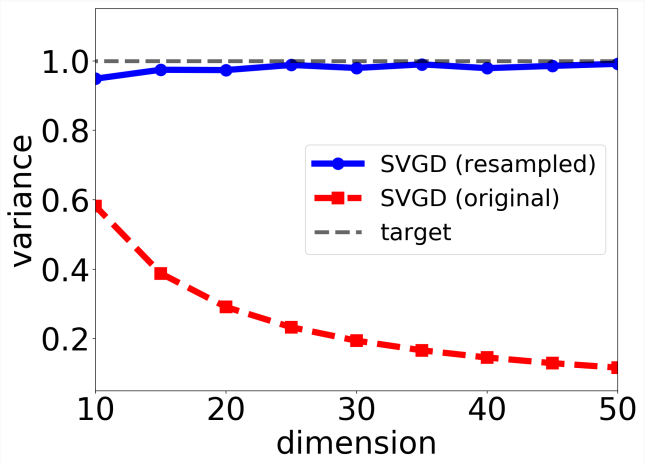
- $-\mathbb{E}_{\boldsymbol{y}\sim p}[\nabla_{\boldsymbol{y}} k(\boldsymbol{x},\boldsymbol{y})] = \mathbb{E}_{\boldsymbol{y}\sim p}[\nabla_{\boldsymbol{y}} \log p(\boldsymbol{y}) k(\boldsymbol{x},\boldsymbol{y})]$
- the latter term tends to have higher variance

Bias from Deterministic Update:

- q is entirely represented by the same set of particles and i.i.d. samples cannot be drawn



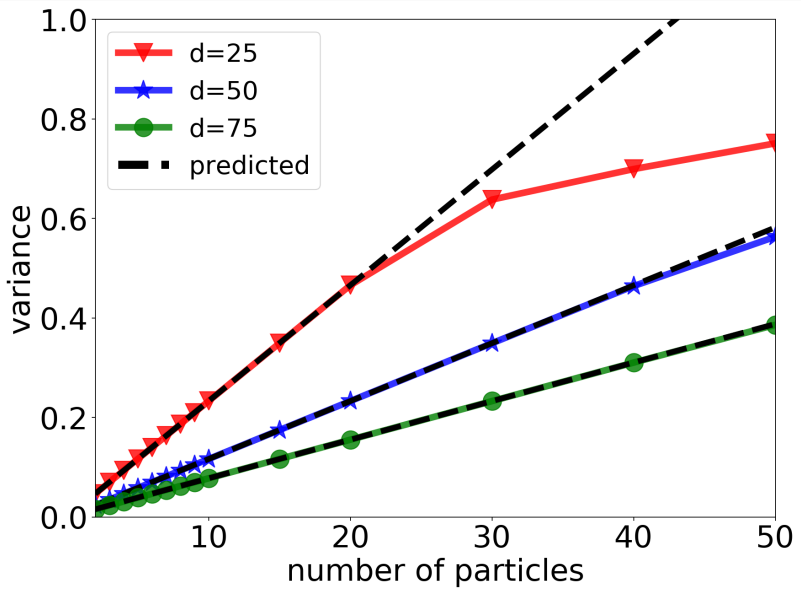
(a) variance.



(b) SVGD (resampled).

A High-dimensional Characterization

Proposition (Informal)
Let $n, d \to \infty$ and $n/d \to \gamma \in (0, 1)$. For learning **unit Gaussian** *target using* **Gaussian RBF kernel** *with bandwidth $\sigma^2 \in \Theta(d)$, given that particles at the fixed point "correlate weakly", then SVGD equilibrates with marginal variance $v^{\text{SVGD}} \to (e-1)^{-1}\gamma$, whereas MMD-descent leads to $v^{\text{MMD}} \to 1$.*



Predicted variance of SVGD.

Reference

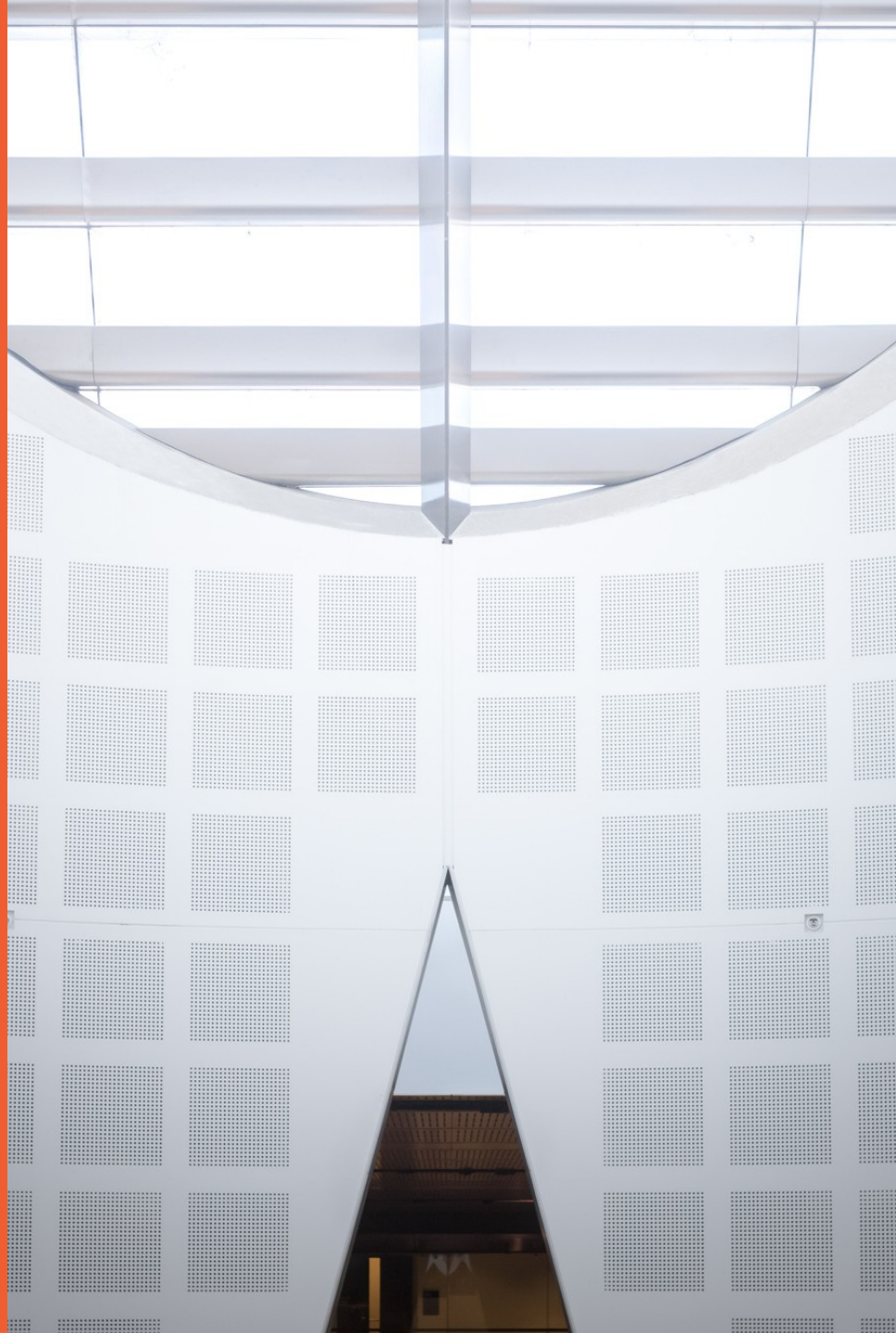

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola.
A kernel two-sample test.
Journal of Machine Learning Research, 13(Mar):723–773, 2012.


Q. Liu and D. Wang.
Stein variational gradient descent: A general purpose bayesian inference algorithm.
In *Advances In Neural Information Processing Systems*, pages 2378–2386, 2016.


J. Zhuo, C. Liu, J. Shi, J. Zhu, N. Chen, and B. Zhang.
Message passing stein variational gradient descent.

Optimal Transport for Distribution Adaptation in Bayesian Hilbert Maps

Anthony Tompkins
Ransalu Senanayake
Fabio Ramos



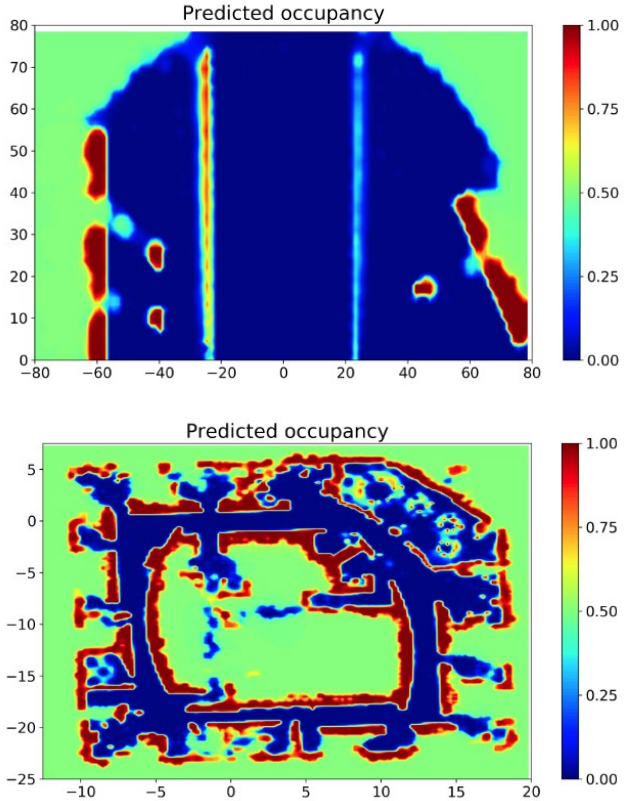
Motivation

In **probabilistic inference** and inference in general,
We often face the problems of

1. **costly optimisation** for new models on new datasets,
2. **picking our priors** well to speed up “learning”.

Problem description

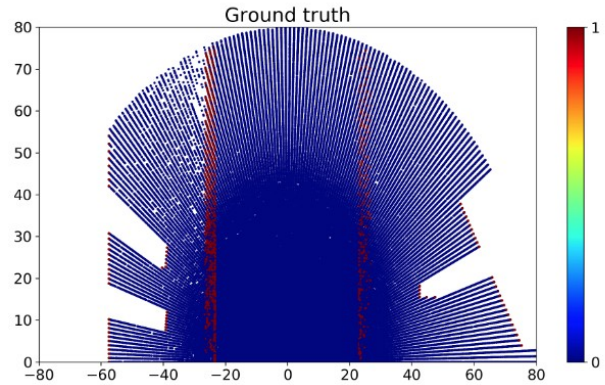
Probabilistic map

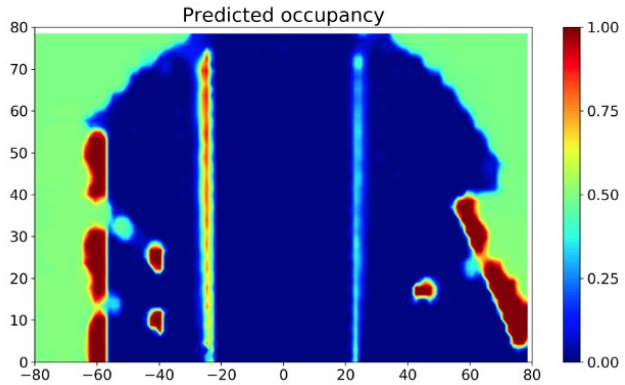


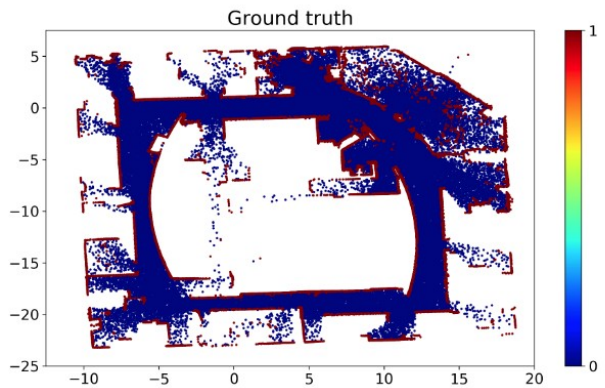
Problem description

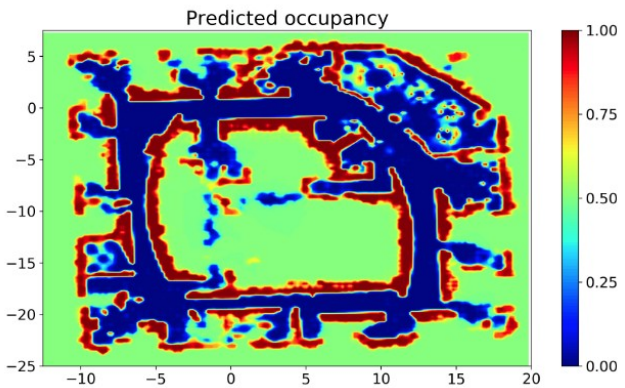
Laser scans

Probabilistic map







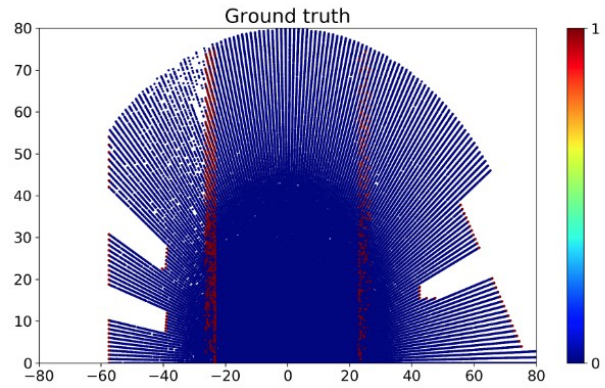


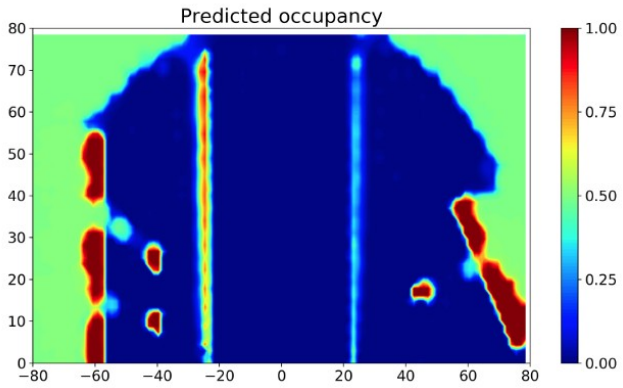
Problem description

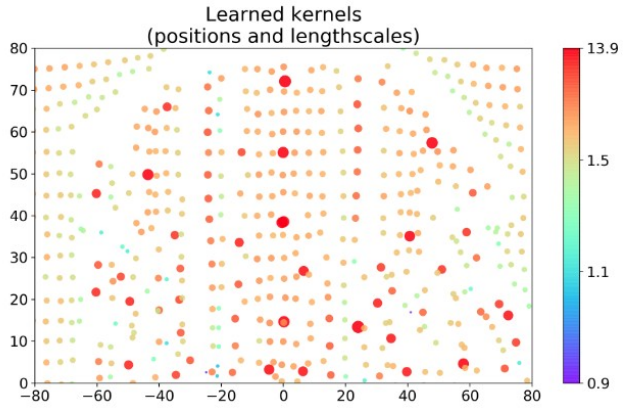
Laser scans

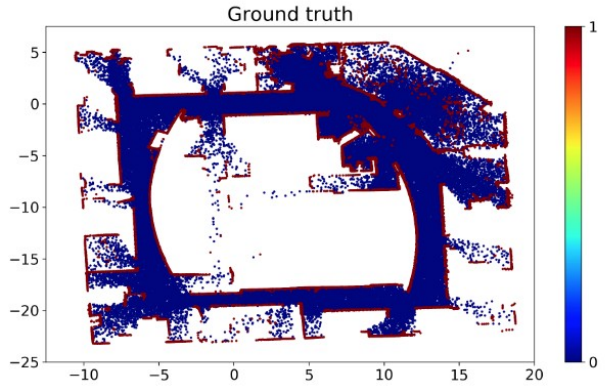
Probabilistic map

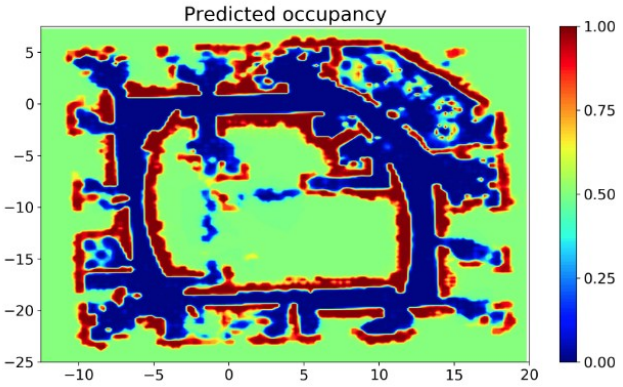
Hyperparameters

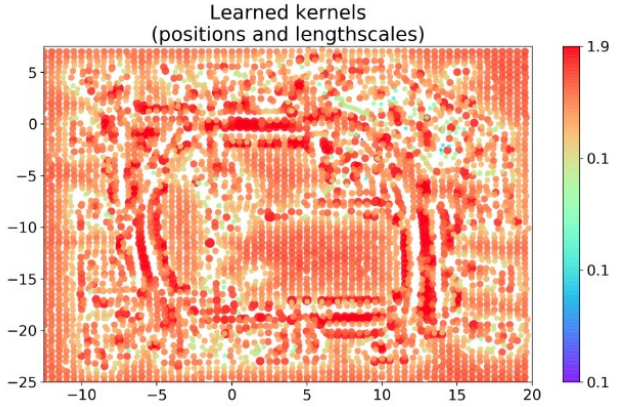












THE UNIVERSITY OF SYDNEY

Problem description

$$\underbrace{\prod_{m=1}^{M} q(w_m)q(l_m^{\text{lon}})q(l_m^{\text{lat}})q(\tilde{\mathbf{x}}_m)}_{\text{factorized variational distribution}} = \underbrace{q(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\substack{\text{variational}\\ \text{distribution}}} \approx \underbrace{p(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}}|\mathbf{x},\mathbf{y})}_{\text{posterior}} \propto \underbrace{p(\mathbf{w})p(\mathbf{l})p(\tilde{\mathbf{x}})}_{\text{priors}} \underbrace{p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\text{likelihood}}.$$

THE UNIVERSITY OF SYDNEY

Problem description

$$\underbrace{\prod_{m=1}^{M} q(w_m)q(l_m^{\text{lon}})q(l_m^{\text{lat}})q(\tilde{\mathbf{x}}_m)}_{\text{factorized variational distribution}} = \underbrace{q(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\substack{\text{variational}\\ \text{distribution}}} \approx \underbrace{p(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}}|\mathbf{x},\mathbf{y})}_{\text{posterior}} \propto \underbrace{p(\mathbf{w})p(\mathbf{l})p(\tilde{\mathbf{x}})}_{\text{priors}} \underbrace{p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\text{likelihood}}.$$

- Our learned model has an approximate variational distribution q
- q has tens of thousands of hyperparameters
- Learning q is a **time consuming process** taking **hours per dataset**

THE UNIVERSITY OF SYDNEY

Problem description

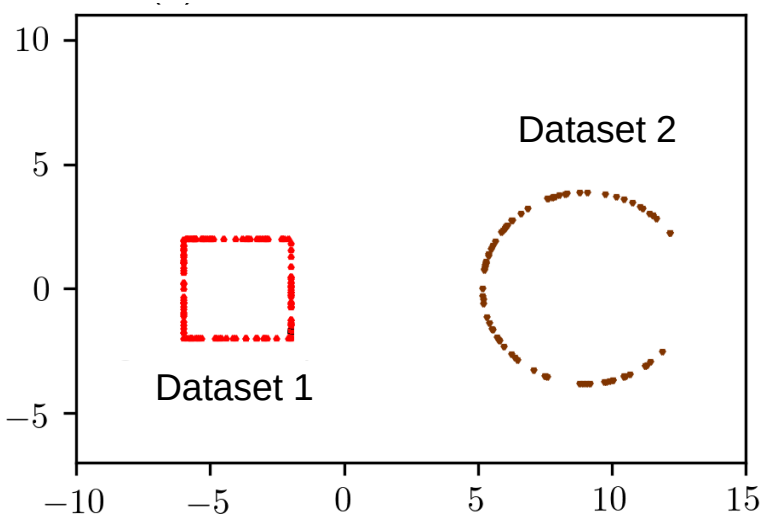
$$\underbrace{\prod_{m=1}^{M} q(w_m)q(l_m^{\text{lon}})q(l_m^{\text{lat}})q(\tilde{\mathbf{x}}_m)}_{\text{factorized variational distribution}} = \underbrace{q(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\substack{\text{variational}\\ \text{distribution}}} \approx \underbrace{p(\mathbf{w},\mathbf{l},\tilde{\mathbf{x}}|\mathbf{x},\mathbf{y})}_{\text{posterior}} \propto \underbrace{p(\mathbf{w})p(\mathbf{l})p(\tilde{\mathbf{x}})}_{\text{priors}} \underbrace{p(\mathbf{y}|\mathbf{x},\mathbf{w},\mathbf{l},\tilde{\mathbf{x}})}_{\text{likelihood}}.$$

- Our learned model has an approximate variational distribution q
- q has tens of thousands of hyperparameters
- Learning q is a **time consuming process** taking **hours per dataset**

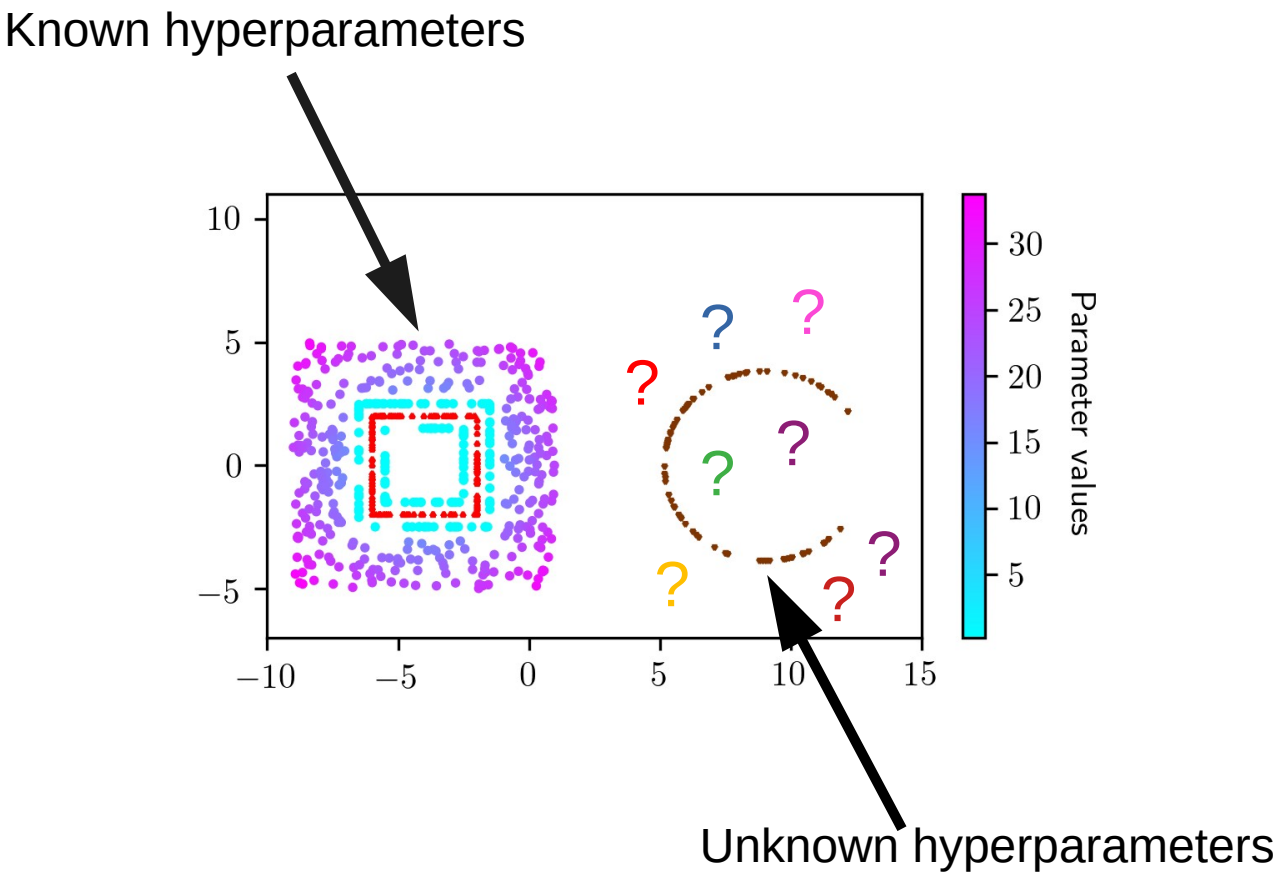
We can transport pre-learned hyperparameters of q in seconds

Method

Consider two laser scan datasets

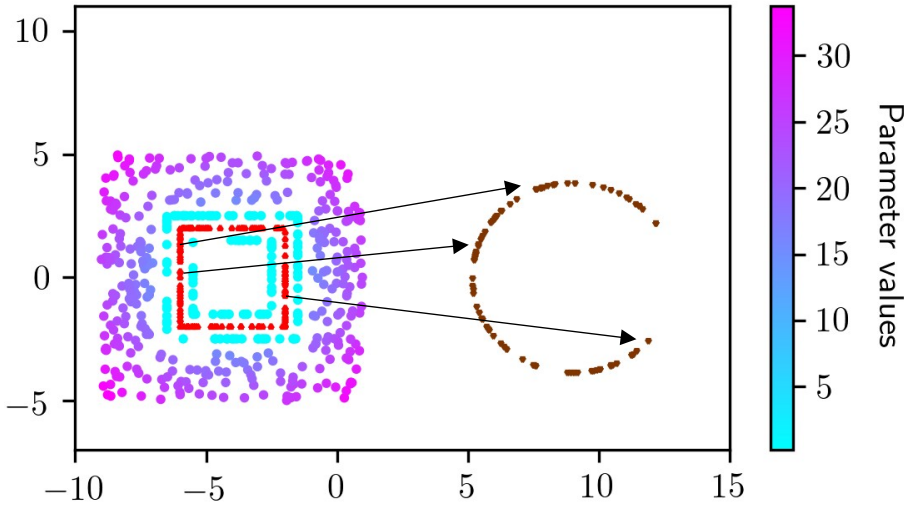


Method


Known hyperparameters
10
5
0
−5
−10
−5
0
5
10
15
?
?
?
?
?
?
?
?
30
25
20
15
10
5
Parameter values
Unknown hyperparameters

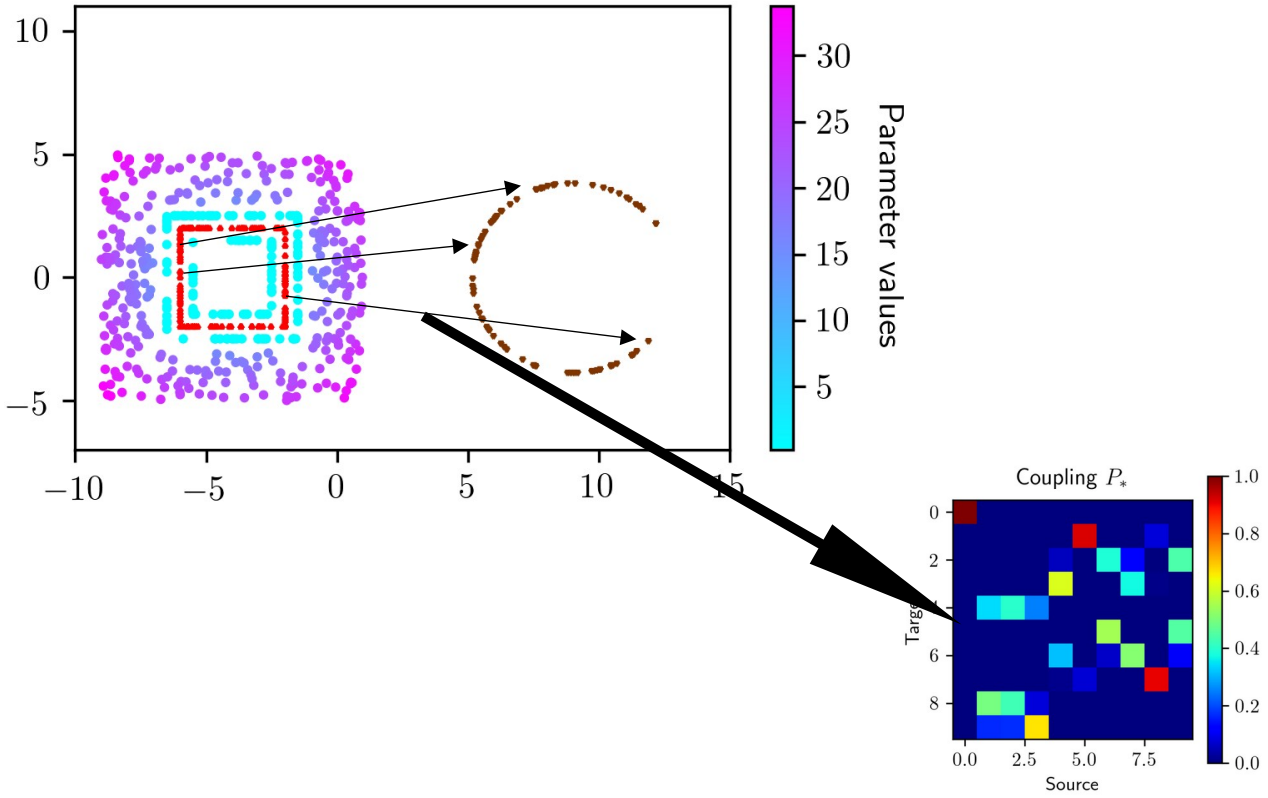
Our approach

We propose to use the theory of **optimal transport** to **transfer posterior distributions** learned in one domain, to a new model in a new domain



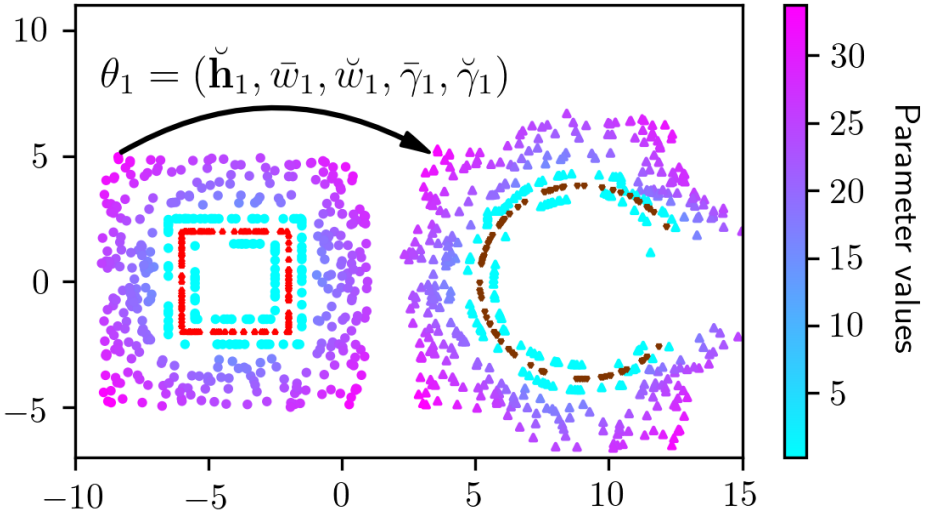
Our approach

We propose to use the theory of **optimal transport** to **transfer posterior distributions** learned in one domain, to a new model in a new domain

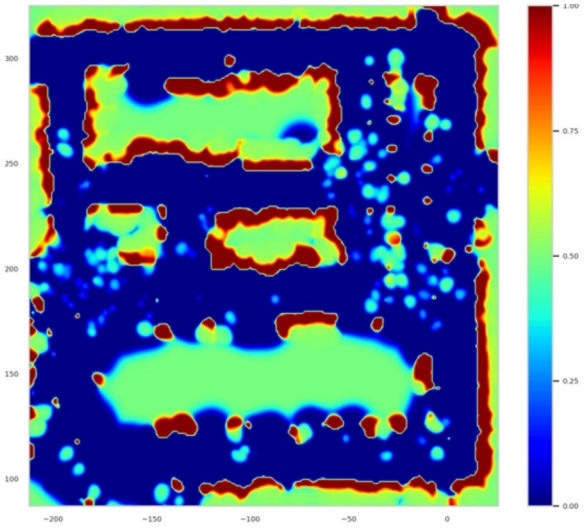


Method

Solution: Transfer the parameter distributions with the coupling information obtained via optimal transport



What kind of maps can POT-HM produce?



Method	Town1
RePOT	0.95
POT	0.85
ABHM	0.77
BHM	0.66