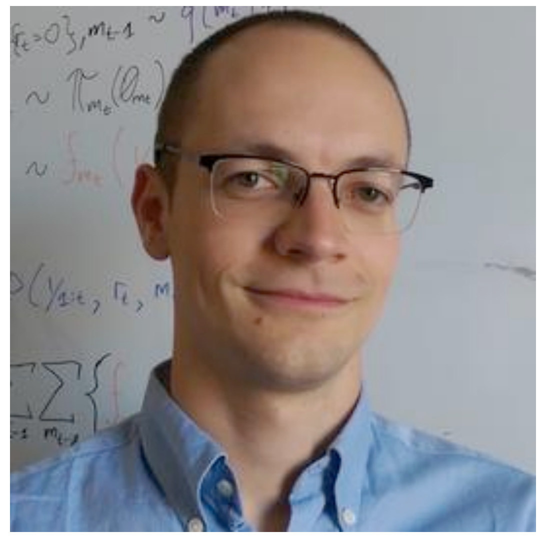
Robust and Conjugate Spatio-Temporal Gaussian Processes

William Laplante
University College London

Matias Altamirano
University College London

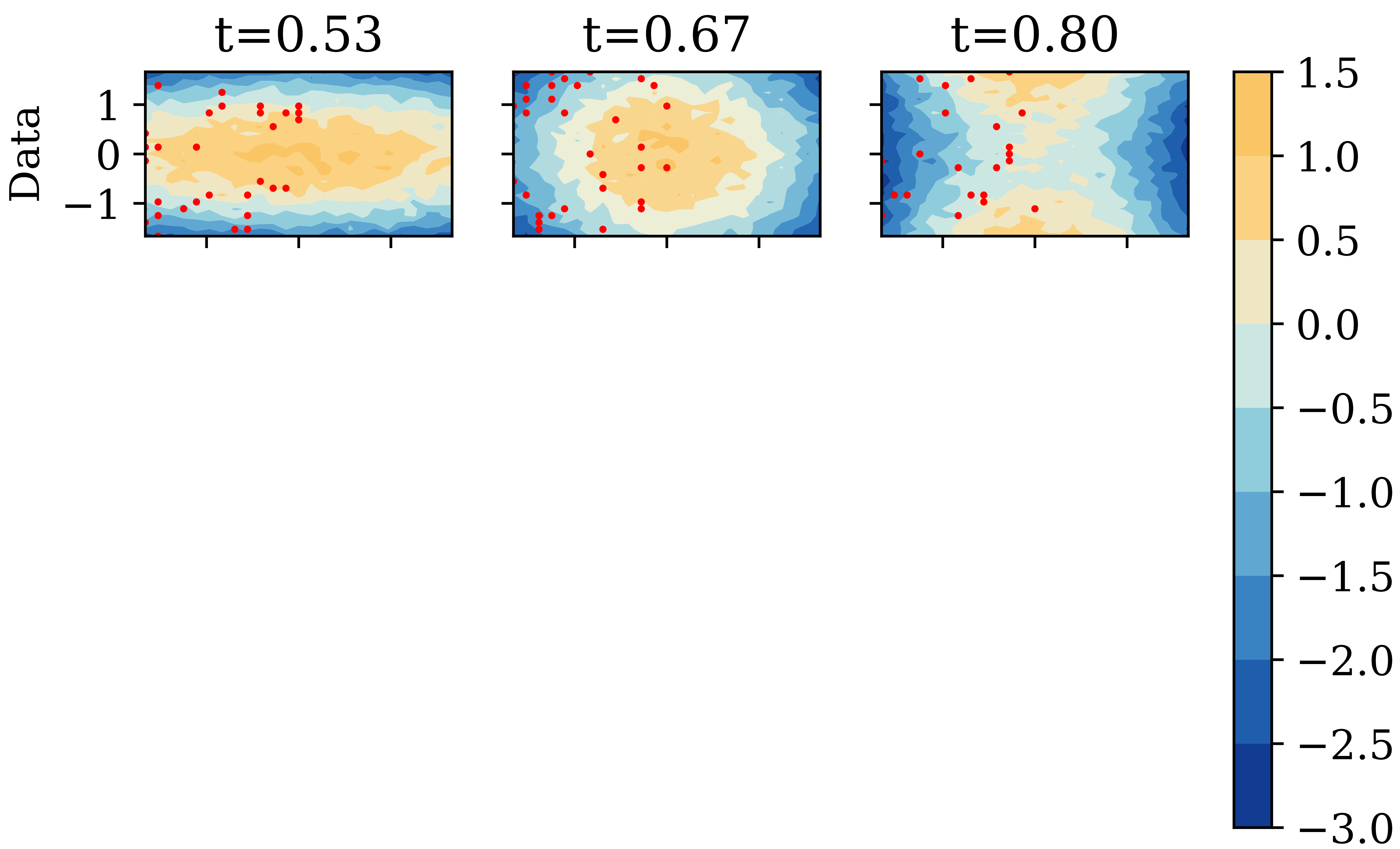
Andrew Duncan
Imperial College London



Jeremias Knoblauch
University College London

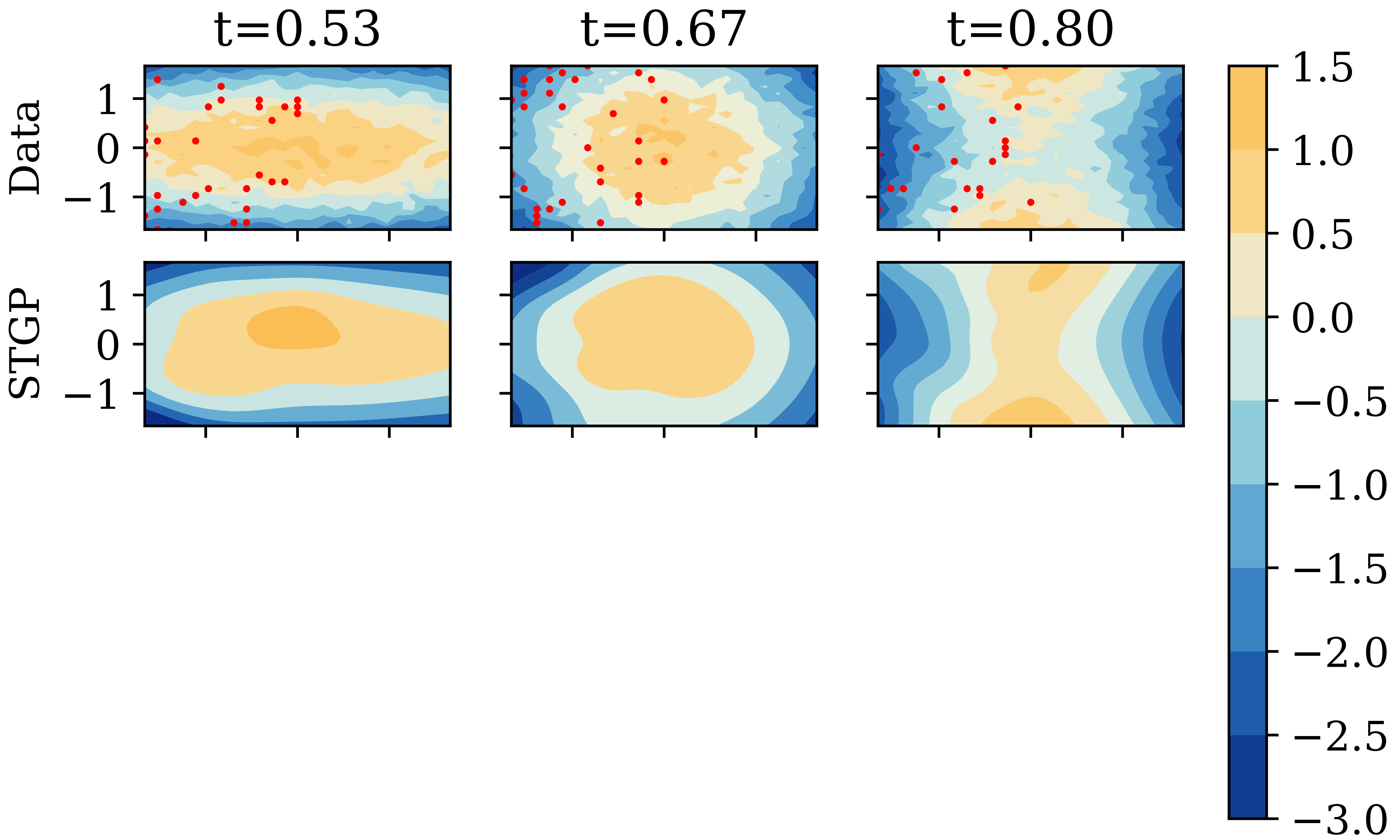
François-Xavier Briol
University College London

Preview...



Preview...

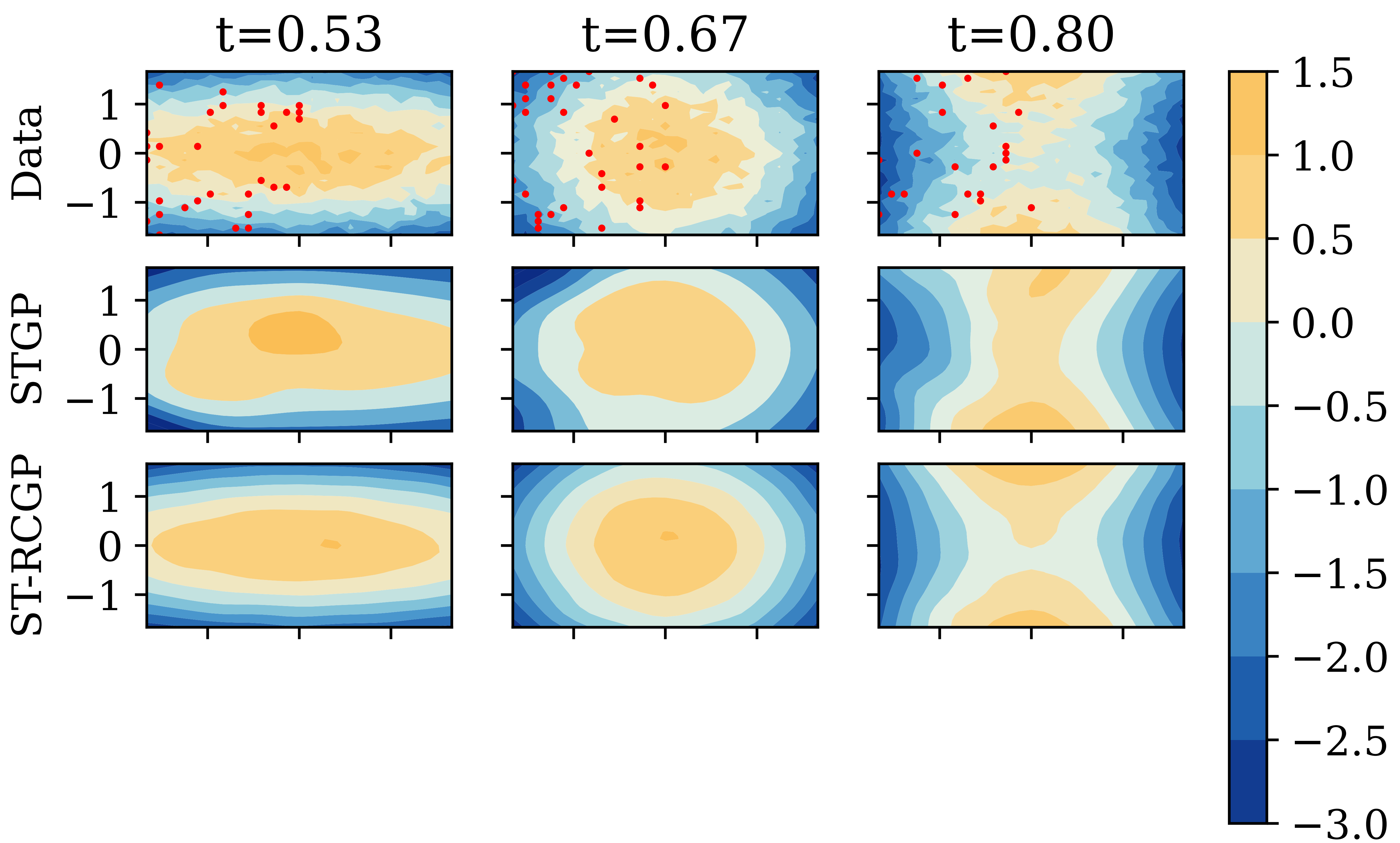
Baseline Method



Preview...

Baseline Method

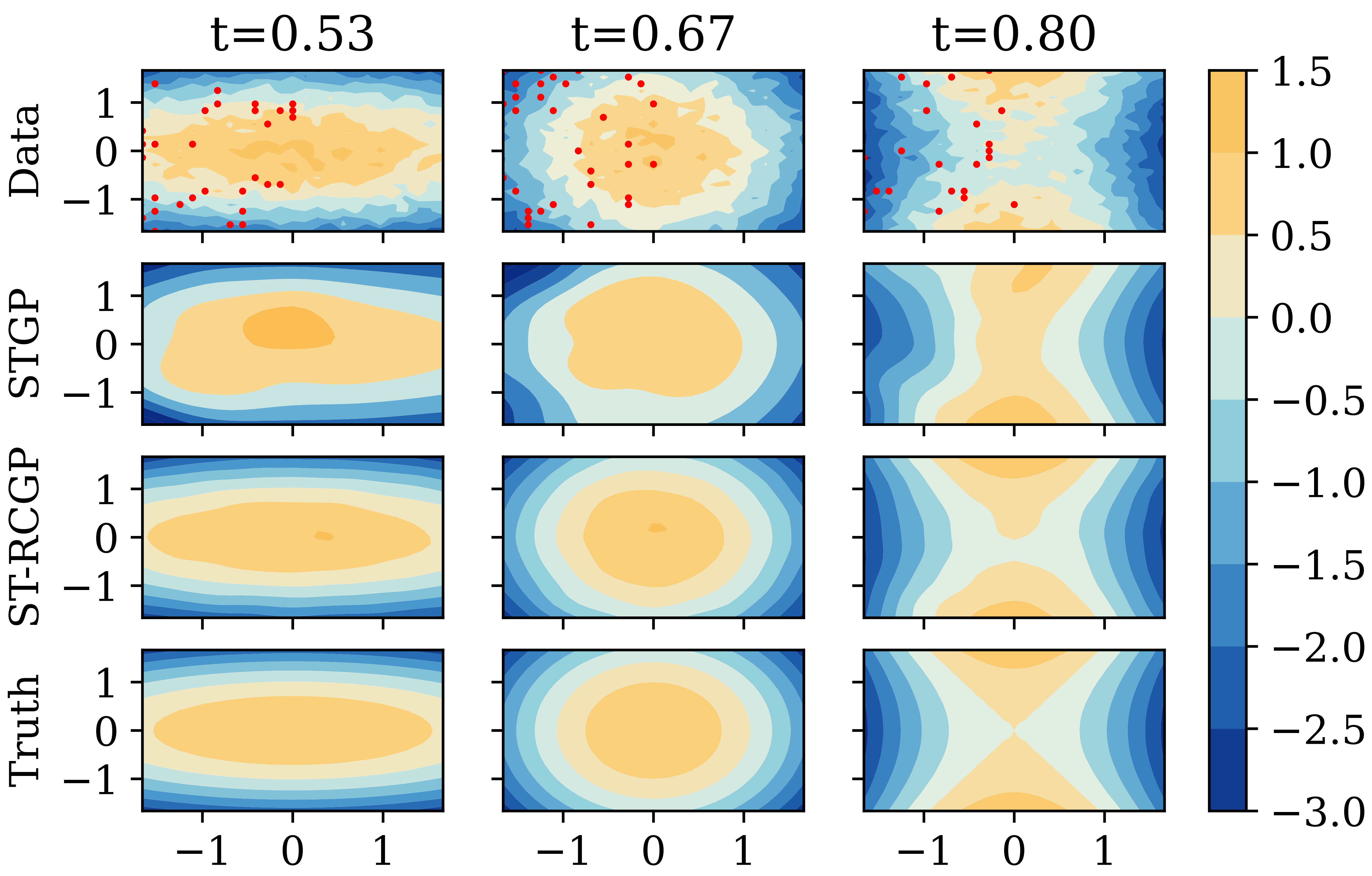
Our Method



Preview...

Baseline Method

Our Method



Background: Gaussian Process Regression

Observations

$\mathcal{D} := \{\mathbf{x}_k, y_k\}_{k=1}^{N} \subseteq \mathcal{X} \times \mathcal{Y}$

Background: Gaussian Process Regression

Observations

$\mathscr{D} := \{\mathbf{x}_k, y_k\}_{k=1}^{N} \subseteq \mathscr{X} \times \mathscr{Y}$

Model

$y_k = f(\mathbf{x}_k) + \epsilon_k,\ f \sim GP(m, \kappa)$

$m : \mathscr{X} \rightarrow \mathbb{R}$ (prior mean)

$\kappa : \mathscr{X} \times \mathscr{X} \rightarrow \mathbb{R}$ (kernel)

Background: Gaussian Process Regression

Observations

$\mathcal{D} := \{\mathbf{x}_k, y_k\}_{k=1}^{N} \subseteq \mathcal{X} \times \mathcal{Y}$

Model

$y_k = f(\mathbf{x}_k) + \epsilon_k, \; f \sim GP(m, \kappa)$

$m : \mathcal{X} \to \mathbb{R}$ (prior mean)

$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (kernel)

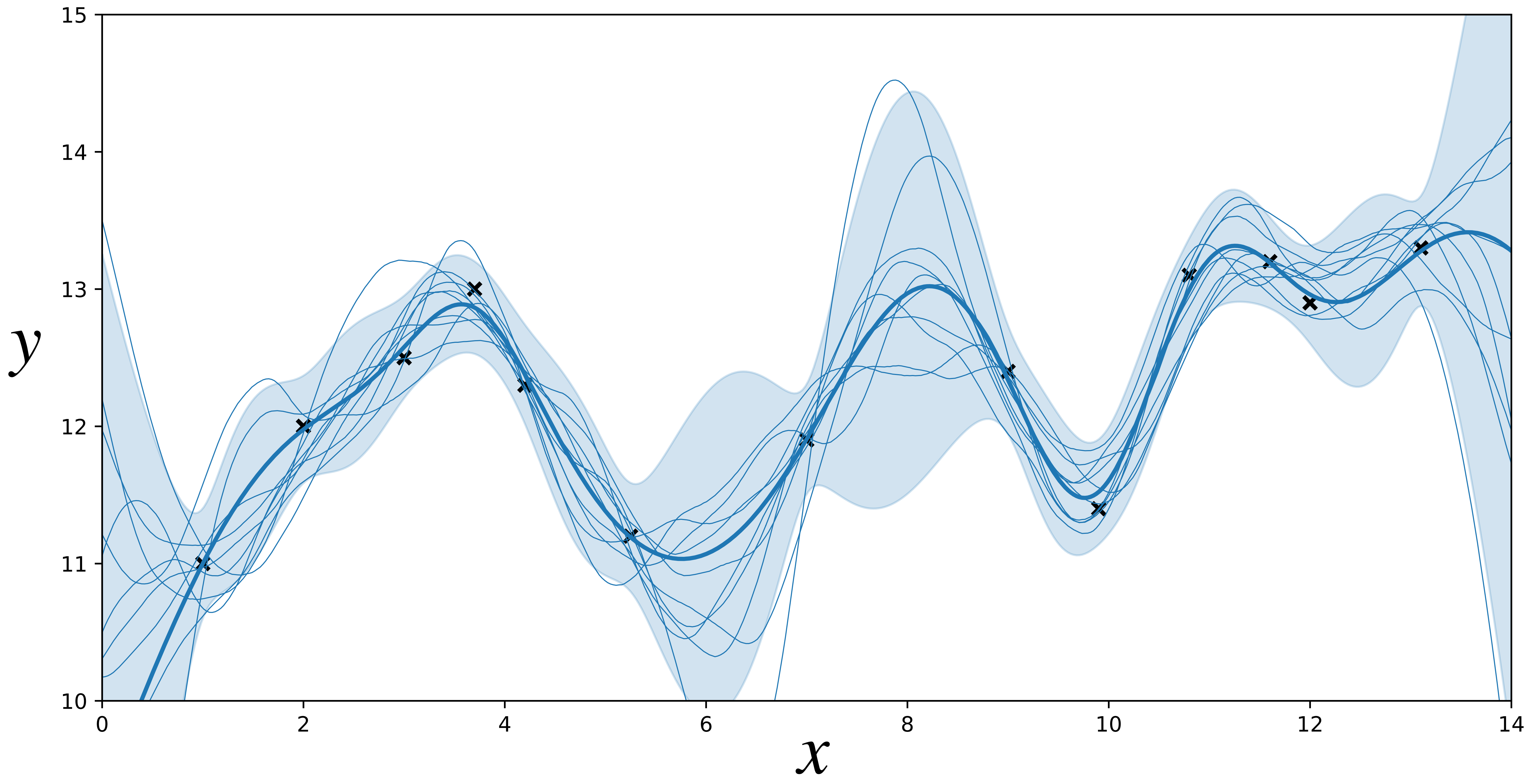


Image taken from: https://www.lancaster.ac.uk/stor-i-student-sites/thomas-newman/2022/05/05/gaussian-processes-in-regression/

Background: Gaussian Process Regression

Observations

$\mathcal{D} := \{\mathbf{x}_k, y_k\}_{k=1}^N \subseteq \mathcal{X} \times \mathcal{Y}$

Model

$y_k = f(\mathbf{x}_k) + \epsilon_k, \; f \sim GP(m, \kappa)$

$m : \mathcal{X} \to \mathbb{R}$ (prior mean)

$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (kernel)

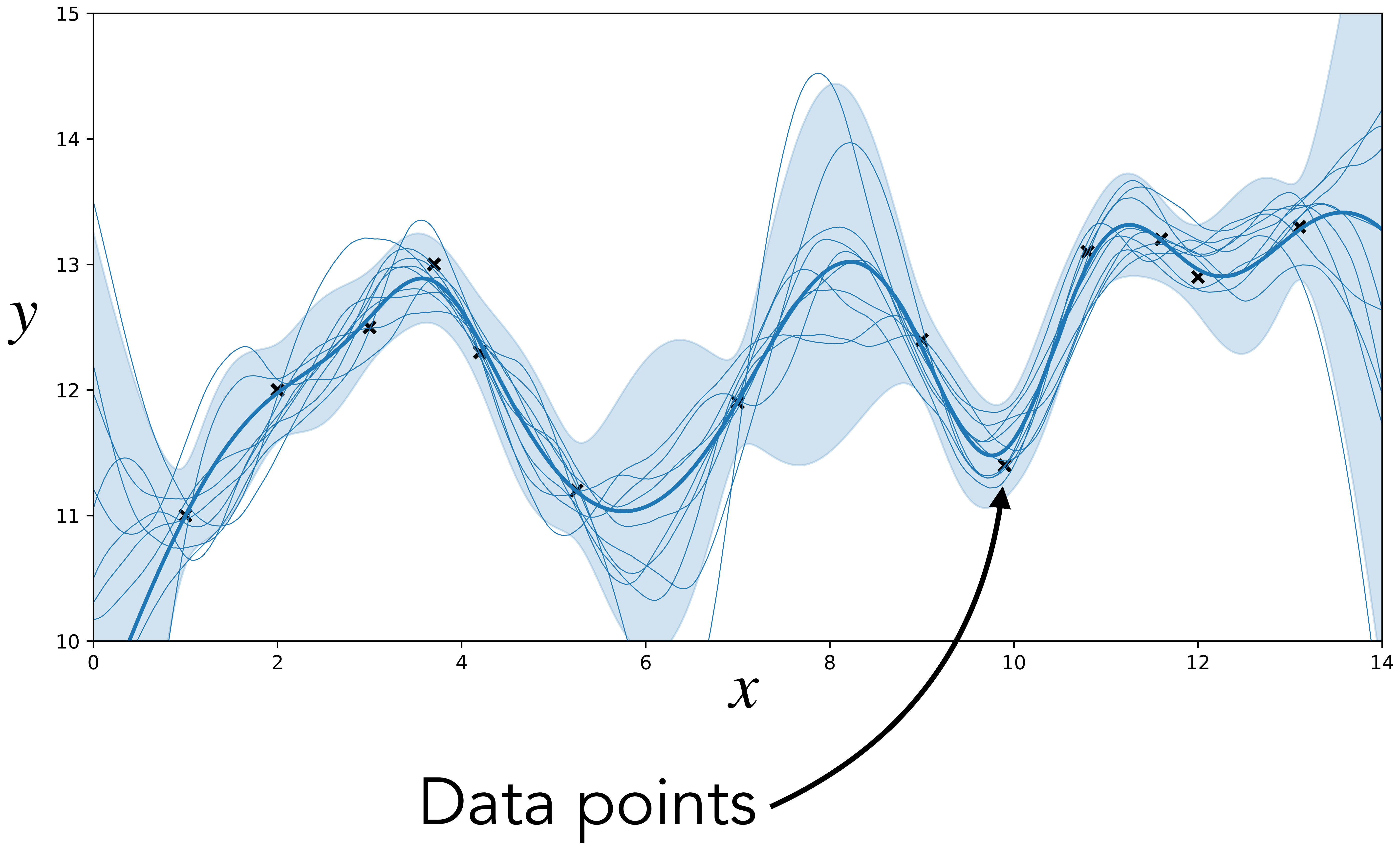


Image taken from: https://www.lancaster.ac.uk/stor-i-student-sites/thomas-newman/2022/05/05/gaussian-processes-in-regression/

Background: Gaussian Process Regression

Observations

$$\mathscr{D} := \{\mathbf{x}_k, y_k\}_{k=1}^{N} \subseteq \mathscr{X} \times \mathscr{Y}$$

Model

$$y_k = f(\mathbf{x}_k) + \epsilon_k, \ f \sim GP(m, \kappa)$$

$m : \mathscr{X} \to \mathbb{R}$ (prior mean)

$\kappa : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ (kernel)

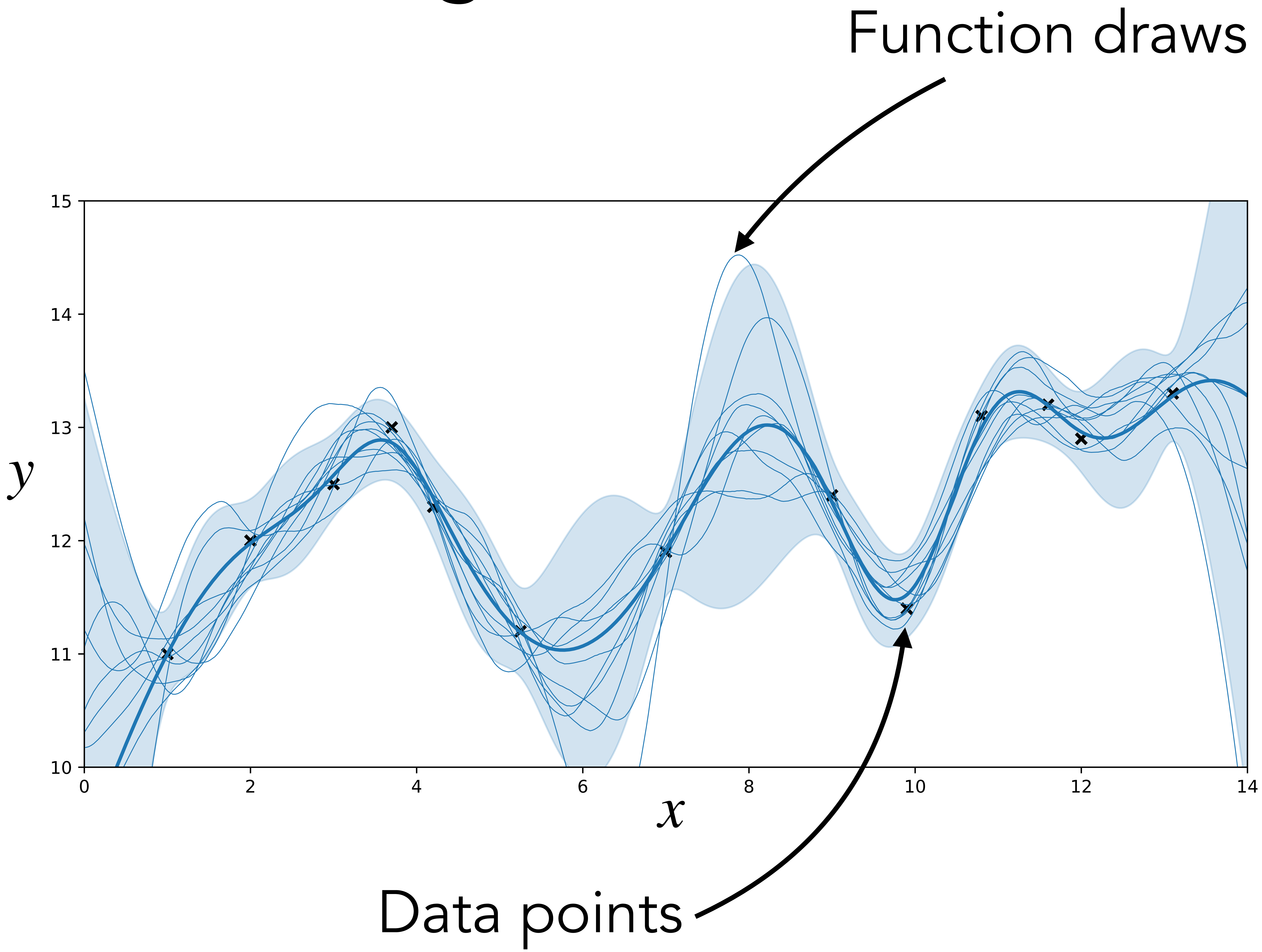


Image taken from: https://www.lancaster.ac.uk/stor-i-student-sites/thomas-newman/2022/05/05/gaussian-processes-in-regression/

Background: Gaussian Process Regression

Restrict f to $\mathscr{D}$

Background: Gaussian Process Regression

Restrict f to $\mathscr{D}$ $\longrightarrow$ $\mathbf{f} := (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\top \sim \mathscr{N}(\mathbf{m}, \mathbf{K})$

Background: Gaussian Process Regression

Restrict f to $\mathscr{D}$ $\longrightarrow$ $\mathbf{f} := (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\top \sim \mathscr{N}(\mathbf{m}, \mathbf{K})$

Background: Gaussian Process Regression

Restrict f to $\mathscr{D}$ $\longrightarrow$ $\mathbf{f} := (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\top \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$

The posterior predictive $p(f(\mathbf{x}_\star) \mid y_{1:N}, \mathbf{x}_{1:N}) \sim \mathcal{N}(\mu^\star_{GP}, \Sigma^\star_{GP})$ with:

Background: Gaussian Process Regression

Restrict f to $\mathscr{D}$ $\longrightarrow$ $\mathbf{f} := (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\top \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$

The posterior predictive $p(f(\mathbf{x}_\star) \mid y_{1:N}, \mathbf{x}_{1:N}) \sim \mathcal{N}(\mu^\star_{GP}, \Sigma^\star_{GP})$ with:

$$\mu^\star_{GP} = m_\star + \mathbf{k}_\star^\top (\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1} (\mathbf{y} - \mathbf{m}) \qquad \Sigma^\star_{GP} = k_{\star\star} - \mathbf{k}_\star^\top (\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1} \mathbf{k}_\star$$

Background: Gaussian Process Regression

Restrict f to $\mathscr{D}$ $\longrightarrow$ $\mathbf{f} := (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^\top \sim \mathcal{N}(\mathbf{m}, \mathbf{K})$

The posterior predictive $p(f(\mathbf{x}_\star) \,|\, y_{1:N}, \mathbf{x}_{1:N}) \sim \mathcal{N}(\mu^\star_{GP}, \Sigma^\star_{GP})$ with:

$$\mu^\star_{GP} = m_\star + \mathbf{k}_\star^\top(\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1}(\mathbf{y} - \mathbf{m}) \qquad \Sigma^\star_{GP} = k_{\star\star} - \mathbf{k}_\star^\top(\mathbf{K} + \sigma^2 \mathbf{I}_N)^{-1}\mathbf{k}_\star$$

Problem! $\mathcal{O}(N^3)$ cost…

Solution to GP Cost for Spatio-Temporal Inputs?

KALMAN FILTERING AND SMOOTHING SOLUTIONS TO TEMPORAL GAUSSIAN PROCESS REGRESSION MODELS

Jouni Hartikainen and Simo Särkkä

Aalto University
Department of Biomedical Engineering and Computational Science
Rakentajanaukio 2, 02150 Espoo, Finland

ABSTRACT

In this paper, we show how temporal (i.e., time-series) Gaussian process regression models in machine learning can be reformulated as linear-Gaussian state space models, which can be solved exactly with classical Kalman filtering theory. The result is an efficient non-parametric learning algorithm, whose computational complexity grows linearly with respect to number of observations. We show how the reformulation can be done for Matérn family of covariance functions analytically and for squared exponential covariance function by applying spectral Taylor series approximation. Advantages of the proposed approach are illustrated with two numerical experiments.

can be used for squared exponential covariance functions. This approach is directly applicable to cases with missing or non-uniformly sampled data, which is a typical problem for methods utilizing the temporal structure of data in covariance representations.

The underlying idea of efficient inference of Gaussian processes using a state space formulation is not new, because the contribution of Kalman's original 1960's article [3] was exactly this kind of re-formulation of the filtering problem of Wiener [4]. The idea of Bayesian modeling of unknown processes as Gaussian processes has also been widely utilized in communications theory [5, 6], but the philosophy differs slightly from that of Gaussian process regression in machine learning [7]. The idea of approximating

Solution to GP Cost for Spatio-Temporal Inputs?

KALMAN FILTERING AND SMOOTHING SOLUTIONS TO TEMPORAL GAUSSIAN PROCESS REGRESSION MODELS

Jouni Hartikainen and Simo Särkkä

Aalto University
Department of Biomedical Engineering and Computational Science
Rakentajanaukio 2, 02150 Espoo, Finland

ABSTRACT

In this paper, we show how temporal (i.e., time-series) Gaussian process regression models in machine learning can be reformulated as linear-Gaussian state space models, which can be solved exactly with classical Kalman filtering theory. The result is an efficient non-parametric learning algorithm, whose computational complexity grows linearly with respect to number of observations. We show how the re-formulation can be done for Matérn family of covariance functions analytically and for squared exponential covariance function by applying spectral Taylor series approximation. Advantages of the proposed approach are illustrated with two numerical experiments.

can be used for squared exponential covariance functions. This approach is directly applicable to cases with missing or non-uniformly sampled data, which is a typical problem for methods utilizing the temporal structure of data in covariance representations.

The underlying idea of efficient inference of Gaussian processes using a state space formulation is not new, because the contribution of Kalman's original 1960's article [3] was exactly this kind of re-formulation of the filtering problem of Wiener [4]. The idea of Bayesian modeling of unknown processes as Gaussian processes has also been widely utilized in communications theory [5, 6], but the philosophy differs slightly from that of Gaussian process regression in machine learning [7]. The idea of approximating

Result: $\mathcal{O}(N^3)$ **cost becomes** $\mathcal{O}(N)$**!**

Spatio-Temporal Gaussian Processes (STGPs)

If $\mathbf{x} = (\mathbf{s}, t)$, GP prior = solution to a **Stochastic Differential Equation (SDE)**

Spatio-Temporal Gaussian Processes (STGPs)

If $\mathbf{x} = (\mathbf{s}, t)$, GP prior = solution to a **Stochastic Differential Equation (SDE)**

Let $\mathbf{z}(\mathbf{s}, t) := \left(f(\mathbf{s}, t), \frac{\partial f(\mathbf{s}, t)}{\partial t}, \frac{\partial^2 f(\mathbf{s}, t)}{\partial t^2}, \ldots, \frac{\partial^\nu f(\mathbf{s}, t)}{\partial t^\nu} \right)^\top$

Spatio-Temporal Gaussian Processes (STGPs)

If $\mathbf{x} = (\mathbf{s}, t)$, GP prior = solution to a **Stochastic Differential Equation (SDE)**

Let $\mathbf{z}(\mathbf{s}, t) := \left(f(\mathbf{s}, t), \frac{\partial f(\mathbf{s}, t)}{\partial t}, \frac{\partial^2 f(\mathbf{s}, t)}{\partial t^2}, \ldots, \frac{\partial^\nu f(\mathbf{s}, t)}{\partial t^\nu} \right)^\mathsf{T}$

Continuous SDE

$$\frac{\partial \mathbf{z}(\mathbf{s}, t)}{\partial t} = \mathbf{F}\mathbf{z}(\mathbf{s}, t) + \mathbf{L}\mathbf{w}(\mathbf{s}, t)$$

Spatio-Temporal Gaussian Processes (STGPs)

If $\mathbf{x} = (\mathbf{s}, t)$, GP prior = solution to a **Stochastic Differential Equation (SDE)**

Let $\mathbf{z}(\mathbf{s}, t) := \left(f(\mathbf{s}, t), \frac{\partial f(\mathbf{s}, t)}{\partial t}, \frac{\partial^2 f(\mathbf{s}, t)}{\partial t^2}, \ldots, \frac{\partial^\nu f(\mathbf{s}, t)}{\partial t^\nu} \right)^\mathsf{T}$

Continuous SDE

$$\frac{\partial \mathbf{z}(\mathbf{s}, t)}{\partial t} = \mathbf{F}\mathbf{z}(\mathbf{s}, t) + \mathbf{L}\mathbf{w}(\mathbf{s}, t)$$

"Discretize" $\Rightarrow$

State-Space Model

$$\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_0)$$
$$\mathbf{z}_k = \mathbf{A}_{k-1}\mathbf{z}_{k-1} + \mathbf{q}_{k-1}, \mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{k-1})$$
$$\mathbf{y}_k = \mathbf{H}\mathbf{z}_k + \epsilon_k$$

Spatio-Temporal Gaussian Processes (STGPs)

If $\mathbf{x} = (\mathbf{s}, t)$, GP prior = solution to a **Stochastic Differential Equation (SDE)**

Let $\mathbf{z}(\mathbf{s}, t) := \left(f(\mathbf{s}, t), \frac{\partial f(\mathbf{s}, t)}{\partial t}, \frac{\partial^2 f(\mathbf{s}, t)}{\partial t^2}, \ldots, \frac{\partial^\nu f(\mathbf{s}, t)}{\partial t^\nu} \right)^\top$

Continuous SDE

$$\frac{\partial \mathbf{z}(\mathbf{s}, t)}{\partial t} = \mathbf{F}\mathbf{z}(\mathbf{s}, t) + \mathbf{L}\mathbf{w}(\mathbf{s}, t)$$

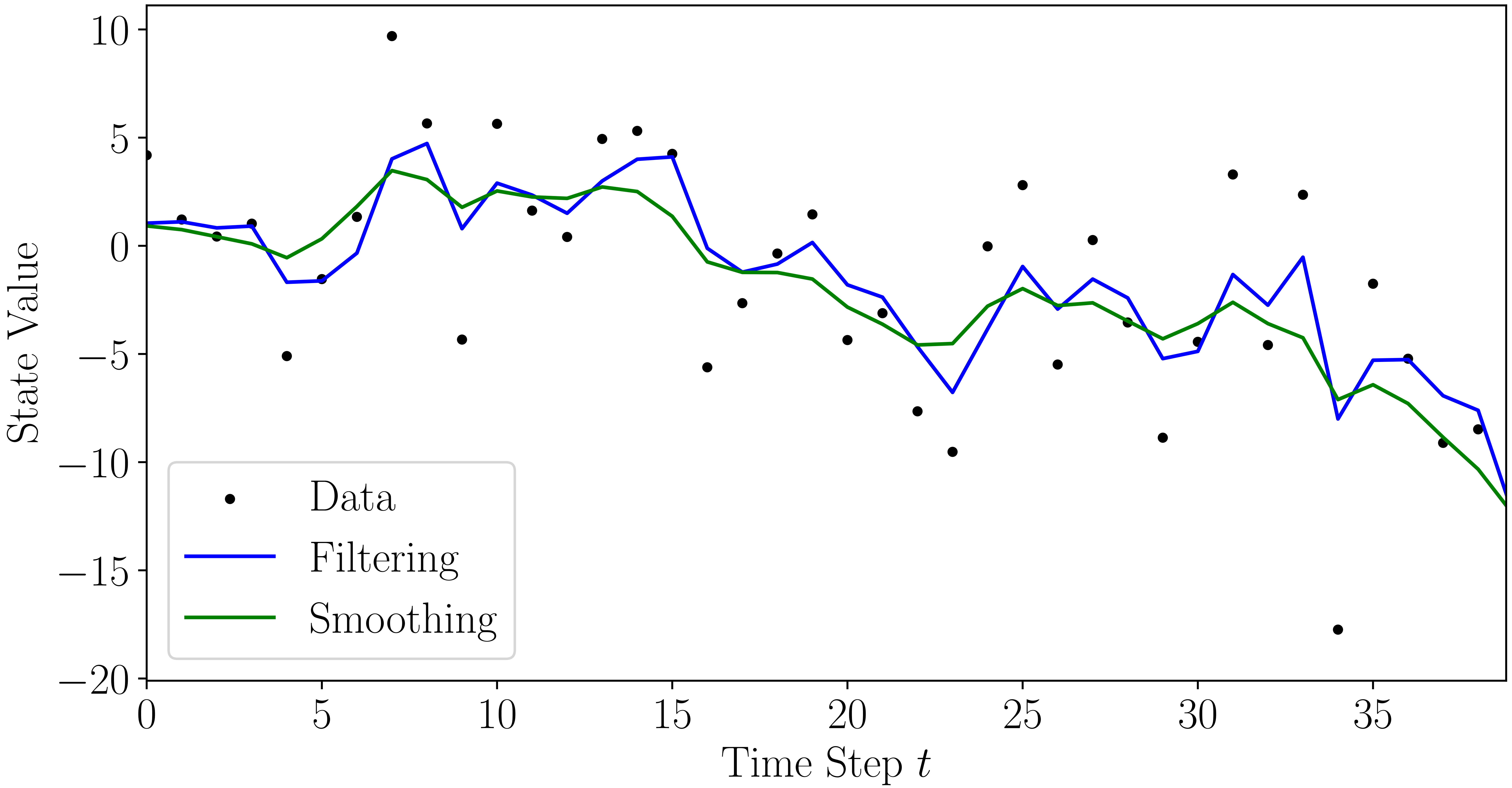
"Discretize" $\Rightarrow$

State-Space Model

$$\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_0)$$
$$\mathbf{z}_k = \mathbf{A}_{k-1}\mathbf{z}_{k-1} + \mathbf{q}_{k-1}, \mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{k-1})$$
$$\mathbf{y}_k = \mathbf{H}\mathbf{z}_k + \epsilon_k$$

Obtain posterior predictive via **filtering & smoothing.**

Filtering & Smoothing



Filtering & Smoothing

1) Filtering: $p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k})$

Filtering & Smoothing

1) Filtering: $p(\mathbf{z}_k \mid \mathbf{y}_{1:k})$ $\longrightarrow$ 2) Smoothing: $p(\mathbf{z}_k \mid \mathbf{y}_{1:N})$

Filtering & Smoothing

1) Filtering: $p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k})$ $\longrightarrow$ 2) Smoothing: $p(\mathbf{z}_k \,|\, \mathbf{y}_{1:N})$

Predict Step:

$p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})$

Filtering

Update Step:

$p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k}) \sim \mathcal{N}(\mathbf{m}_{k|k}, \mathbf{P}_{k|k})$

Filtering & Smoothing

1) Filtering: $p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k})$ → 2) Smoothing: $p(\mathbf{z}_k \,|\, \mathbf{y}_{1:N})$

Predict Step:

$p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})$

Filtering

Update Step:

$p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k}) \sim \mathcal{N}(\mathbf{m}_{k|k}, \mathbf{P}_{k|k})$

Here, we assume
$\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{n_s(\nu+1)})$

Filtering & Smoothing

1) Filtering: $p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k})$ → 2) Smoothing: $p(\mathbf{z}_k \,|\, \mathbf{y}_{1:N})$

Predict Step:

$$p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})$$

Filtering

Update Step:

$$p(\mathbf{z}_k \,|\, \mathbf{y}_{1:k}) \sim \mathcal{N}(\mathbf{m}_{k|k}, \mathbf{P}_{k|k})$$

Here, we assume
$\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{n_s(\nu+1)})$

Why do we care? Linear-in-time cost!

Filtering & Smoothing

1) Filtering: $p(\mathbf{z}_k \mid \mathbf{y}_{1:k})$ → 2) Smoothing: $p(\mathbf{z}_k \mid \mathbf{y}_{1:N})$

Predict Step:

$p(\mathbf{z}_k \mid \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})$

Filtering

Update Step:

$p(\mathbf{z}_k \mid \mathbf{y}_{1:k}) \sim \mathcal{N}(\mathbf{m}_{k|k}, \mathbf{P}_{k|k})$

Here, we assume $\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{n_s(\nu+1)})$

Why do we care? Linear-in-time cost!

Note on Smoothing: Throughout we use the Rauch-Tung-Striebel (RTS) smoother. Closed-form, for linear Gaussian models

Summary: STGPs

Used for **any** temporal data lying on a grid; not just “spatial” points!

Summary: STGPs

Used for **any** temporal data lying on a grid; not just “spatial” points!

Benefits:

- Reformulate GPs in state-space model setting for **linear-in-time cost**
- Replicate **exactly** GP posterior (i.e. no approximations)

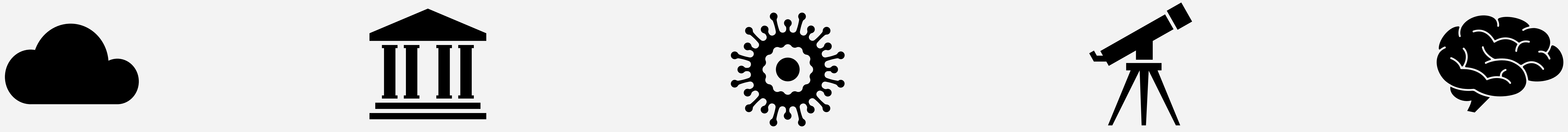
Summary: STGPs

Used for **any** temporal data lying on a grid; not just “spatial” points!

Benefits:

- Reformulate GPs in state-space model setting for **linear-in-time cost**
- Replicate **exactly** GP posterior (i.e. no approximations)

Applications



STGPs are great! What's the problem then?

Problems With STGPs

Noise typically distributed as $\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ →

Great for conjugacy and comp. cost!

Terrible for robustness against outliers…

Problems With STGPs

Noise typically distributed as $\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ → Great for conjugacy and comp. cost!

Terrible for robustness against outliers…

Solution? *Approximate* methods: variational inference (VI), Laplace approximation, EP

Problems With STGPs

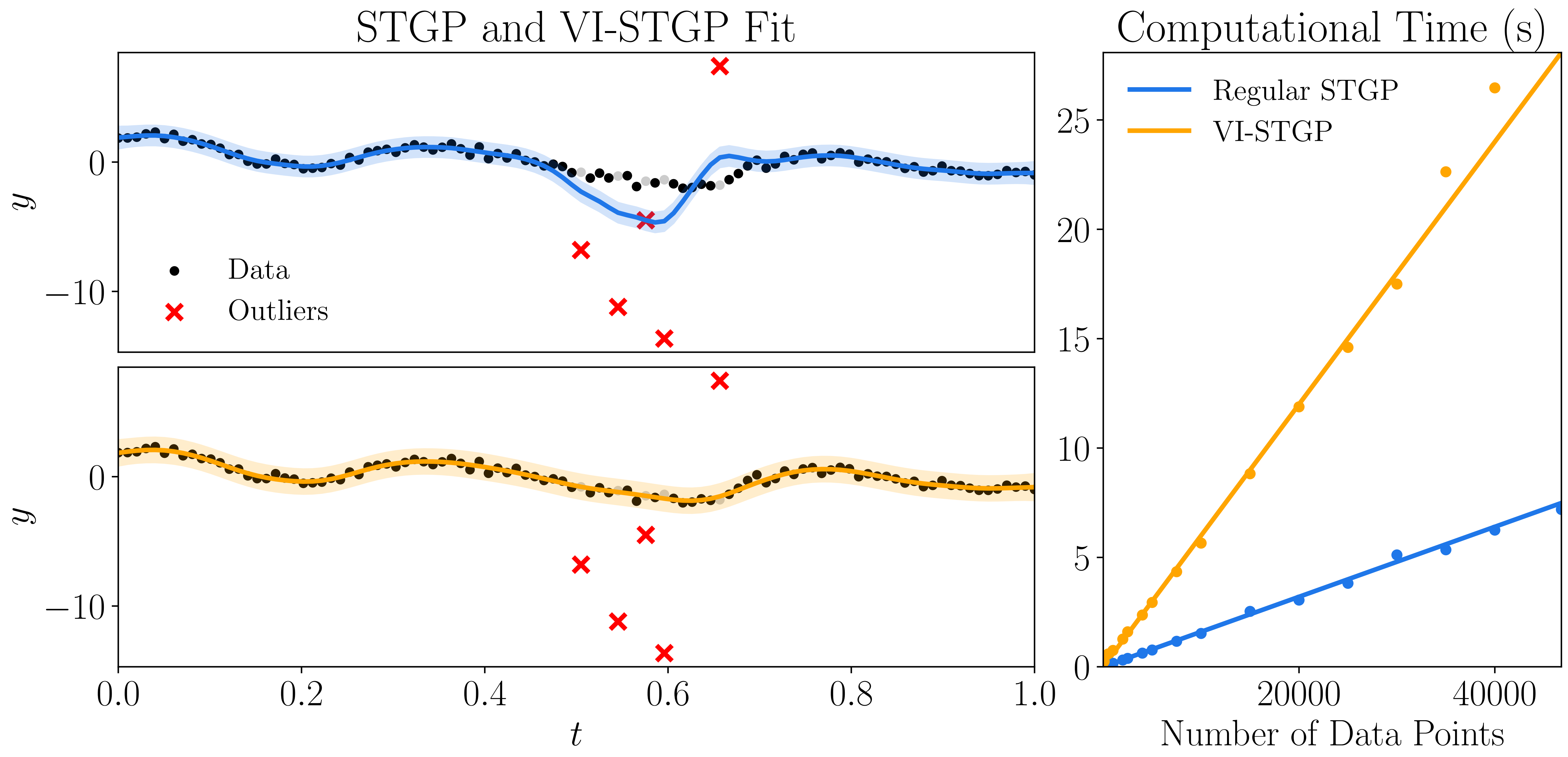
Noise typically distributed as $\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$

Great for conjugacy and comp. cost!

Terrible for robustness against outliers…

Solution? *Approximate* methods: variational inference (VI), Laplace approximation, EP

Tradeoff between robustness & comp. cost

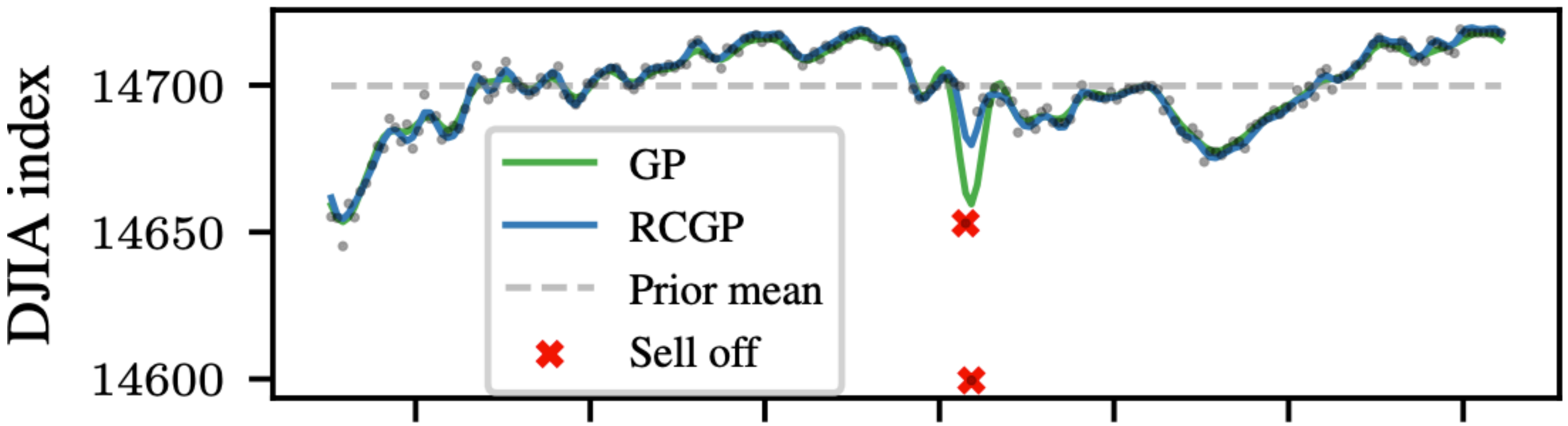


Solution To This Tradeoff?

We adapt the *robust and conjugate GP* (RCGP) from Altamirano et al. (2024).

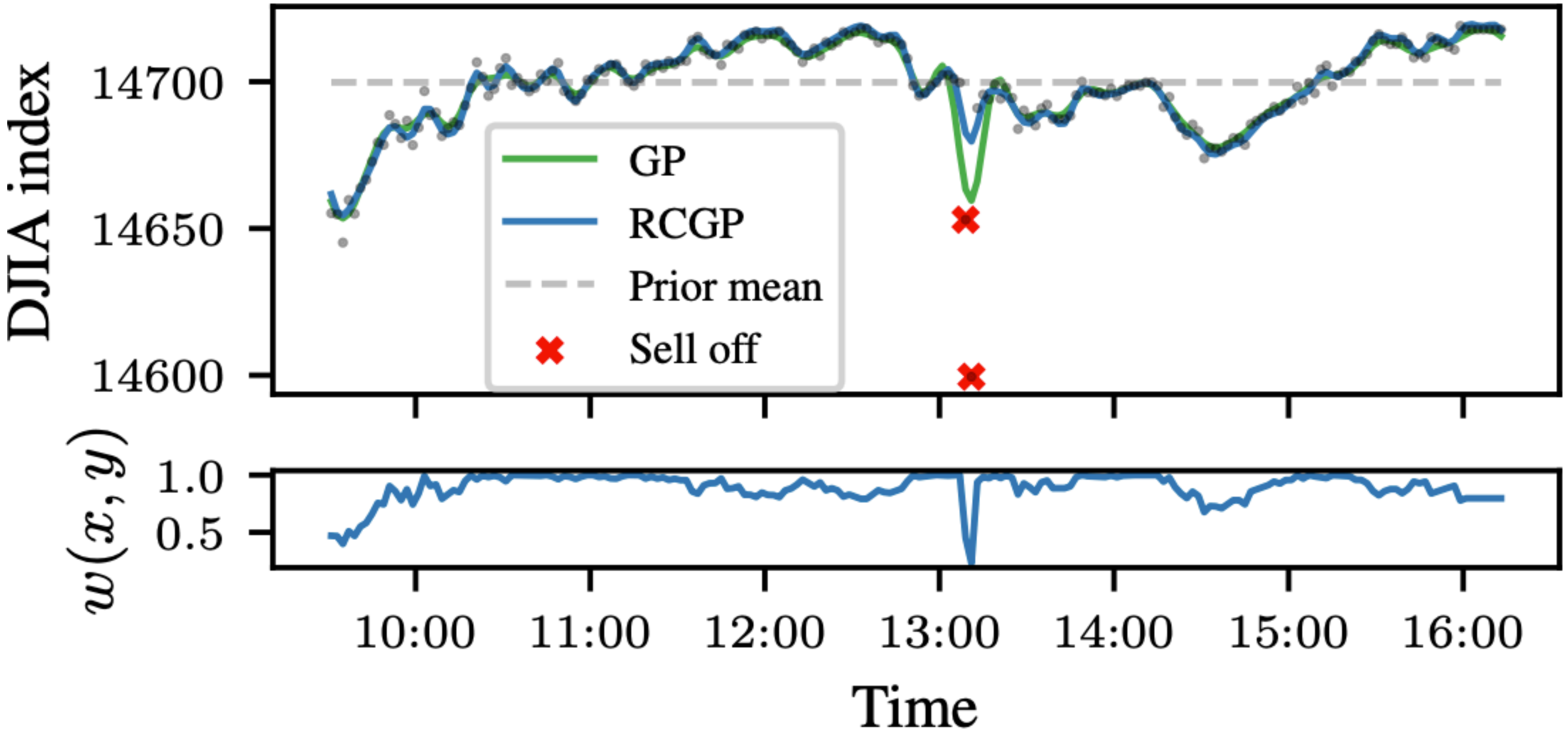
Solution To This Tradeoff?

We adapt the *robust and conjugate GP* (RCGP) from Altamirano et al. (2024).



Solution To This Tradeoff?

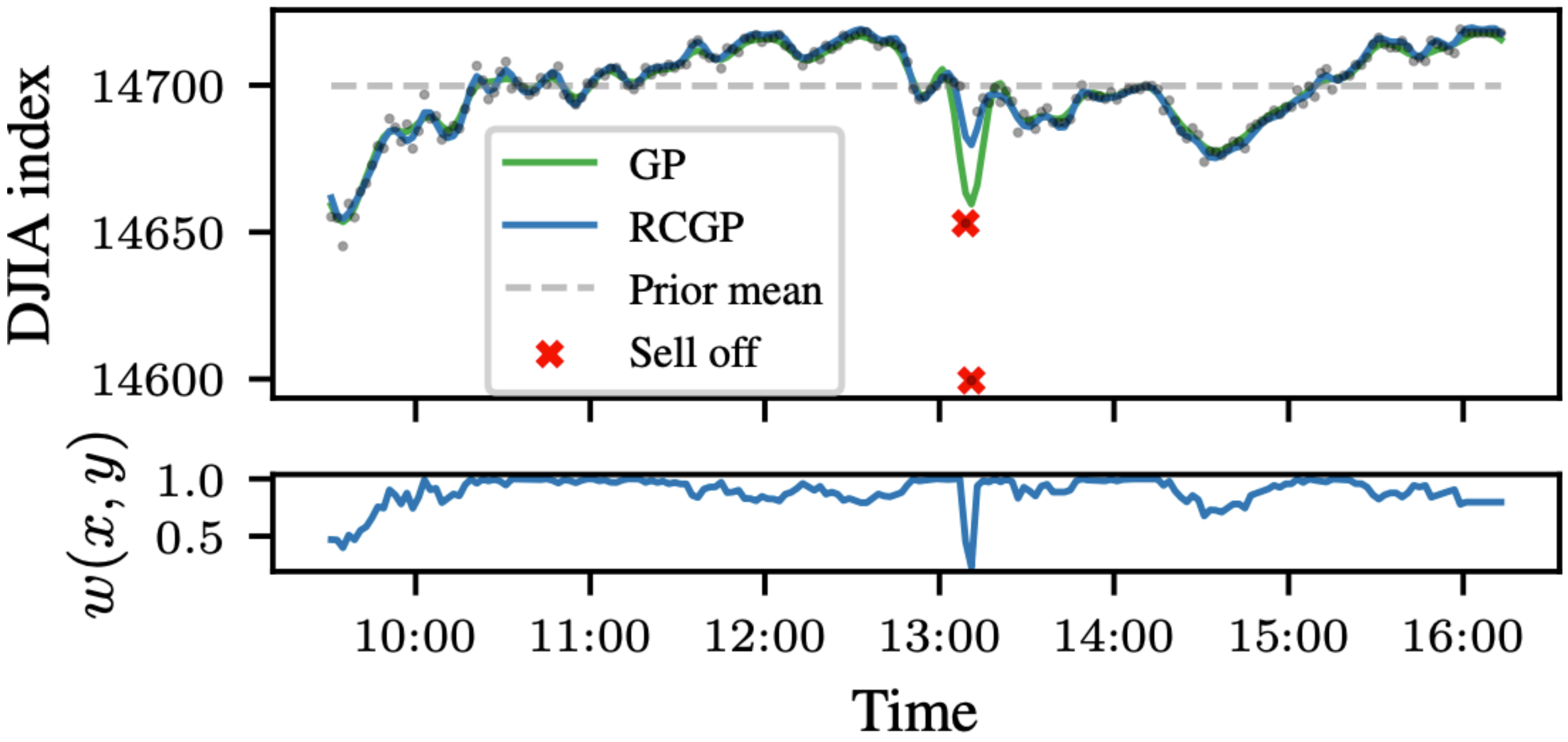
We adapt the *robust and conjugate GP* (RCGP) from Altamirano et al. (2024).



"Weight Function"

Solution To This Tradeoff?

We adapt the *robust and conjugate GP* (RCGP) from Altamirano et al. (2024).



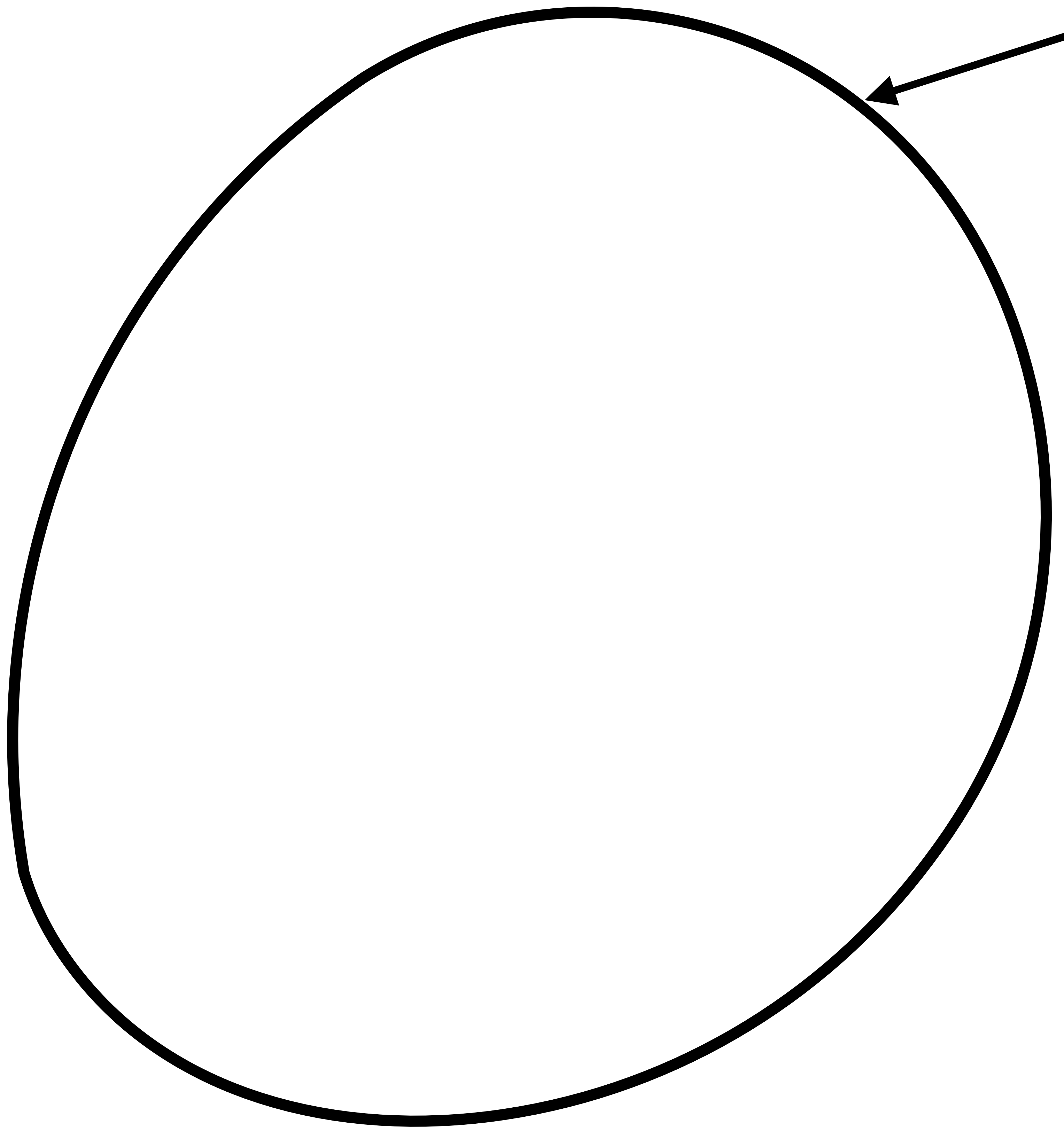
"Weight Function"

Key Idea: Introduce a weight function $w : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ that **downweights outliers**

RCGP's backbone: Generalised Bayesian Inference

$$\{\mathscr{B} : \underbrace{\mathscr{P}(\mathscr{F})}_{\text{space of priors}} \times \overbrace{\mathbb{R}^N}^{\text{data space}} \to \underbrace{\mathscr{P}(\mathscr{F})}_{\text{space of posteriors}}\}$$

Possible belief updates



RCGP's backbone: Generalised Bayesian Inference

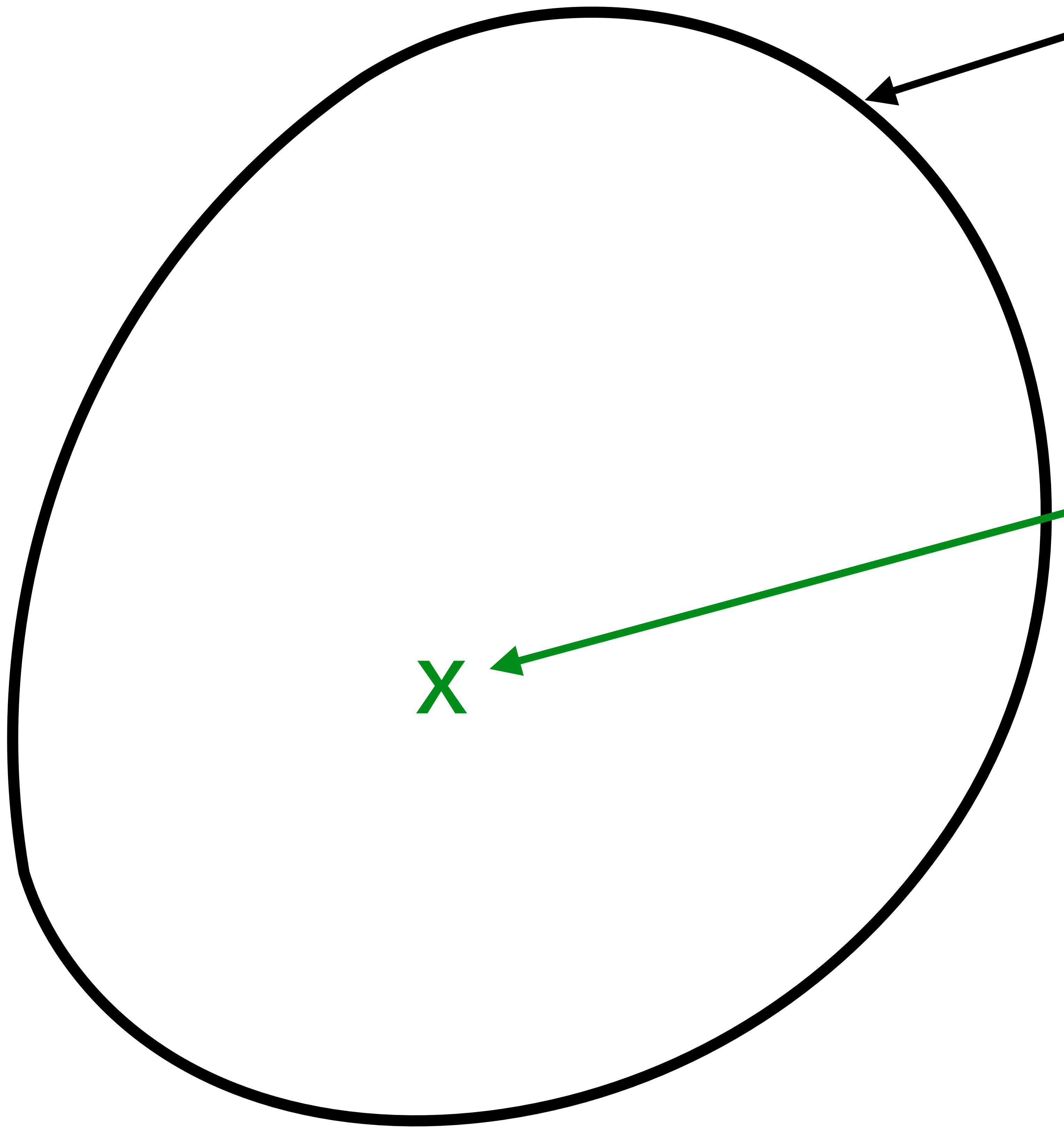
data space

$$\{\mathscr{B} : \mathscr{P}(\mathscr{F}) \times \mathbb{R}^N \to \mathscr{P}(\mathscr{F})\}$$

space of priors

space of posteriors

Possible belief updates



Standard Bayes' Posterior

$$\boxed{p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{x}) \cdot p(\mathbf{f} \,|\, \mathbf{x})}$$

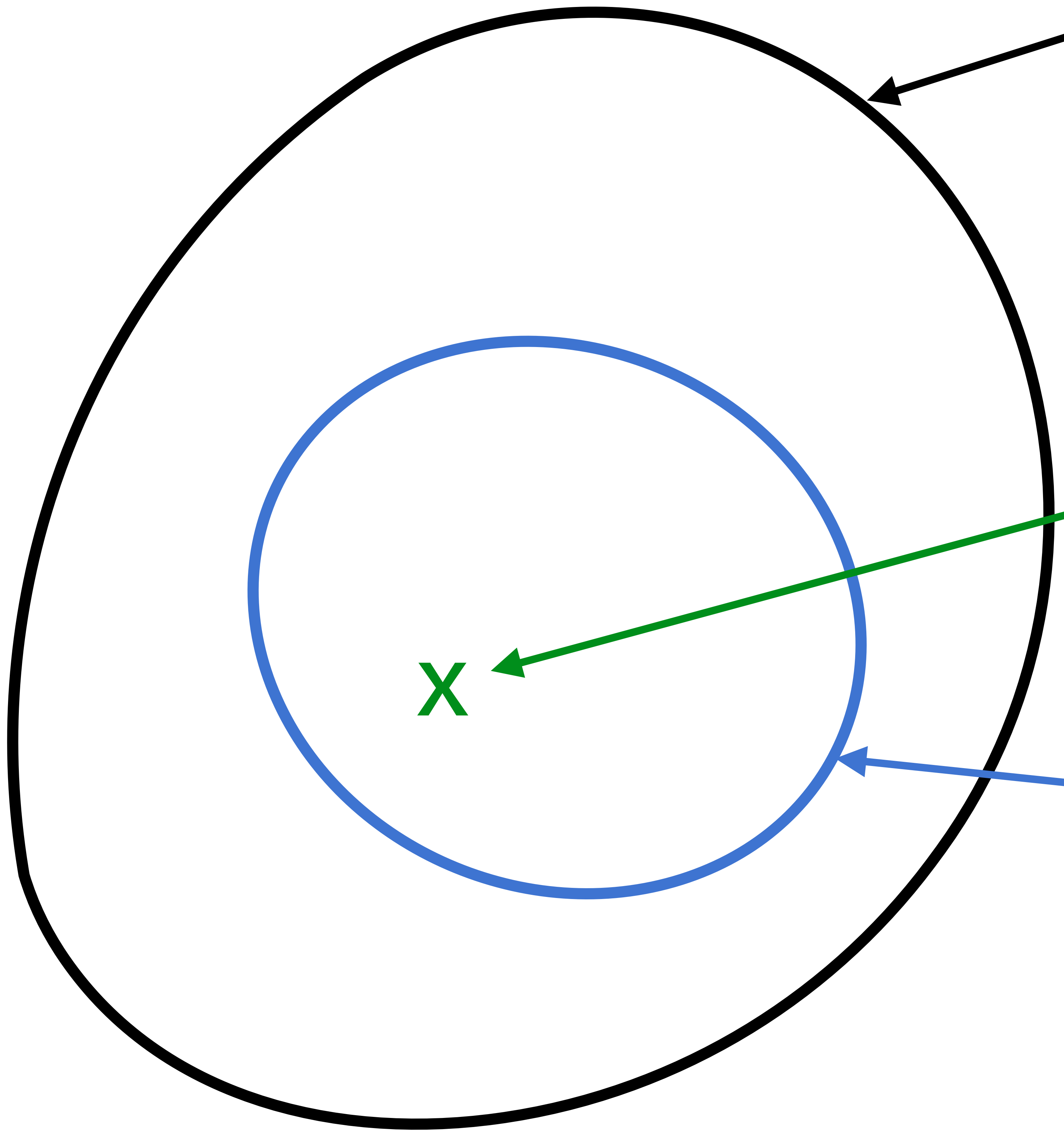
RCGP's backbone: Generalised Bayesian Inference

data space

$$\{\mathscr{B} : \underbrace{\mathscr{P}(\mathscr{F})}_{\text{space of priors}} \times \overbrace{\mathbb{R}^N}^{\text{data space}} \to \underbrace{\mathscr{P}(\mathscr{F})}_{\text{space of posteriors}}\}$$

space of priors space of posteriors

Possible belief updates



Standard Bayes' Posterior

$$p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{x}) \cdot p(\mathbf{f} \,|\, \mathbf{x})$$

Generalised Bayes' Posterior

$$p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-NL_N(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \cdot p(\mathbf{f} \,|\, \mathbf{x})$$

RCGP's backbone: Generalised Bayesian Inference

$$\boxed{p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-NL_N(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \cdot p(\mathbf{f} \,|\, \mathbf{x})}$$

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

RCGP's backbone: Generalised Bayesian Inference

$$\boxed{p(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) \propto \exp\left(-NL_N(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \cdot p(\mathbf{f} \mid \mathbf{x})}$$

Goal: select L to obtain robustness and tractability

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

RCGP's backbone: Generalised Bayesian Inference

$$\boxed{p(\mathbf{f} \mid \mathbf{y}, \mathbf{x}) \propto \exp\left(-NL_N(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \cdot p(\mathbf{f} \mid \mathbf{x})}$$

Goal: select L to obtain robustness and tractability

Under mild conditions $p(\mathbf{f} \mid \mathbf{y}, \mathbf{x})$ concentrates around $\mathbf{f}^\star = \mathrm{argmin}_{\mathbf{f} \in \mathbb{R}^N} L_N(\mathbf{f}, \mathbf{y}, \mathbf{x})$

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

RCGP's backbone: Generalised Bayesian Inference

$$\boxed{p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x}) \propto \exp\left(-NL_N(\mathbf{f}, \mathbf{y}, \mathbf{x})\right) \cdot p(\mathbf{f} \,|\, \mathbf{x})}$$

Goal: select L to obtain robustness and tractability

Under mild conditions $p(\mathbf{f} \,|\, \mathbf{y}, \mathbf{x})$ concentrates around $\mathbf{f}^\star = \mathrm{argmin}_{\mathbf{f} \in \mathbb{R}^N} L_N(\mathbf{f}, \mathbf{y}, \mathbf{x})$

Divergence-based losses are a good option! $D(\mathbb{P}, \mathbb{Q}) \geq 0$ $D(\mathbb{P}, \mathbb{Q}) = 0 \iff \mathbb{P} = \mathbb{Q}$

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

Divergence Used in RCGP: (Weighted) Score-matching

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

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

Altamirano, M., Briol, F.X. and Knoblauch, J. (2023). Robust and scalable bayesian online changepoint detection. *International Conference on Machine Learning* 642-663.

Divergence Used in RCGP: (Weighted) Score-matching

The score-matching divergence is given by:

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

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

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

Altamirano, M., Briol, F.X. and Knoblauch, J. (2023). Robust and scalable bayesian online changepoint detection. *International Conference on Machine Learning* 642-663.

Divergence Used in RCGP: (Weighted) Score-matching

The score-matching divergence is given by:

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

We consider a weighted generalisation based on $w : \mathscr{X} \to \mathbb{R}$:

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

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

Altamirano, M., Briol, F.X. and Knoblauch, J. (2023). Robust and scalable bayesian online changepoint detection. *International Conference on Machine Learning* 642-663.

Divergence Used in RCGP: (Weighted) Score-matching

The score-matching divergence is given by:

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

We consider a weighted generalisation based on $w : \mathscr{X} \to \mathbb{R}$:

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

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

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

Altamirano, M., Briol, F.X. and Knoblauch, J. (2023). Robust and scalable bayesian online changepoint detection. *International Conference on Machine Learning* 642-663.

RCGP: Putting It All Together

The generalised posterior in a GP regression setting is given by

$$p(\mathbf{f} \mid y_{1:N}, \mathbf{x}_{1:N}) \propto p(\mathbf{f} \mid \mathbf{x}_{1:N}) \ \exp\left\{ -NL_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) \right\}$$

RCGP: Putting It All Together

The generalised posterior in a GP regression setting is given by

$$p(\mathbf{f} | y_{1:N}, \mathbf{x}_{1:N}) \propto p(\mathbf{f} \mid \mathbf{x}_{1:N}) \ \exp\Big\{ -NL_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) \Big\}$$

$L_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) = -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \ln p(y_i \mid f(\mathbf{x}_i))$ recovers Bayes, where $p(y_i \mid f(\mathbf{x}_i)) = \mathcal{N}(f(\mathbf{x}_i), \sigma^2)$

RCGP: Putting It All Together

The generalised posterior in a GP regression setting is given by

$$p(\mathbf{f} \mid y_{1:N}, \mathbf{x}_{1:N}) \propto p(\mathbf{f} \mid \mathbf{x}_{1:N}) \ \exp\left\{ -NL_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) \right\}$$

$L_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) = -\frac{1}{N}\sum_{i=1}^{N} \ln p(y_i \mid f(\mathbf{x}_i))$ recovers Bayes, where $p(y_i \mid f(\mathbf{x}_i)) = \mathcal{N}(f(\mathbf{x}_i), \sigma^2)$

RCGP: Putting It All Together

The generalised posterior in a GP regression setting is given by

$$p(\mathbf{f} \mid y_{1:N}, \mathbf{x}_{1:N}) \propto p(\mathbf{f} \mid \mathbf{x}_{1:N}) \; \exp\left\{ -NL_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) \right\}$$

$L_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) = -\frac{1}{N}\sum_{i=1}^{N} \ln p(y_i \mid f(\mathbf{x}_i))$ recovers Bayes, where $p(y_i \mid f(\mathbf{x}_i)) \cancel{= \mathcal{N}}(f(\mathbf{x}_i), \sigma^2)$

Instead , with the score $s(\mathbf{x}, y) := -\nabla_y \ln p(y \mid f(\mathbf{x}))$ and a weight function w,

RCGP: Putting It All Together

The generalised posterior in a GP regression setting is given by

$$p(\mathbf{f} \,|\, y_{1:N}, \mathbf{x}_{1:N}) \propto p(\mathbf{f} \mid \mathbf{x}_{1:N}) \ \exp\left\{ -NL_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) \right\}$$

$L_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) = -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \ln p(y_i \mid f(\mathbf{x}_i))$ recovers Bayes, where $p(y_i \mid f(\mathbf{x}_i)) \cancel{=} \mathcal{N}(f(\mathbf{x}_i), \sigma^2)$

Instead , with the score $s(\mathbf{x}, y) := -\nabla_y \ln p(y \mid f(\mathbf{x}))$ and a weight function w,

$$L_N^{\mathrm{RCGP}}(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) = \frac{1}{N}\sum_{i=1}^{N} (w(\mathbf{x}_i, y_i) \cdot s(\mathbf{x}_i, y_i))^2 + 2\,\nabla_y (w(\mathbf{x}_i, y_i)^2 \cdot s(\mathbf{x}_i, y_i)) = \mathbf{f}^\top A_N \mathbf{f} + b_N^\top \mathbf{f} + c_N$$

RCGP: Putting It All Together

The generalised posterior in a GP regression setting is given by

$$p(\mathbf{f} \,|\, y_{1:N}, \mathbf{x}_{1:N}) \propto p(\mathbf{f} \mid \mathbf{x}_{1:N}) \;\exp\Big\{ -NL_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) \Big\}$$

$L_N(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) = -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \ln p(y_i \mid f(\mathbf{x}_i))$ recovers Bayes, where $p(y_i \mid f(\mathbf{x}_i)) \cancel{= \mathcal{N}}(f(\mathbf{x}_i), \sigma^2)$

Instead , with the score $s(\mathbf{x}, y) := -\nabla_y \ln p(y \mid f(\mathbf{x}))$ and a weight function w,

$$L_N^{\mathrm{RCGP}}(\mathbf{f}, y_{1:N}, \mathbf{x}_{1:N}) = \frac{1}{N}\sum_{i=1}^{N} (w(\mathbf{x}_i, y_i) \cdot s(\mathbf{x}_i, y_i))^2 + 2\nabla_y (w(\mathbf{x}_i, y_i)^2 \cdot s(\mathbf{x}_i, y_i)) = \mathbf{f}^\top A_N \mathbf{f} + b_N^\top \mathbf{f} + c_N$$

Therefore, $p(\mathbf{f} \mid y_{1:N}, \mathbf{x}_{1:N})$ depends on weights (robust), and is conjugate with GP prior!

Key Idea of RCGPs: The Weight Function

Weight function used is the Inverse Multi-Quadratic (IMQ) kernel

$$w_{IMQ}(\mathbf{x}, y) := \beta\left(1 + \frac{(y - \gamma(\mathbf{x}))^2}{c(\mathbf{x})^2}\right)^{-1/2} \quad \longleftrightarrow \quad \text{“Statistically efficient”}$$

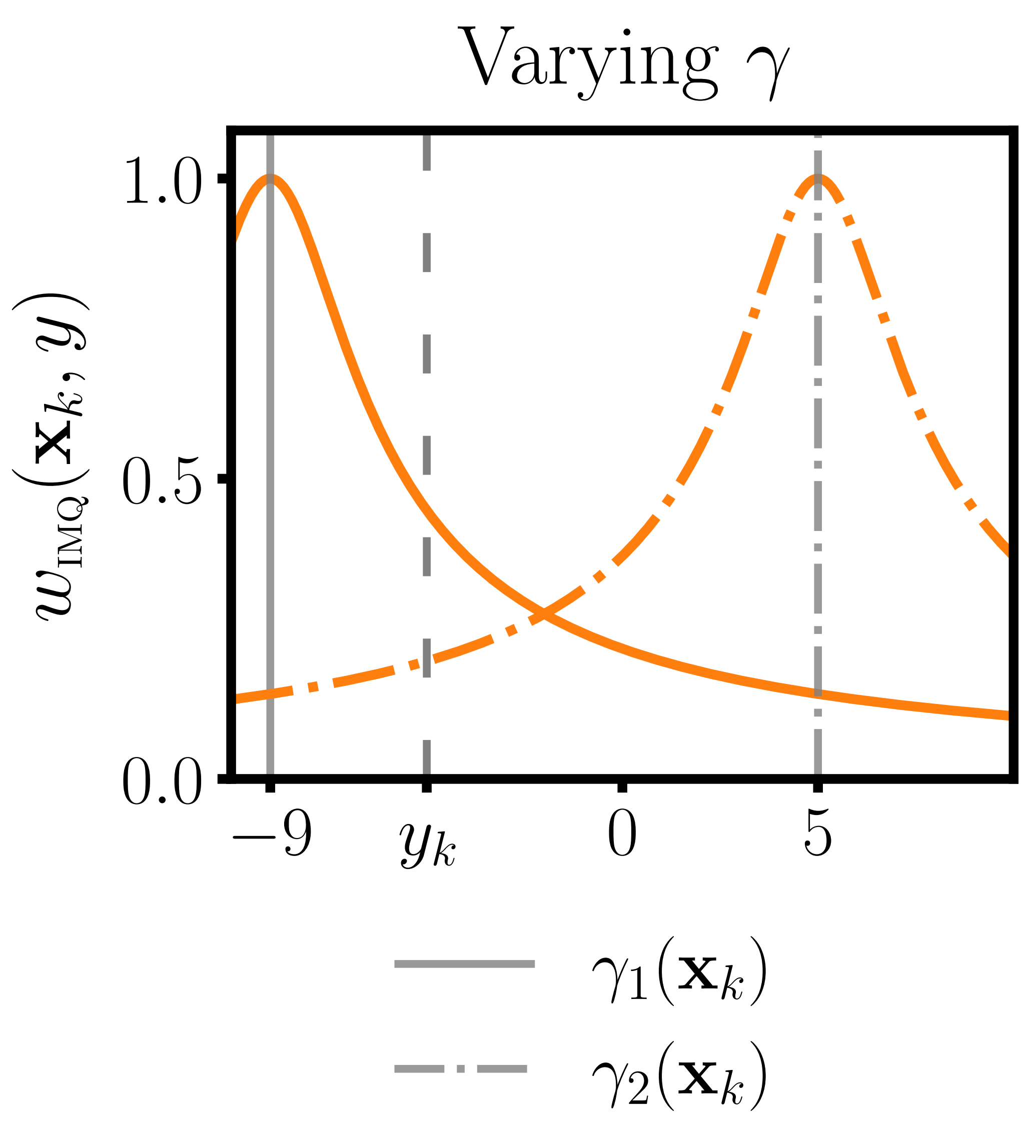
Key Idea of RCGPs: The Weight Function

Weight function used is the Inverse Multi-Quadratic (IMQ) kernel

$$w_{IMQ}(\mathbf{x}, y) := \beta \left(1 + \frac{(y - \gamma(\mathbf{x}))^2}{c(\mathbf{x})^2} \right)^{-1/2}$$

"Statistically efficient"

$\gamma(\mathbf{x})$: "Centering function"



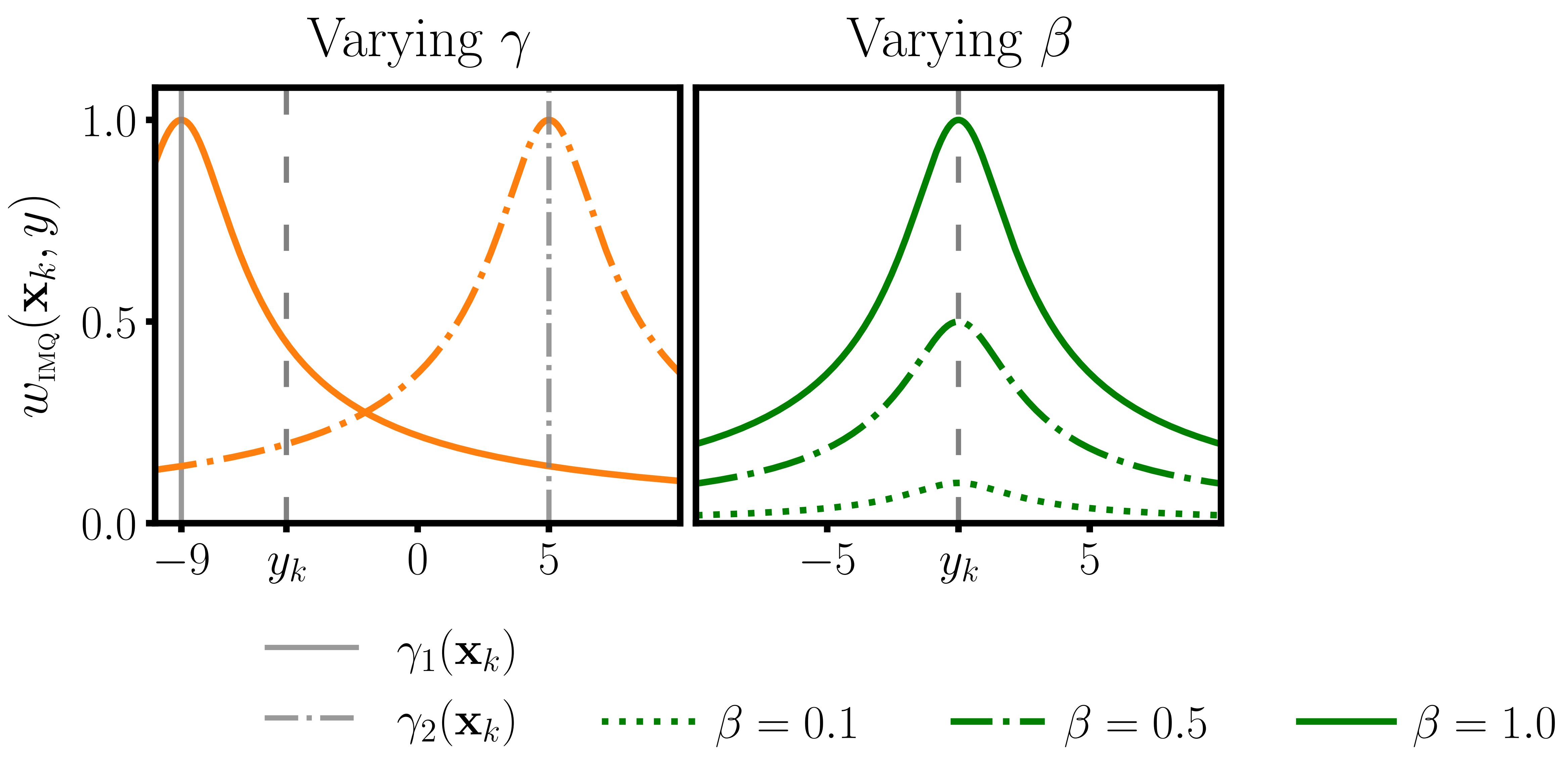
Key Idea of RCGPs: The Weight Function

Weight function used is the Inverse Multi-Quadratic (IMQ) kernel

$$w_{IMQ}(\mathbf{x}, y) := \beta \left(1 + \frac{(y - \gamma(\mathbf{x}))^2}{c(\mathbf{x})^2} \right)^{-1/2} \quad \longleftrightarrow \quad \text{"Statistically efficient"}$$

$\gamma(\mathbf{x})$: "Centering function"

β: "Learning rate"



Key Idea of RCGPs: The Weight Function

Weight function used is the Inverse Multi-Quadratic (IMQ) kernel

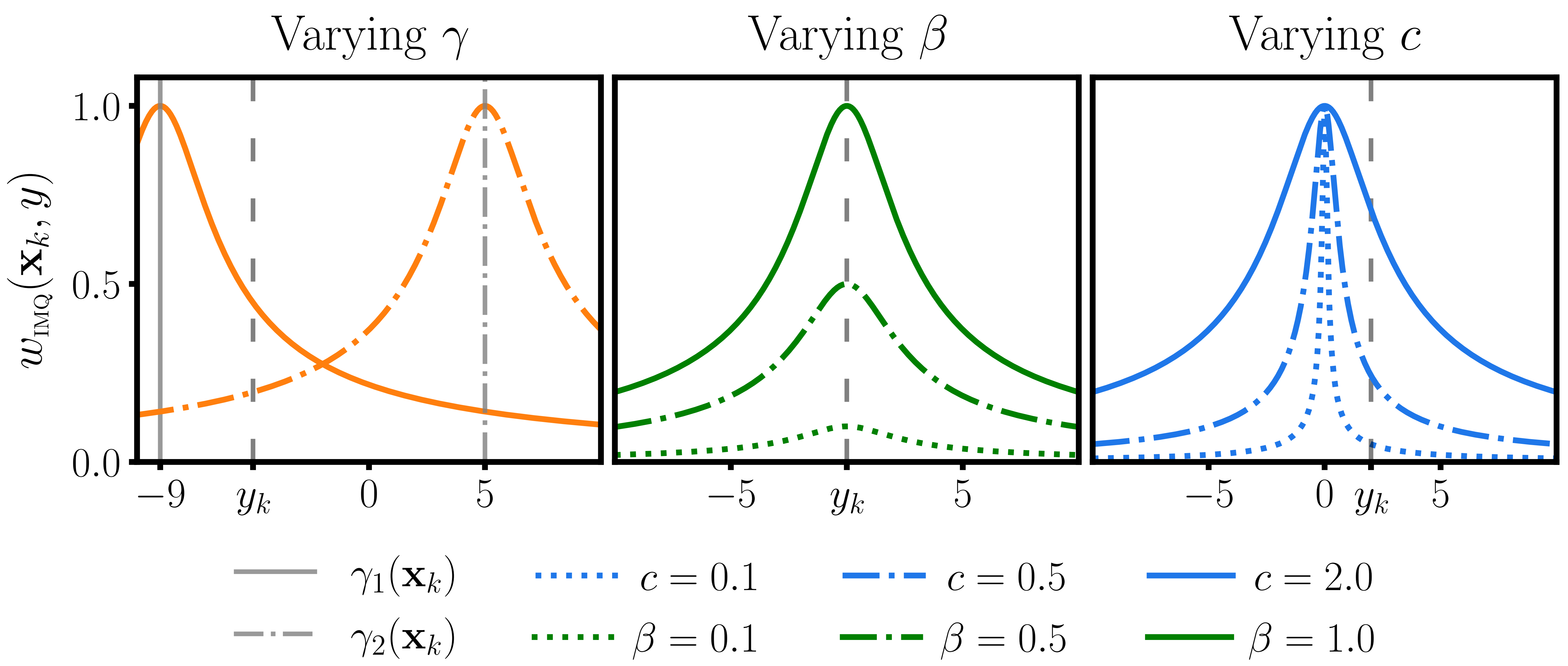
$$w_{IMQ}(\mathbf{x}, y) := \beta \left(1 + \frac{(y - \gamma(\mathbf{x}))^2}{c(\mathbf{x})^2} \right)^{-1/2}$$

"Statistically efficient"

$\gamma(\mathbf{x})$: "Centering function"

β: "Learning rate"

$c(\mathbf{x})$: "Shrinking function"



<u>Exact</u> Translation to State-Space Setting: ST-RCGP

Inference: Filtering Update Equations

Exact Translation to State-Space Setting: ST-RCGP

Inference: Filtering Update Equations

Before (STGP)

$$\mathbf{P}_{k|k} := \left(\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}^\top \sigma^{-2} \mathbf{I}_{n_s(\nu+1)} \mathbf{H}\right)^{-1}$$

$$\mathbf{K}_k := \mathbf{P}_{k|k} \mathbf{H}^\top \sigma^{-2} \mathbf{I}_{n_s(\nu+1)}$$

$$\mathbf{m}_{k|k} := \mathbf{m}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{f}}_k),$$

Exact Translation to State-Space Setting: ST-RCGP

Inference: Filtering Update Equations

Before (STGP)

$$\mathbf{P}_{k|k} := \left(\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}^\top \sigma^{-2} \mathbf{I}_{n_s(\nu+1)} \mathbf{H}\right)^{-1}$$

$$\mathbf{K}_k := \mathbf{P}_{k|k} \mathbf{H}^\top \sigma^{-2} \mathbf{I}_{n_s(\nu+1)}$$

$$\mathbf{m}_{k|k} := \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{f}}_k),$$

After (ST-RCGP)

$$\mathbf{P}_{k|k}^{GB} := \left(\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}^\top \sigma^{-2} \mathbf{J}_{\mathbf{w}_k}^{-1} \mathbf{H}\right)^{-1}$$

$$\mathbf{K}_k^{GB} := \mathbf{P}_{k|k}^{GB} \mathbf{H}^\top \sigma^{-2} \mathbf{J}_{\mathbf{w}_k}^{-1}$$

$$\mathbf{m}_{k|k}^{GB} := \mathbf{m}_{k|k-1} + \mathbf{K}_k^{GB} (\mathbf{y}_k - \hat{\mathbf{f}}_{w_k}),$$

Exact Translation to State-Space Setting: ST-RCGP

Inference: Filtering Update Equations

Before (STGP)

$$\mathbf{P}_{k|k} := \left(\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}^\top \sigma^{-2} \mathbf{I}_{n_s(\nu+1)} \mathbf{H}\right)^{-1}$$

$$\mathbf{K}_k := \mathbf{P}_{k|k} \mathbf{H}^\top \sigma^{-2} \mathbf{I}_{n_s(\nu+1)}$$

$$\mathbf{m}_{k|k} := \mathbf{m}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{f}}_k),$$

After (ST-RCGP)

$$\mathbf{P}_{k|k}^{GB} := \left(\mathbf{P}_{k|k-1}^{-1} + \mathbf{H}^\top \sigma^{-2} \mathbf{J}_{\mathbf{w}_k}^{-1} \mathbf{H}\right)^{-1}$$

$$\mathbf{K}_k^{GB} := \mathbf{P}_{k|k}^{GB} \mathbf{H}^\top \sigma^{-2} \mathbf{J}_{\mathbf{w}_k}^{-1}$$

$$\mathbf{m}_{k|k}^{GB} := \mathbf{m}_{k|k-1} + \mathbf{K}_k^{GB}(\mathbf{y}_k - \hat{\mathbf{f}}_{w_k}),$$

Where $\mathbf{J}_{\mathbf{w}_k} := \operatorname{diag}\left(\frac{\sigma^2}{2}\mathbf{w}_k^{-2}\right)$ and $\hat{\mathbf{f}}_{w_k} := \hat{\mathbf{f}}_k + \sigma^2 \nabla_y \log(\mathbf{w}_k^2)$

For weights $\mathbf{w}_k = (w(\mathbf{x}_{k,1}, y_{k,1}), \ldots, w(\mathbf{x}_{k,n_s}, y_{k,n_s}))^\top$

Spatio-temporal RCGPs now have linear-in-time cost!

Spatio-temporal RCGPs now have linear-in-time cost!

Are we done?

Spatio-temporal RCGPs now have linear-in-time cost!

Are we done?

Not quite…

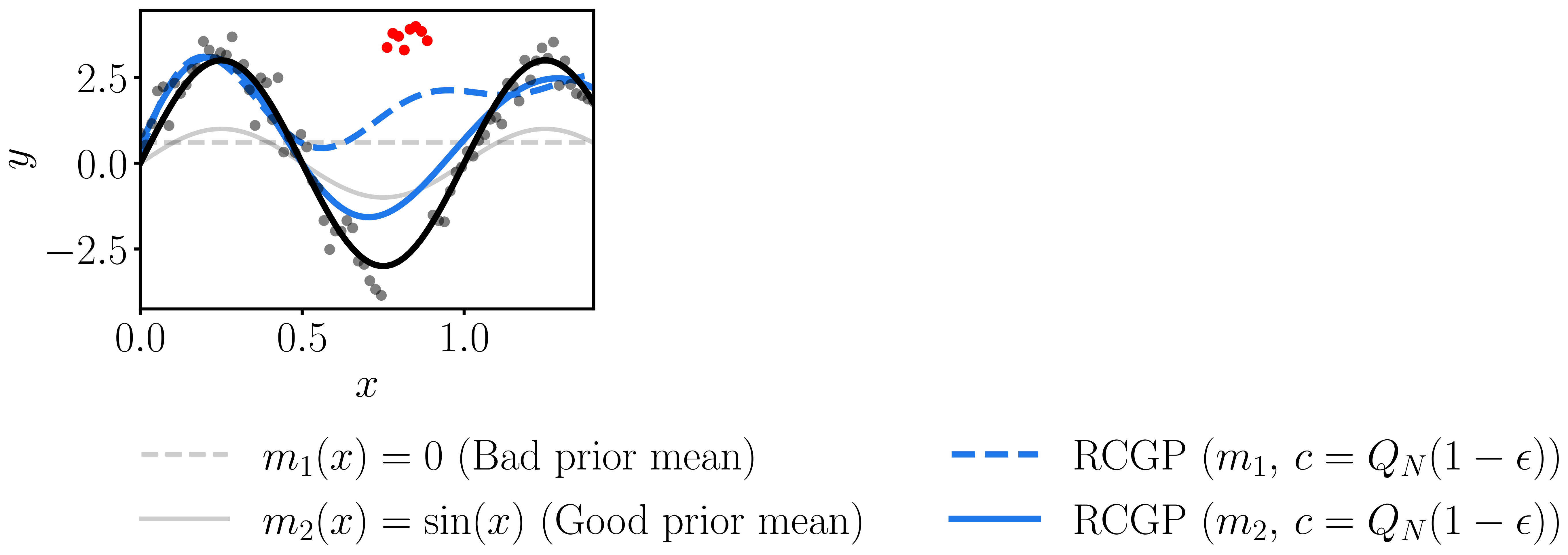
RCGP Problems Bleeding Into Our Method…

The weight function in the RCGP introduces new parameters. Gives rise to **3 issues:**

RCGP Problems Bleeding Into Our Method...

The weight function in the RCGP introduces new parameters. Gives rise to **3 issues:**

Issue #1: Sensitivity to GP prior mean specification (unreliable centering function)

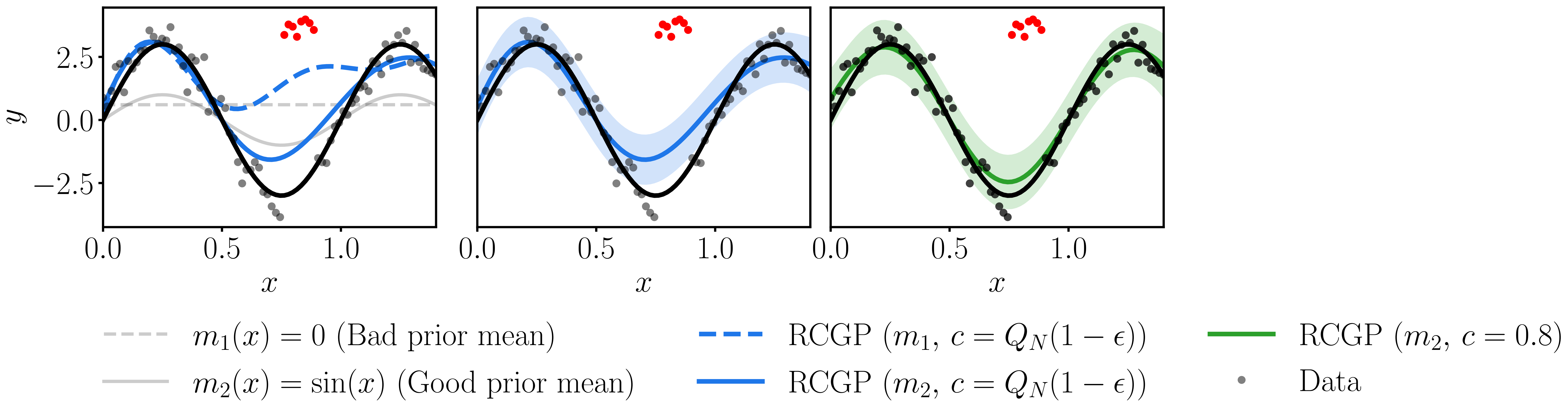


RCGP Problems Bleeding Into Our Method…

The weight function in the RCGP introduces new parameters. Gives rise to **3 issues:**

Issue #1: Sensitivity to GP prior mean specification (unreliable centering function)

Issue #2: Difficult selection of shrinking function (must choose by hand)



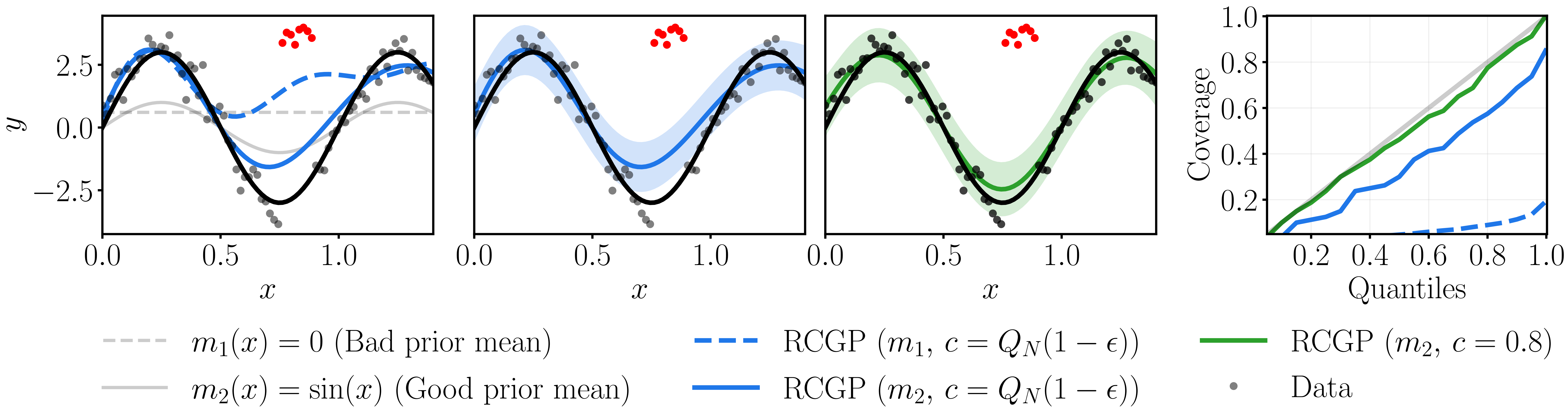
RCGP Problems Bleeding Into Our Method...

The weight function in the RCGP introduces new parameters. Gives rise to **3 issues:**

Issue #1: Sensitivity to GP prior mean specification (unreliable centering function)

Issue #2: Difficult selection of shrinking function (must choose by hand)

Issue #3: Poor uncertainty quantification (when parameters are wrongly specified)



The Remedy Enabled by Sequential Inference

We use the Generalised Bayes filtering predictive $p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\hat{\mathbf{f}}, \hat{\mathbf{S}})$, where

$$\hat{\mathbf{f}} := \mathbf{H}\mathbf{m}_{k|k-1} \qquad \hat{\mathbf{S}} := \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \sigma^2 \mathbf{I}_{n_s}$$

The Remedy Enabled by Sequential Inference

We use the Generalised Bayes filtering predictive $p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\hat{\mathbf{f}}, \hat{\mathbf{S}})$, where

$$\hat{\mathbf{f}} := \mathbf{H}\mathbf{m}_{k|k-1} \qquad \hat{\mathbf{S}} := \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \sigma^2\mathbf{I}_{n_s}$$

We choose:

Centering Function

$$\gamma := \hat{\mathbf{f}}$$

The Remedy Enabled by Sequential Inference

We use the Generalised Bayes filtering predictive $p(\mathbf{y}_k \,|\, \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\hat{\mathbf{f}}, \hat{\mathbf{S}})$, where

$$\hat{\mathbf{f}} := \mathbf{H}\mathbf{m}_{k|k-1} \qquad\qquad \hat{\mathbf{S}} := \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \sigma^2\mathbf{I}_{n_s}$$

We choose:

Centering Function

$$\gamma := \hat{\mathbf{f}}$$

Shrinking Function

$$\mathbf{c} := diag(\hat{\mathbf{S}})$$

The Remedy Enabled by Sequential Inference

We use the Generalised Bayes filtering predictive $p(\mathbf{y}_k \,|\, \mathbf{y}_{1:k-1}) \sim \mathcal{N}(\hat{\mathbf{f}}, \hat{\mathbf{S}})$, where

$$\hat{\mathbf{f}} := \mathbf{H}\mathbf{m}_{k|k-1} \qquad\qquad \hat{\mathbf{S}} := \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^\top + \sigma^2 \mathbf{I}_{n_s}$$

We choose:

Centering Function

$$\gamma := \hat{\mathbf{f}}$$

Shrinking Function

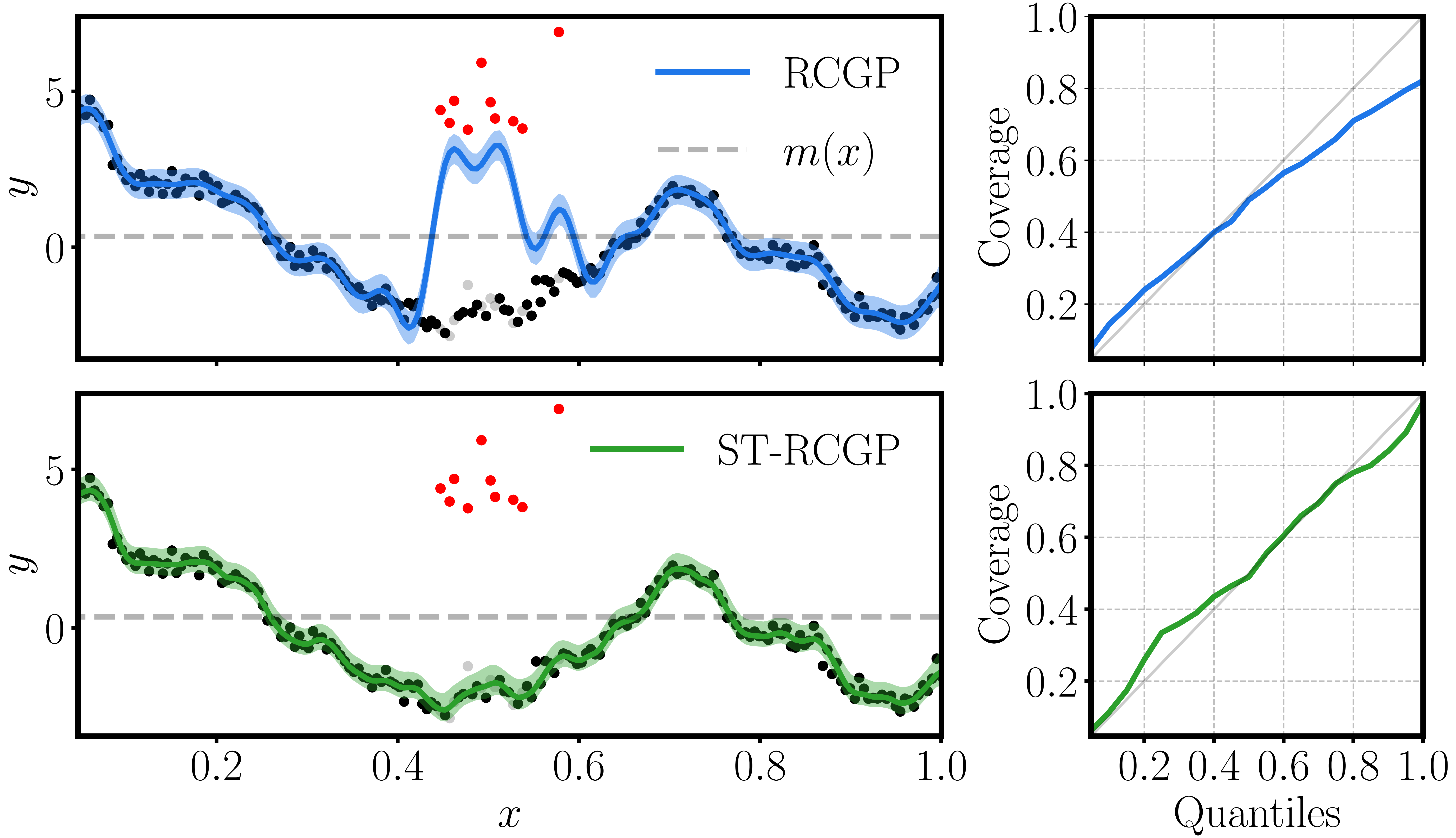
$$\mathbf{c} := diag(\hat{\mathbf{S}})$$

"Learning Rate"

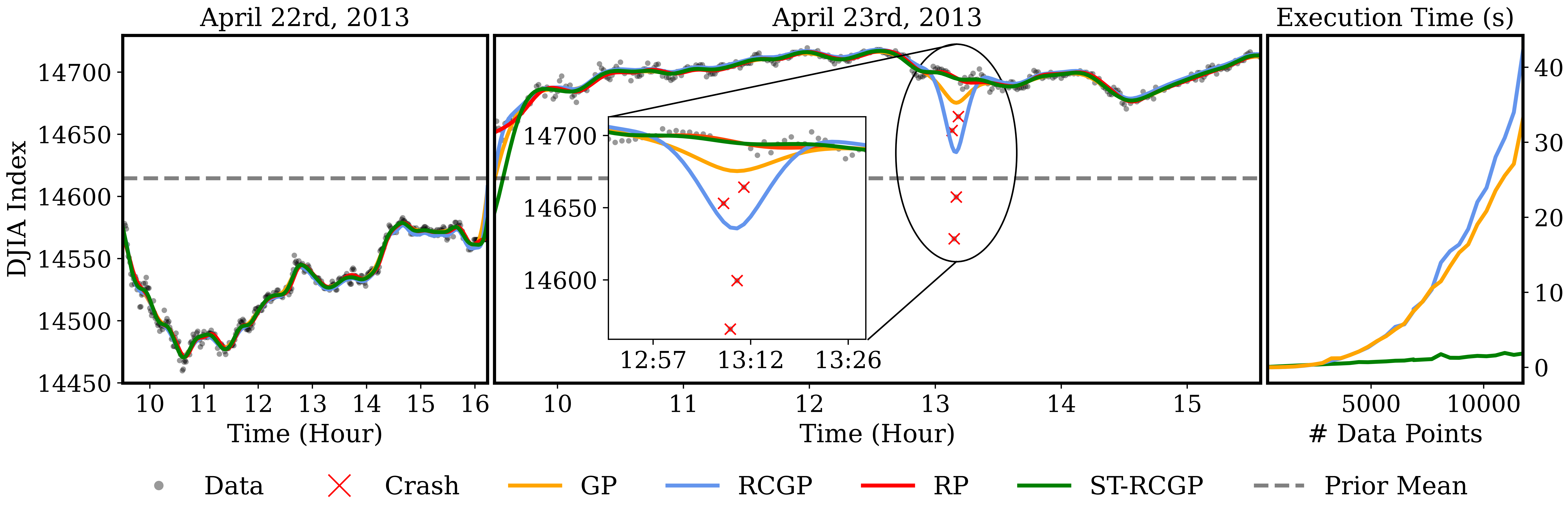
$$\beta := \sigma/\sqrt{2}$$

The Remedy Enabled by Sequential Inference

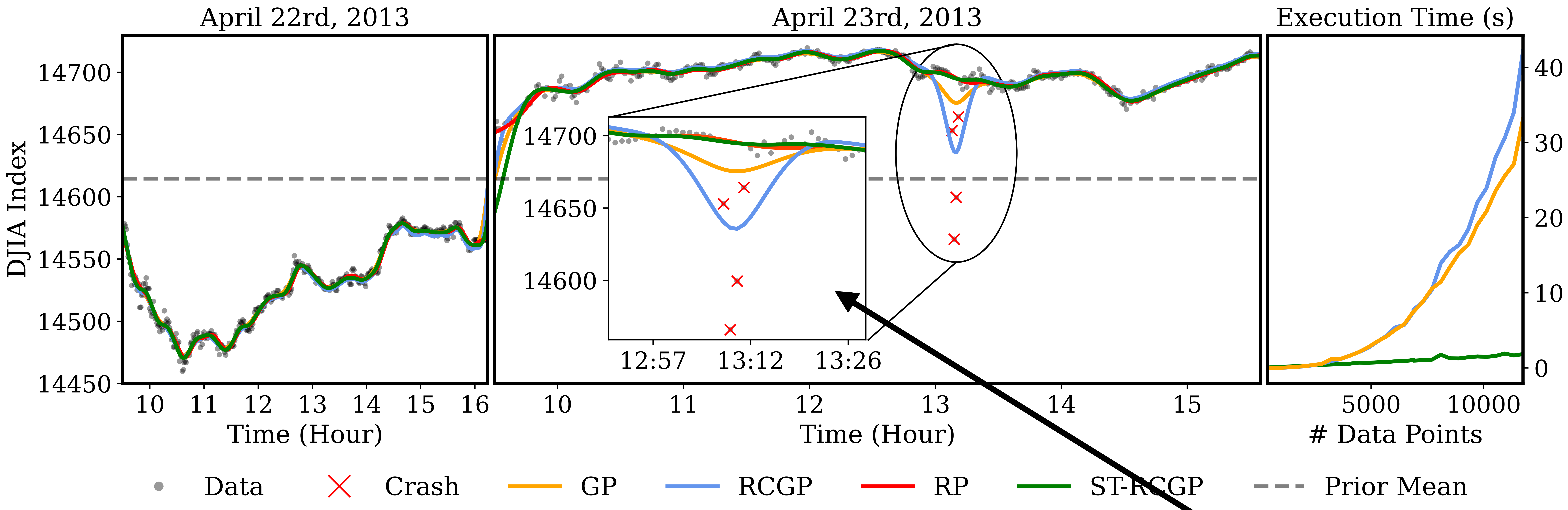
Result... Great!



Experiment 1: Robustness During Financial Crashes



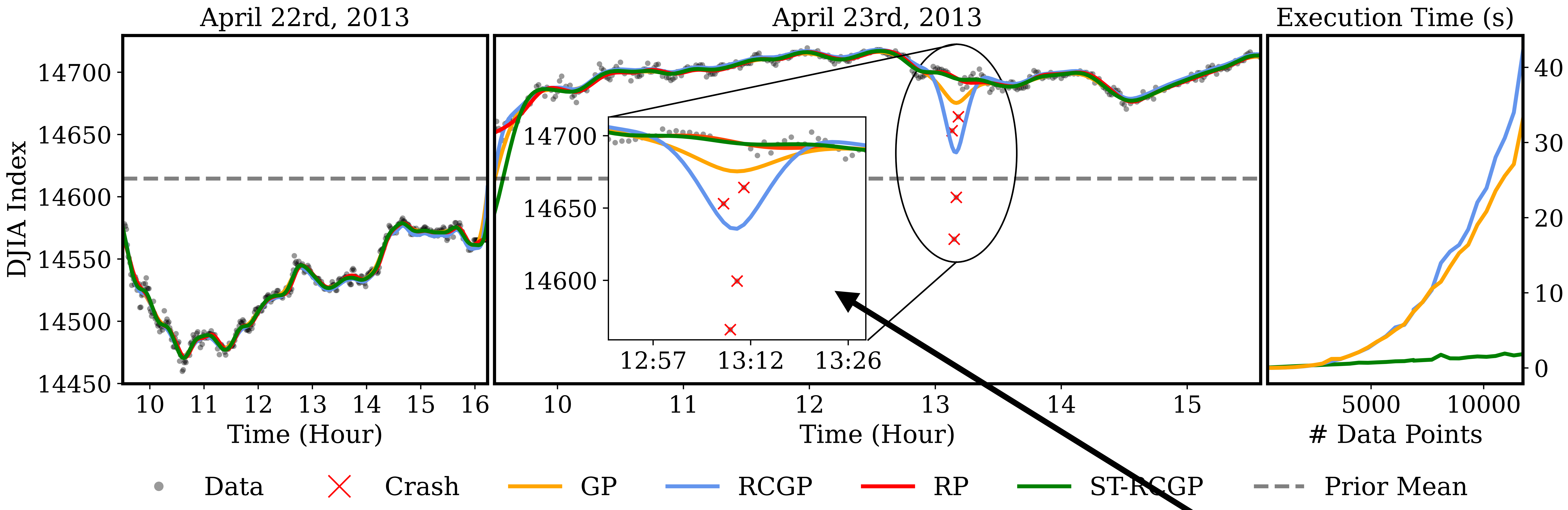
Experiment 1: Robustness During Financial Crashes



Associated Press's Twitter account **hacked!**

Experiment 1: Robustness During Financial Crashes

GP: Not robust to crash…

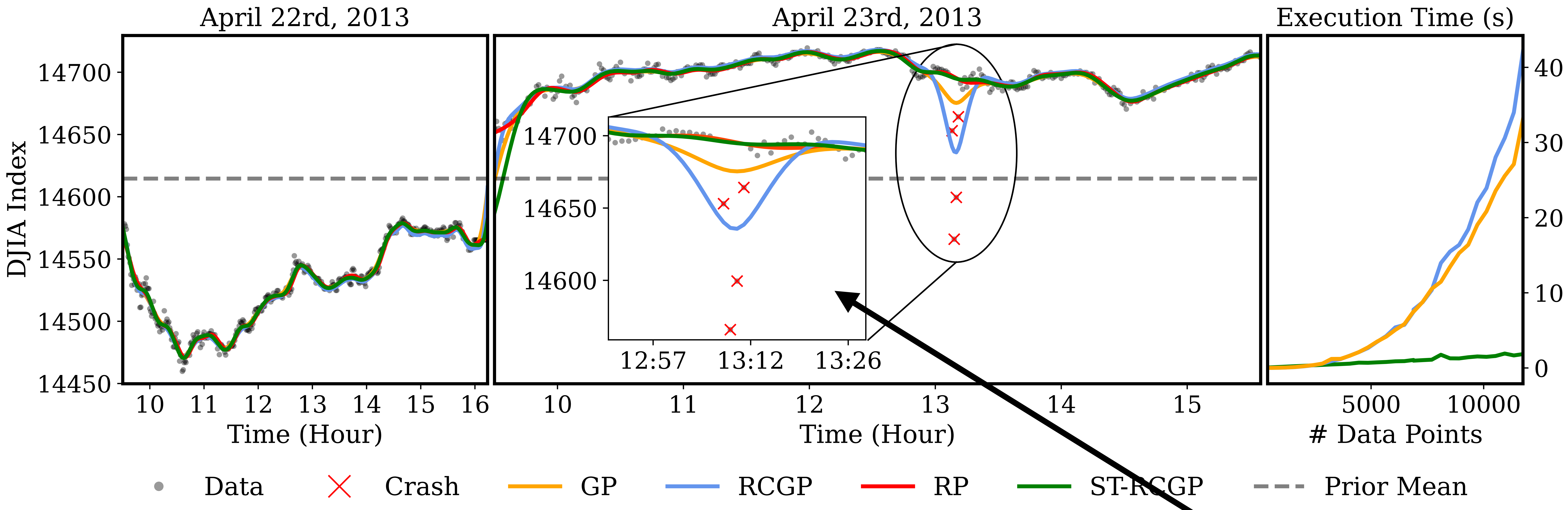


Associated Press's Twitter account **hacked!**

Experiment 1: Robustness During Financial Crashes

GP: Not robust to crash…

RCGP: Even worse than GP! Why? **Issue #1**



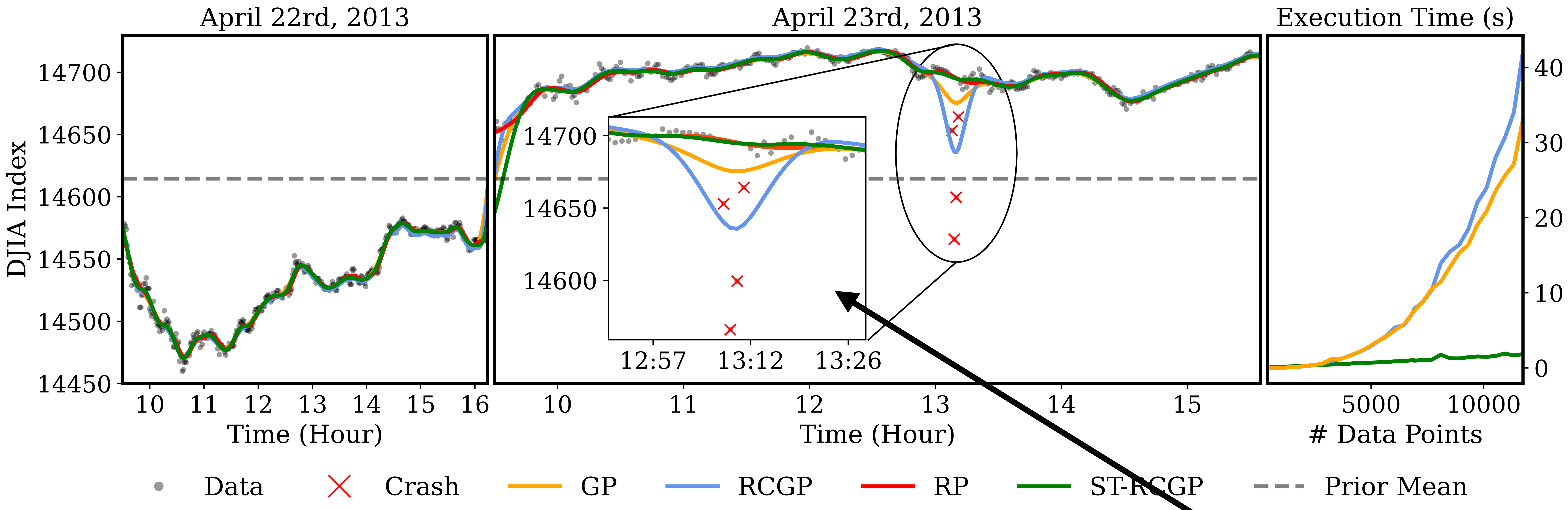
Associated Press's Twitter account **hacked!**

Experiment 1: Robustness During Financial Crashes

GP: Not robust to crash…

Relevance pursuit (RP): Robust but slow; $\mathcal{O}(KN^3)$

RCGP: Even worse than GP! Why? **Issue #1**



Associated Press's Twitter account **hacked!**

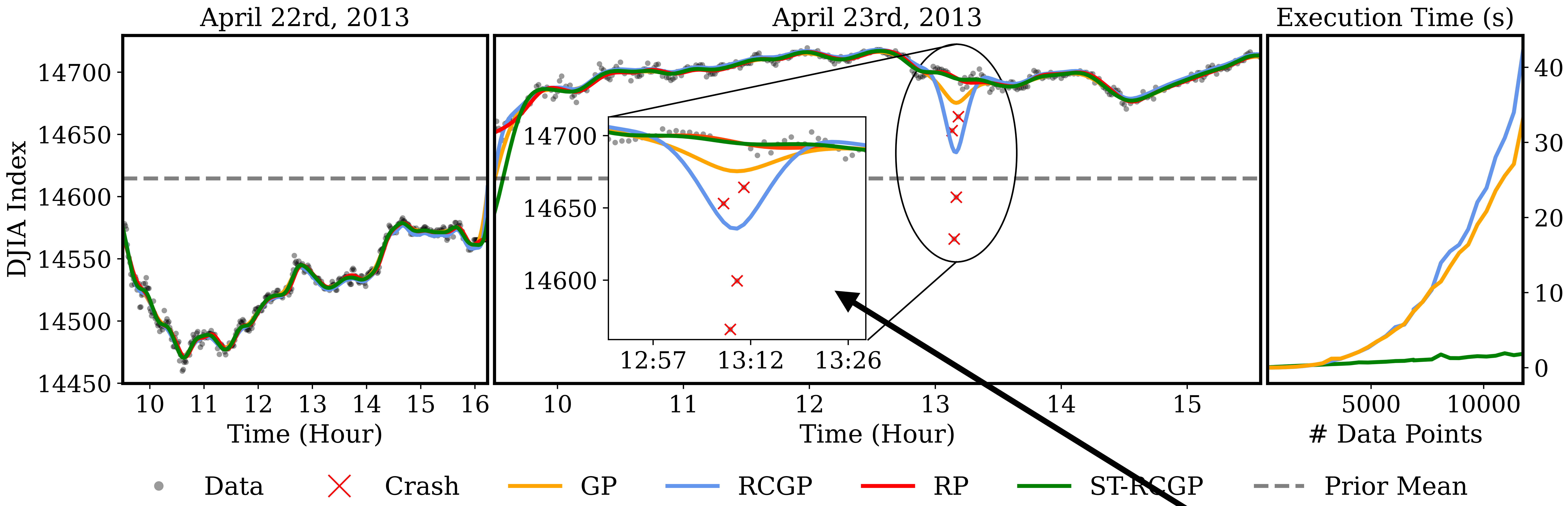
Experiment 1: Robustness During Financial Crashes

GP: Not robust to crash…

Relevance pursuit (RP): Robust but slow; $\mathcal{O}(KN^3)$

RCGP: Even worse than GP! Why? **Issue #1**

ST-RCGP: Robust AND fast! How? **Adaptive.**



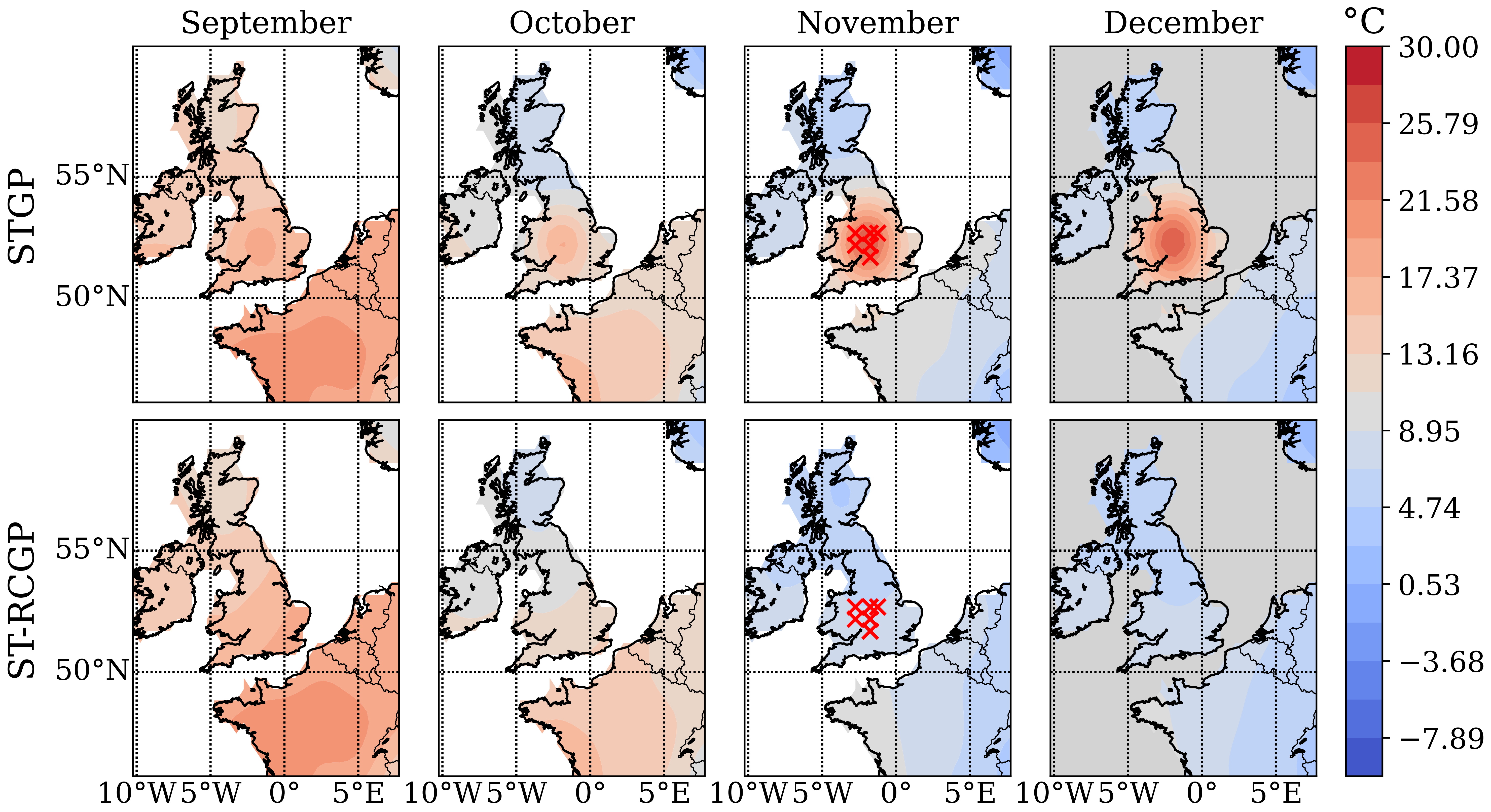
Associated Press's Twitter account **hacked!**

Experiment 2: Weather Forecasting

- Temperature data: Jan 2022 - Dec 2023; 571 spatial locations.
- We induce ***focussed outliers*** (e.g. nearby weather stations break)

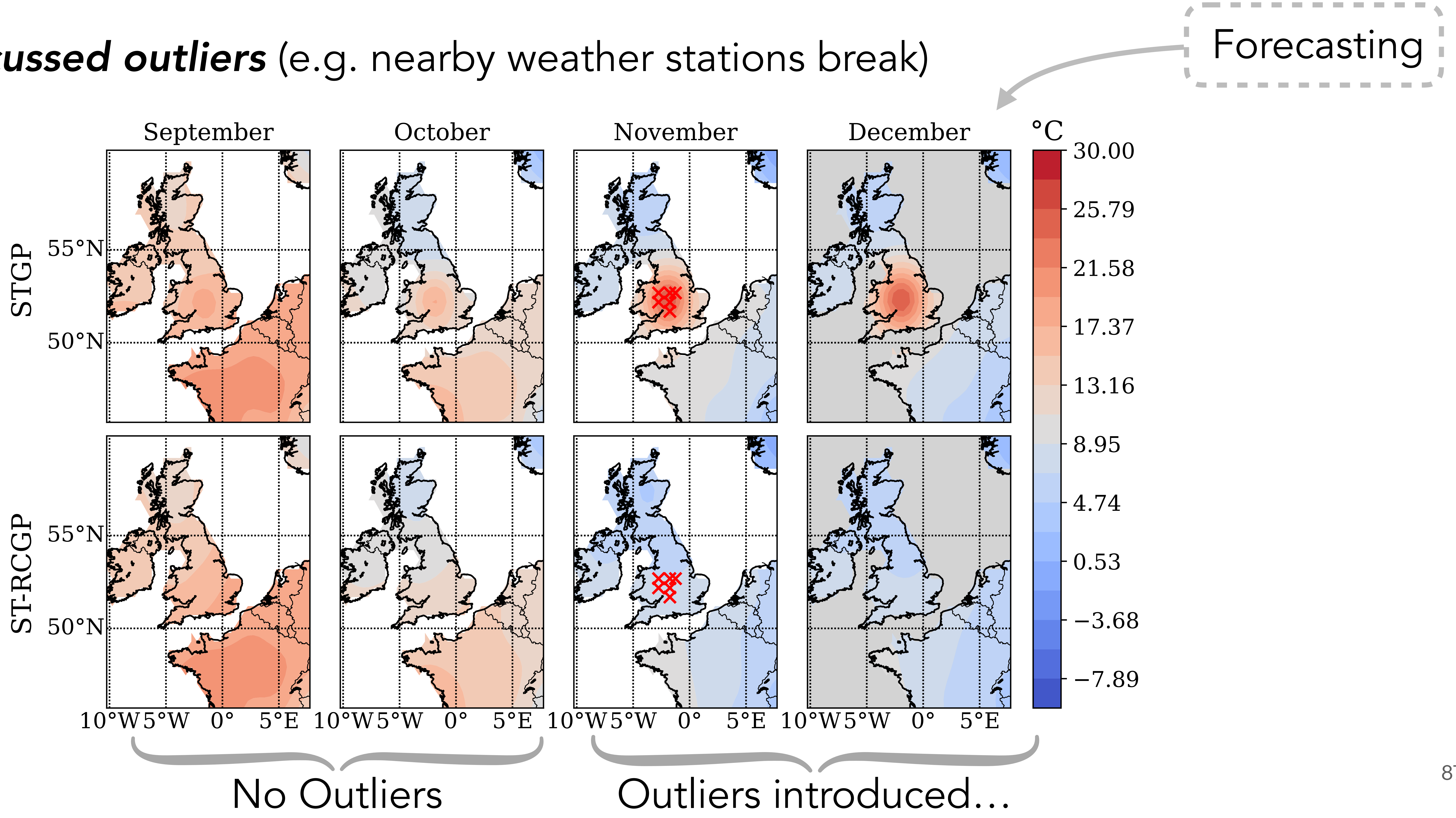
Experiment 2: Weather Forecasting

- Temperature data: Jan 2022 - Dec 2023; 571 spatial locations.
- We induce ***focussed outliers*** (e.g. nearby weather stations break)



Experiment 2: Weather Forecasting

- Temperature data: Jan 2022 - Dec 2023; 571 spatial locations.
- We induce ***focussed outliers*** (e.g. nearby weather stations break)

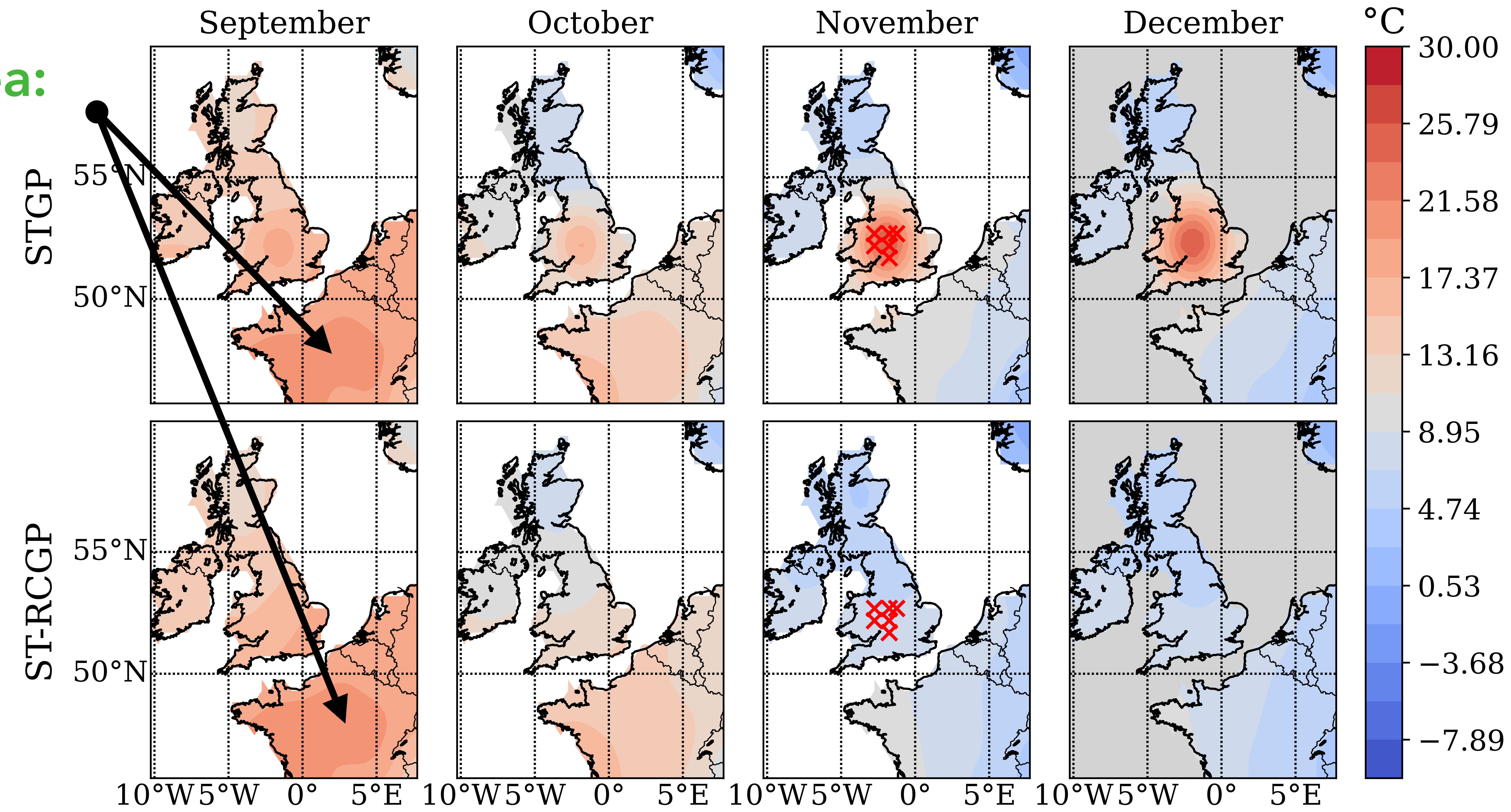


Experiment 2: Weather Forecasting

- Temperature data: Jan 2022 - Dec 2023; 571 spatial locations.
- We induce ***focussed outliers*** (e.g. nearby weather stations break)

Forecasting

Outlier-Free Area:
≈ Performance



No Outliers

Outliers introduced...

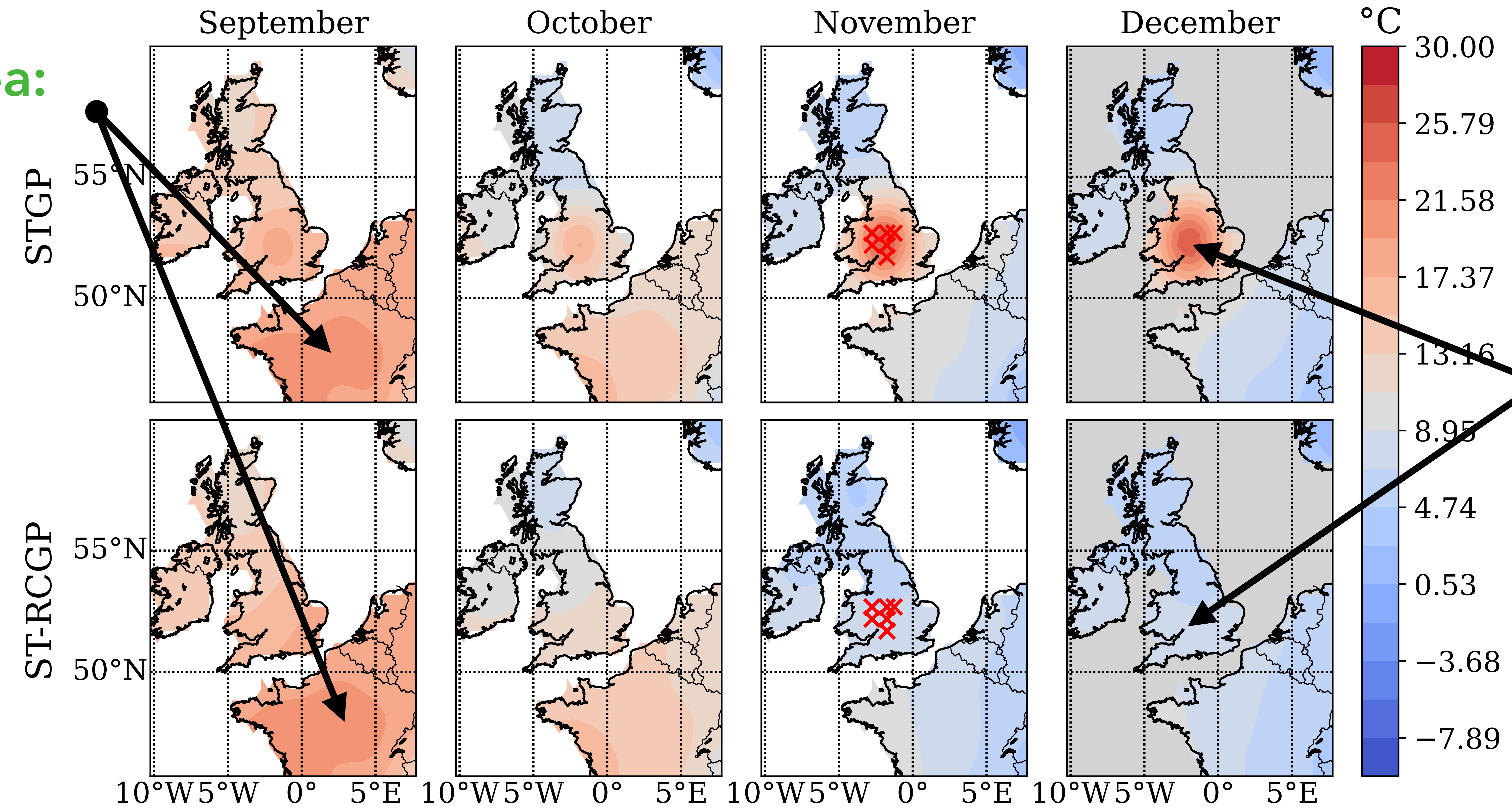
Experiment 2: Weather Forecasting

- Temperature data: Jan 2022 - Dec 2023; 571 spatial locations.
- We induce ***focussed outliers*** (e.g. nearby weather stations break)

Forecasting

Outlier-Free Area:

≈ Performance



Outlier Area:

ST-RCGP ≫ STGP

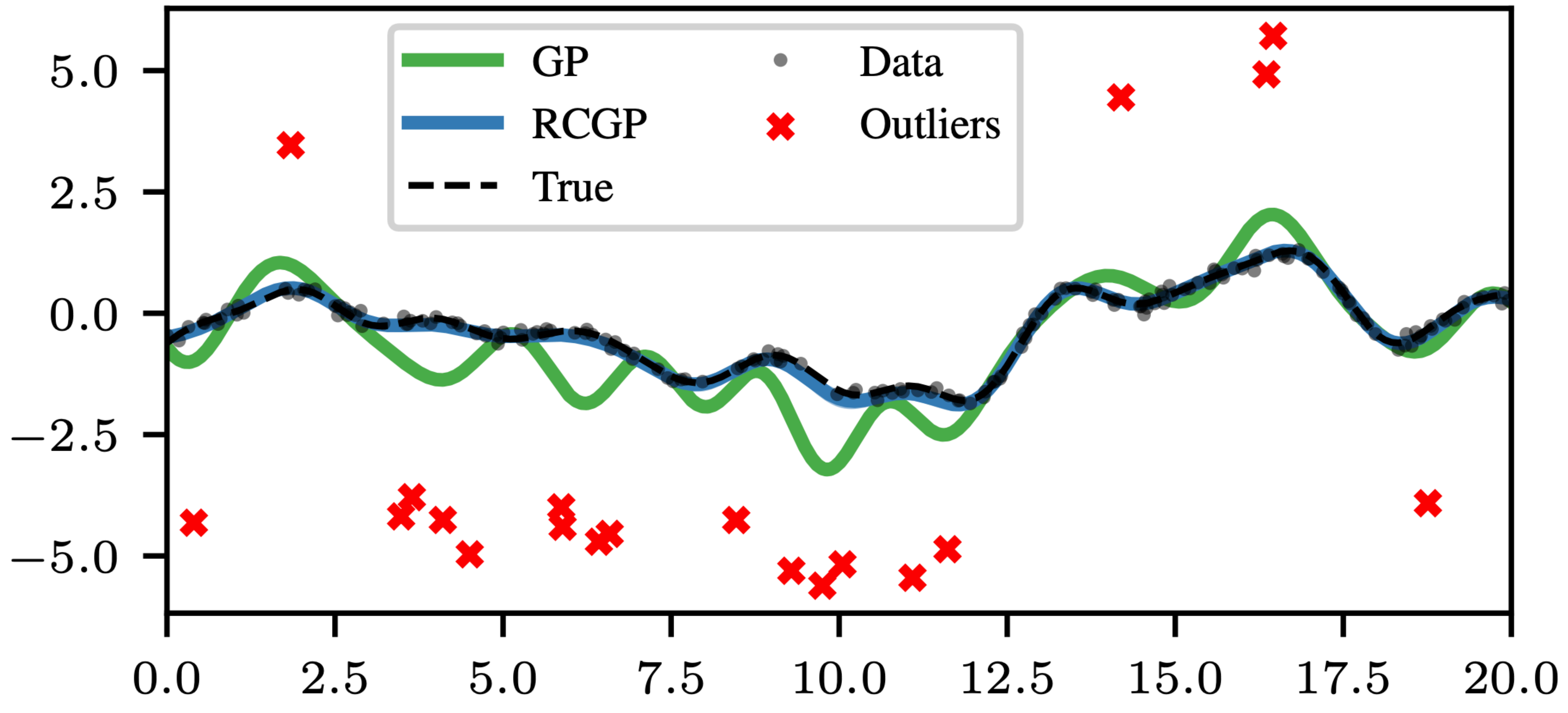
$\mathcal{O}$(ST-RCGP) ≈ $\mathcal{O}$(STGP)

No Outliers

Outliers introduced…

Related Work: RCGP

Robust and Conjugate Gaussian Process Regression

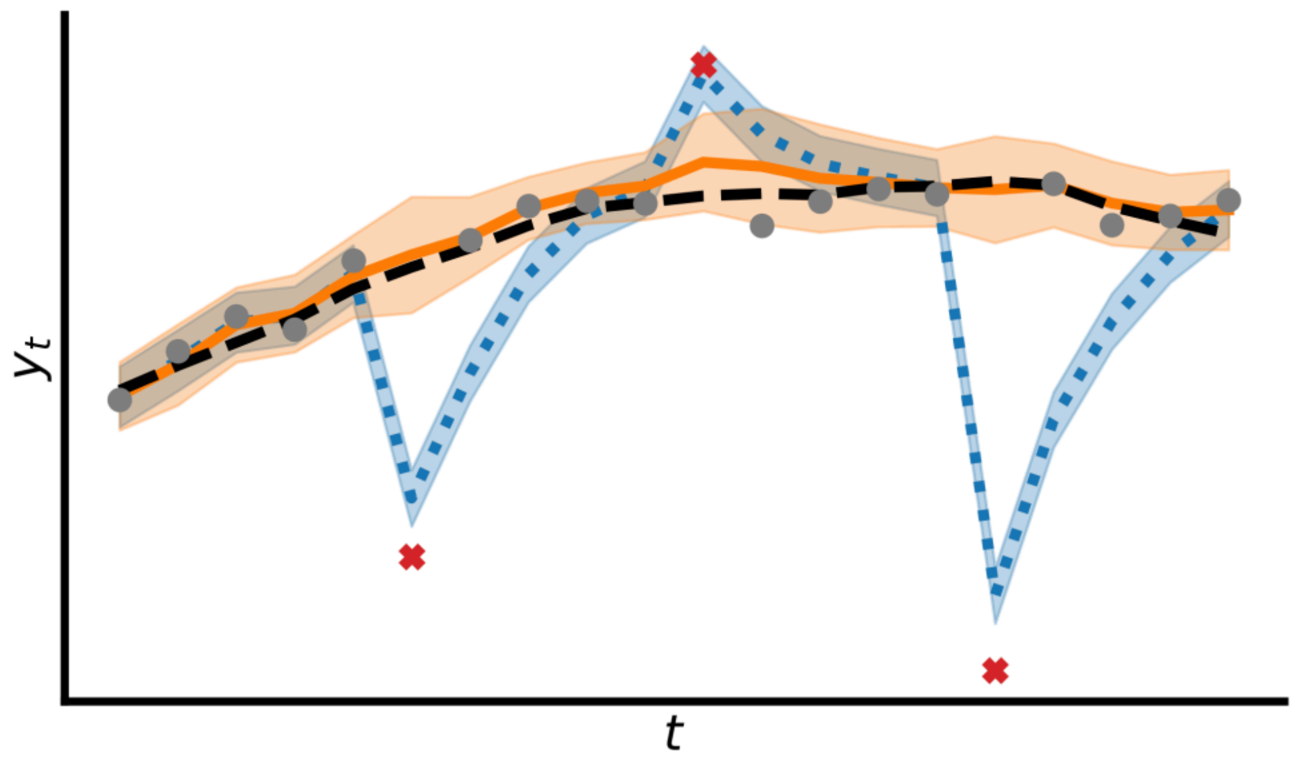


Scan to download RCGP paper!

See Altamirano, M., Briol, F.-X., Knoblauch,F. (2024). Robust and Conjugate Gaussian Process Regression in ICML

Related Work: Robust Kalman Filtering

Outlier-Robust Kalman Filter



Scan to download the full paper

See Also:
Towards Robust Inference for Bayesian Filtering of Linear Gaussian Dynamical Systems Subject to Additive Change

Scan to download the full paper

See Duran-Martin, G., Altamirano, M., Shestopaloff, A. Y., Sánchez-Betancourt, L., Knoblauch, J., Jones, M., ... & Murphy, K. (2024). Outlier-robust Kalman Filtering through Generalised Bayes In ICML

See Reimann H. (2024). Towards Robust Inference for Bayesian Filtering of Linear Gaussian Dynamical Systems Subject to Additive Change. (MSc thesis)

Related Work: Variational STGPs

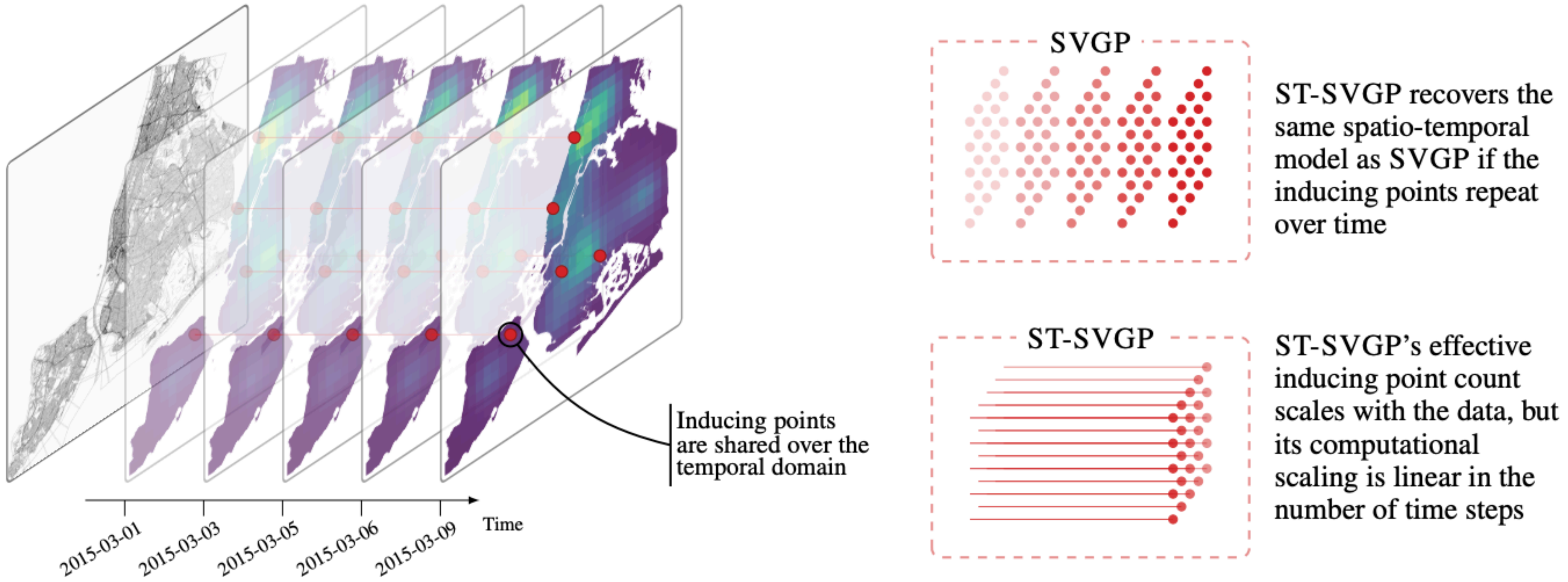


Figure 1: A demonstration of the spatio-temporal sparse variational GP (ST-SVGP) applied to crime count data in New York. ST-SVGP tracks spatial points over time via spatio-temporal filtering. The colourmap is the posterior mean, and the red dots are spatial inducing points. The diagram shows the difference between how inducing points are treated in ST-SVGP and SVGP.

Scan to download the full paper

See Hamelijnck, O., Wilkinson, William J., Loppi, Niki A., Solin, A., Damoulas, T. (2021). Spatio-Temporal Variational Gaussian Processes In NeurIPS

Summary

Scan to download
the ST-RCGP paper!

1. **Fix problematic issues with RCGPs in spatio-temporal settings**

2. **Provide an outlier-robust <u>and</u> efficient spatio-temporal GP**

See Laplante, W., Altamirano, M., Duncan, A., Knoblauch, J., Briol, F.-X. (2025). Robust and Conjugate Spatio-Temporal Gaussian Processes.

Any Questions?