Sampling from Gaussian Process Posteriors using Stochastic Gradient Descent

CBL Research Talk 07 / 06 / 2023

Javier Antorán

UNIVERSITY OF CAMBRIDGE

The team

Andy Lin

Shreyas Padhy

Dave Janz

Miguel Hernández-Lobato

Alex Terenin

Talk Outline

1. Whirlwind introduction to Gaussian Processes
2. The computational cost of inference and popular approximations
3. Sampling from GPs with SGD
4. Analysis of what SGD does in this setting
5. Experiments: regression and Bayesian optimisation

I must warn you

This talk applies the linear model inference of Antoran, Padhy, et al 2022 to GPs. The content

- Is of limited novelty (R1)

> **Clarity, Quality, Novelty And Reproducibility:**
> 1. Novelty is somehow limited. The main contribution is that the authors simply apply EM algorithm to scale inference and hyperparameter selection in Gaussian linear regression. It would be more novel if the EM algorithm was improved.

- Of minor interest (R2)

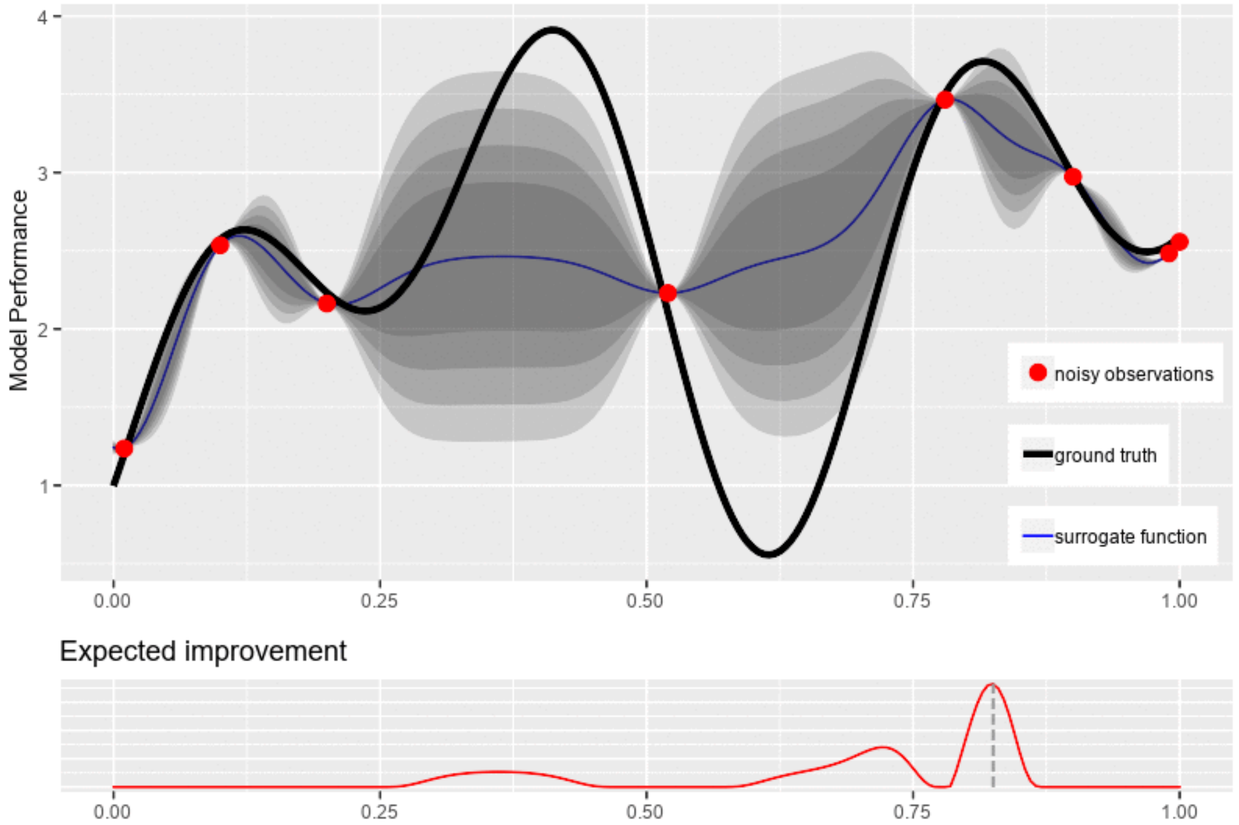
> - The approach is interesting and combines a lot of disparate ideas to make this the linearization + Laplace approximation scalable. Some of these ideas may be of (minor) independent interest.

- Is just a combination of some non-groundbreaking clever tricks (AC)

> **Justification For Why Not Higher Score:**
> The paper is well-written and the experiments are well done. It uses a combination of clever tricks to scale the Laplace method, which may not be groundbreaking individually but are effective when combined. For this reason, I do not recommend a higher score.

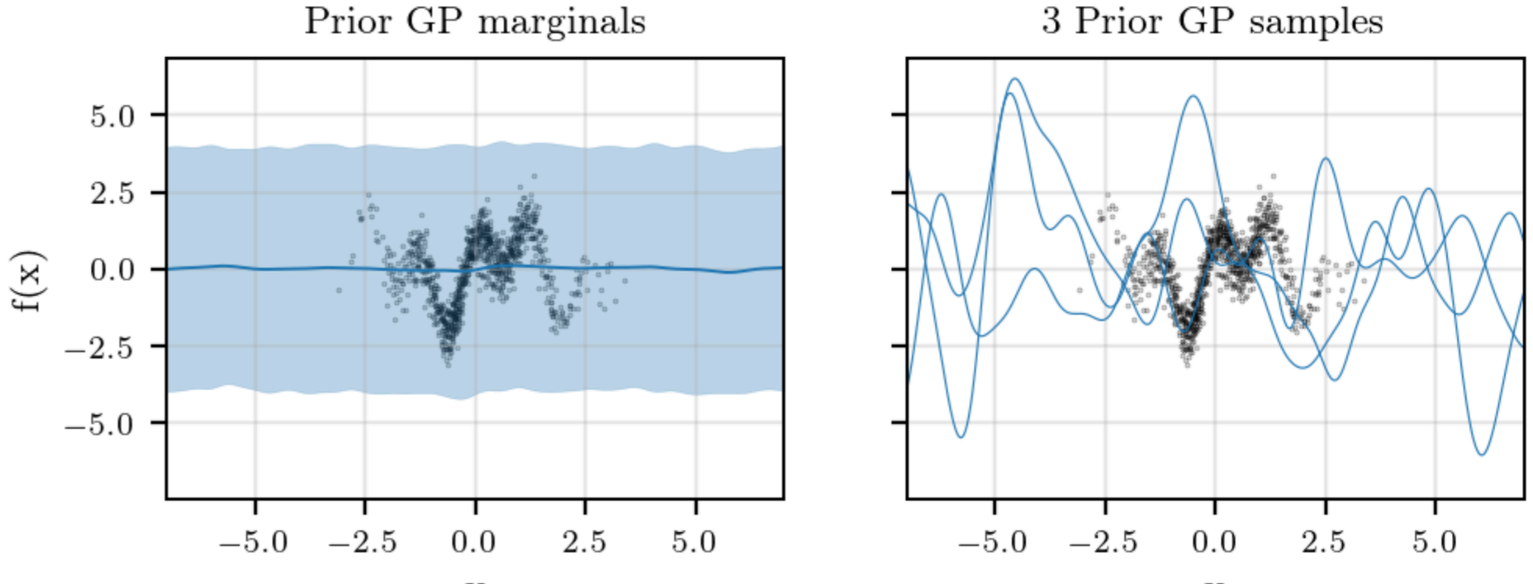
Gaussian Process

- Flexible model class in which exact inference is tractable!
  - Provides uncertainty estimates together with predictions
  - State of the art tool for sequential decision making



Gaussian Process — the Bayesian model

- Bayesian generative model: $\mathbf{y} = f(\mathbf{x}) + \epsilon$ with $\mathbf{y} \in R^N$
- The function $f : X \to R$ is assumed to be sampled from a GP. $f \sim GP(\mu, k)$
- We take the mean function to be $\mu(\,\cdot\,) = 0$ and $k(\,\cdot\,, \cdot')$ is the covariance kernel.
  - We evaluate k at the train data to obtain the Kernel matrix $K_{\mathbf{xx}} = [k(x_i, x_j)]_{i,j=0,\ldots,N}$
- We assume Gaussian observation noise $\epsilon \sim N(0, \Sigma)$ (assume $\Sigma = \sigma^2 I_N$)



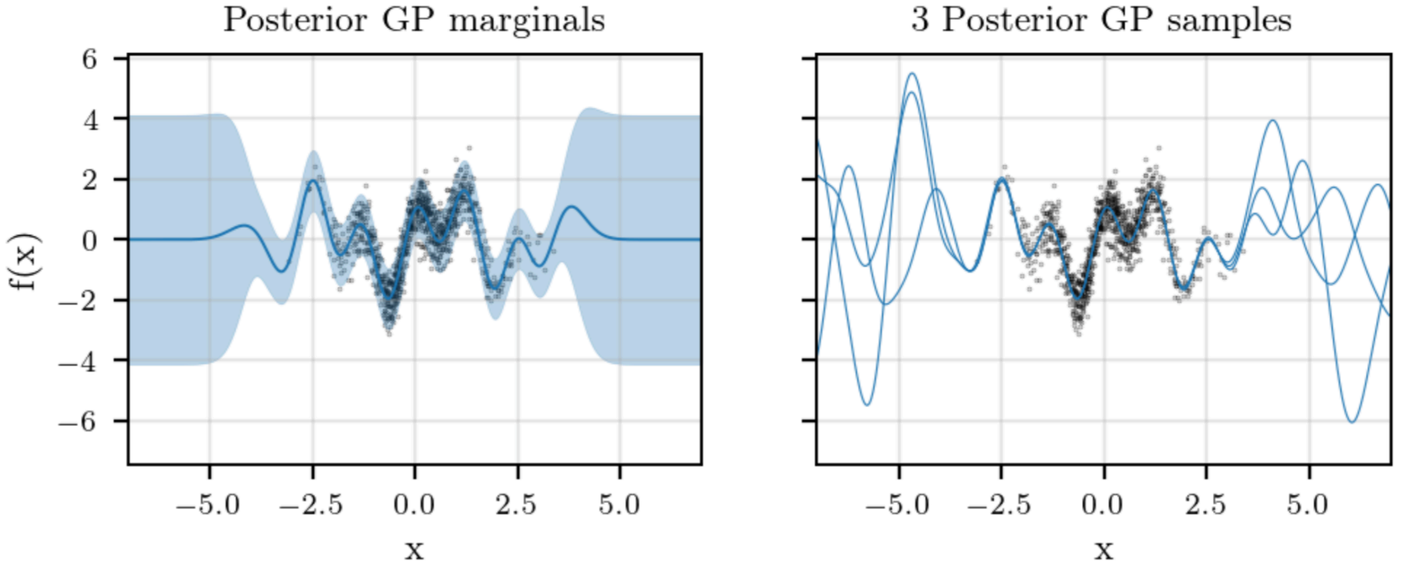
Gaussian Process — posterior inference

- Traditional formulation: The posterior is a the GP $f \,|\, \mathbf{y} \sim GP(\mu_{f|\mathbf{y}}\ k_{f|\mathbf{y}})$ with

$$\mu_{f|\boldsymbol{y}}(\,\cdot\,) = \mathbf{K}_{(\cdot)\boldsymbol{x}}\left(\mathbf{K}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}\right)^{-1}\boldsymbol{y} \quad k_{f|\boldsymbol{y}}(\,\cdot\,,\cdot\,') = \mathbf{K}_{(\cdot,\cdot')} - \mathbf{K}_{(\cdot)\boldsymbol{x}}\left(\mathbf{K}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}\right)^{-1}\mathbf{K}_{\boldsymbol{x}(\,\cdot\,')}$$

- Pathwise formulation: Posterior functions are given by updating prior samples $f \sim GP(\mu, k)$ as

$$(f \mid \boldsymbol{y})(\,\cdot\,) = f(\,\cdot\,) + \mathbf{K}_{(\cdot)\boldsymbol{x}}\left(\mathbf{K}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}\right)^{-1}(\boldsymbol{y} - f(\boldsymbol{x}) - \boldsymbol{\varepsilon}) \quad \boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}) \quad f \sim \mathrm{GP}(\mu, k)$$



Cubic Computational Cost

$$\mu_{f|\boldsymbol{y}}(\,\cdot\,) = \mathbf{K}_{(\cdot)\boldsymbol{x}}\left(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma}\right)^{-1}\boldsymbol{y} \qquad k_{f|\boldsymbol{y}}(\,\cdot\,,\cdot\,') = \mathbf{K}_{(\cdot,\cdot')} - \mathbf{K}_{(\cdot)\boldsymbol{x}}\left(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma}\right)^{-1}\mathbf{K}_{\boldsymbol{x}(\,\cdot\,')}$$

$$(f \mid \boldsymbol{y})(\,\cdot\,) = f(\,\cdot\,) + \mathbf{K}_{(\cdot)\boldsymbol{x}}\left(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma}\right)^{-1}(\boldsymbol{y} - f(\boldsymbol{x}) - \boldsymbol{\varepsilon}) \qquad \boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}) \qquad f \sim \mathrm{GP}(\mu, k)$$

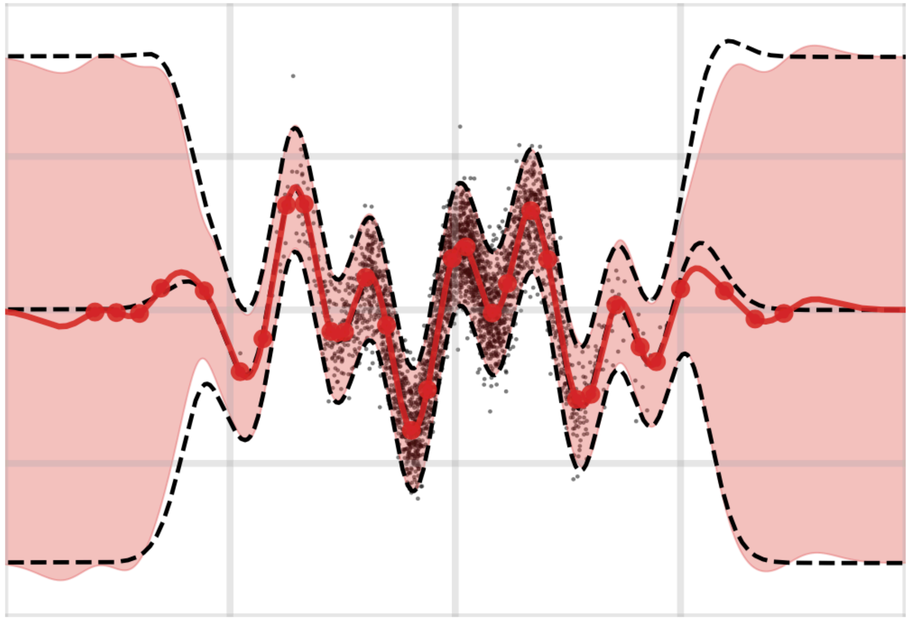
If we want to get anything done, we need to invert or solve against $K_{\mathbf{xx}} + \Sigma \in R^{N \times N}$ which has cost $O(N^3)$

On a A100 GPU largest problem we can deal with is $N = 50k$

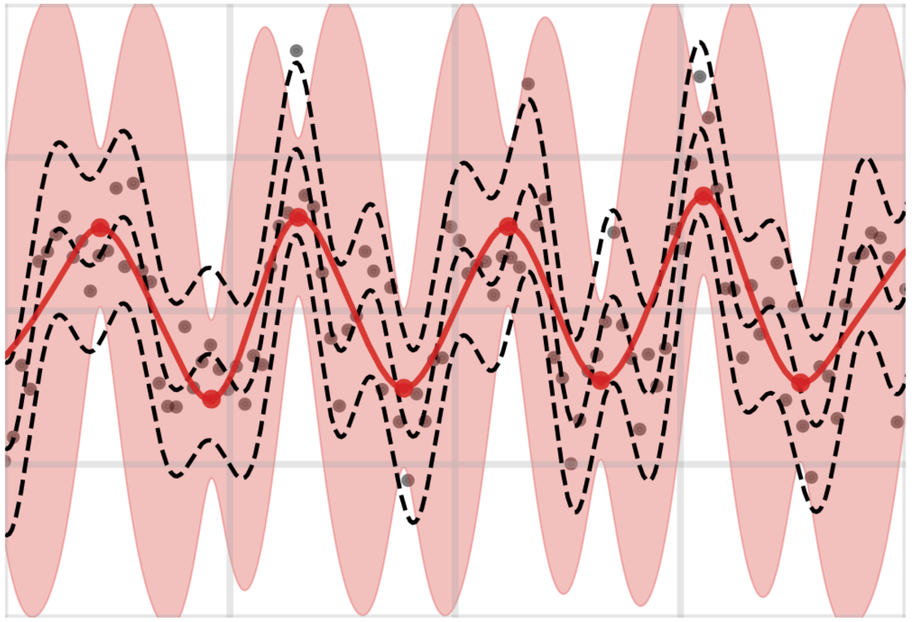
Variational Inference (Titsias 2009, Hensman et. al. 2013)

- Idea: Data can be summarised by a set of **"Inducing points"**
  - Cost is $O(M^3)$ for M the number of inducing points

Infill Asymptotics



Large Domain Asymptotics



----- exact GP —— approximations

Conjugate Gradients (Gibbs & Mackay, 1996, Wang et. al. 2019)

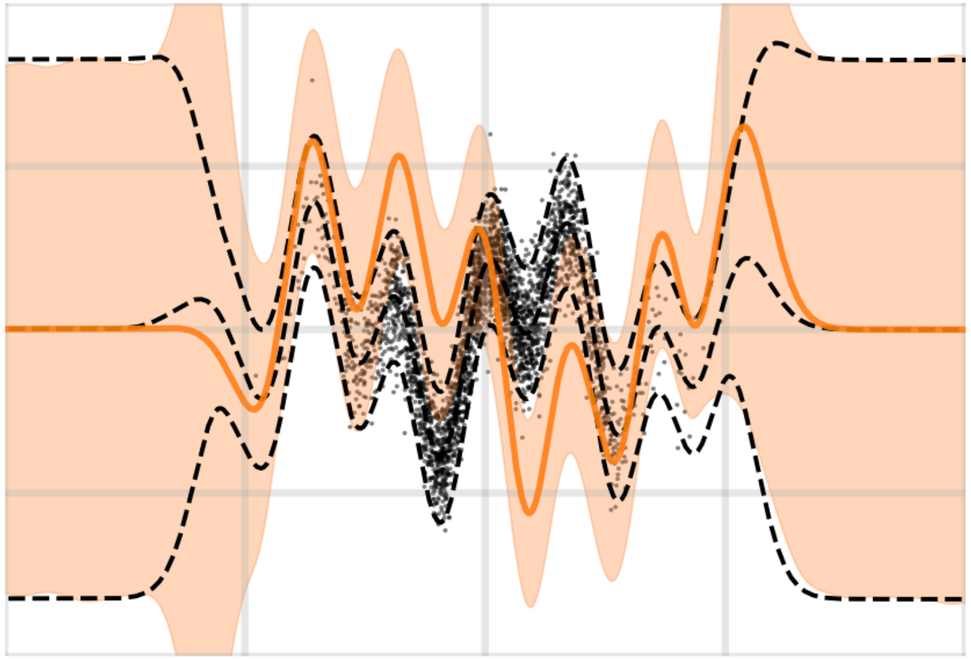
- CG is the most popular approach to solving linear systems $(K + \Sigma)^{-1} b$
  - Iterative method, each step requires a matrix multiplication with $K_{\mathbf{xx}} + \Sigma \in R^{N \times N}$ (cost $O(N^2)$)
  - Algorithm converges in at most N steps but in practise for some tolerance ϵ

$$O\left(\sqrt{\text{cond}(K+\Sigma)}\ \log \frac{\text{cond}(K+\Sigma)\|b\|}{\epsilon}\right) \qquad \text{cond}(K+\Sigma)) = \frac{\lambda_{\text{max}}(K+\Sigma)}{\lambda_{\text{min}}(K+\Sigma)}$$

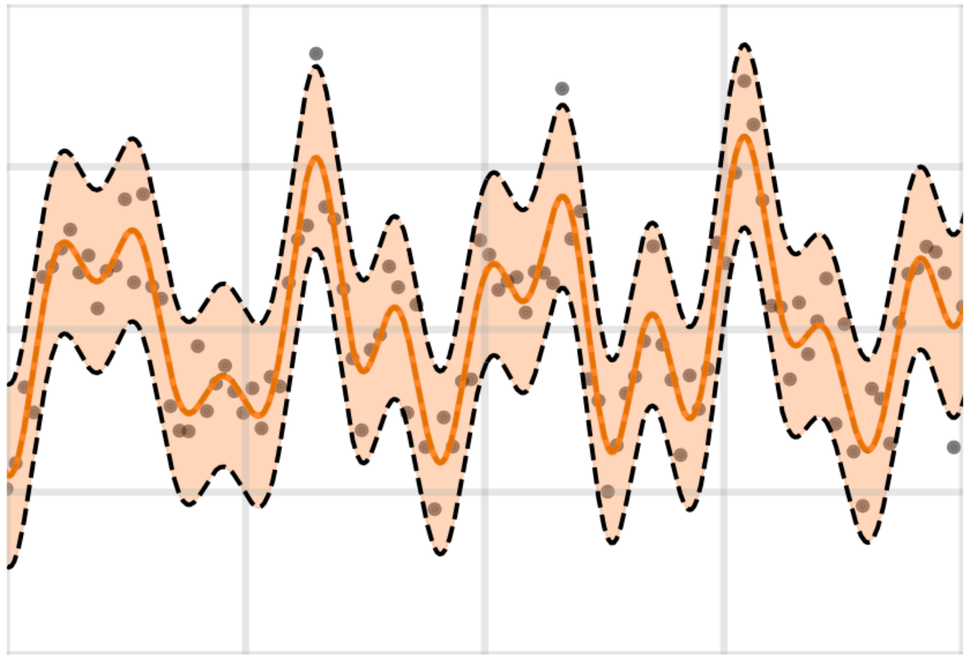
Conjugate Gradients — performance depends on conditioning

- Infill Asymptotics: Redundant data, Kernel matrix is very ill-conditioned
- Large Domain Asymptotics: Data is non-redundant, Kernel matrix better conditioned

Infill Asymptotics



Large Domain Asymptotics

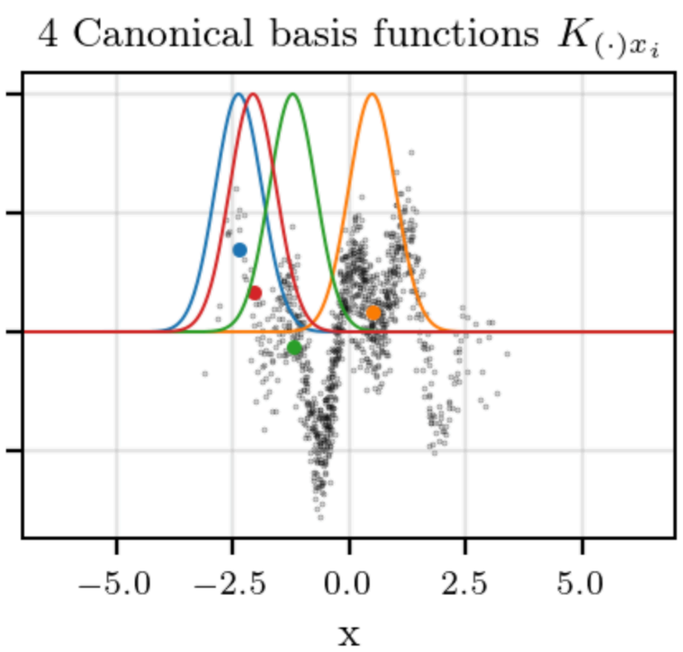


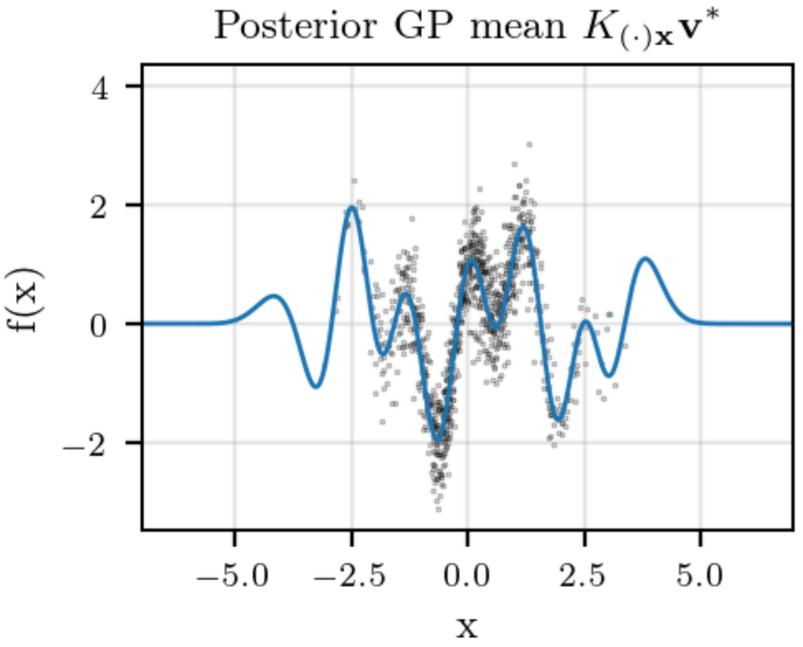
----- exact GP —— approximations

SAMPLING WITH SGD

Sampling as optimisation of representer weights

$$\mu_{f|y}(\,\cdot\,) = \mathbf{K}_{(\cdot)x} \underbrace{\left(\mathbf{K}_{xx} + \mathbf{\Sigma}\right)^{-1} y}_{\mathbf{v}^\star}$$





$$\boldsymbol{v}^* = \underset{\boldsymbol{v}\in\mathbb{R}^N}{\arg\min} \sum_{i=1}^{N} \frac{\left(y_i - \mathbf{K}_{x_i x}\boldsymbol{v}\right)^2}{\Sigma_{ii}} + \|\boldsymbol{v}\|^2_{\mathbf{K}_{xx}}$$

Minibatch estimation and Fourier features

- **Data fit term:** Making predictions $\mathbf{K}_{(\cdot)\boldsymbol{x}}\boldsymbol{v}$ is O(N) and we have N training points $\rightarrow O(N^2)$ naively
  - Can use minibatch estimator to reduce to $O(N)$
- **Regulariser term $\|\boldsymbol{v}\|^2_{\mathbf{K}_{\boldsymbol{xx}}}$:** Naively $O(N^2)$ to construct and collapse $\mathbf{K}_{\boldsymbol{xx}}$
  - We use an unbiased random Fourier feature approximation with L features. Since L is arbitrary, we have $O(N)$

$$\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}\in\mathbb{R}^N} \sum_{i=1}^{N} \frac{\left(y_i - \mathbf{K}_{x_i\boldsymbol{x}}\boldsymbol{v}\right)^2}{\Sigma_{ii}} + \|\boldsymbol{v}\|^2_{\mathbf{K}_{\boldsymbol{xx}}} \longrightarrow \frac{N}{D}\sum_{i}^{D} \frac{\left(y_i - \mathbf{K}_{x_i\boldsymbol{x}}\boldsymbol{v}\right)^2}{\Sigma_{ii}} + \sum_{\ell=1}^{L}\left(\boldsymbol{v}^T\phi_\ell(\boldsymbol{x})\right)^2$$

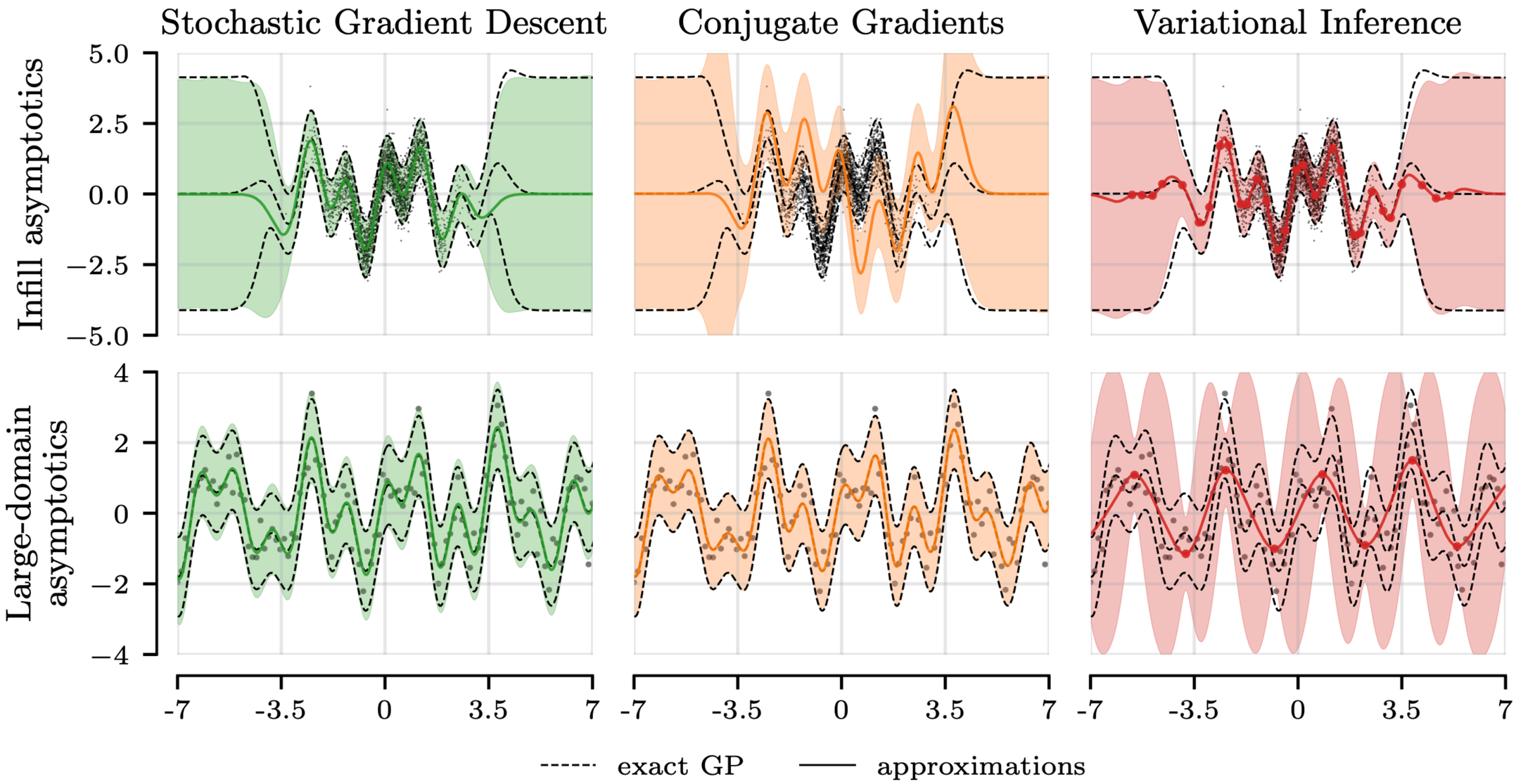
*We can do O(M) with inducing points— not included in slide

This extends to posterior samples in pathwise form as

$$(f \mid \boldsymbol{y})(\,\cdot\,) = \underbrace{\mathbf{K}_{(\cdot)\boldsymbol{x}} \left(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma}\right)^{-1} \boldsymbol{y}}_{\text{mean } \mu_{f|\boldsymbol{y}}(\cdot)} + \underbrace{f(\,\cdot\,) - \mathbf{K}_{(\cdot)\boldsymbol{x}} \left(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma}\right)^{-1} (f(\boldsymbol{x}) + \boldsymbol{\varepsilon})}_{\text{0-mean posterior sample}}$$

$$\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}) \quad f \sim \mathrm{GP}(\mu, k)$$

What does SGD do in practise?

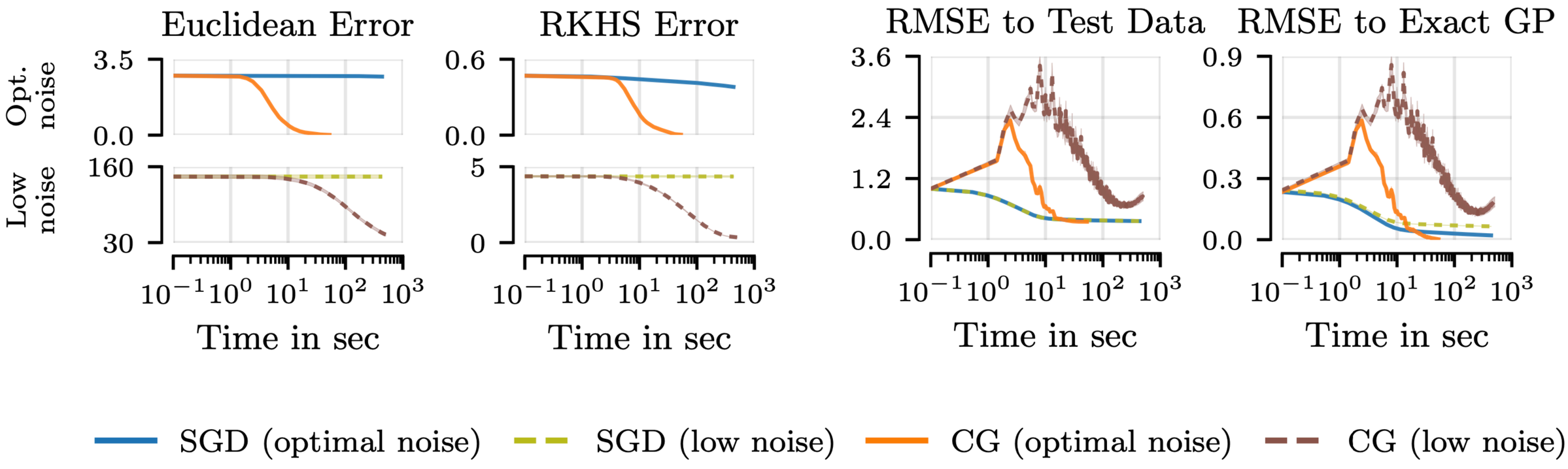


Is SGD converging to the right solution?

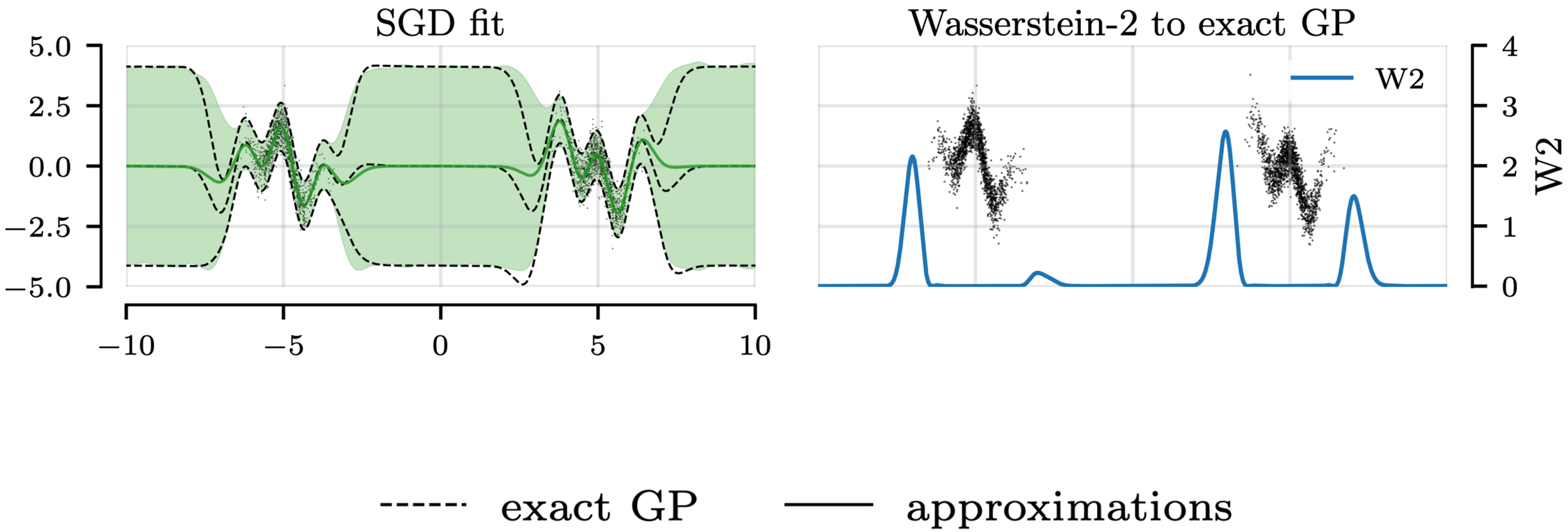
Elevators dataset (≈16k points)

In representer weight space: **NO**

In function space: **YES ????**



What does SGD do in practise? — cont.

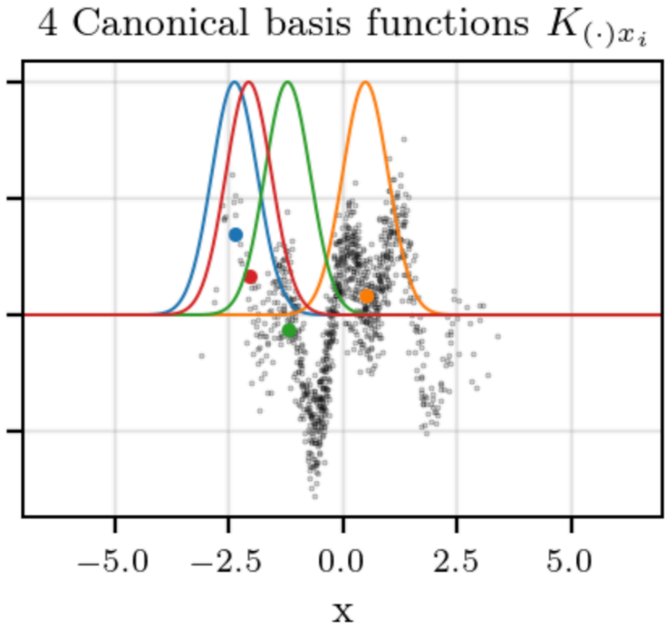


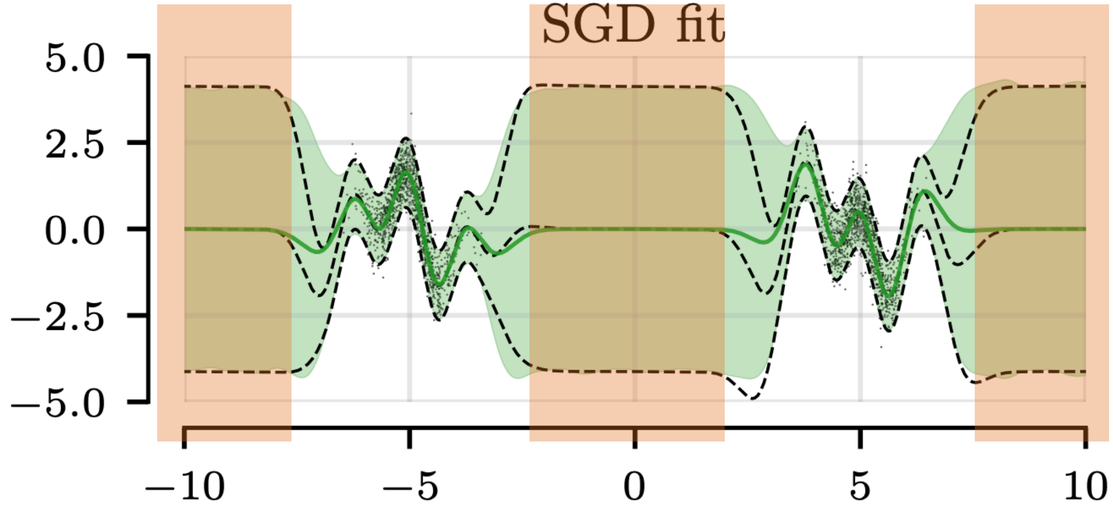
What is going on here? — far-away region

As we move far away from the train data: $K_{(\cdot),x} \to 0$ and

$$(f \mid \boldsymbol{y})(\,\cdot\,) = f(\,\cdot\,) + \underbrace{\mathbf{K}_{(\cdot)\boldsymbol{x}} \left(\mathbf{K}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}\right)^{-1} \boldsymbol{y}}_{\to 0} + \underbrace{-\mathbf{K}_{(\cdot)\boldsymbol{x}} \left(\mathbf{K}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}\right)^{-1} (f(\boldsymbol{x}) + \boldsymbol{\varepsilon})}_{\to 0}$$

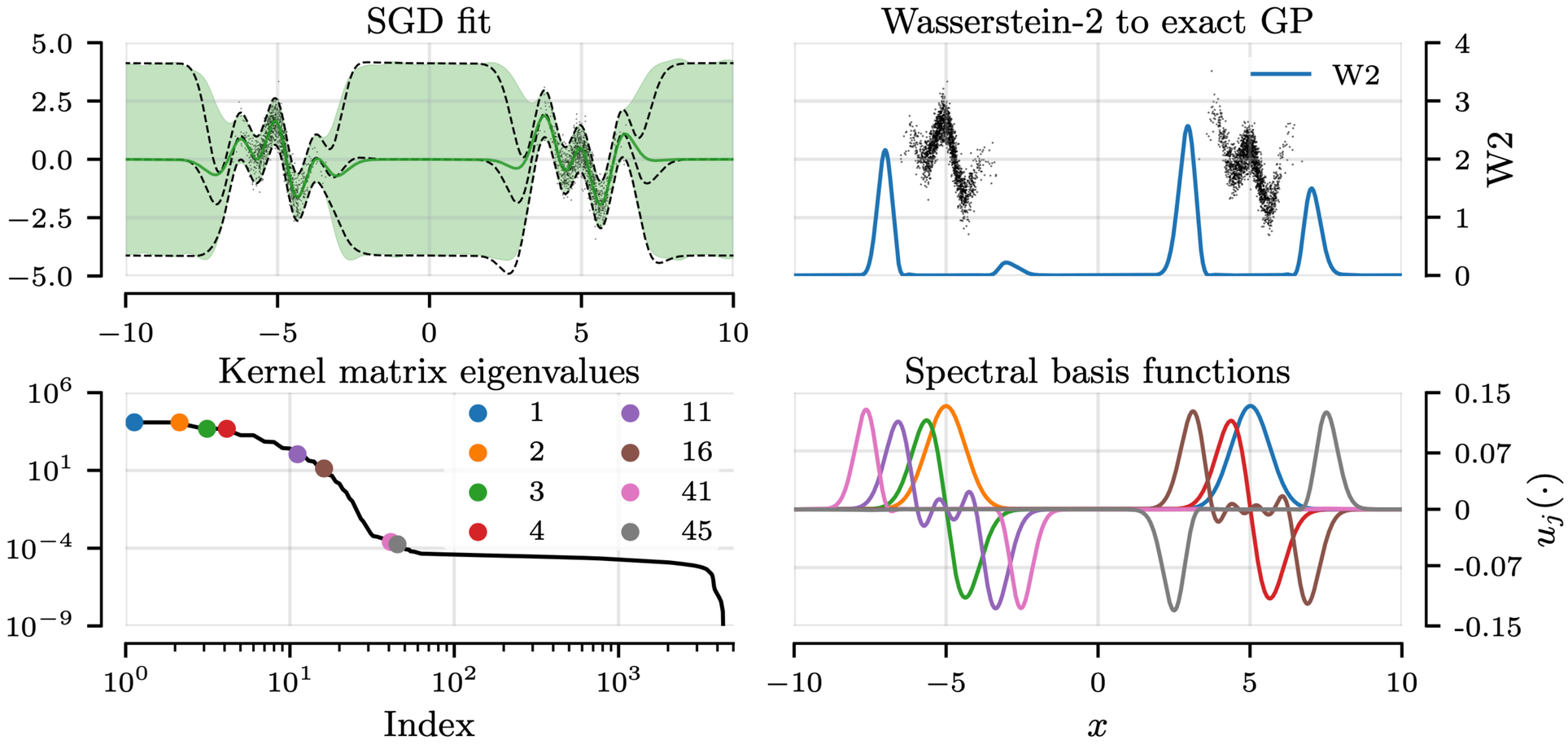
So we revert to the prior!





What is going on here? — spectral basis functions

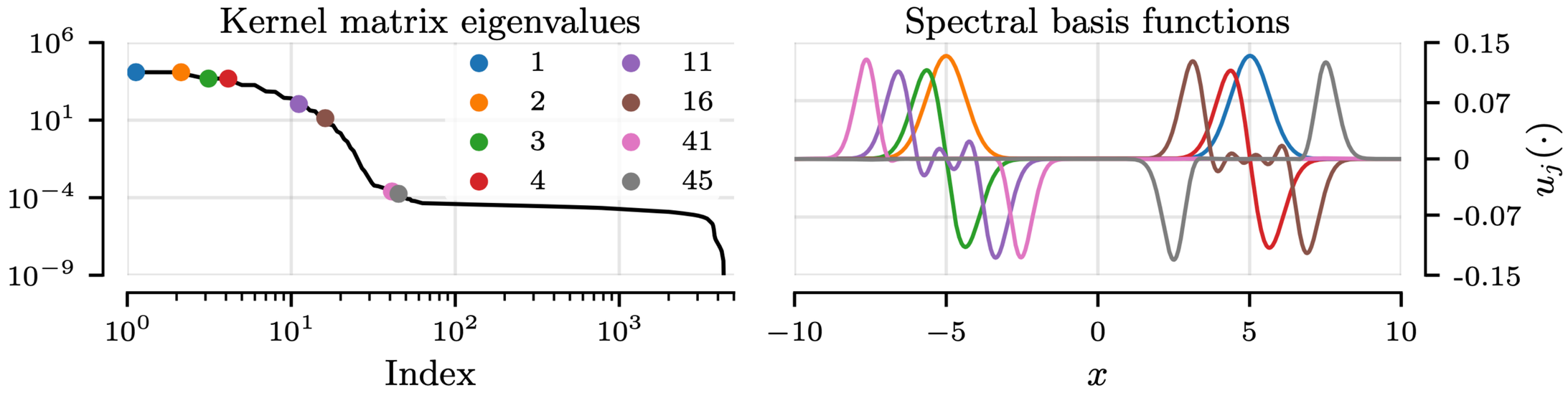
$$(K_{xx} + \Sigma) = U \Lambda U^T \qquad u^{(i)}(\,\cdot\,) = \sum_{j=1}^{N} \frac{U_{ji}}{\sqrt{\lambda_i}} k\left(x_j, \cdot\right)$$



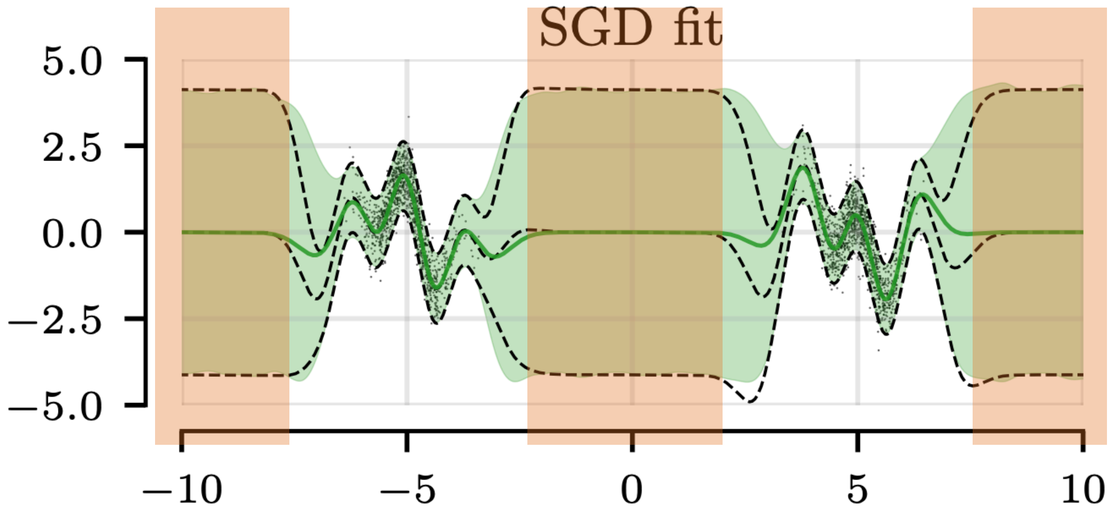
What is going on here? — interpolation region

Proposition 1. *Let $\delta > 0$. Let $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$ for $\sigma^2 > 0$. Let μ_{SGD} be the predictive mean obtained by Polyak-averaged SGD after t steps, starting from an initial set of representer weights equal to zero, and using a sufficiently small learning rate of $0 < \eta < \frac{\sigma^2}{\lambda_1(\lambda_1+\sigma^2)}$. Assume the stochastic estimate of the gradient is G-sub-Gaussian. Then, with probability $1-\delta$, we have for $i = 1, .., N$ that*

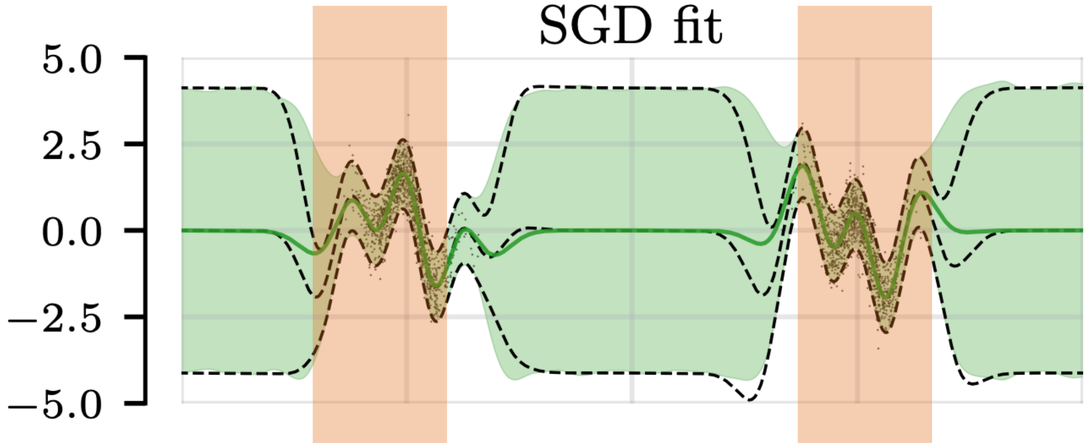
$$\left\| \mathrm{proj}_{u^{(i)}} \mu_{f|\boldsymbol{y}} - \mathrm{proj}_{u^{(i)}} \mu_{\mathrm{SGD}} \right\|_{H_k} \leq \frac{1}{\sqrt{\lambda_i t}} \left(\frac{\|\boldsymbol{y}\|_2}{\eta\sigma^2} + G\sqrt{2\eta\sigma^2 \log \frac{N}{\delta}} \right). \tag{16}$$



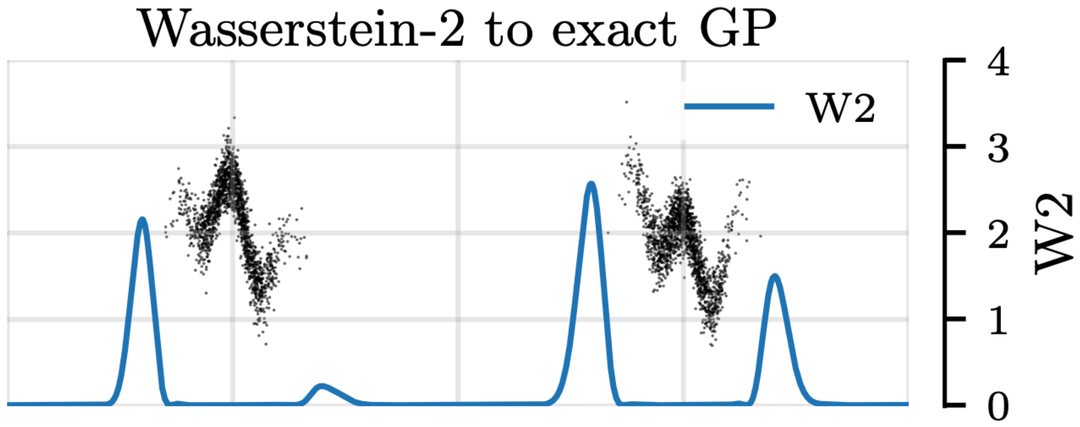
Finding the approximation error



Error is not where the data is

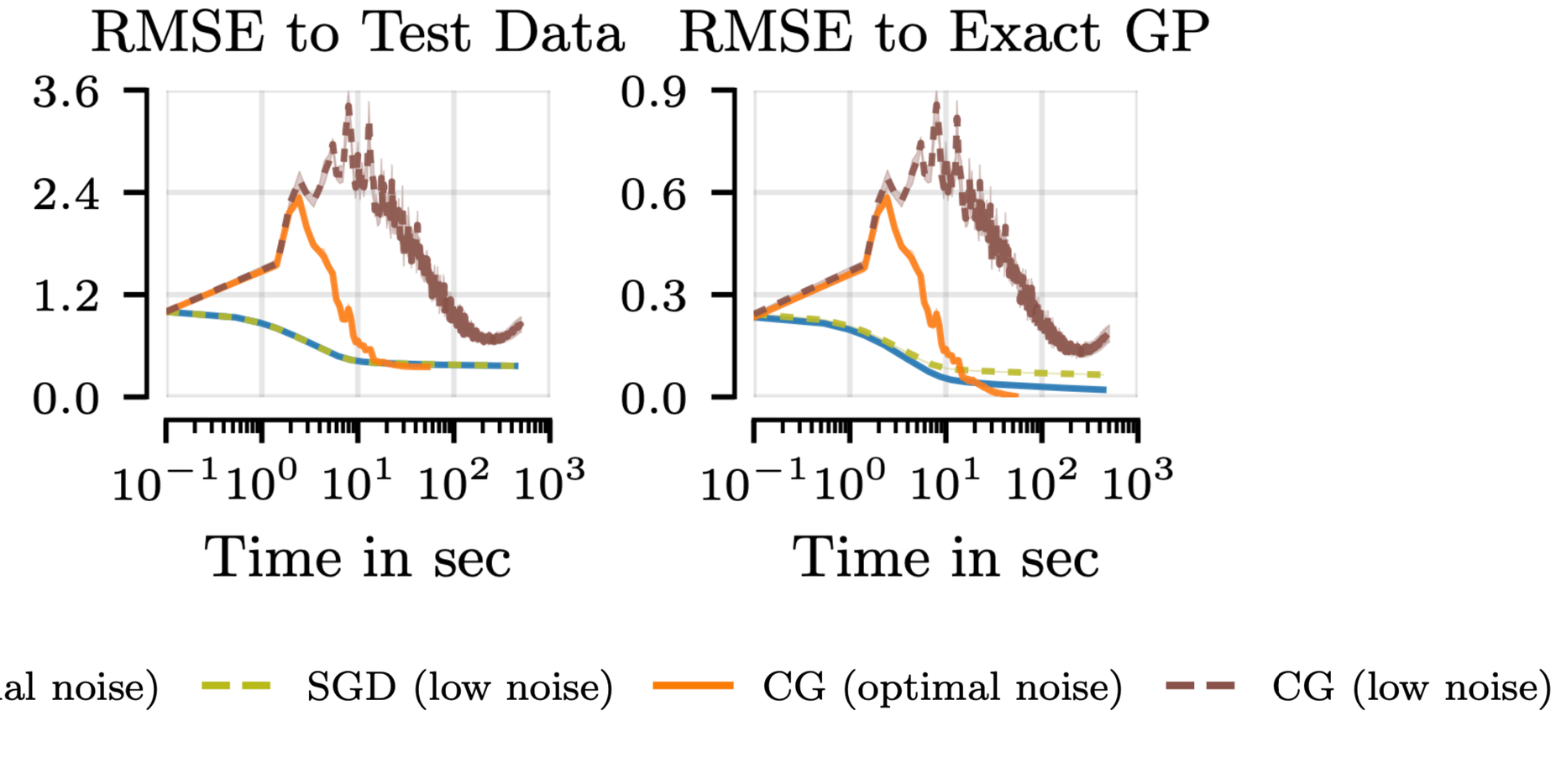


Error is in the nearby-extrapolation region



Resistance to ill-conditioning — no need for adding extra noise

Elevators dataset (≈16k points)



Experiments: regression on datasets of increasing size

- Advantage of $O(N)$ vs $O(N^2 \log \text{cond})$ starts to kick in around 50k training points

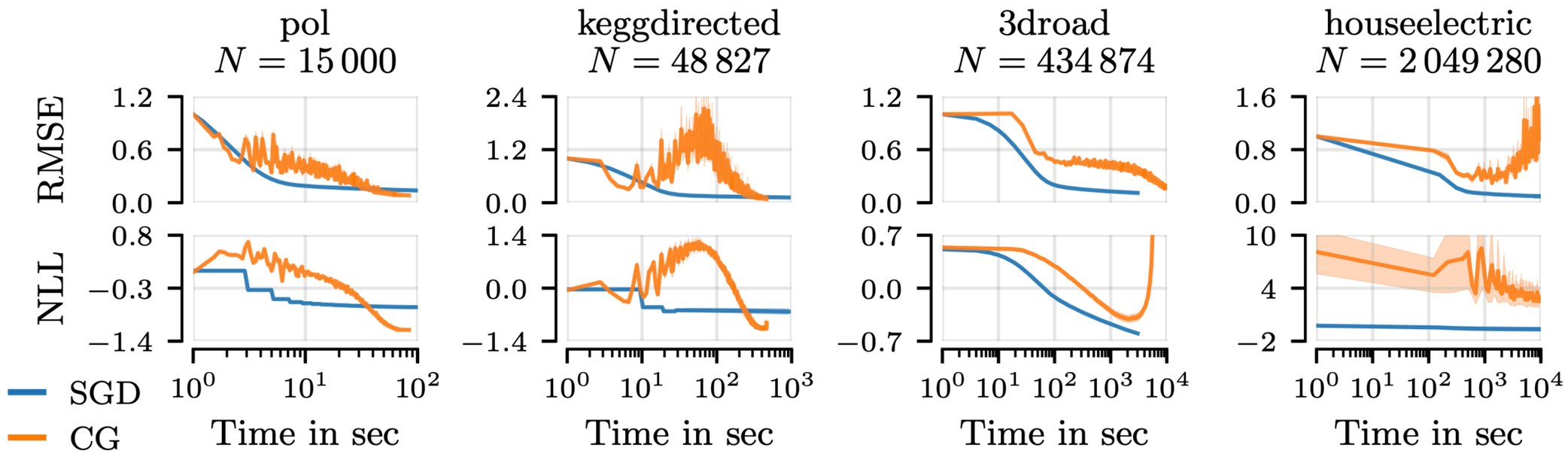
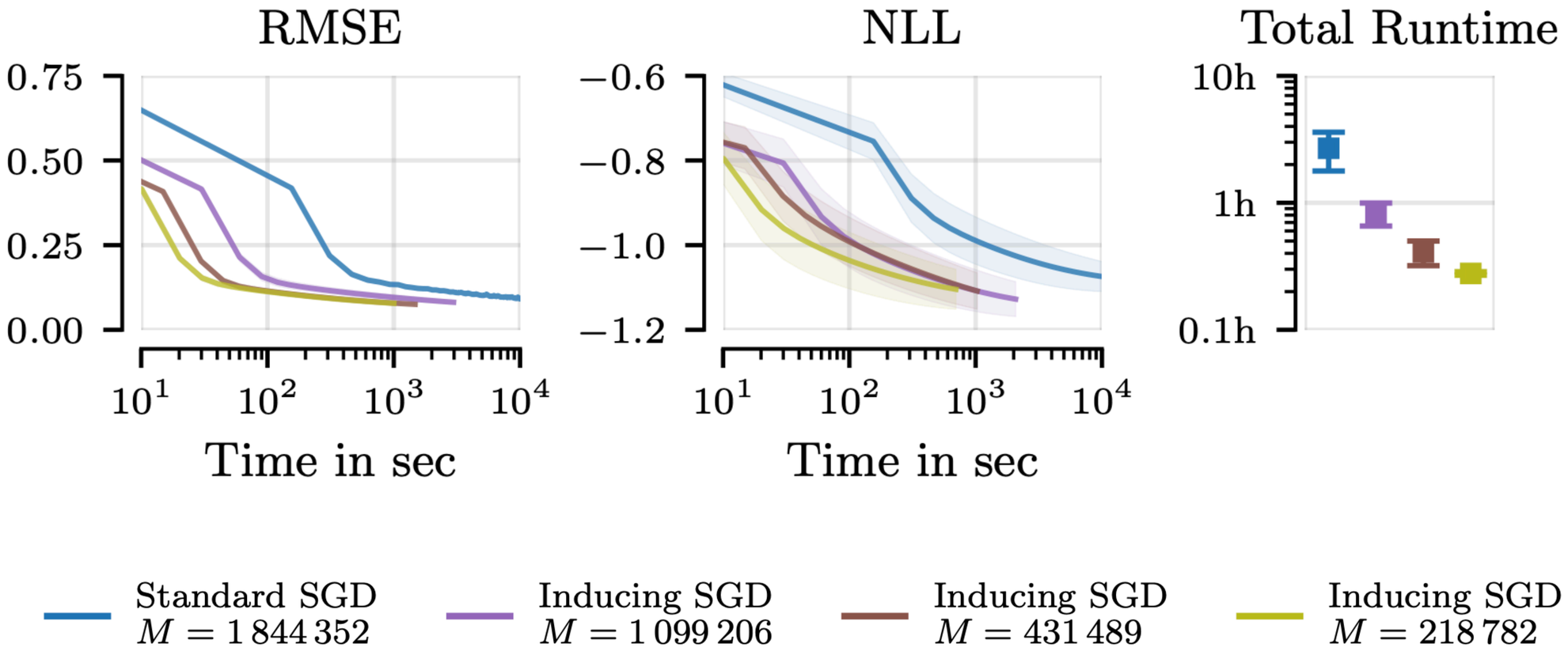


Figure 5: Test RMSE and NLL as a function of compute time on a TPUv2 core for CG and SGD.

Inducing point variant of SGD

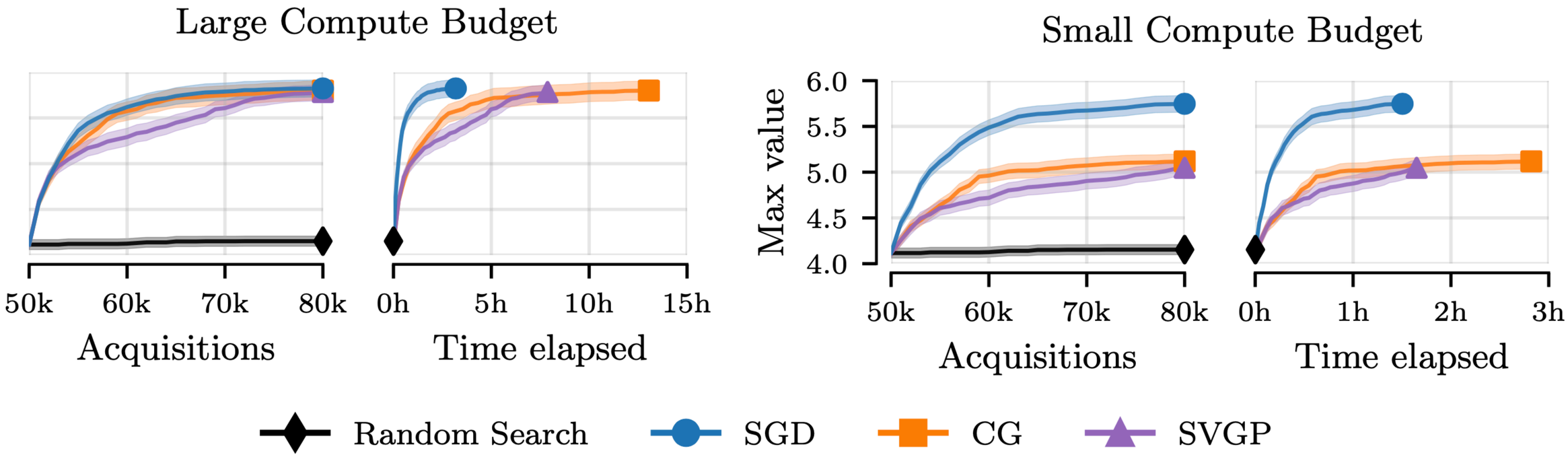
- Inducing points take us from O(N) to O(M) and we are free to choose number of inducing points M and their location



Bayesian Optimisation: Thompson Sampling

Setup:

- True function is a GP prior samples and our models are well-specified
- We consider 5 length scales and 10 seeds for each lenthscale
- We start with 50k uniformly sampled points and we collect 1000 points for 30 steps



QUESTIONS?