Robust and Conjugate Gaussian Process Regression

Dr François-Xavier Briol
Department of Statistical Science
University College London

Gaussian process regression

- **Regression problem:** Let $f : \mathscr{X} \to \mathbb{R}$ be some unknown function of interest. we have access to data $\{x_i, y_i\}_{i=1}^n$ where:

$$y_i = f(x_i) + \epsilon_i$$

- Two main assumptions:

$$f \sim GP(m, k)$$

"Prior"

$$\epsilon_i \sim N(0, \sigma^2)$$

"Likelihood/ Observation Model"

Why Gaussian processes?

Why Gaussian processes?

1. A very **flexible and interpretable model** through the choice of prior mean function m and covariance k function (e.g. smoothness, periodicity, sparsity, etc…).

Why Gaussian processes?

1. A very **flexible and interpretable model** through the choice of prior mean function m and covariance k function (e.g. smoothness, periodicity, sparsity, etc…).

2. We get a posterior on f which quantifies **epistemic uncertainty**.

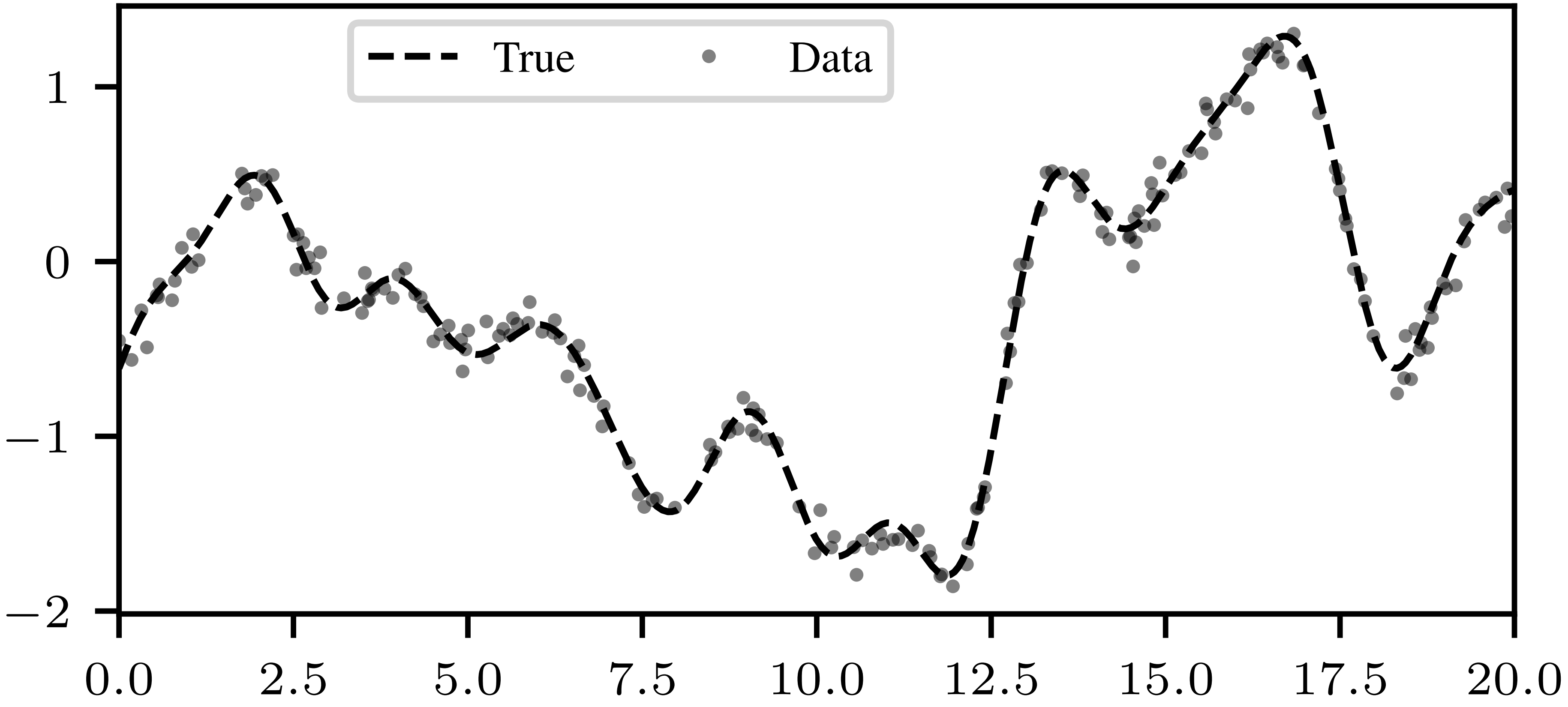
Why Gaussian processes?

1. A very **flexible and interpretable model** through the choice of prior mean function m and covariance k function (e.g. smoothness, periodicity, sparsity, etc…).

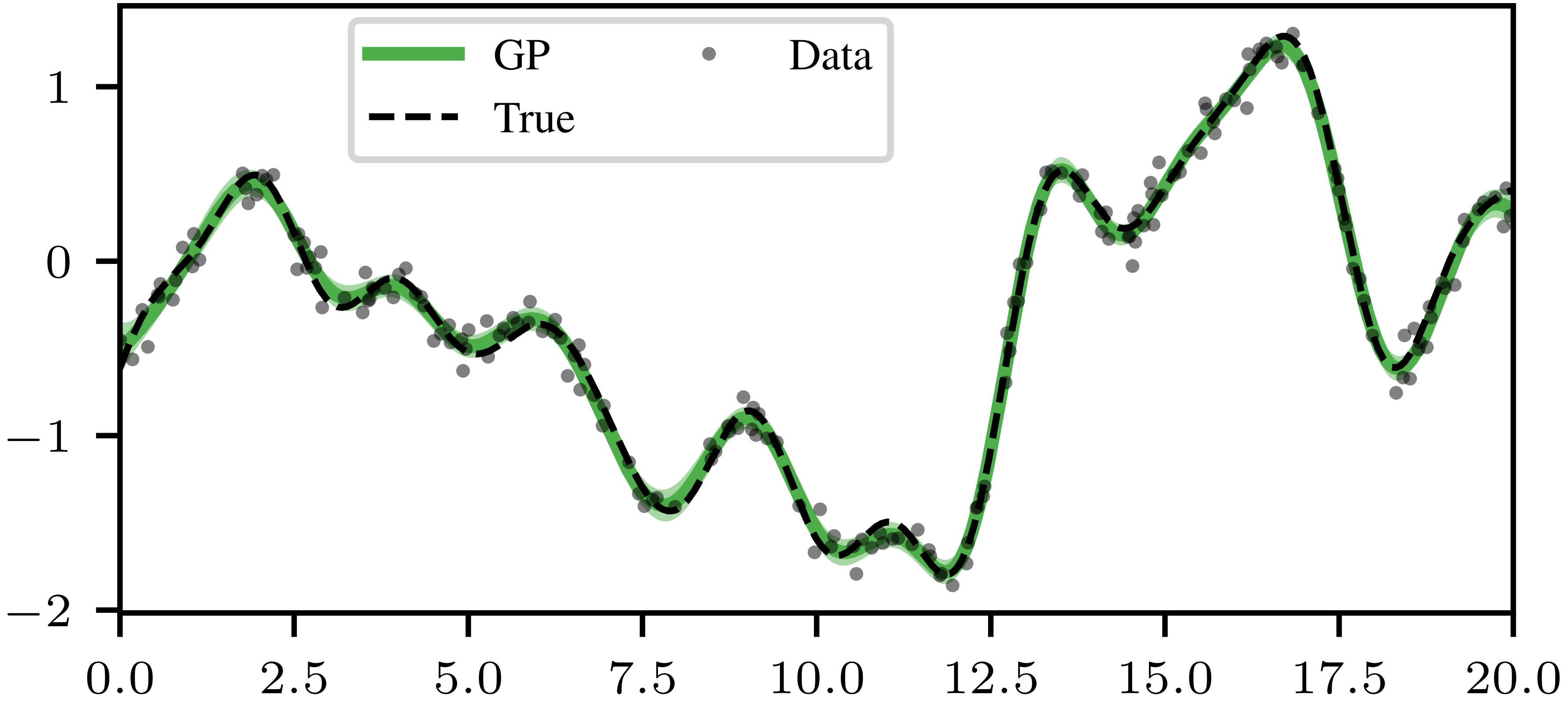
2. We get a posterior on f which quantifies **epistemic uncertainty**.

3. We can do **exact conditioning** through Gaussian conjugacy! We therefore don't need to do any approximation of the posterior!

A synthetic problem

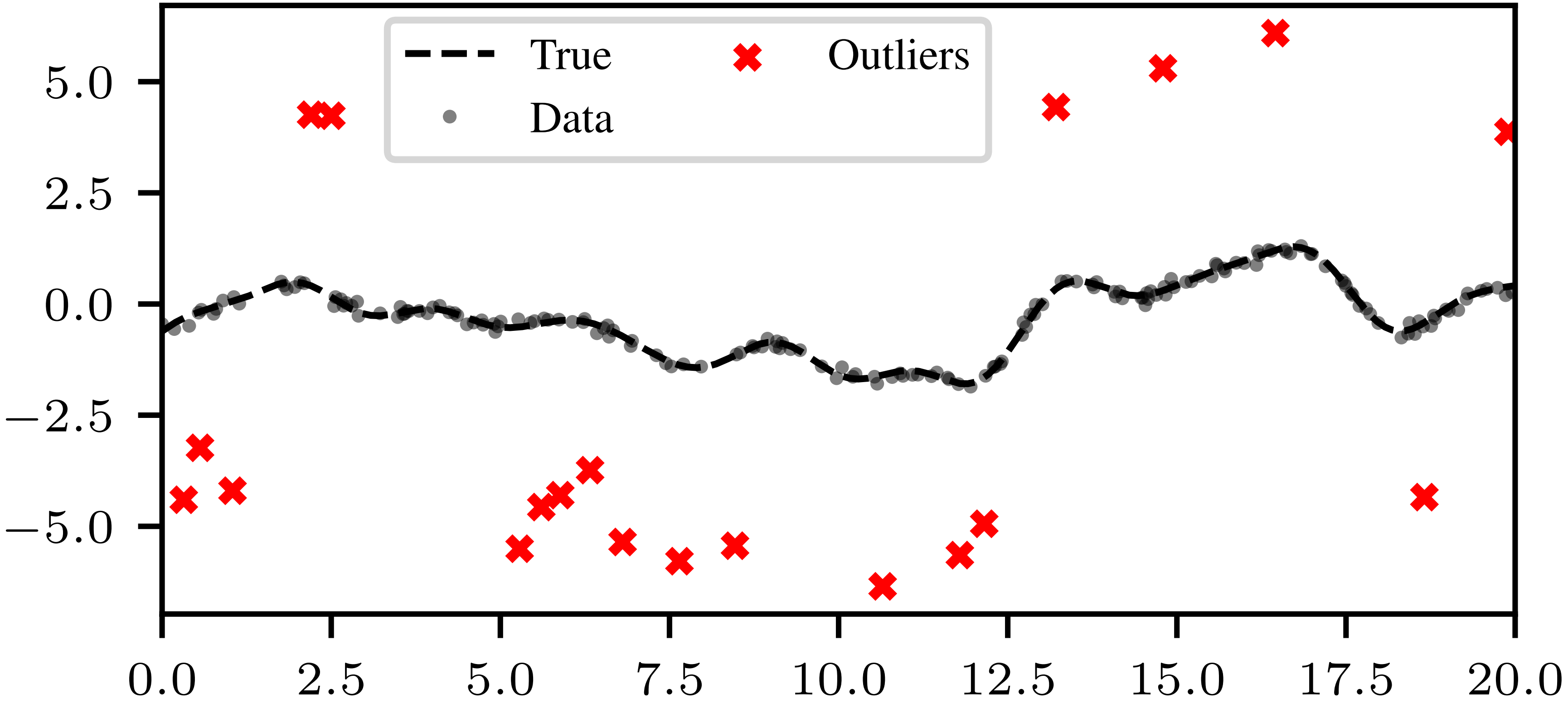


GP regression on the synthetic problem



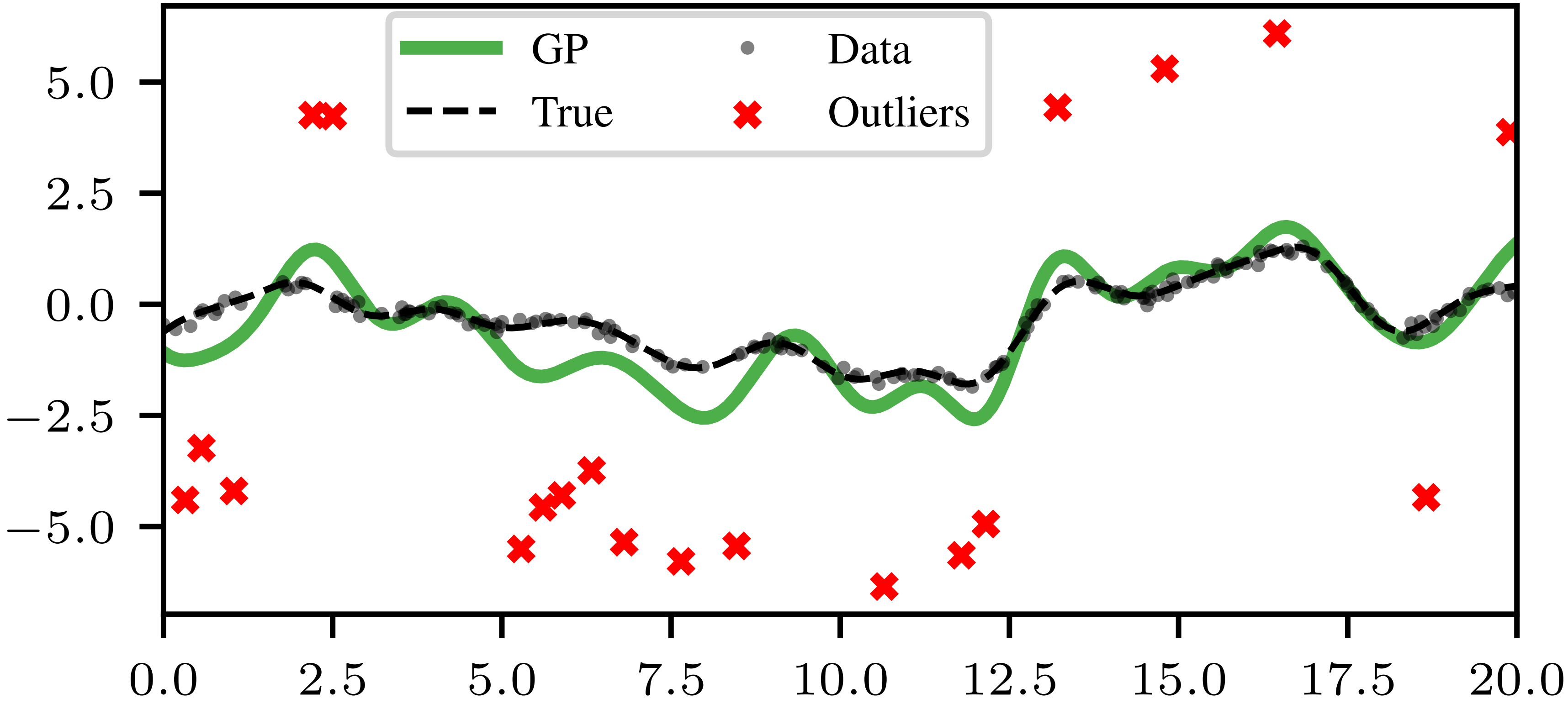
[I am being a bad Bayesian by plotting only the mean… sorry….]

Regression in the “real world”



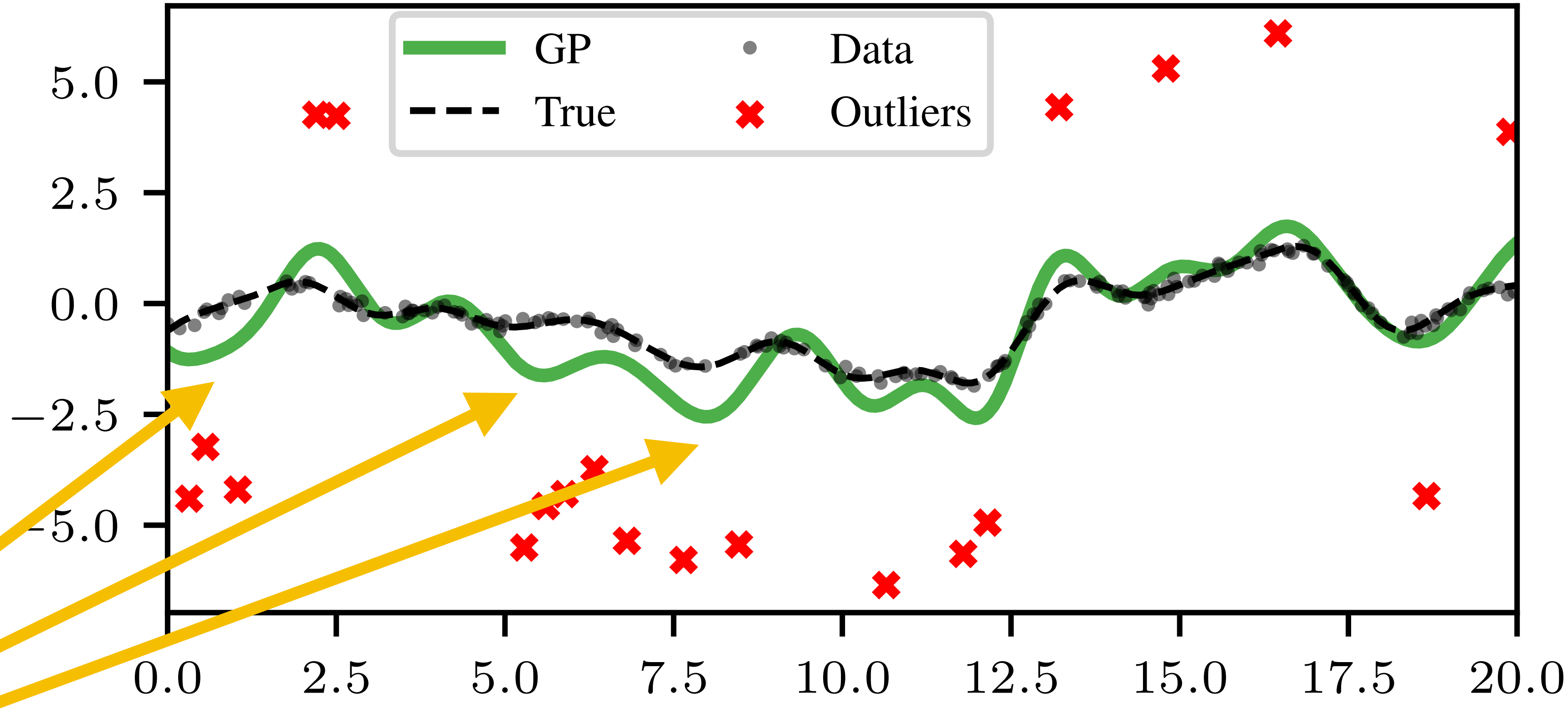
$\epsilon_i \sim \cancel{N}(0,\sigma^2)$

GP regression in the "real world"



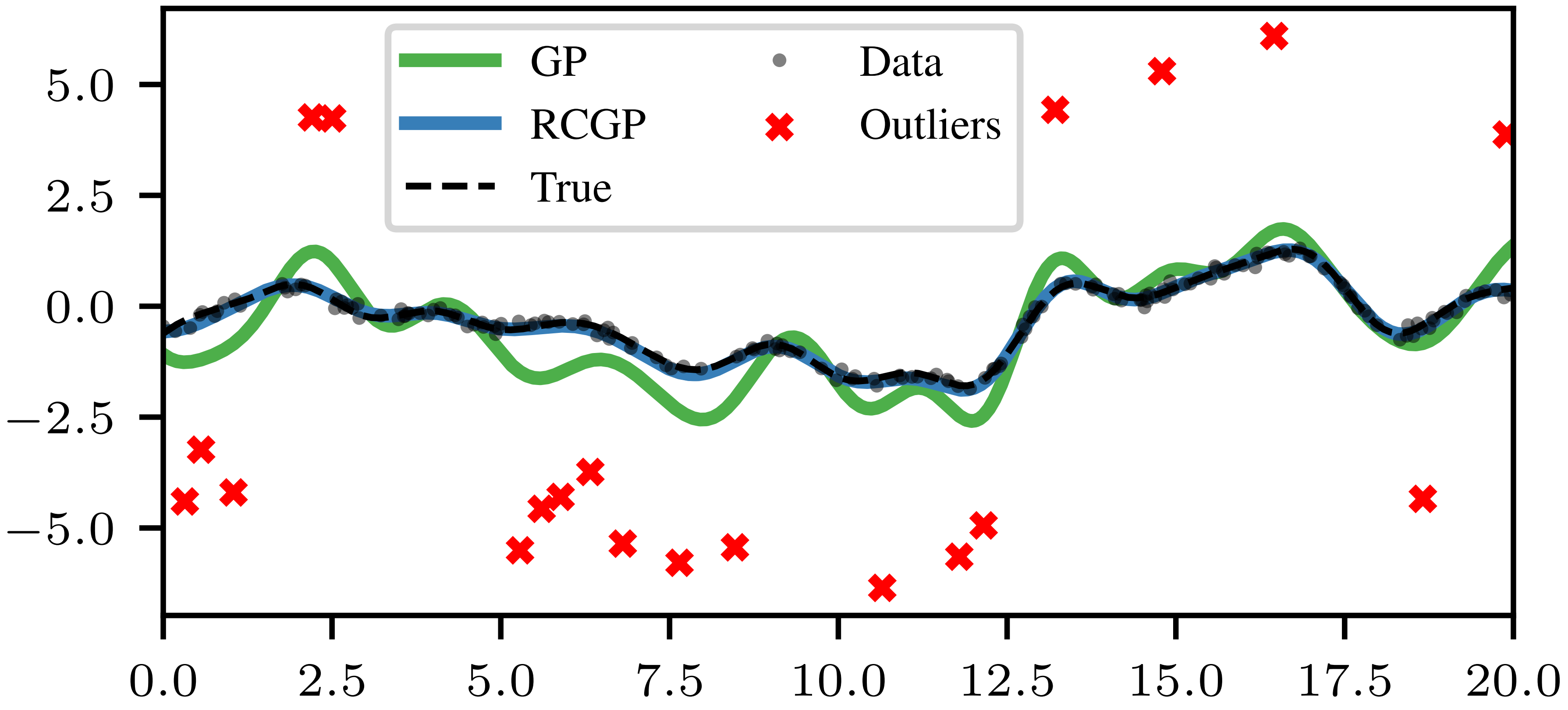
We assumed $\epsilon_i \sim N(0,\sigma^2)$ but its wrong…

GP regression in the "real world"



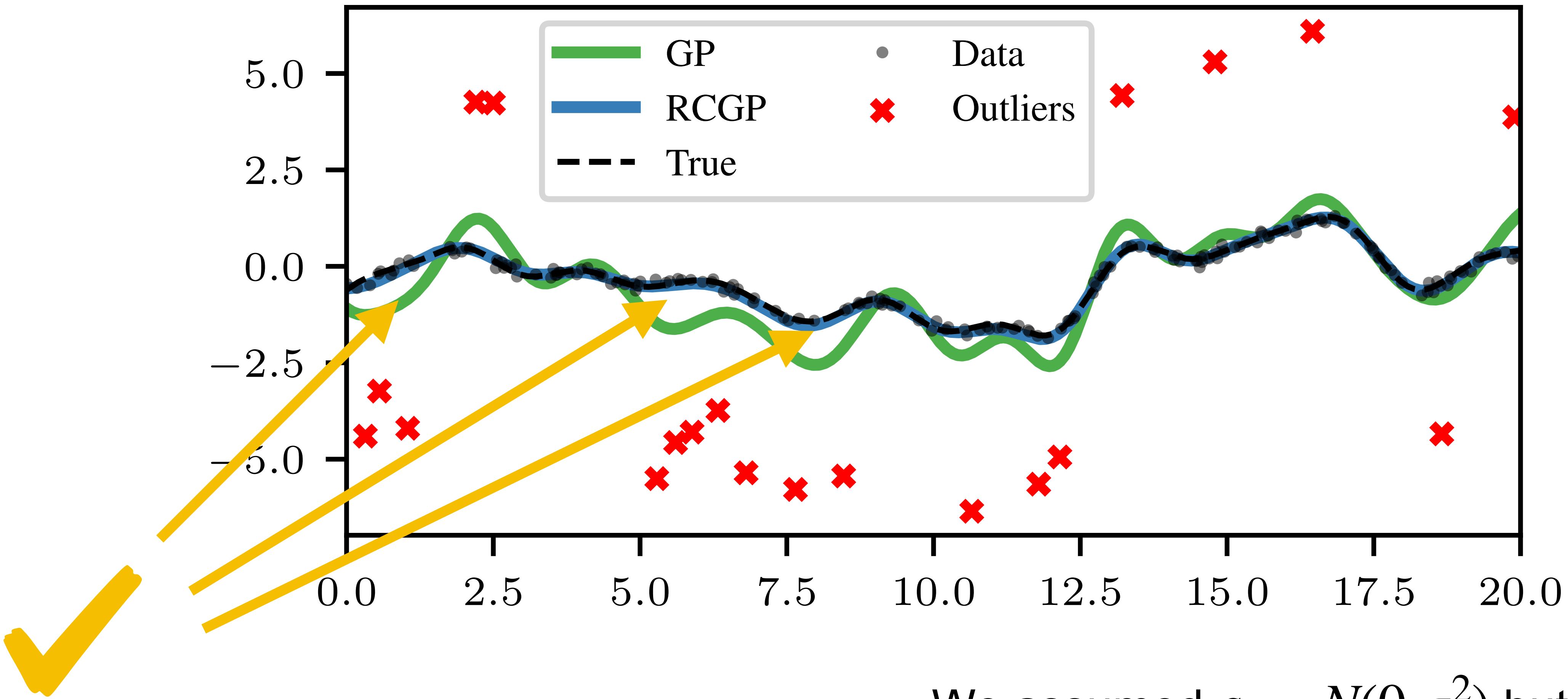
We assumed $\epsilon_i \sim N(0,\sigma^2)$ but its wrong…

Our goal: robust GP regression



We assumed $\epsilon_i \sim N(0,\sigma^2)$ but its wrong…

Our goal: robust GP regression



We assumed $\epsilon_i \sim N(0,\sigma^2)$ but its wrong…

Existing work

Existing work

Gaussian process regression with Student-t likelihood

Jarno Vanhatalo
Department of Biomedical Engineering and Computational Science
Helsinki University of Technology
Finland
jarno.vanhatalo@tkk.fi

Pasi Jylänki
Department of Biomedical Engineering and Computational Science
Helsinki University of Technology
Finland
pasi.jylanki@tkk.fi

Aki Vehtari
Department of Biomedical Engineering and Computational Science
Finland
Helsinki University of Technology
aki.vehtari@tkk.fi

IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. 55, NO. 9, SEPTEMBER 2008

Gaussian Process Robust Regression for Noisy Heart Rate Data

Oliver Stegle*, Sebastian V. Fallert, David J. C. MacKay, and Søren Brage

Corruption-Tolerant Gaussian Process Bandit Optimization

Ilija Bogunovic
ETH Zürich

Andreas Krause
ETH Zürich

Jonathan Scarlett
National University of Singapore

Identification of robust Gaussian Process Regression with noisy input using EM algorithm

Atefeh Daemi, Yousef Alipouri, Biao Huang *
Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta, T6G 1H9, Canada

Robust Gaussian process modeling using EM algorithm

Rishik Ranjan[a], Biao Huang[a,*], Alireza Fatehi[a,b]
[a] *Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Alberta, Canada T6G 2G6*
[b] *APAC Research Group, Industrial Control Center of Excellence, Faculty of Electrical Engineering, K.N. Toosi University of Technology, Tehran 16317-14191, Iran*

Robust Bayesian Optimization with Student-t Likelihood

Ruben Martinez-Cantin RUBEN@SIGOPT.COM
SigOpt Inc.
Centro Universitario de la Defensa, Zaragoza
Michael McCourt MCCOURT@SIGOPT.COM
Kevin Tee KEVIN@SIGOPT.COM
SigOpt Inc.

Robust Gaussian Process Regression with a Bias Model

Chiwoo Park CPARK5@FSU.EDU
Department of Industrial and Manufacturing Engineering
Florida State University
Tallahassee, FL 32310, USA

Robust and Scalable Gaussian Process Regression and Its Applications

Yifan Lu[1], Jiayi Ma[1*], Leyuan Fang[2], Xin Tian[1], and Junjun Jiang[3]
[1] Wuhan University, China [2] Hunan University, China [3] Harbin Institute of Technology, China
{lyf048, xin.tian}@whu.edu.cn, {jyma2010, fangleyuan}@gmail.com, jiangjunjun@hit.edu.

Robust Regression with Twinned Gaussian Processes

Andrew Naish-Guzman & Sean Holden
Computer Laboratory
University of Cambridge
Cambridge, CB3 0FD. United Kingdom
{agpn2,sbh11}@cl.cam.ac.uk

ROBUST GAUSSIAN PROCESS REGRESSION WITH HUBER LIKELIHOOD
*

BY POOJA ALGIKAR[1,a], LAMINE MILI[2,b]

Robust Gaussian Process Regression with the Trimmed Marginal Likelihood

Daniel Andrade[1] **Akiko Takeda**[2,3]

Robust Gaussian process regression with G-confluent likelihood

Martin Lindfors *,** Tianshi Chen ** Christian A. Naesseth ***

Robust Gaussian process regression based on iterative trimming

Zhao-Zhou Li [a,*], Lu Li [b,c], Zhengyi Shao [b,d]

Existing work

- There are two main categories:

1. **Extended models:** i.e. use more flexible likelihood model to ensure that the outliers are well modelled. Examples include Student-t, mixtures, Laplace, etc…

$$\epsilon \sim P \neq N(0, \sigma^2)$$

2. **Outlier detection/removal:** i.e. find the outliers, remove them, then fit a standard GP model (with Gaussian observations) to the rest of the data.

Issues with existing work

- The main issue with all of the methods above is that they are **very slow**!
- This is because they all **break Gaussian conjugacy** and so we must resort to approximate methods such as MCMC, Laplace or Variational Bayes.

Issues with existing work

- The main issue with all of the methods above is that they are **very slow**!
- This is because they all **break Gaussian conjugacy** and so we must resort to approximate methods such as MCMC, Laplace or Variational Bayes.

	GP	t-GP	m-GP	
Synthetic	1.5 (0.1)	2.2 (0.0)	3.0 (0.0)	$n = 300, d = 1$
Boston	1.9 (0.5)	30.7 (6.1)	16.7 (1.7)	$n = 506, d = 13$
Energy	3.8 (0.9)	34.0 (11)	33.8 (0.3)	$n = 768, d = 8$
Yacht	1.6 (0.3)	5.6 (0.7)	4.5 (0.4)	$n = 308, d = 6$

Table: Fitting time in second, including time for hyper parameter optimisation.

Issues with existing work

- The main issue with all of the methods above is that they are **very slow**!
- This is because they all **break Gaussian conjugacy** and so we must resort to approximate methods such as MCMC, Laplace or Variational Bayes.

	GP	t-GP	m-GP	
Synthetic	1.5 (0.1)	2.2 (0.0)	3.0 (0.0)	$n = 300, d = 1$
Boston	1.9 (0.5)	30.7 (6.1)	16.7 (1.7)	$n = 506, d = 13$
Energy	3.8 (0.9)	34.0 (11)	33.8 (0.3)	$n = 768, d = 8$
Yacht	1.6 (0.3)	5.6 (0.7)	4.5 (0.4)	$n = 308, d = 6$

Table: Fitting time in second, including time for hyper parameter optimisation.

Being Gaussian for convenience…

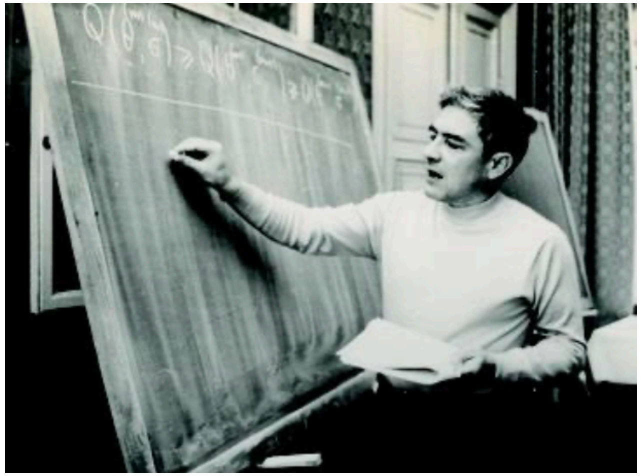
I would argue most practitioners just ignore that they have a misspecified likelihood and run with it anyway!

Being Gaussian for convenience…

I would argue most practitioners just ignore that they have a misspecified likelihood and run with it anyway!

"Gauss was fully aware that his main reason for assuming an underlying normal distribution […] was mathematical, **i.e. computational, convenience**"

"This raises a question which could have been asked by Gauss […] **What happens if the true distribution deviates slightly from the assumed normal one?**"



Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, *35*(1), 73–101.

This talk:

Robust and Conjugate Gaussian Process Regression

Matias Altamirano [1] François-Xavier Briol [1] Jeremias Knoblauch [1]

Appeared as a **spotlight paper** (top 3% of papers) at **ICML 2024**!

Bayesian inference for regression

- In standard GP regression, we do:

$$p(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) \propto p(\mathbf{y} \mid \mathbf{f}, \mathbf{x}) \times p(\mathbf{f} \mid \mathbf{x})$$

$$\underbrace{p(\mathbf{f} \mid \mathbf{y}, \mathbf{x})}_{\text{Posterior}} \propto \underbrace{p(\mathbf{y} \mid \mathbf{f}, \mathbf{x})}_{\text{Likelihood}} \times \underbrace{p(\mathbf{f} \mid \mathbf{x})}_{\text{Prior}}$$

$$\mathbf{x} = (x_1, \ldots, x_n)^\top$$
$$\mathbf{f} = (f(x_1), \ldots, f(x_n))^\top$$
$$\mathbf{y} = (y_1, \ldots, y_n)^\top$$

Generalised Bayesian inference for regression

- In standard GP regression, we do:

Posterior Likelihood Prior

$$p(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) \propto p(\mathbf{y} \mid \mathbf{f}, \mathbf{x}) \times p(\mathbf{f} \mid \mathbf{x})$$

- We take a generalised Bayesian approach and do:

$$p^L(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \mid \mathbf{x})$$

Generalised Posterior

Loss function

Prior

Standard vs Generalised Bayesian inference

$$p^L(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \,|\, \mathbf{x})$$

- Standard Bayes is recovered by taking

$$L_n(\mathbf{f}, \mathbf{y}, \mathbf{x}) = -\frac{1}{n} \log p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{x})$$

Standard vs Generalised Bayesian inference

$$p^L(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \,|\, \mathbf{x})$$

- Standard Bayes is recovered by taking

$$L_n(\mathbf{f}, \mathbf{y}, \mathbf{x}) = -\frac{1}{n} \log p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{x})$$

- This is **optimal**, but **only when the model is well-specified**; i.e. when $\epsilon \sim N(0,\sigma^2)$!

Standard vs Generalised Bayesian inference

$$p^L(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \,|\, \mathbf{x})$$

- Standard Bayes is recovered by taking

$$L_n(\mathbf{f}, \mathbf{y}, \mathbf{x}) = -\frac{1}{n} \log p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{x})$$

- This is **optimal**, but **only when the model is well-specified**; i.e. when $\epsilon \sim N(0,\sigma^2)$!

→ **Key Question:** What should we do when this is not the case??

Generalised Bayesian inference

$$p^L(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \,|\, \mathbf{x})$$

Bissiri, P., Holmes, C., & Walker, S. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *78*, 1103–1130.

Knoblauch, J., Jewson, J., & Damoulas, T. (2022). An optimization-centric view on Bayes' rule: reviewing and generalizing variational inference. *Journal of Machine Learning Research*, *23*(132), 1–109.

Generalised Bayesian inference

$$p^L(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \,|\, \mathbf{x})$$

- We can choose the loss function to induce **robustness to mild model misspecification**.

Bissiri, P., Holmes, C., & Walker, S. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *78*, 1103–1130.

Knoblauch, J., Jewson, J., & Damoulas, T. (2022). An optimization-centric view on Bayes' rule: reviewing and generalizing variational inference. *Journal of Machine Learning Research*, *23*(132), 1–109.

Generalised Bayesian inference

$$p^L(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \,|\, \mathbf{x})$$

- We can choose the loss function to induce **robustness to mild model misspecification**.
- Common choice is a loss based on a divergence:

$$\mathscr{D}\left(\frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}, p_f\right)$$

Data-generating process; here a Gaussian

Bissiri, P., Holmes, C., & Walker, S. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *78*, 1103–1130.

Knoblauch, J., Jewson, J., & Damoulas, T. (2022). An optimization-centric view on Bayes' rule: reviewing and generalizing variational inference. *Journal of Machine Learning Research*, *23*(132), 1–109.

Generalised Bayesian inference

$$p^L(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) \propto \exp\left(-nL_n(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \times p(\mathbf{f} \mid \mathbf{x})$$

- We can choose the loss function to induce **robustness to mild model misspecification**.
- Common choice is a loss based on a divergence:

$$\mathcal{D}\left(\frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}, p_f\right)$$

Data-generating process; here a Gaussian

- In this talk, we will also choose the loss function for computational convenience!

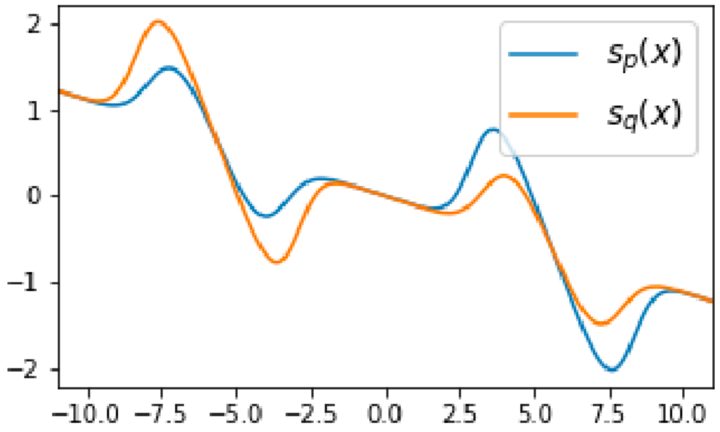
Bissiri, P., Holmes, C., & Walker, S. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *78*, 1103–1130.

Knoblauch, J., Jewson, J., & Damoulas, T. (2022). An optimization-centric view on Bayes' rule: reviewing and generalizing variational inference. *Journal of Machine Learning Research*, *23*(132), 1–109.

Score-matching and generalisations

- The score-matching divergence is given by:

$$D(p\,||\,q) := \mathbb{E}_{Y\sim q}[\|\nabla_y \log p(Y) - \nabla_y \log q(Y)\|_2^2]$$



[1] Hyvärinen, A. (2006). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, *6*, 695–708.

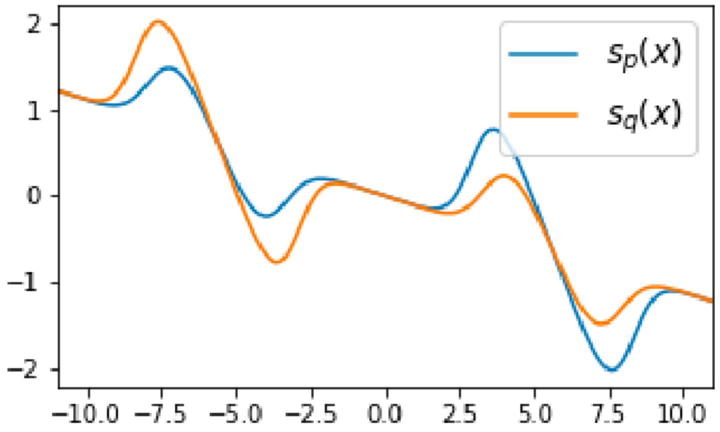
Score-matching and generalisations

- The score-matching divergence is given by:

$$D(p\,||\,q) := \mathbb{E}_{Y\sim q}[\|\nabla_y \log p(Y) - \nabla_y \log q(Y)\|_2^2]$$

- We consider a weighted generalisation:

$$D(p\,||\,q) := \mathbb{E}_{Y\sim q}[\|w(Y)(\nabla_y \log p(Y) - \nabla_y \log q(Y))\|_2^2]$$



[1] Hyvärinen, A. (2006). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, *6*, 695–708.

[2] Barp, A., Briol, F.-X., Duncan, A. B., Girolami, M., & Mackey, L. (2019). Minimum Stein discrepancy estimators. *Neural Information Processing Systems*, 12964–12976.

Score-matching and generalisations

- For regression setting, we need to extend this divergence (now $w : \mathcal{X} \times \mathbb{R} \to \mathbb{R}$):

$$D(p\,||\,q) := \mathbb{E}_{X \sim q_x}\left[\mathbb{E}_{Y \sim q(\cdot|X)}\left[\left\| w(X,Y)(\nabla_y \log p(Y|X) - \nabla_y \log q(Y|X)) \right\|_2^2\right]\right]$$

Score-matching and generalisations

- For regression setting, we need to extend this divergence (now $w : \mathcal{X} \times \mathbb{R} \to \mathbb{R}$):

$$D(p\,||\,q) := \mathbb{E}_{X \sim q_x}\left[\mathbb{E}_{Y \sim q(\cdot|X)}\left[\| w(X, Y)(\nabla_y \log p(Y|X) - \nabla_y \log q(Y|X)) \|_2^2\right]\right]$$

- With integration by part and replacing q by our samples, we get that:

$$D(p\,||\,q_n) = L_n^w(\mathbf{f}, \mathbf{y}, \mathbf{x}) + C$$

$$= \frac{1}{n}\sum_{i=1}^{n}\Big((w(x_i, y_i)\nabla_y \log p(y_i|x_i))^2 + 2\nabla_y (w(x_i, y_i)^2 \nabla_y \log p(y_i|x_i))\Big) + C$$

Likelihood

RCGPs are conjugate!

- Suppose $f \sim GP(m, k)$ and $\epsilon \sim N(0,\sigma^2 I_n)$, then the GP and RCGP posteriors are:

<u>Standard GP</u>

$$p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu, \Sigma)$$

$$\mu = \mathbf{m} + K(K + \sigma^2 I_n)^{-1}(\mathbf{y} - \mathbf{m})$$

$$\Sigma = K(K + \sigma^2 I_n)^{-1}\sigma^2 I_n$$

$K_{ij} = k(x_i, x_j)$

Identity matrix

RCGPs are conjugate!

- Suppose $f \sim GP(m, k)$ and $\epsilon \sim N(0, \sigma^2 I_n)$, then the GP and RCGP posteriors are:

<u>Standard GP</u>

$$p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu, \Sigma)$$

$$\mu = \mathbf{m} + K(K + \sigma^2 I_n)^{-1}(\mathbf{y} - \mathbf{m})$$

$$\Sigma = K(K + \sigma^2 I_n)^{-1}\sigma^2 I_n$$

<u>RCGP</u>

$$p^w(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu^R, \Sigma^R)$$

$$\mu^R = \mathbf{m} + K(K + \sigma^2 J_{\mathbf{w}})^{-1}(\mathbf{y} - \mathbf{m}_{\mathbf{w}})$$

$$\Sigma^R = K(K + \sigma^2 J_{\mathbf{w}})^{-1}\sigma^2 J_{\mathbf{w}}$$

RCGPs are conjugate!

- Suppose $f \sim GP(m, k)$ and $\epsilon \sim N(0, \sigma^2 I_n)$, then the GP and RCGP posteriors are:

Standard GP

$$p(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu, \Sigma)$$

$$\mu = \mathbf{m} + K(K + \sigma^2 I_n)^{-1}(\mathbf{y} - \mathbf{m})$$

$$\Sigma = K(K + \sigma^2 I_n)^{-1} \sigma^2 I_n$$

RCGP

$$p^w(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu^R, \Sigma^R)$$

$$\mu^R = \mathbf{m} + K(K + \sigma^2 J_{\mathbf{w}})^{-1}(\mathbf{y} - \mathbf{m}_{\mathbf{w}})$$

$$\Sigma^R = K(K + \sigma^2 J_{\mathbf{w}})^{-1} \sigma^2 J_{\mathbf{w}}$$

$$J_{\mathbf{w}} = \mathrm{diag}(\mathbf{w}^{-2}) \qquad \mathbf{m}_{\mathbf{w}} = \mathbf{m} + \sigma^2 \nabla_y \log(\mathbf{w}^2)$$

RCGPs generalise existing GPs

- Suppose $f \sim GP(m, k)$ and $\epsilon \sim N(0, \sigma^2 I_n)$, then the RCGP posterior is:

$$p^w(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu^R, \Sigma^R)$$

$$\mu^R = \mathbf{m} + K(K + \sigma^2 J_{\mathbf{w}})^{-1}(\mathbf{y} - \mathbf{m}_{\mathbf{w}})$$

$$\Sigma^R = K(K + \sigma^2 J_{\mathbf{w}})^{-1} \sigma^2 J_{\mathbf{w}}$$

$$J_{\mathbf{w}} = \text{diag}(\mathbf{w}^{-2}) \qquad \mathbf{m}_{\mathbf{w}} = \mathbf{m} + \sigma^2 \nabla_y \log(\mathbf{w}^2)$$

RCGPs generalise existing GPs

- Suppose $f \sim GP(m, k)$ and $\epsilon \sim N(0, \sigma^2 I_n)$, then the RCGP posterior is:

$$p^w(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu^R, \Sigma^R)$$

$$\mu^R = \mathbf{m} + K(K + \sigma^2 J_{\mathbf{w}})^{-1}(\mathbf{y} - \mathbf{m}_{\mathbf{w}})$$

$$\Sigma^R = K(K + \sigma^2 J_{\mathbf{w}})^{-1} \sigma^2 J_{\mathbf{w}}$$

$$J_{\mathbf{w}} = \mathrm{diag}(\mathbf{w}^{-2}) \qquad \mathbf{m}_{\mathbf{w}} = \mathbf{m} + \sigma^2 \nabla_y \log(\mathbf{w}^2)$$

→ Taking $w(x, y) = \sigma/\sqrt{2}$ **recovers standard GPs.**

RCGPs generalise existing GPs

- Suppose $f \sim GP(m, k)$ and $\epsilon \sim N(0, \sigma^2 I_n)$, then the RCGP posterior is:

$$p^w(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu^R, \Sigma^R)$$

$$\mu^R = \mathbf{m} + K(K + \sigma^2 J_{\mathbf{w}})^{-1}(\mathbf{y} - \mathbf{m}_{\mathbf{w}})$$

$$\Sigma^R = K(K + \sigma^2 J_{\mathbf{w}})^{-1} \sigma^2 J_{\mathbf{w}}$$

$$J_{\mathbf{w}} = \mathrm{diag}(\mathbf{w}^{-2}) \qquad \mathbf{m}_{\mathbf{w}} = \mathbf{m} + \sigma^2 \nabla_y \log(\mathbf{w}^2)$$

→ Taking $w(x, y) = \sigma/\sqrt{2}$ **recovers standard GPs.**

→ Taking $w(x, y) = \sigma(x)/\sqrt{2}$ **recovers heteroscedastic GPs.**

RCGPs generalise existing GPs

- Suppose $f \sim GP(m, k)$ and $\epsilon \sim N(0, \sigma^2 I_n)$, then the RCGP posterior is:

$$p^w(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) = N(\mathbf{f}; \mu^R, \Sigma^R)$$

$$\mu^R = \mathbf{m} + K(K + \sigma^2 J_{\mathbf{w}})^{-1}(\mathbf{y} - \mathbf{m}_{\mathbf{w}})$$

$$\Sigma^R = K(K + \sigma^2 J_{\mathbf{w}})^{-1} \sigma^2 J_{\mathbf{w}}$$

$$J_{\mathbf{w}} = \mathrm{diag}(\mathbf{w}^{-2}) \qquad \mathbf{m}_{\mathbf{w}} = \mathbf{m} + \sigma^2 \nabla_y \log(\mathbf{w}^2)$$

→ Taking $w(x, y) = \sigma/\sqrt{2}$ **recovers standard GPs.**

→ Taking $w(x, y) = \sigma(x)/\sqrt{2}$ **recovers heteroscedastic GPs.**

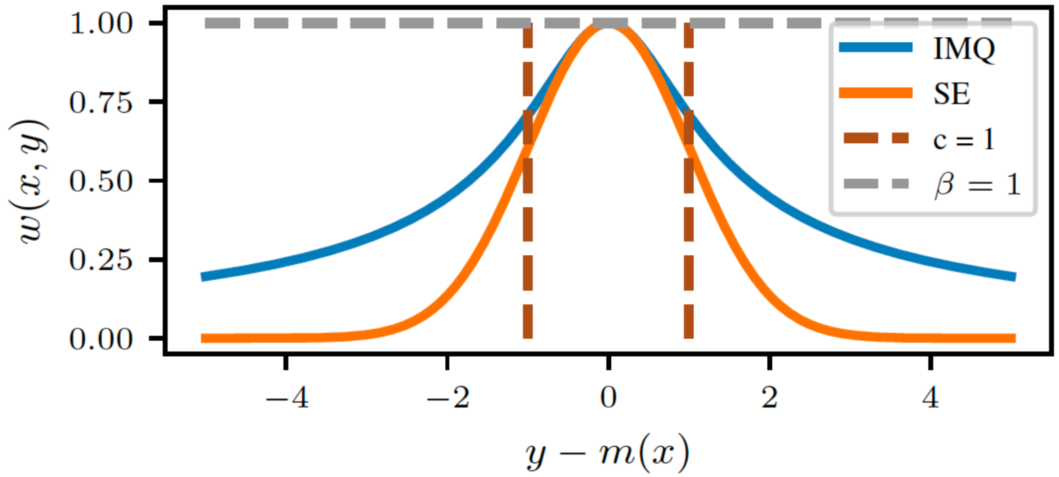
→ We will choose $w(x, y)$ differently to induce robustness….

Down-weighting outliers

$$w(x, y) = \left(1 + \frac{(y - m(x))^2}{c^2} \right)^{-\frac{1}{2}}$$

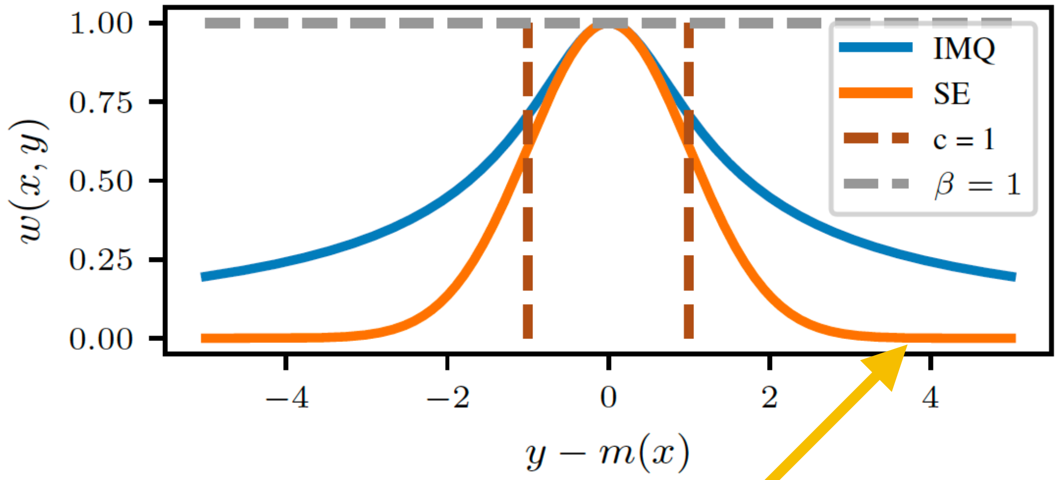
Down-weighting outliers

$$w(x, y) = \left(1 + \frac{(y - m(x))^2}{c^2}\right)^{-\frac{1}{2}}$$



Down-weighting outliers

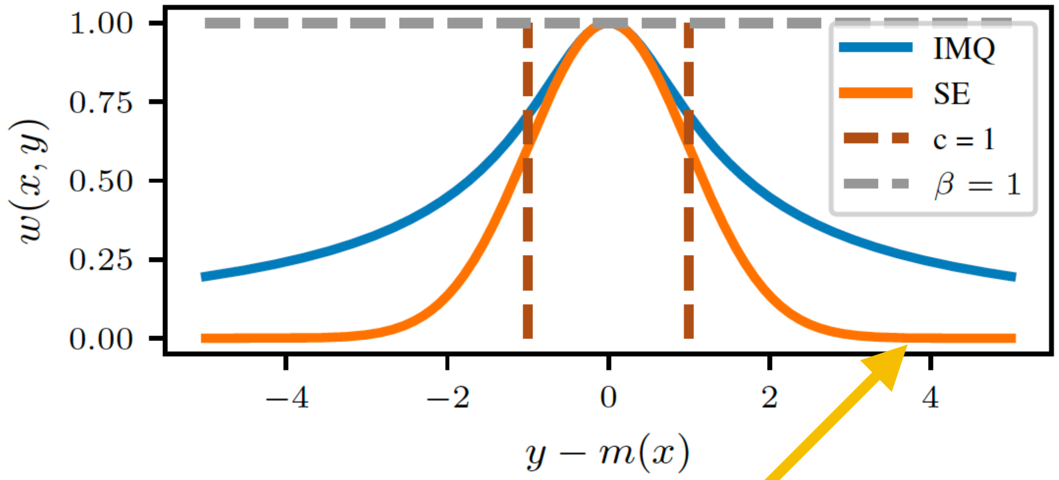
$$w(x, y) = \left(1 + \frac{(y - m(x))^2}{c^2}\right)^{-\frac{1}{2}}$$

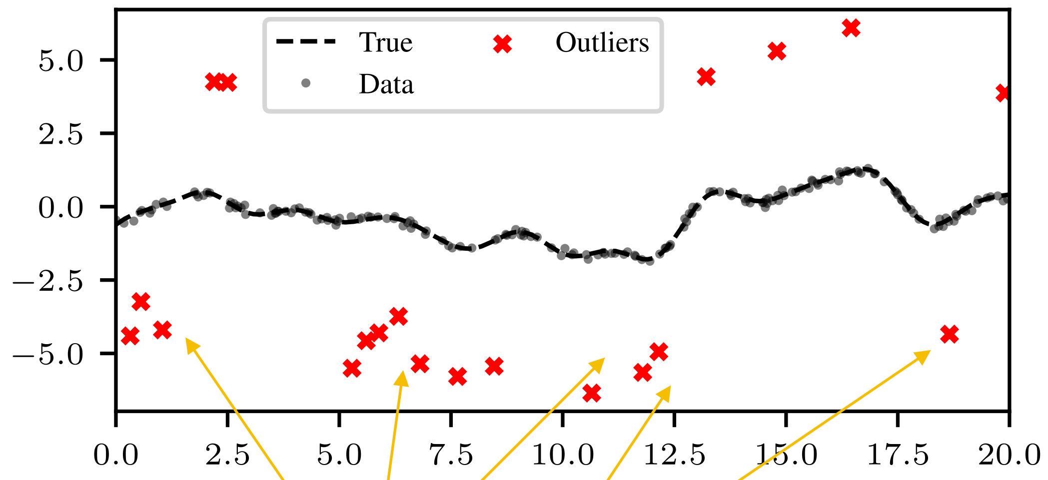


We down-weight extreme observations…but not too much…

Down-weighting outliers

$$w(x, y) = \left(1 + \frac{(y - m(x))^2}{c^2}\right)^{-\frac{1}{2}}$$





Low weights!

We down-weight extreme observations…but not too much…

Measuring outlier-robustness

- The posterior influence function measures the impact of a single outlier on the posterior:

$$\text{PIF}(y_m^c, D) = \text{KL}\left(p(f \mid D), p(f \mid D_m^c)\right)$$

$$D = \{x_i, y_i\}_{i=1}^n \qquad D_m^c = (D \setminus \{x_m, y_m\}) \cup \{x_m, y_m^c\}$$

Measuring outlier-robustness

- The posterior influence function measures the impact of a single outlier on the posterior:

$$\text{PIF}(y_m^c, D) = \text{KL}\left(p(f \,|\, D), p(f \,|\, D_m^c)\right)$$

$$D = \{x_i, y_i\}_{i=1}^n \qquad D_m^c = (D \backslash \{x_m, y_m\}) \cup \{x_m, y_m^c\}$$

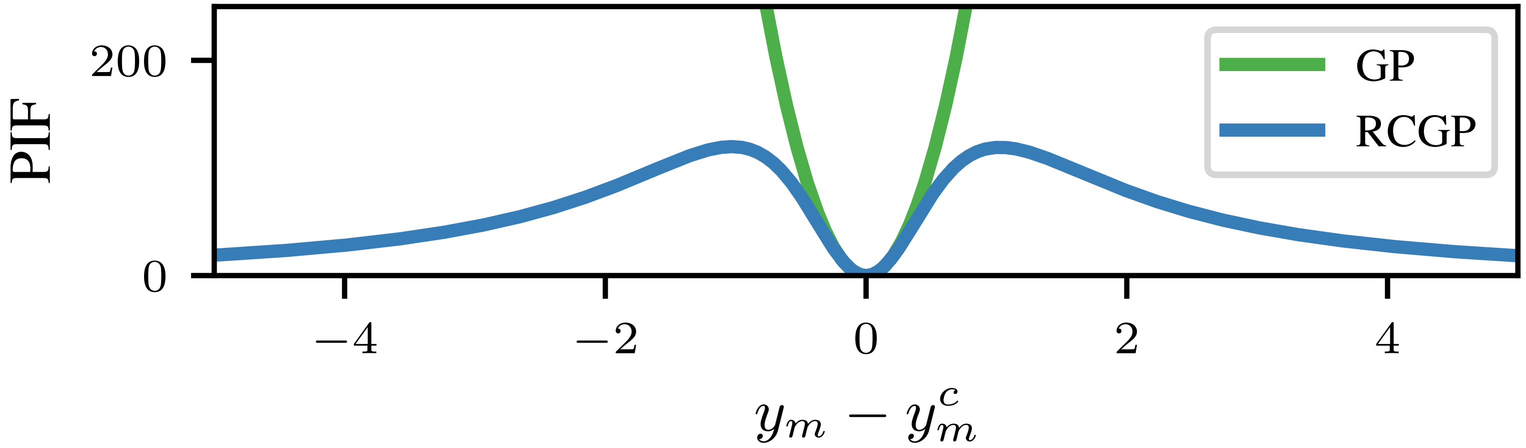
- Sadly…

$$\sup_{y_m^c} \text{PIF}_{\text{GP}}(y_m^c, D) = \infty$$

RCGPs are provably outlier-robust

- **Theorem (informal):** Suppose $w(x, y) = (1 + (y - m(x))^2/c^2)^{-\frac{1}{2}}$ for some $c > 0$, then RCGPs are robust since:

$$\sup_{y_m^c} \text{PIF}_{\text{RCGP}}(y_m^c, D) < \infty$$



Hyperparameter selection

- The standard approach for selecting hyper parameters is to do empirical Bayes and **maximise the marginal likelihood**.

Hyperparameter selection

- The standard approach for selecting hyper parameters is to do empirical Bayes and **maximise the marginal likelihood**.

- This of course does not make sense when the likelihood is wrong!

Hyperparameter selection

- The standard approach for selecting hyper parameters is to do empirical Bayes and **maximise the marginal likelihood**.
- This of course does not make sense when the likelihood is wrong!
- Our alternative is to do **leave-one-out cross-validation**

$$\hat{\sigma}^2, \hat{\theta} = \arg\max_{\sigma^2, \theta} \left\{ \sum_{i=1}^{n} \log p^w(y_i \,|\, \mathbf{x}, \mathbf{y}_{-i}, \theta, \sigma^2) \right\},$$

Hyperparameter selection

- The standard approach for selecting hyper parameters is to do empirical Bayes and **maximise the marginal likelihood**.

- This of course does not make sense when the likelihood is wrong!

- Our alternative is to do **leave-one-out cross-validation**

$$\hat{\sigma}^2, \hat{\theta} = \arg\max_{\sigma^2, \theta} \left\{ \sum_{i=1}^{n} \log p^w(y_i \,|\, \mathbf{x}, \mathbf{y}_{-i}, \theta, \sigma^2) \right\},$$

- This can be done efficiently through clever linear algebra tricks and gradient-based optimisation.

Performance when well-specified (MAE)

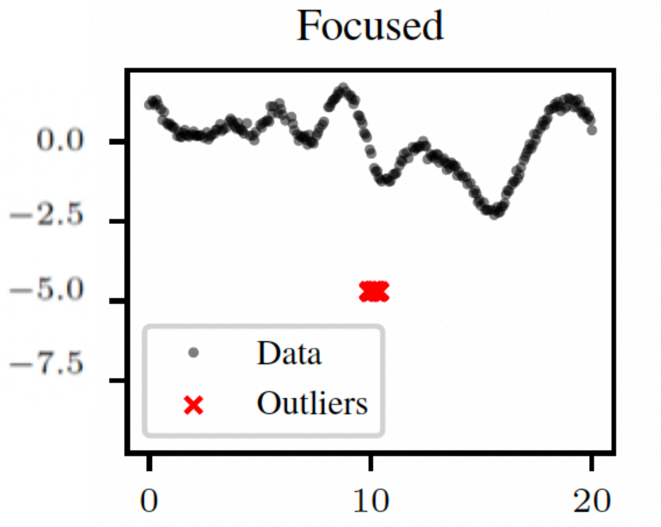
	GP	RCGP	t-GP	m-GP
		No Outliers		
Synthetic	**0.09 (0.00)**	**0.09 (0.00)**	**0.09 (0.00)**	0.33 (0.00)
Boston	**0.19 (0.01)**	**0.19 (0.01)**	**0.19 (0.01)**	0.28 (0.00)
Energy	0.03 (0.00)	**0.02 (0.00)**	0.03 (0.00)	0.61 (0.00)
Yacht	0.02 (0.01)	0.02 (0.01)	**0.01 (0.00)**	0.33 (0.00)

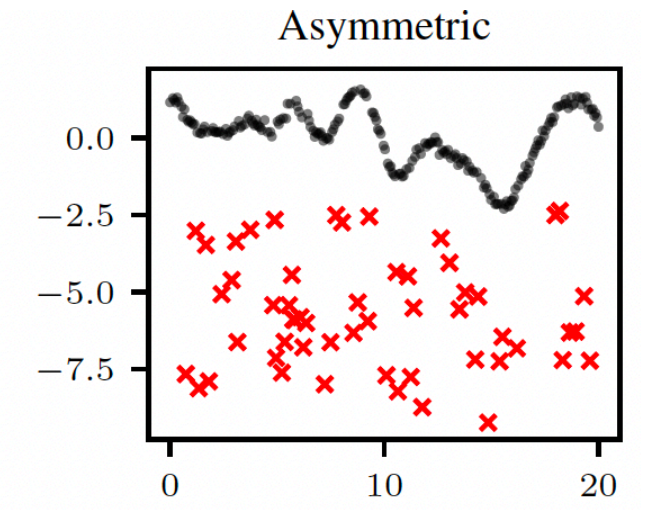
GPs and RCGPs are comparable when the model is well-specified!

Performance when misspecified (MAE)

	GP	RCGP	t-GP	m-GP
		Focused Outliers		
Synthetic	0.19 (0.00)	**0.15 (0.00)**	0.18 (0.00)	0.23 (0.00)
Boston	0.23 (0.06)	**0.22 (0.01)**	0.27 (0.00)	0.27 (0.00)
Energy	0.03 (0.04)	**0.02 (0.00)**	0.03 (0.05)	0.24 (0.00)
Yacht	0.26 (0.15)	**0.10 (0.14)**	0.20 (0.04)	0.24 (0.00)
		Asymmetric Outliers		
Synthetic	1.14 (0.00)	0.63 (0.00)	1.06 (0.00)	**0.61 (0.00)**
Boston	0.63 (0.02)	**0.49 (0.00)**	0.52 (0.00)	0.52 (0.00)
Energy	0.54 (0.02)	0.44 (0.04)	0.42 (0.02)	**0.41 (0.00)**
Yacht	0.54 (0.06)	**0.35 (0.02)**	0.41 (0.00)	0.40 (0.00)

RCGPs are robust!





RCGPs are fast!

(Time in seconds, incl. hyper parameter optimisation)

	GP	RCGP	t-GP	m-GP
Synthetic	1.5 (0.1)	1.2 (0.0)	2.2 (0.0)	3.0 (0.0)
Boston	1.9 (0.5)	5.1 (0.9)	30.7 (6.1)	16.7 (1.7)
Energy	3.8 (0.9)	4.6 (2.0)	34.0 (11)	33.8 (0.3)
Yacht	1.6 (0.3)	2.1 (0.2)	5.6 (0.7)	4.5 (0.4)

RCGPs are much faster than other robust alternatives!

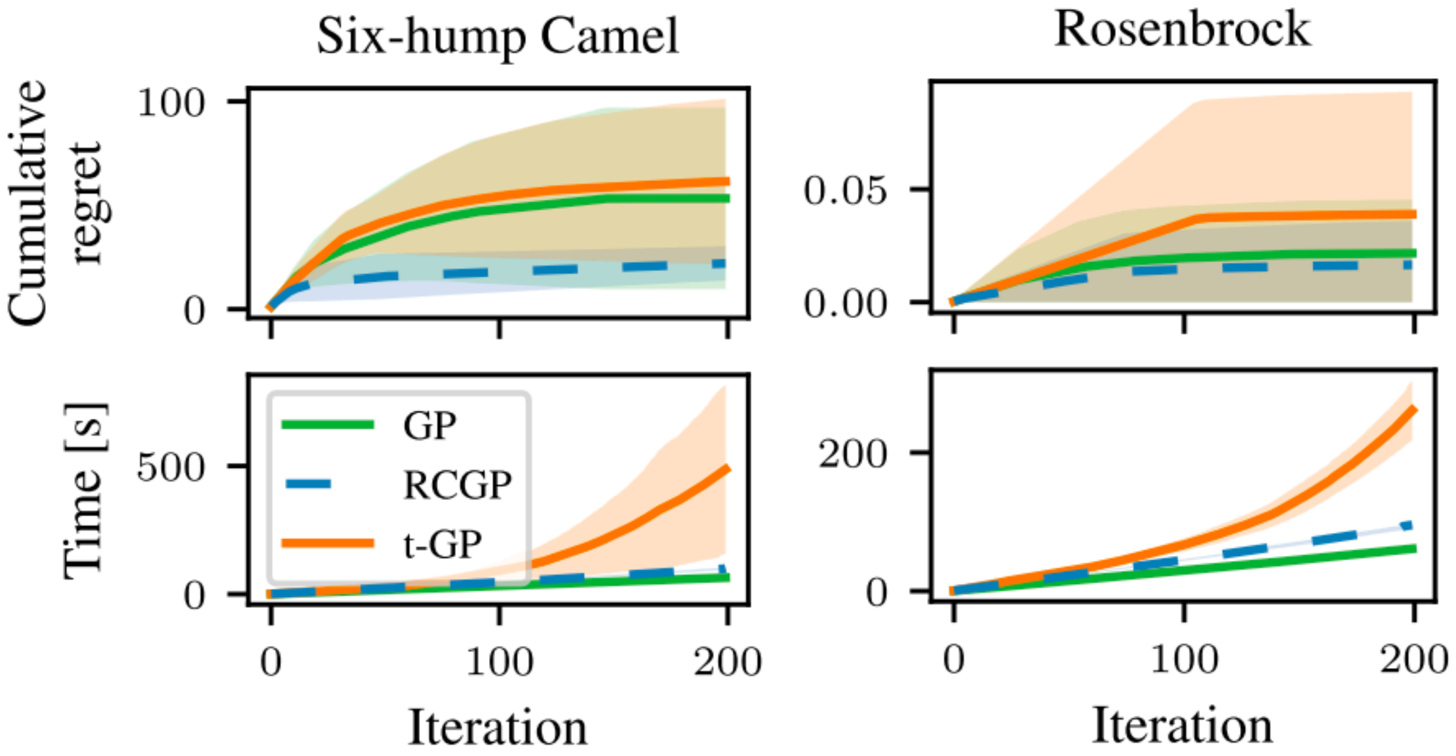
RCGPs are roughly as fast as GPs

	GP	RCGP	t-GP	m-GP
Synthetic	1.5 (0.1)	1.2 (0.0)	2.2 (0.0)	3.0 (0.0)
Boston	1.9 (0.5)	5.1 (0.9)	30.7 (6.1)	16.7 (1.7)
Energy	3.8 (0.9)	4.6 (2.0)	34.0 (11)	33.8 (0.3)
Yacht	1.6 (0.3)	2.1 (0.2)	5.6 (0.7)	4.5 (0.4)

Most of the difference between GP and RCGP comes down to adaptive optimisers for hyper parameter optimisation

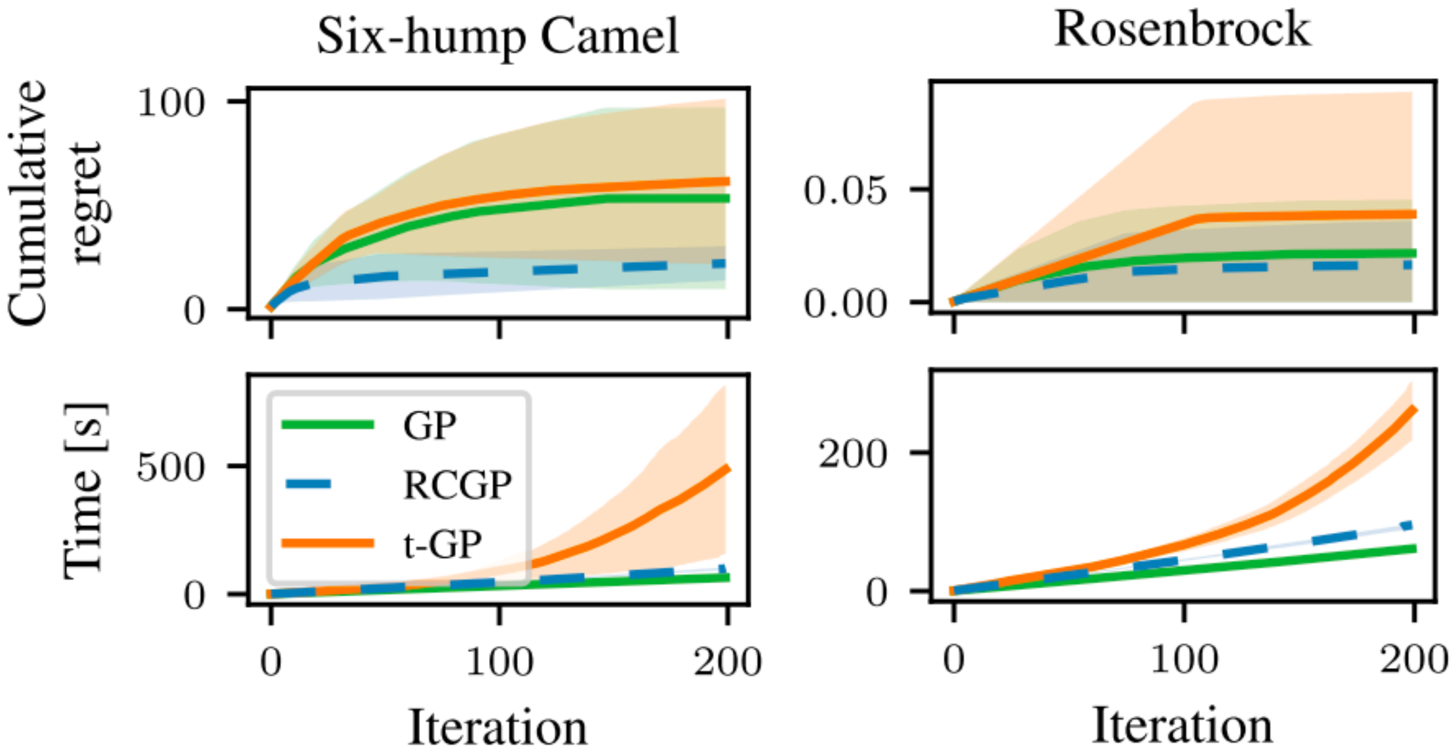
Robust Bayesian Optimisation

- In Bayesian optimisation, the GP posterior is used to create an acquisition function. Our RCGPs naturally lead to robust acquisition functions!



Robust Bayesian Optimisation

- In Bayesian optimisation, the GP posterior is used to create an acquisition function. Our RCGPs naturally lead to robust acquisition functions!

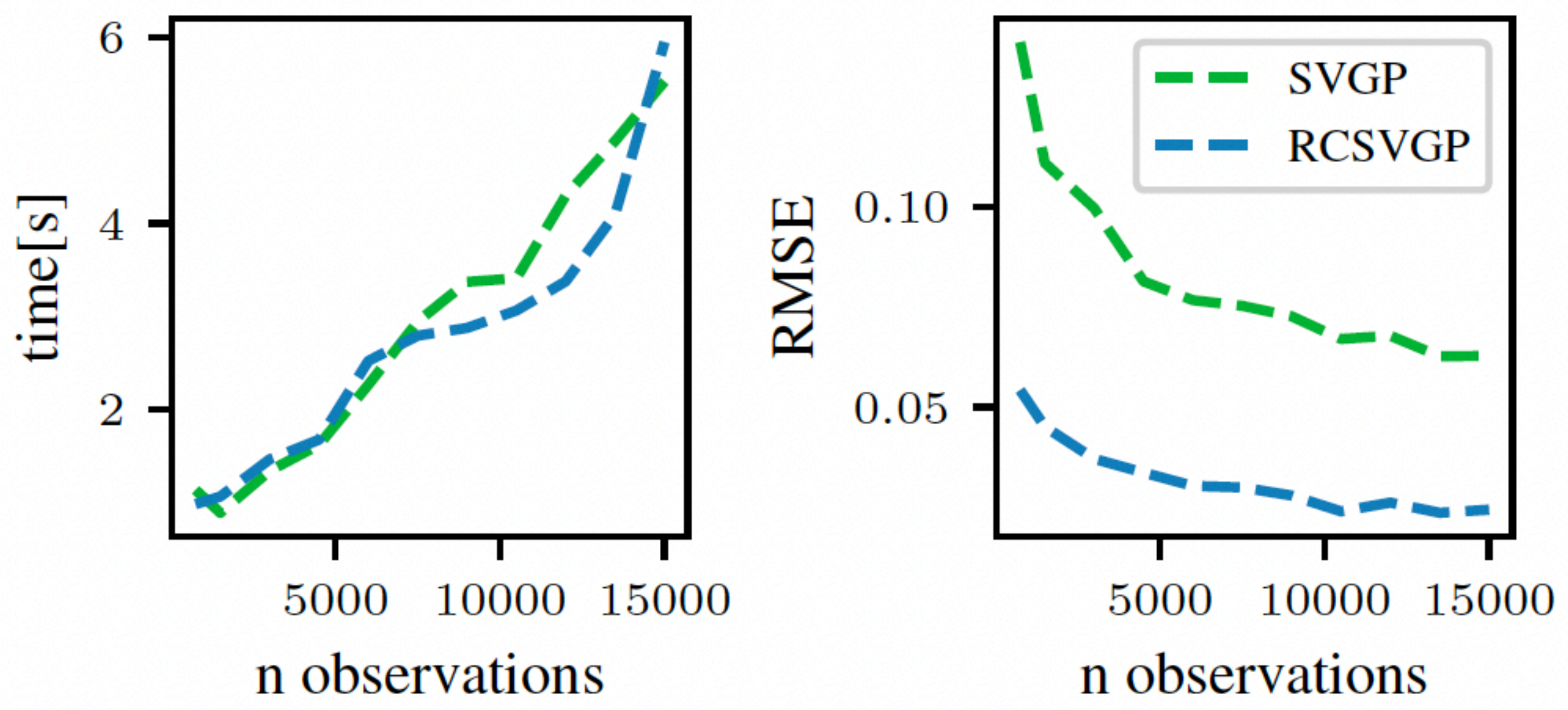


Robust!

Fast!

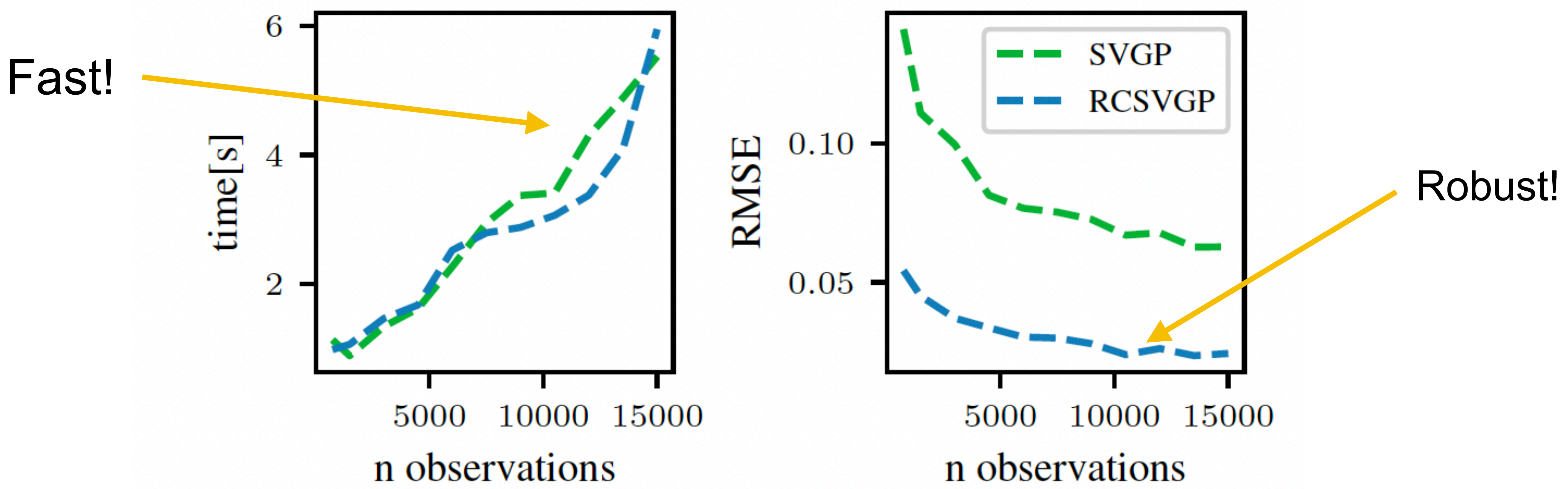
Robust SVGPs

- Sparse Variational GPs (SVGPs) is an approximate GP method which reduces significantly the cost of GPs from $O(n^3)$ to $O(nm^2)$ where m is small. Our approach naturally leads to a robust version!



Robust SVGPs

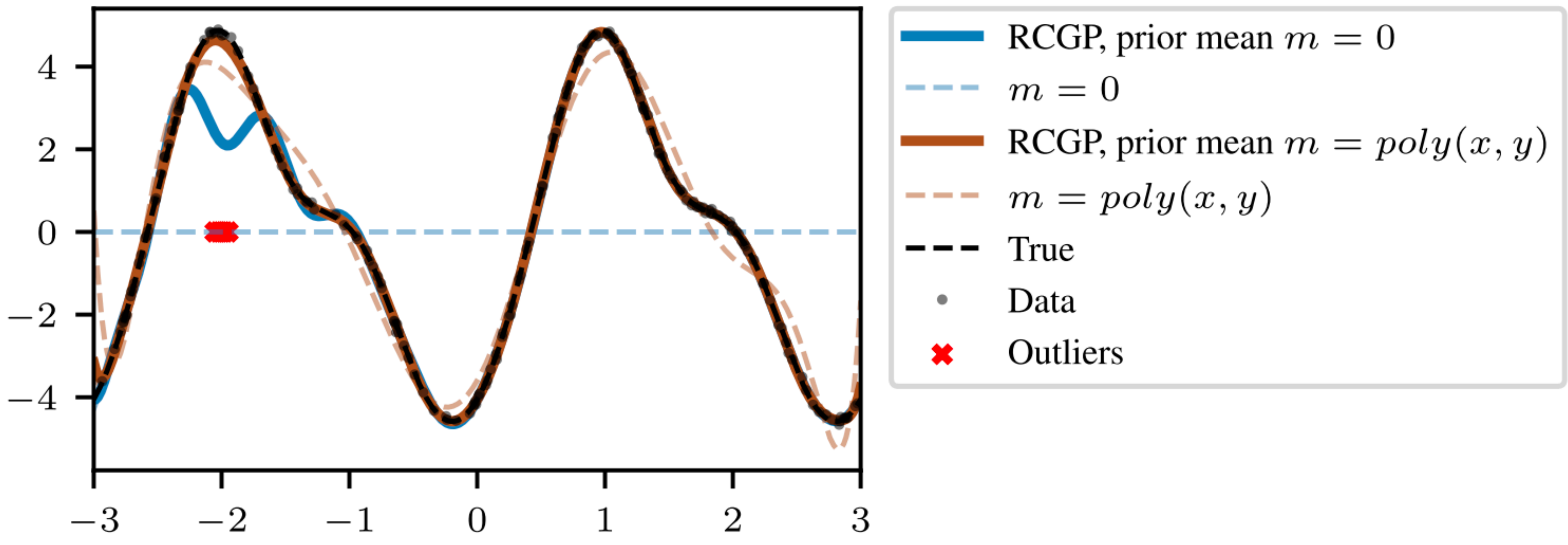
- Sparse Variational GPs (SVGPs) is an approximate GP method which reduces significantly the cost of GPs from $O(n^3)$ to $O(nm^2)$ where m is small. Our approach naturally leads to a robust version!



A drawback of the current approach

- It relies heavily on having a good mean function….

$$w(x,y) = \left(1 + \frac{(y - m(x))^2}{c^2}\right)^{-\frac{1}{2}}$$

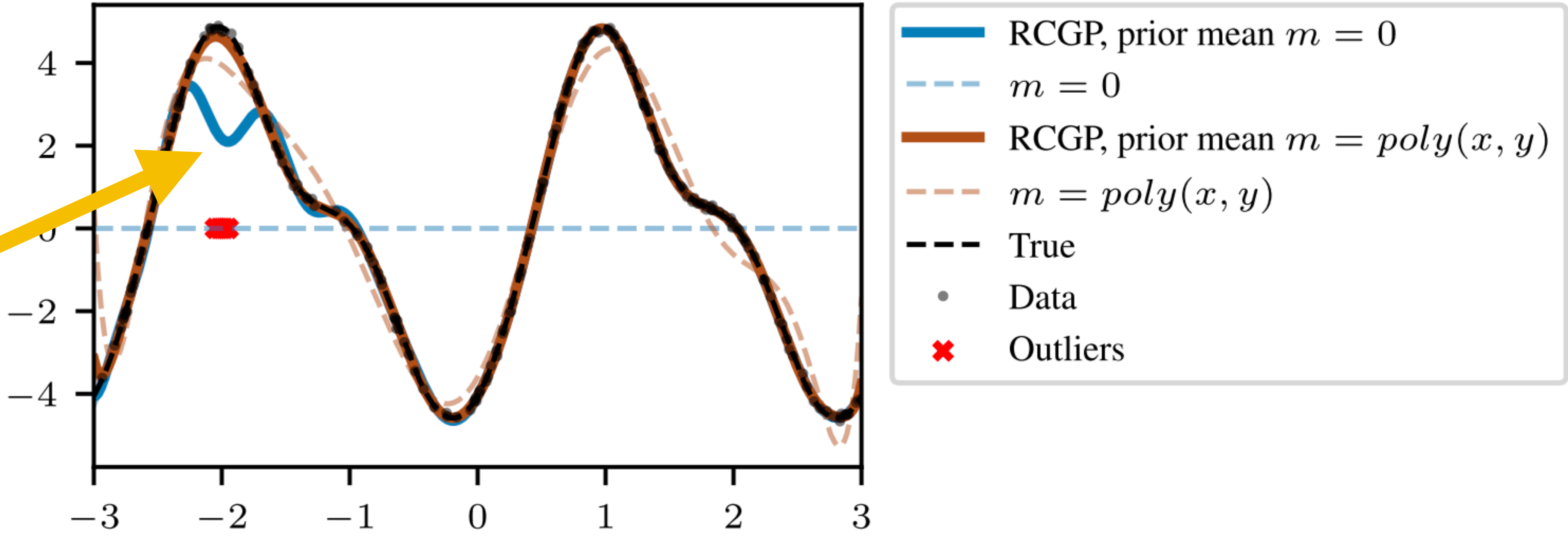


A drawback of the current approach

- It relies heavily on having a good mean function….

$$w(x,y) = \left(1 + \frac{(y - m(x))^2}{c^2}\right)^{-\frac{1}{2}}$$

!!!!!!

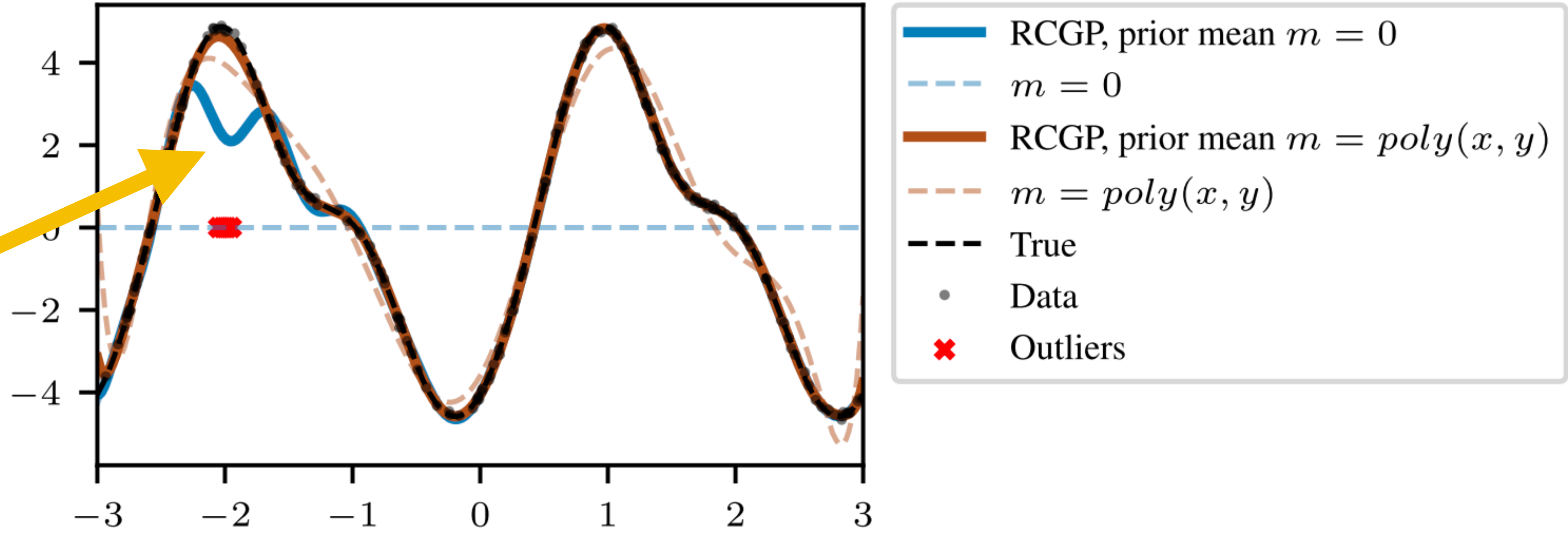


A drawback of the current approach

- It relies heavily on having a good mean function….

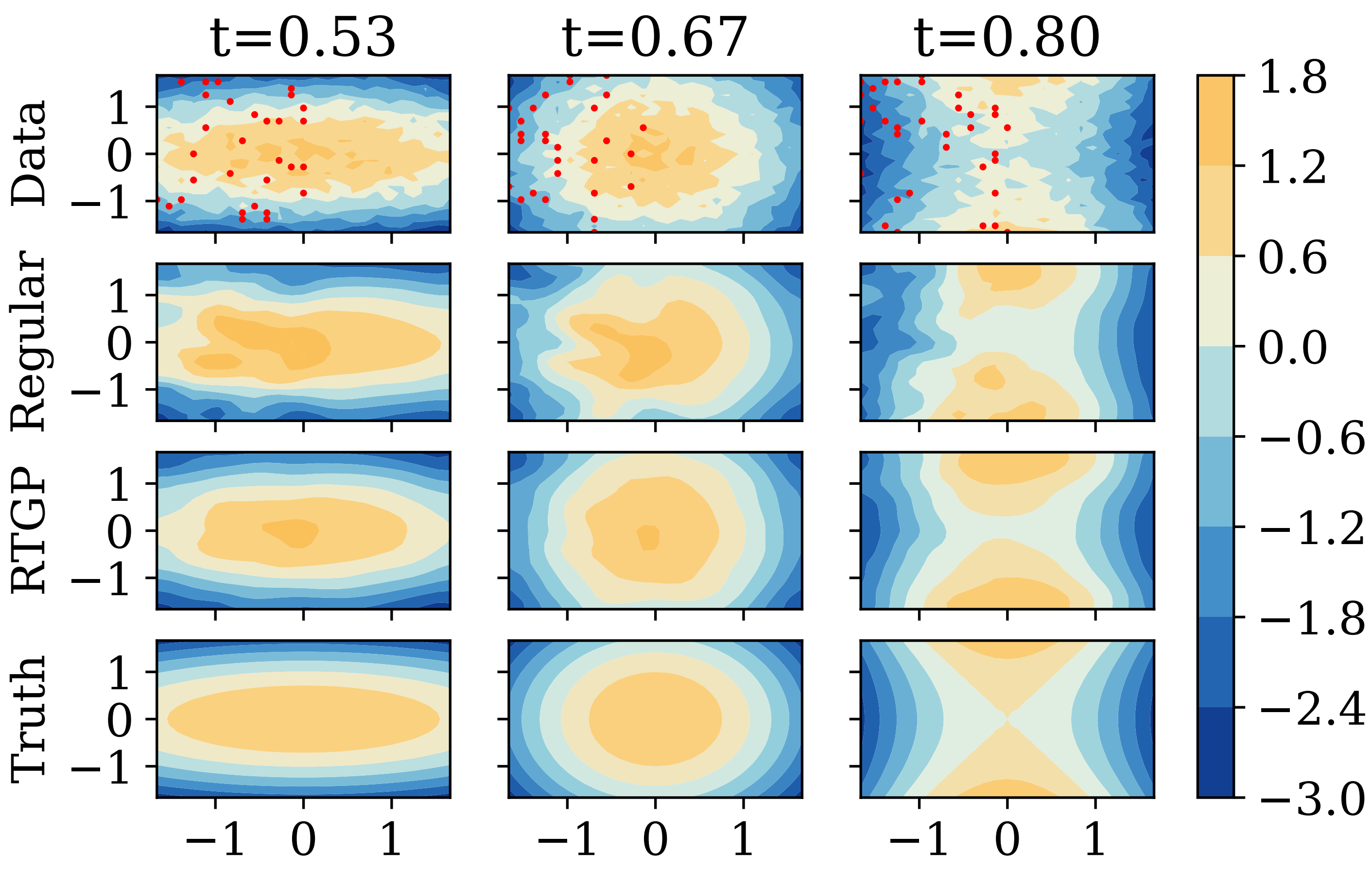
$$w(x, y) = \left(1 + \frac{(y - m(x))^2}{c^2}\right)^{-\frac{1}{2}}$$

!!!!!!



- **Potential fixes:** use a robust parametric model to fit the prior mean function first!

Linear-time spatio-temporal GPs



Paper on arXiv soon….

The cost is $O(n)$ where n is the number of time points + much easier to pick weights!

Conclusion

Conclusion

- With careful choices of loss functions, Generalised Bayes can bring both **robustness** and **computational efficiency**!

Conclusion

- With careful choices of loss functions, Generalised Bayes can bring both **robustness** and **computational efficiency**!

Conclusion

- With careful choices of loss functions, Generalised Bayes can bring both **robustness** and **computational efficiency**!

- RCGPs are an example in the case of GP regression where we get **both robustness and conjugacy**, something no other competitor has managed!

Conclusion

- With careful choices of loss functions, Generalised Bayes can bring both **robustness** and **computational efficiency**!

- RCGPs are an example in the case of GP regression where we get **both robustness and conjugacy**, something no other competitor has managed!

Conclusion

- With careful choices of loss functions, Generalised Bayes can bring both **robustness** and **computational efficiency**!

- RCGPs are an example in the case of GP regression where we get **both robustness and conjugacy**, something no other competitor has managed!

- RCGPs can be developed for any case where standard GPs, and could hence be used for multi-output GPs, multi-fidelity GPs, GPs with derivative or integral information, etc…

Conclusion

- With careful choices of loss functions, Generalised Bayes can bring both **robustness** and **computational efficiency**!

- RCGPs are an example in the case of GP regression where we get **both robustness and conjugacy**, something no other competitor has managed!

- RCGPs can be developed for any case where standard GPs, and could hence be used for multi-output GPs, multi-fidelity GPs, GPs with derivative or integral information, etc…

Conclusion

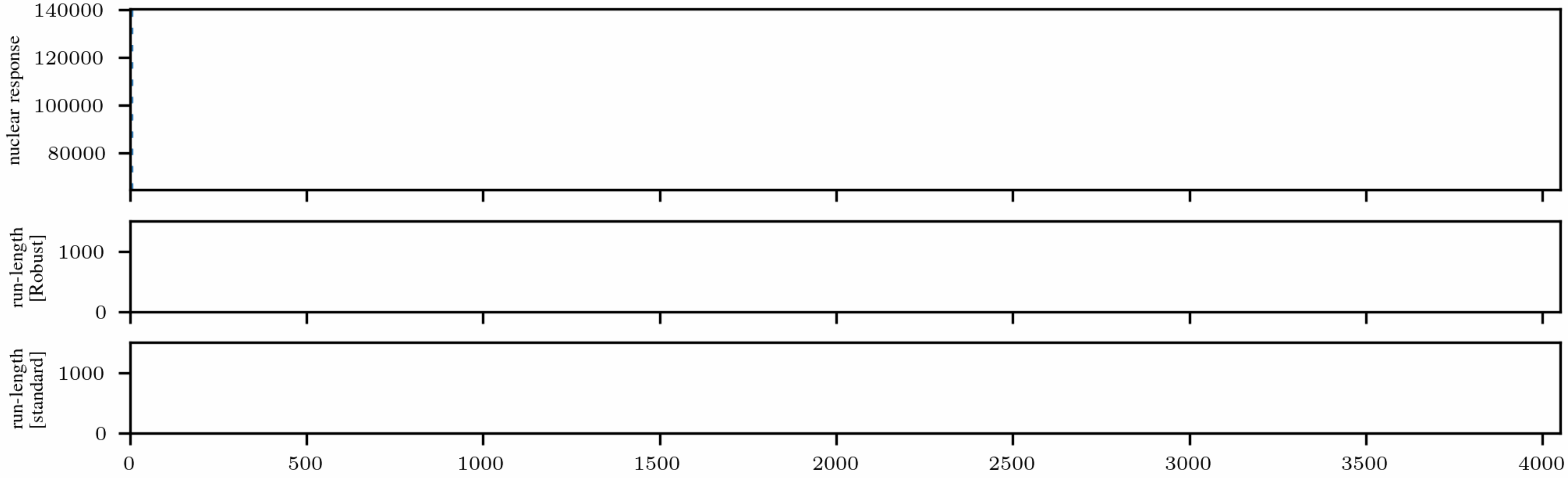
- With careful choices of loss functions, Generalised Bayes can bring both **robustness** and **computational efficiency**!

- RCGPs are an example in the case of GP regression where we get **both robustness and conjugacy**, something no other competitor has managed!

- RCGPs can be developed for any case where standard GPs, and could hence be used for multi-output GPs, multi-fidelity GPs, GPs with derivative or integral information, etc…

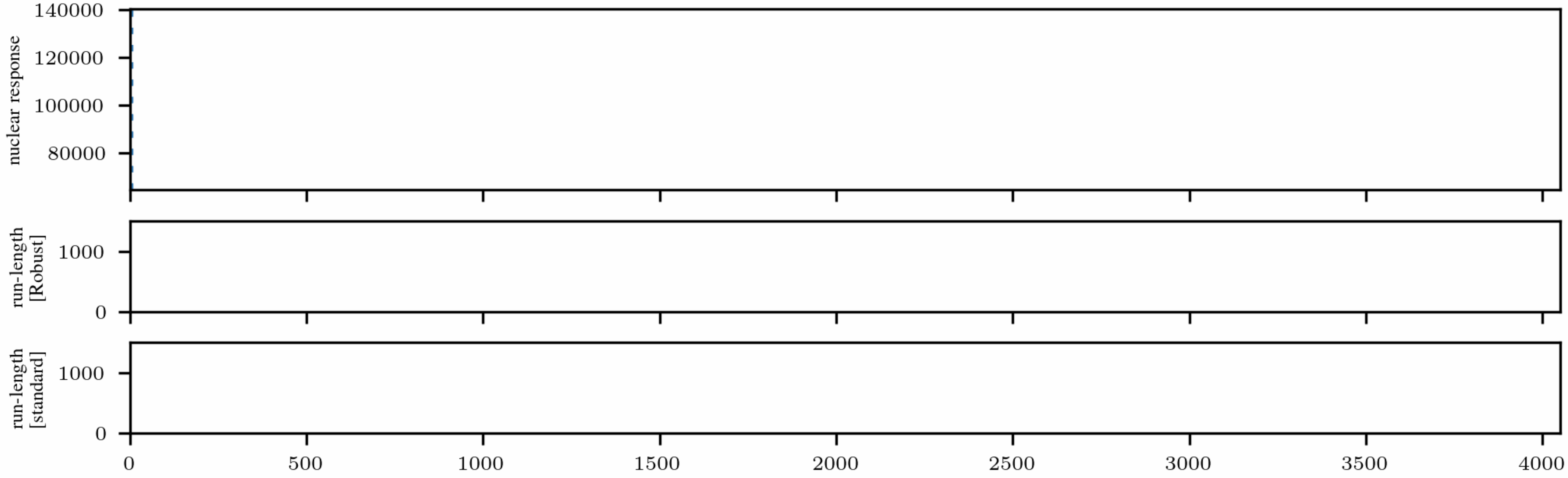
- This type of approach is also useful way beyond the GP world….!

Related work (online change point detection)



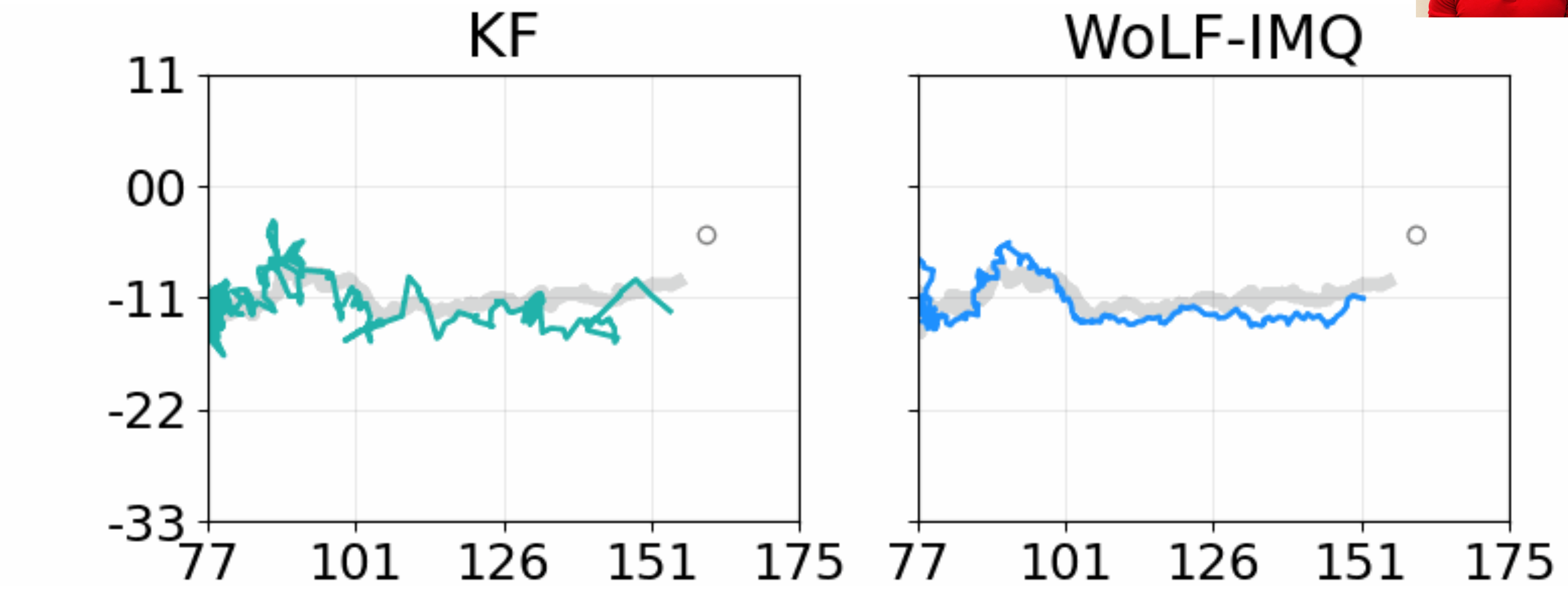
Altamirano, M., Briol, F.-X., & Knoblauch, J. (2023). Robust and scalable Bayesian online changepoint detection. ICML, 642–663.

Related work (online change point detection)



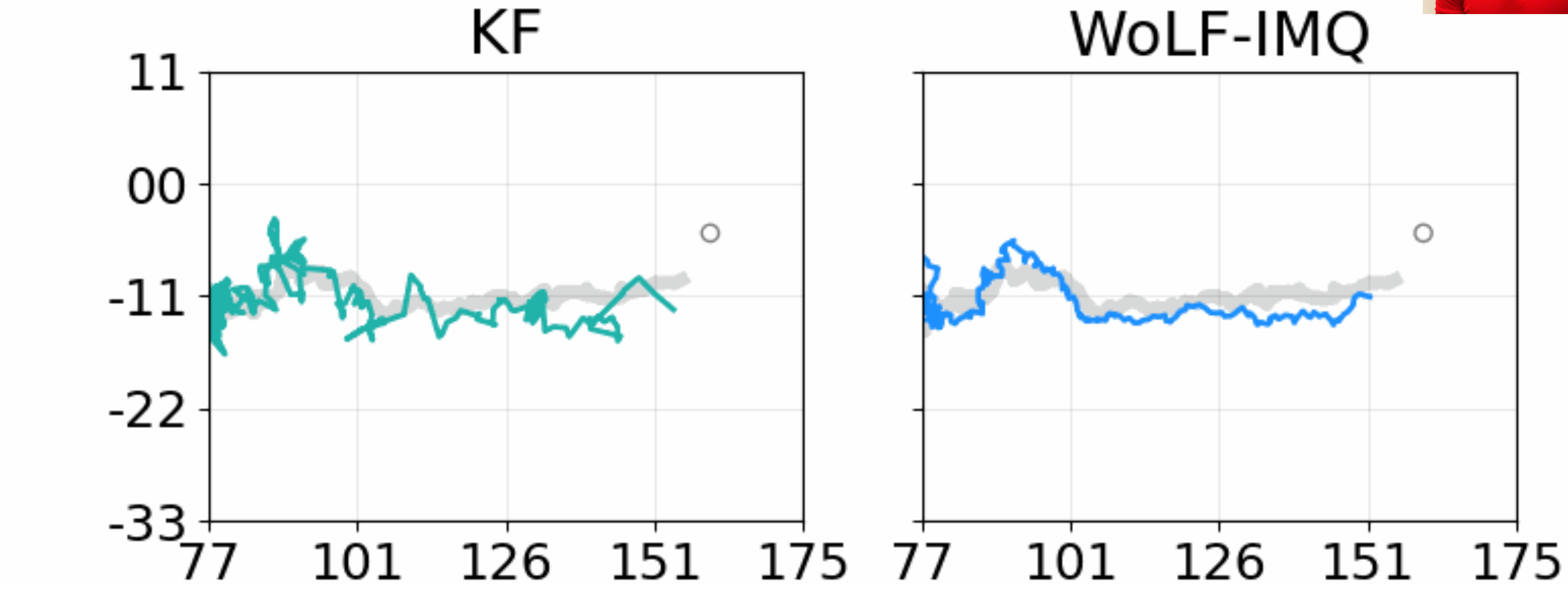
Altamirano, M., Briol, F.-X., & Knoblauch, J. (2023). Robust and scalable Bayesian online changepoint detection. ICML, 642–663.

Related work (Kalman filtering)



Duran-Martin, G., Altamirano, M., Shestopaloff, A. Y., Sanchez-Betancourt, L., Knoblauch, J., Jones, M., *Briol, F-X.* & Murphy, K. (2024). *Outlier-robust Kalman filtering through generalised Bayes*. ICML,12138-12171.

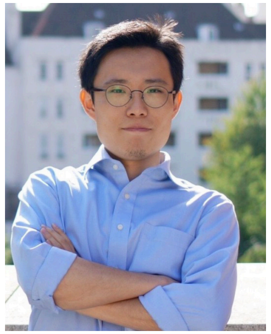
Related work (Kalman filtering)



Duran-Martin, G., Altamirano, M., Shestopaloff, A. Y., Sanchez-Betancourt, L., Knoblauch, J., Jones, M., *Briol, F-X.* & Murphy, K. (2024). *Outlier-robust Kalman filtering through generalised Bayes*. ICML,12138-12171.

Related work (intractable likelihoods)



- Robust and conjugate generalised Bayes for **continuous doubly intractable models**!

  Matsubara, T., Knoblauch, J., Briol, F.-X., & Oates, C. J. (2022). Robust generalised Bayesian inference for intractable likelihoods. JRSBB, 84(3), 997–1022.

- Robust (non-conjugate but fast!) generalised Bayes for **discrete doubly intractable models**.

  Matsubara, T., Knoblauch, J., Briol, F.-X., & Oates, C. J. (2023). Generalised Bayesian inference for discrete intractable likelihood. JASA, to appear.

Any Questions?

Robust and Conjugate Gaussian Process Regression

Matias Altamirano [1] **François-Xavier Briol** [1] **Jeremias Knoblauch** [1]