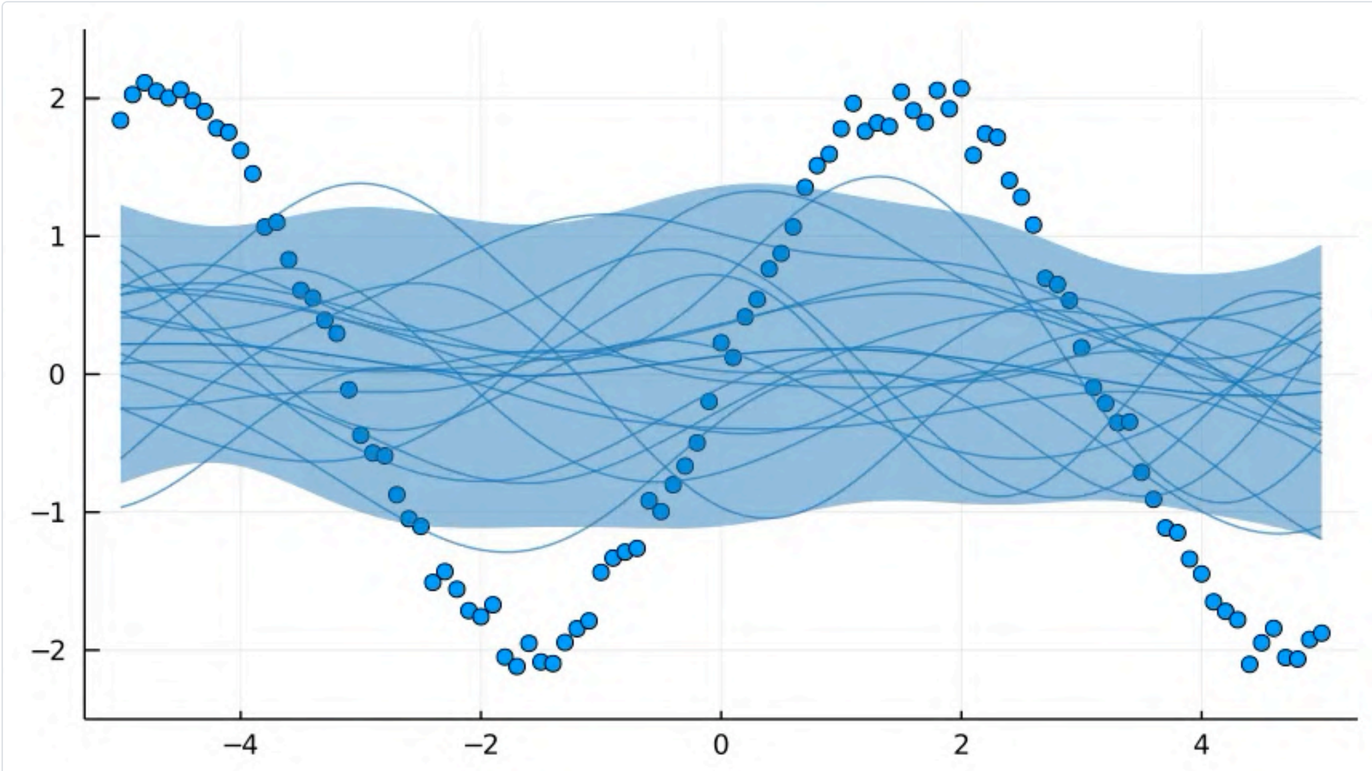
Talk for ETH Zürich

Sampling from Gaussian Process Posteriors using Stochastic Gradient Descent

UNIVERSITY OF CAMBRIDGE

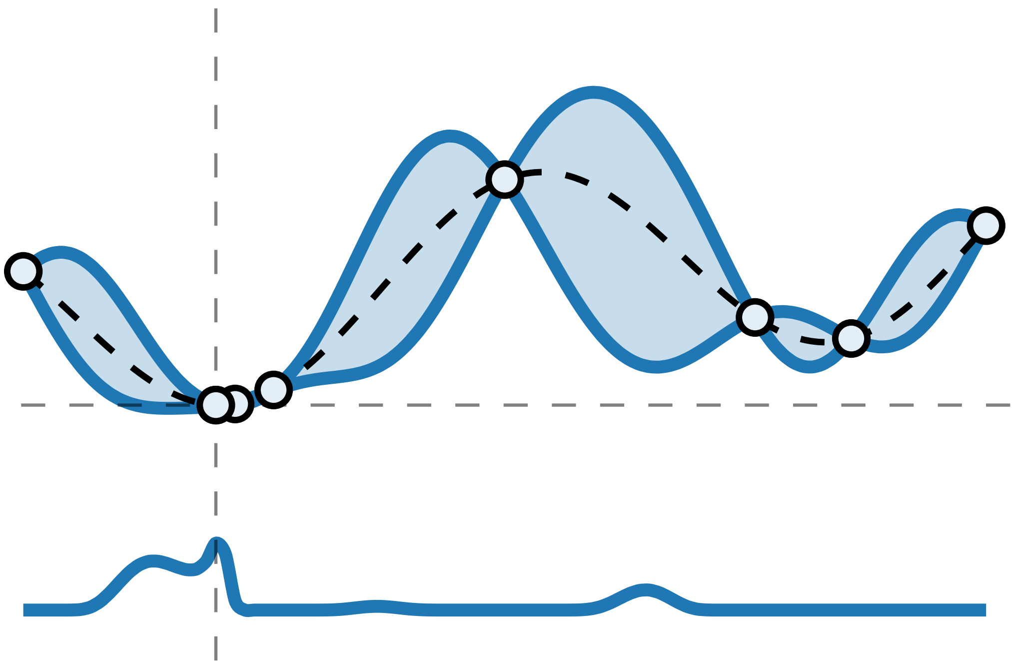
Alexander Terenin
https://avt.im/ · @avt_im

Gaussian Processes



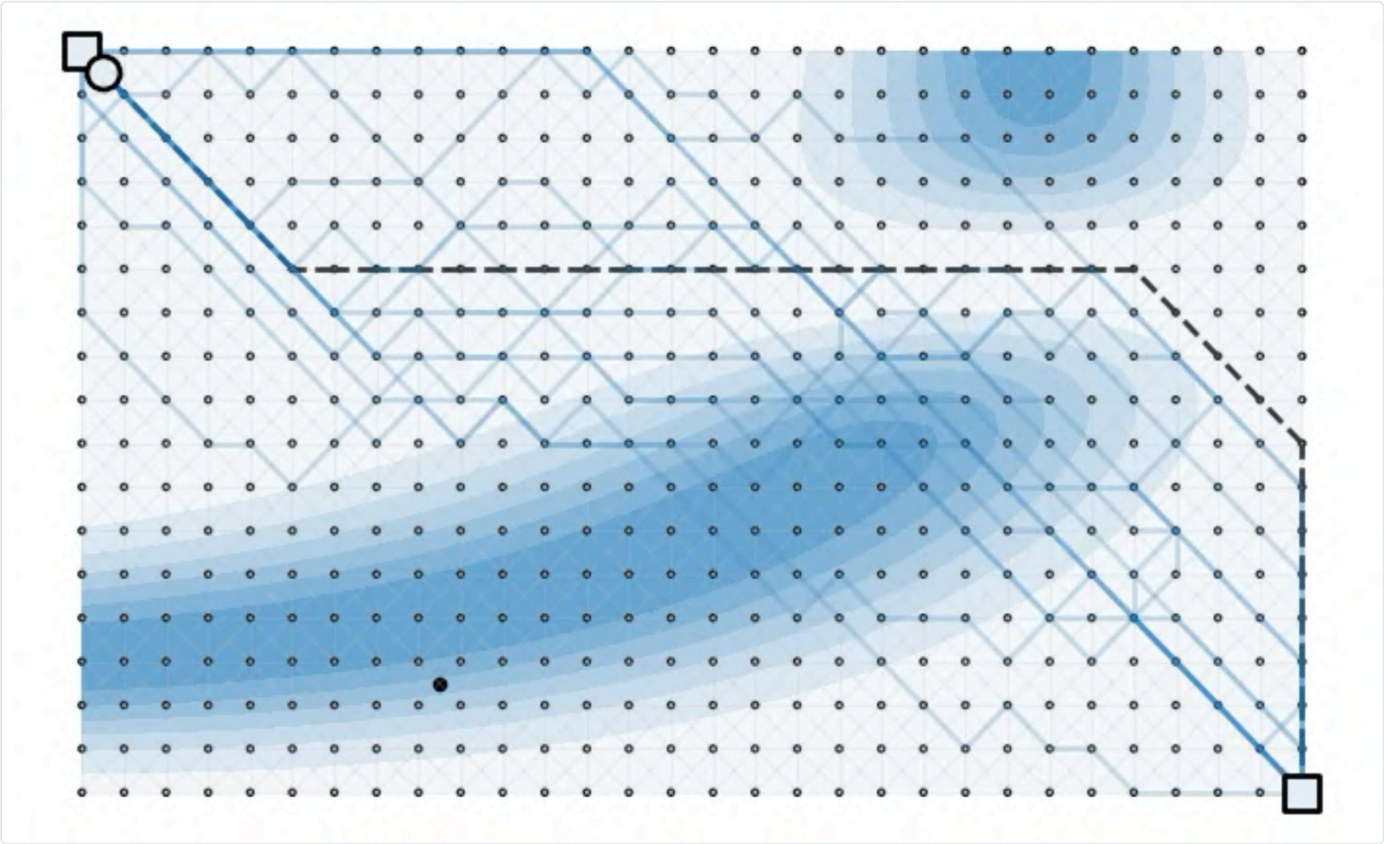
Probabilistic formulation provides uncertainty

Bayesian Optimization

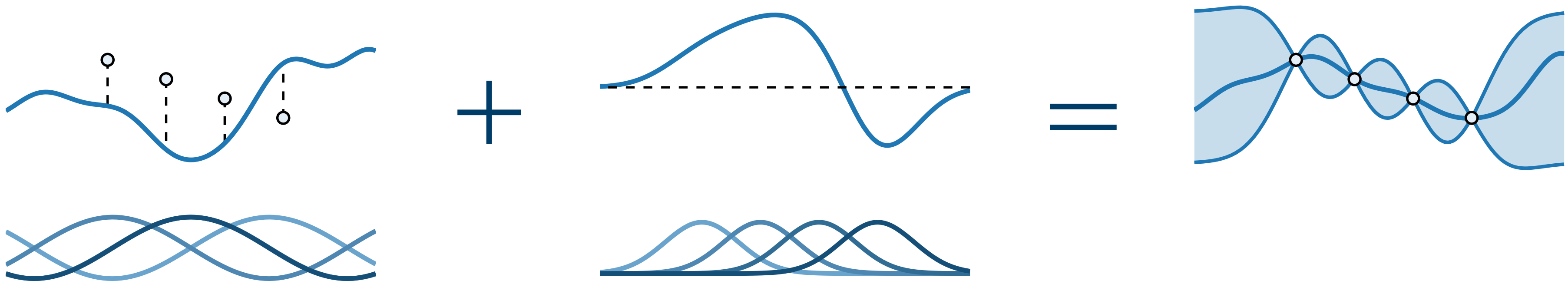


Automatic explore-exploit tradeoff

From Bayesian Optimization to Bayesian Interactive Decision-making



Pathwise Conditioning



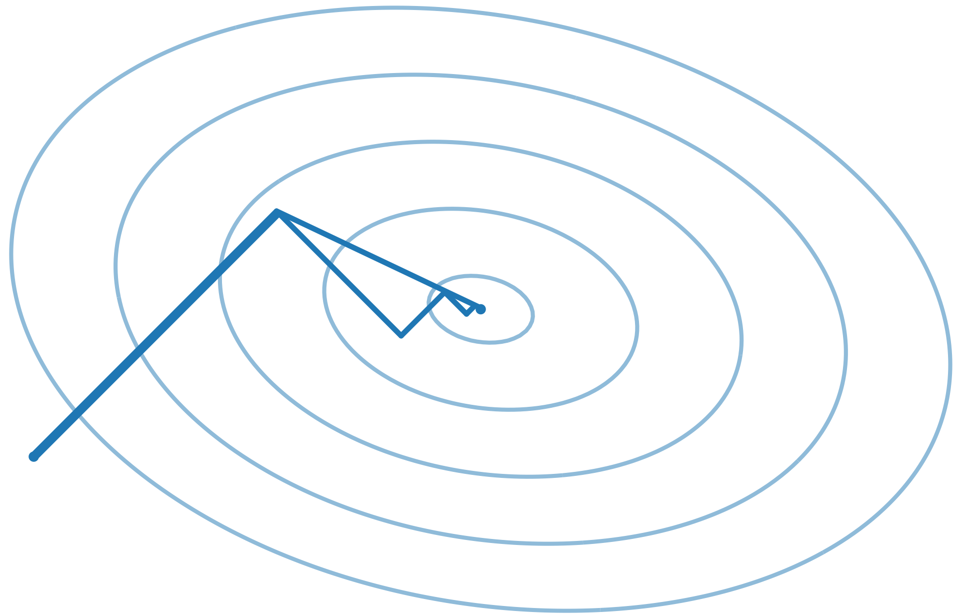
prior term | data term | posterior Gaussian process

$$(f \mid \boldsymbol{y})(\cdot) = f(\cdot) + \sum_{i=1}^{N} v_i k(x_i, \cdot) \qquad \boldsymbol{v} = \mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}(\boldsymbol{y} - f(\boldsymbol{x}))$$

$\boldsymbol{v}$: representer weights

$k(x_i, \cdot)$: canonical basis functions

Conjugate Gradients



Refinement of gradient descent for solving linear systems $\mathbf{A}^{-1}\boldsymbol{b}$

Convergence rate is much faster than gradient descent

Precise rate depends mainly on $\mathbf{cond}(\mathbf{A})$

Numerical Stability

Condition number: quantifies difficulty of solving $\mathbf{A}^{-1}\boldsymbol{b}$

$$\operatorname{cond}(\mathbf{A}) = \lim_{\varepsilon \to 0} \sup_{\|\boldsymbol{\delta}\| \leq \varepsilon \|\boldsymbol{b}\|} \frac{\left\|\mathbf{A}^{-1}(\boldsymbol{b} + \boldsymbol{\delta}) - \mathbf{A}^{-1}\boldsymbol{b}\right\|_2}{\varepsilon \left\|\mathbf{A}^{-1}\boldsymbol{b}\right\|_2} = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}$$

$\lambda_{\min}, \lambda_{\max}$: eigenvalues

Condition Numbers of Kernel Matrices

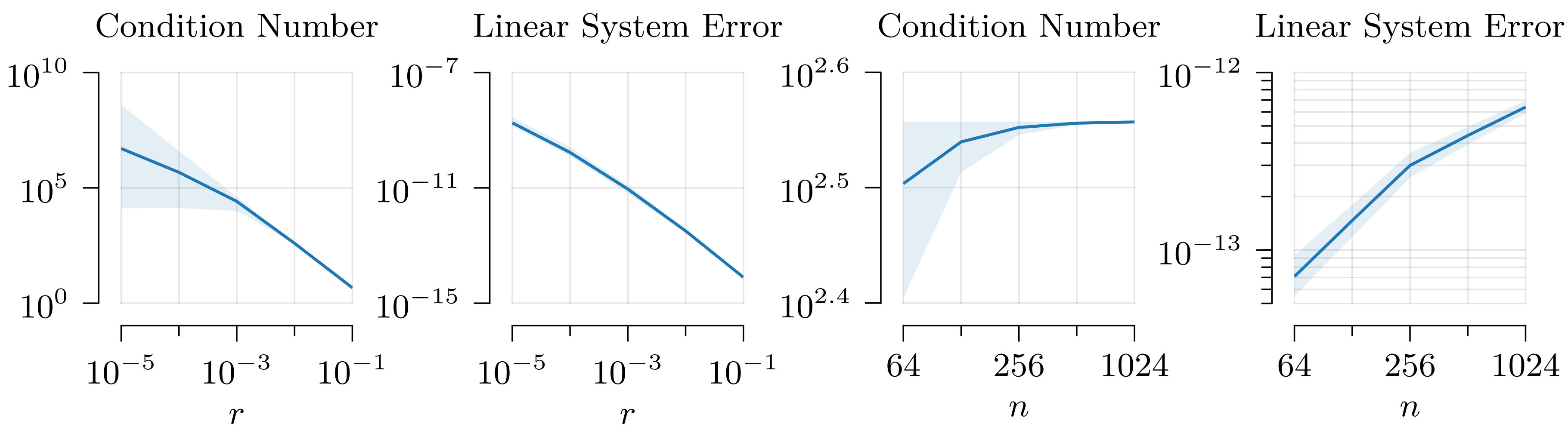
Are kernel matrices always well-conditioned? **No.**

One-dimensional time series on grid: Kac–Murdock–Szegö matrix

$$\mathbf{K}_{\boldsymbol{xx}} = \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{n-1} \\ \rho & 1 & \rho & \dots & \rho^{n-2} \\ \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \dots & 1 \end{pmatrix}$$

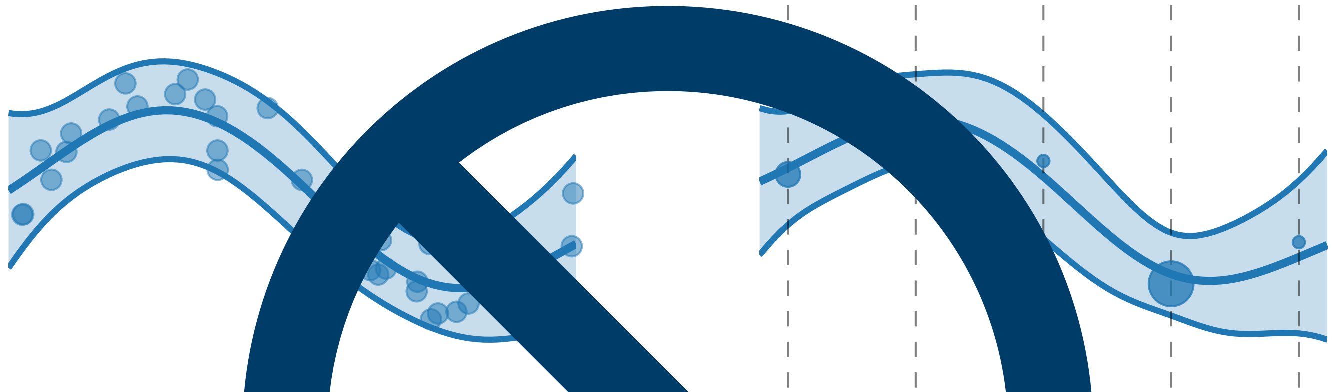
for which $\frac{1+2\rho+2\rho\varepsilon+\rho^2}{1-2\rho-2\rho\varepsilon+\rho^2} \leq \operatorname{cond}(\mathbf{K}_{\boldsymbol{xx}}) \leq \frac{(1+\rho)^2}{(1-\rho)^2}$, where $\varepsilon = \frac{\pi^2}{N+1}$.

Condition Numbers of Kernel Matrices



Problem: too much correlation ⇝ *points too close by*

Minimum Separation



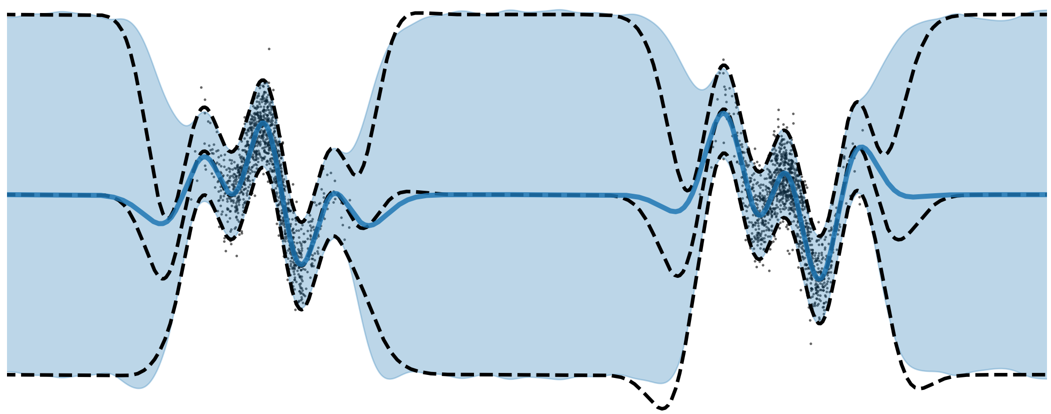
Separation: minimum distance between distinct z_i and z_j

Proposition. Assuming spatial decay and stationarity, separation controls $\mathrm{cond}(\mathbf{K}_{zz})$ uniformly in M.

Idea: use this to select numerically stable inducing points

Sampling from Gaussian Process Posteriors using Stochastic Gradient Descent

Jihao Andreas Lin,* Javier Antorán,* Shreyas Padhy,*
David Janz, José Miguel Hernández-Lobato, Alexander Terenin



*equal contribution

Have you tried stochastic gradient descent?

Conventional wisdom in deep learning:

- SGD variants are empirically often the best optimization algorithms
- ADAM is extremely effective, even on non-convex problems
- Minibatch-based training critical part of scalability

Why not try it out for Gaussian process posterior sample paths?

Gaussian Process Posteriors via Randomized Optimization Objectives

Split into posterior mean and uncertainty reduction terms

$$
\begin{aligned}
(f \mid \boldsymbol{y})(\cdot) &= f(\cdot) + \mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma})^{-1}(\boldsymbol{y} - f(\boldsymbol{x}) - \boldsymbol{\varepsilon}) \\
&= f(\cdot) + \sum_{i=1}^{N} v_i^* k(x_i, \cdot) + \sum_{i=1}^{N} \alpha_i^* k(x_i, \cdot)
\end{aligned}
$$

where

$$
\begin{aligned}
\boldsymbol{v}^* &= \operatorname*{arg\,min}_{\boldsymbol{v} \in \mathbb{R}^N} \sum_{i=1}^{N} \frac{(y_i - \mathbf{K}_{x_i \boldsymbol{x}} \boldsymbol{v})^2}{\Sigma_{ii}} + \boldsymbol{v}^T \mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{v} \\
\boldsymbol{\alpha}^* &= \operatorname*{arg\,min}_{\boldsymbol{\alpha} \in \mathbb{R}^N} \sum_{i=1}^{N} \frac{(f(x_i) + \varepsilon_i - \mathbf{K}_{x_i \boldsymbol{x}} \boldsymbol{\alpha})^2}{\Sigma_{ii}} + \boldsymbol{\alpha}^T \mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} \boldsymbol{\alpha}.
\end{aligned}
$$

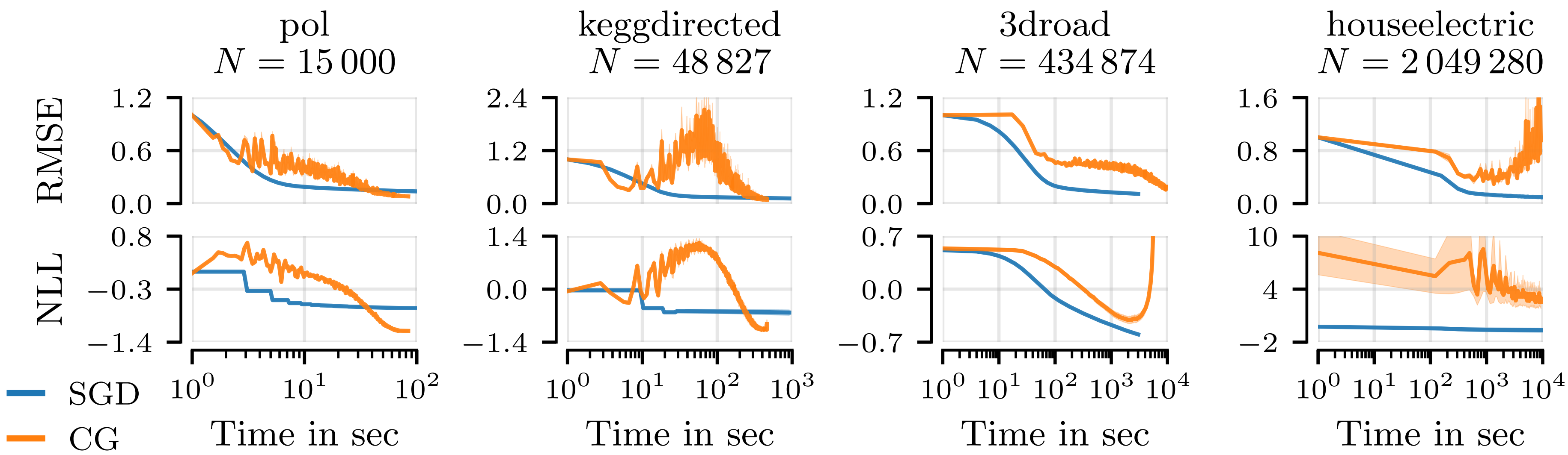
Gaussian Process Posteriors via Randomized Optimization Objectives

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

$$\boldsymbol{\alpha}^* = \operatorname*{arg\,min}_{\boldsymbol{\alpha}\in\mathbb{R}^N} \sum_{i=1}^{N} \frac{(f(x_i) + \varepsilon_i - \mathbf{K}_{x_i\boldsymbol{x}}\boldsymbol{\alpha})^2}{\Sigma_{ii}} + \boldsymbol{\alpha}^T\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\alpha}.$$

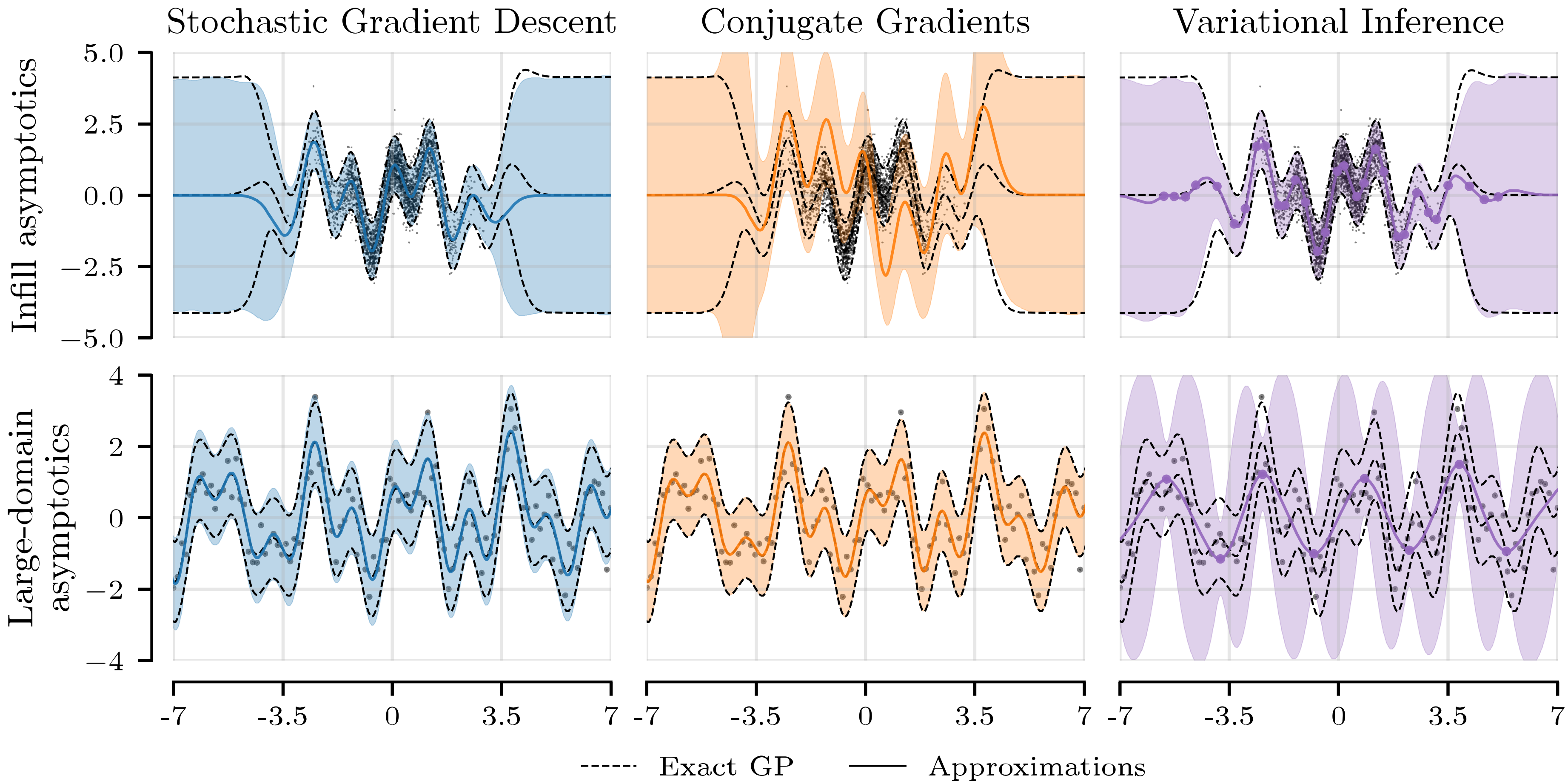
First term: apply mini-batch estimation
Second term: evaluate stochastically via random Fourier features
Variance reduction trick: shift ε_i into regularizer
Use stochastic gradient descent with Polyak averaging

Stochastic Gradient Descent



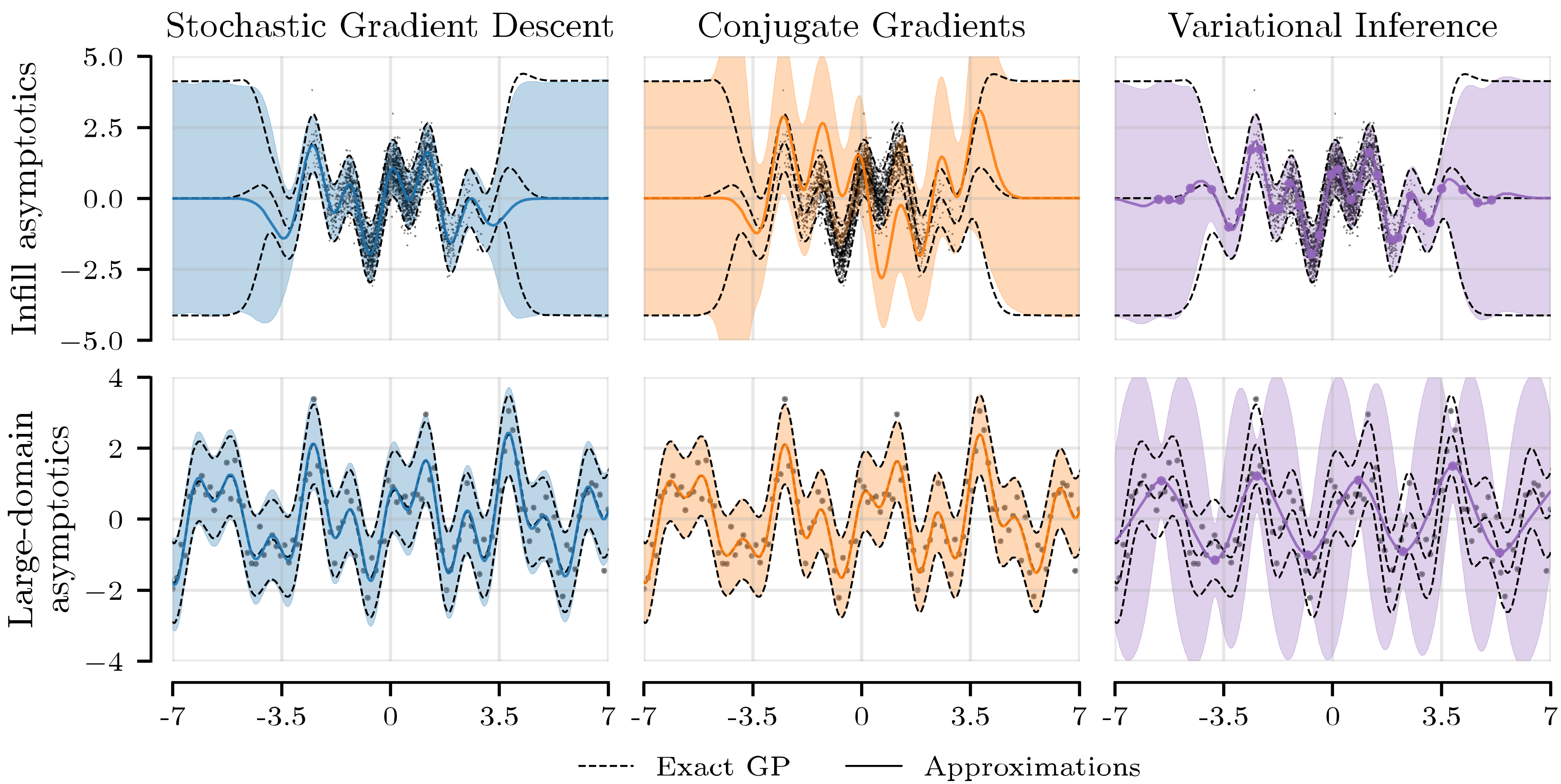
It works better than conjugate gradients on test data? *Wait, what?*

What happens in one dimension?



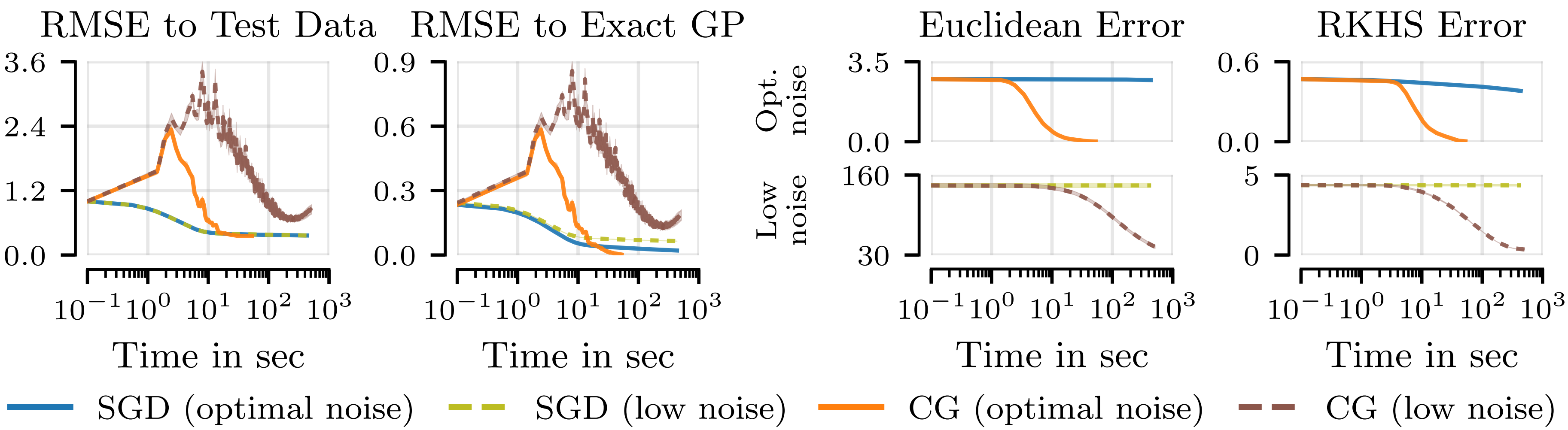
Performance depends on data-generation asymptotics

What happens in one dimension?



SGD does not converge to the correct solution, but still produces reasonable error bars $\rightsquigarrow$ implicit bias?

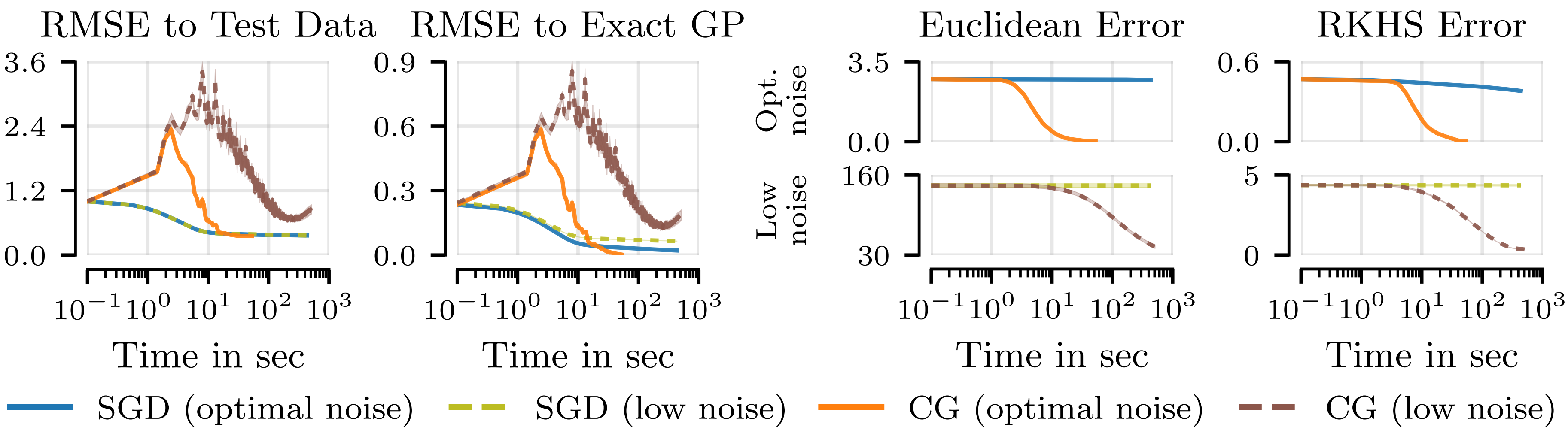
Convergence: Euclidean and RKHS Norms



No convergence in representer weight space or in the reproducing kernel Hilbert space

Good test performance

Convergence: Euclidean and RKHS Norms

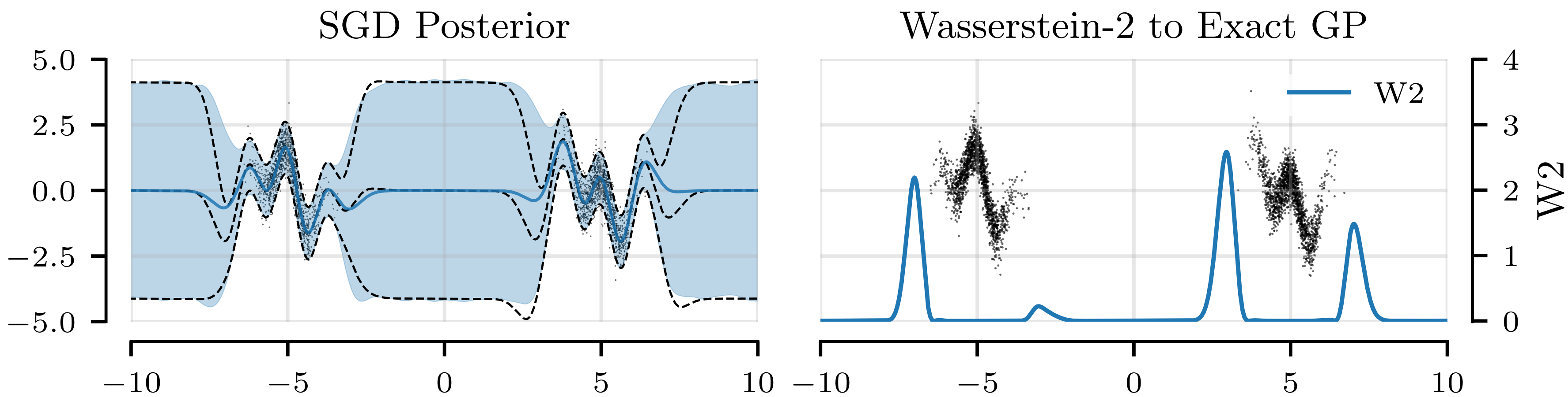


Performance not significantly affected by noise

Unstable optimization problem ⇝ benign non-convergence

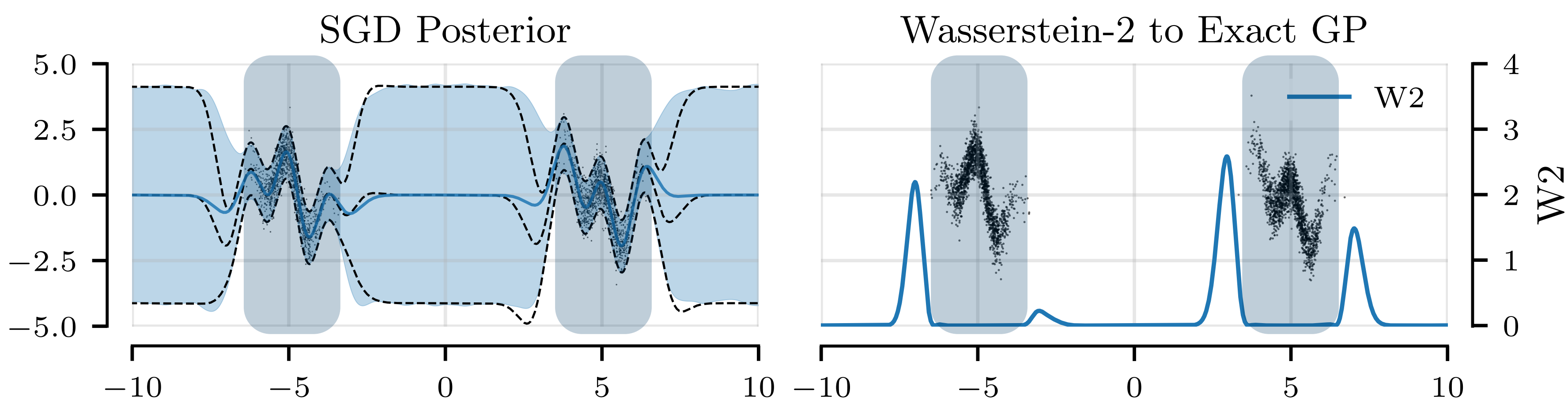
⇝ implicit bias

Where does SGD's implicit bias affect predictions?



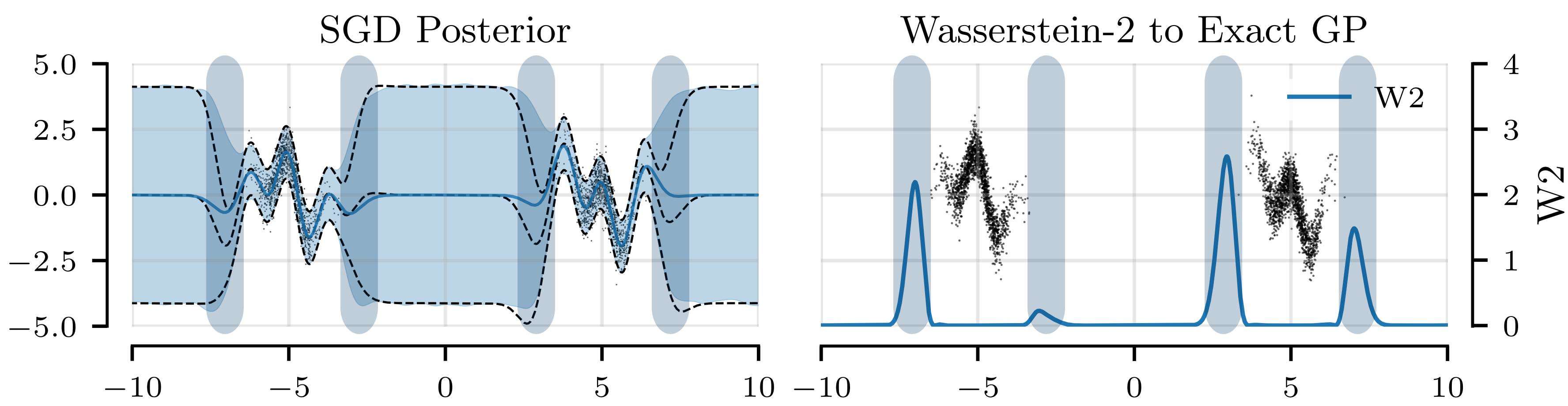
Error seems to concentrate away from data, but not too far away?

Where does SGD's implicit bias affect predictions?



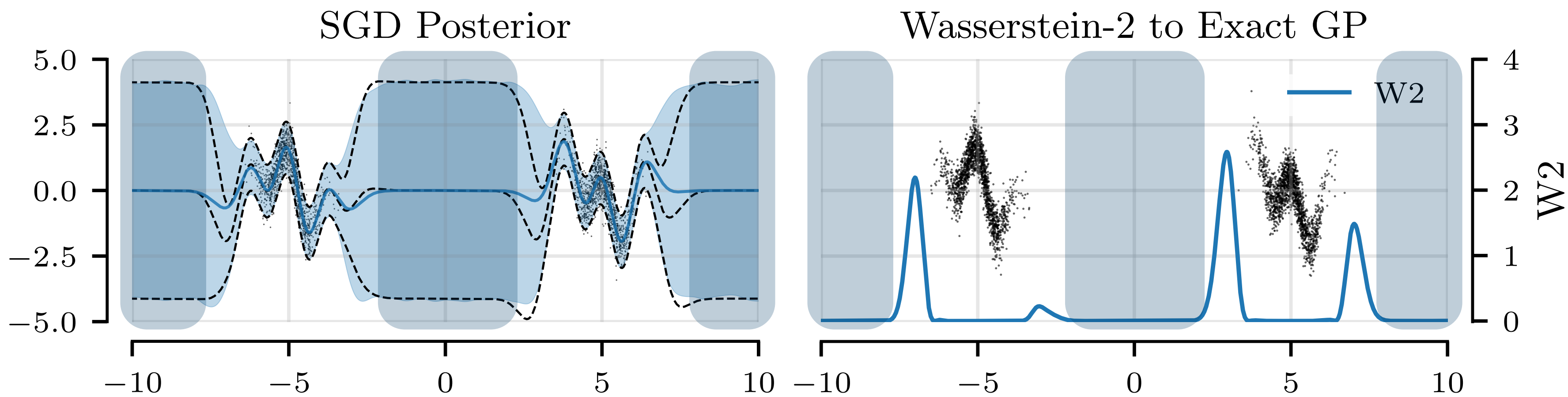
Interpolation region

Where does SGD's implicit bias affect predictions?



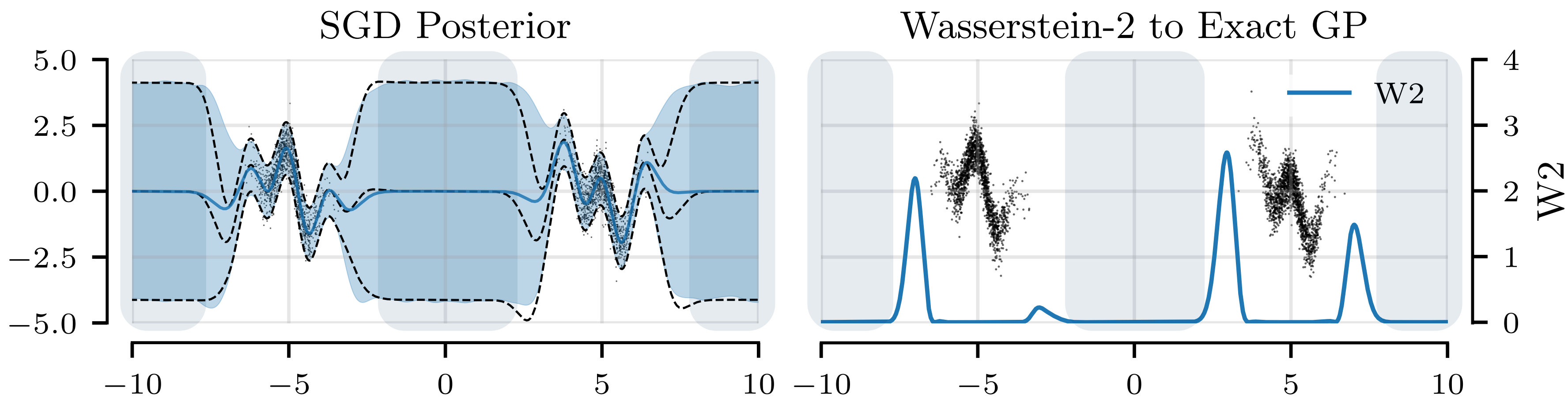
Extrapolation region

Where does SGD's implicit bias affect predictions?



Far-away region

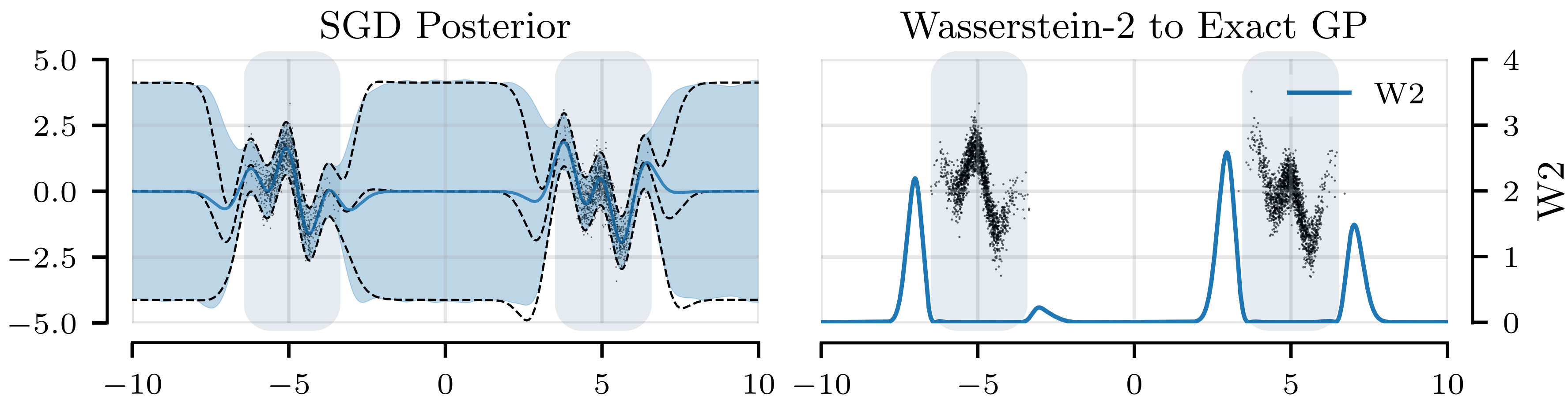
The Far-away Region



$$(\boldsymbol{f} \mid \boldsymbol{y})(\cdot) = \boldsymbol{f}(\cdot) + \sum_{i=1}^{N} \boldsymbol{v}_i \boldsymbol{k}(\boldsymbol{x}_i, \cdot)$$

Kernel decays in space $\rightsquigarrow$ predictions revert to prior

The Interpolation Region



Low approximation error where data is dense

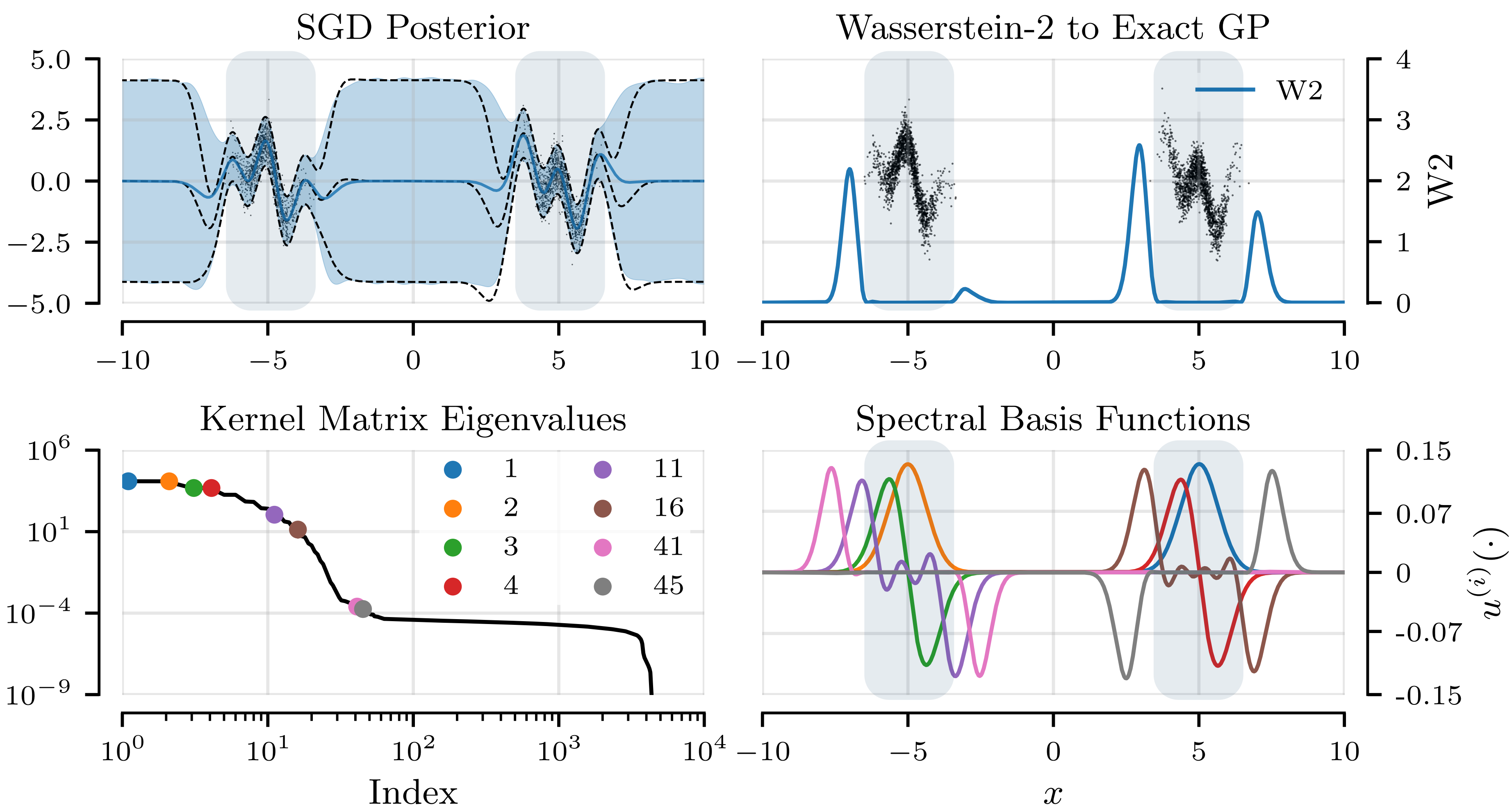
The Interpolation Region

Idea: maybe SGD (a) converges fast on a subspace, and (b) obtains something *arbitrary but benign* on the rest of the space?

Let $\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ be the eigendecomposition of the kernel matrix. Define the *spectral basis functions*

$$u^{(i)}(\cdot) = \sum_{j=1}^{N} \frac{U_{ji}}{\sqrt{\lambda_i}} k(x_j, \cdot).$$

The Interpolation Region



Large-eigenvalue spectral basis functions concentrate on data

The Interpolation Region

Proposition. With probability $1-\delta$, we have

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

η: learning rate

σ^2: noise variance

λ_i: kernel matrix eigenvalues

G: gradient's sub-Gaussian coefficient

SGD converges fast with respect to top spectral basis functions

The Interpolation Region

Where do the top spectral basis functions concentrate?

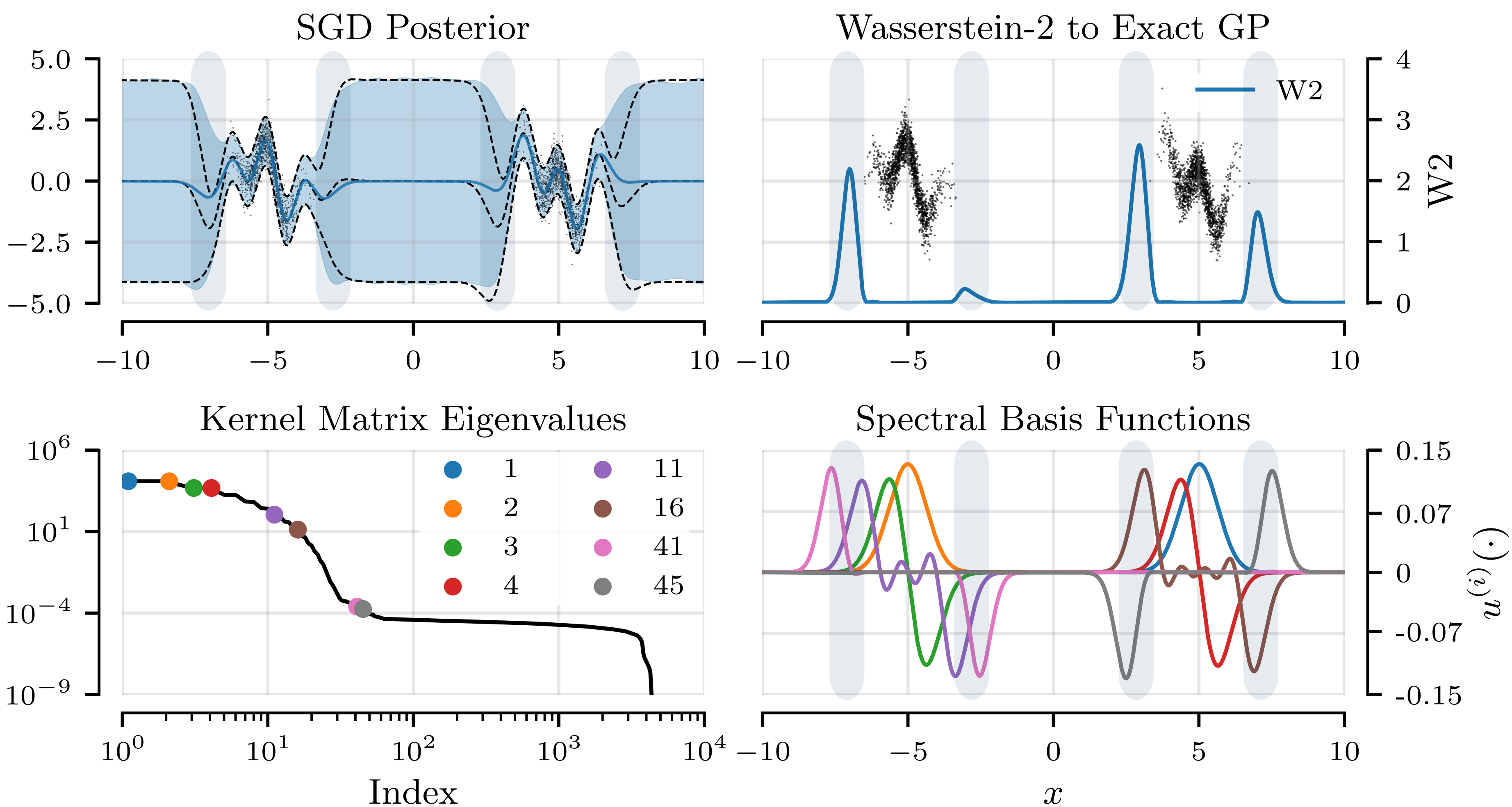
- Idea: lift Courant–Fischer eigenvector characterization to the RKHS

Proposition. The spectral basis functions can be written

$$u^{(i)}(\cdot) = \operatorname*{arg\,max}_{u \in H_k} \left\{ \sum_{i=1}^{N} u(x_i)^2 : \begin{array}{c} \|u\|_{H_k} = 1 \\ \langle u, u^{(j)} \rangle_{H_k} = 0, \forall j < i \end{array} \right\}.$$

Spectral basis functions concentrate on the data *as much as possible,* while remaining orthogonal to those with larger eigenvalues

The Extrapolation Region



High error: where small-eigenvalue spectral basis functions concentrate

SGD's implicit bias for Gaussian processes

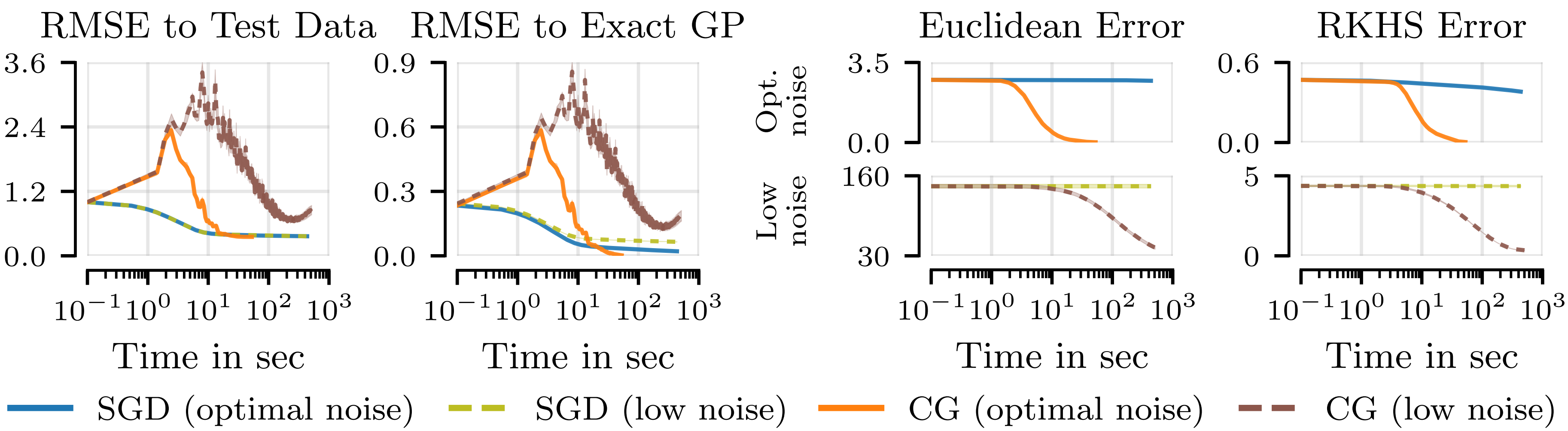
Implicit bias: SGD converges quickly near the data, and causes no harm far from the data

Error concentrates in regions (a) without much data, which also (b) aren't located too far from the data:

- Lack of data $\rightsquigarrow$ predictions are mostly arbitrary
- Empirically: functions shrink to prior faster than exact posterior

Benign non-convergence $\rightsquigarrow$ robustness to instability

Performance



Conjugate gradients: non-monotonic test error, in spite of monotonic convergence

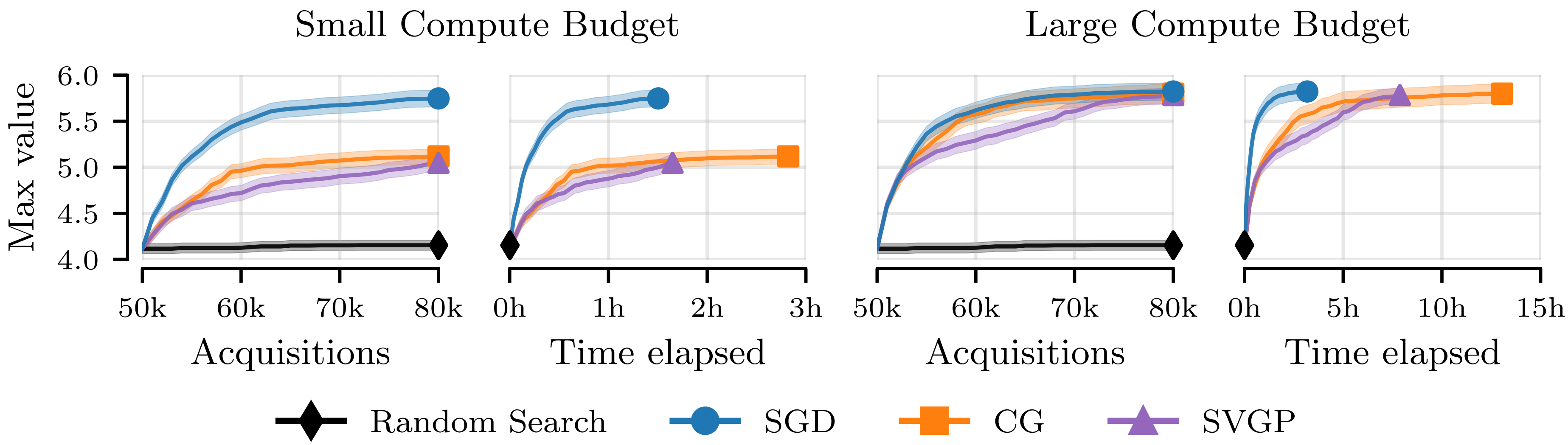
SGD: almost always monotonic decrease in test error

Performance

	Dataset	POL	ELEVATORS	BIKE	PROTEIN	KEGGDIR	3DROAD	SONG	BUZZ	HOUSEELEC
	N	15000	16599	17379	45730	48827	434874	515345	583250	2049280
RMSE	SGD	0.13 ± 0.00	0.38 ± 0.00	0.11 ± 0.00	**0.51 ± 0.00**	0.12 ± 0.00	**0.11 ± 0.00**	**0.80 ± 0.00**	0.42 ± 0.01	**0.09 ± 0.00**
	CG	**0.08 ± 0.00**	**0.35 ± 0.00**	**0.04 ± 0.00**	**0.50 ± 0.00**	**0.08 ± 0.00**	0.15 ± 0.01	0.85 ± 0.03	1.41 ± 0.08	0.87 ± 0.14
	SVGP	0.10 ± 0.00	0.37 ± 0.00	0.08 ± 0.00	0.62 ± 0.00	0.10 ± 0.00	0.64 ± 0.01	0.82 ± 0.00	**0.34 ± 0.00**	0.10 ± 0.02
RMSE†	SGD	**0.13 ± 0.00**	**0.38 ± 0.00**	0.11 ± 0.00	**0.51 ± 0.00**	**0.12 ± 0.00**	**0.11 ± 0.00**	**0.80 ± 0.00**	**0.42 ± 0.01**	**0.09 ± 0.00**
	CG	0.16 ± 0.01	0.68 ± 0.09	**0.05 ± 0.01**	3.03 ± 0.23	9.79 ± 1.06	0.34 ± 0.02	0.83 ± 0.02	5.66 ± 1.14	0.93 ± 0.19
	SVGP	—	—	—	—	—	—	—	—	—
Hours	SGD	0.06 ± 0.00	0.06 ± 0.00	0.09 ± 0.00	0.12 ± 0.00	0.25 ± 0.00	0.46 ± 0.19	3.67 ± 0.24	5.78 ± 1.02	2.69 ± 0.91
	CG	0.04 ± 0.01	**0.03 ± 0.01**	0.05 ± 0.00	0.15 ± 0.03	0.21 ± 0.03	1.42 ± 0.60	3.25 ± 0.04	5.85 ± 0.80	2.62 ± 0.01
	SVGP	**0.04 ± 0.00**	**0.04 ± 0.00**	**0.04 ± 0.00**	**0.04 ± 0.00**	**0.04 ± 0.00**	**0.04 ± 0.00**	**0.05 ± 0.00**	**0.04 ± 0.00**	**0.04 ± 0.00**
NLL	SGD	-0.70 ± 0.02	0.47 ± 0.00	-0.48 ± 0.08	0.64 ± 0.01	-0.62 ± 0.07	**-0.60 ± 0.00**	**1.21 ± 0.00**	0.83 ± 0.07	**-1.09 ± 0.04**
	CG	**-1.17 ± 0.01**	**0.38 ± 0.00**	**-2.62 ± 0.06**	**0.62 ± 0.01**	**-0.92 ± 0.10**	16.27 ± 0.45	1.36 ± 0.07	2.38 ± 0.08	2.07 ± 0.58
	SVGP	-0.64 ± 0.02	0.44 ± 0.00	-1.47 ± 0.02	0.94 ± 0.01	**-0.89 ± 0.03**	0.94 ± 0.03	1.23 ± 0.00	**0.29 ± 0.04**	-0.94 ± 0.13

Strong predictive performance at sufficient scale

Parallel Thompson Sampling



Uncertainty: strong decision-making performance

Reflections

Numerical analysis conventional wisdom:

- Don't run gradient descent on quadratic objectives
  - It's slow, conjugate gradient works much better
- If CG is slow then your problem is unstable
  - Unstable problems are ill-posed, you should reformulate

This work: a very different way of looking at things

- Don't solve the linear system approximately if the solution isn't inherently needed
- Instead of a well-posed problem, a *well-posed subproblem* might be good enough
- Try SGD! It might work well, including for counterintuitive reasons
- A kernel matrix's eigenvectors carry information about data-density
  - To see this, adopt a function-analytic view given by the spectral basis functions

Thank you!

https://avt.im/ · @avt_im

J. A. Lin,* J. Antorán,* S. Padhy,* D. Janz, J. M. Hernández-Lobato, A. Terenin. Sampling from Gaussian Process Posteriors using Stochastic Gradient Descent. *arXiv:2306.11589*, 2023.

A. Terenin,* D. R. Burt,* A. Artemev, S. Flaxman, M. van der Wilk, C. E. Rasmussen, H. Ge. Numerically Stable Sparse Gaussian Processes via Minimum Separation using Cover Trees. *arXiv: 2210.07893*, 2022.

J. T. Wilson,* V. Borovitskiy,* P. Mostowsky,* A. Terenin,* M. P. Deisenroth. Efficiently Sampling Functions from Gaussian Process Posteriors. *International Conference on Machine Learning*, 2020. **Honorable Mention for Outstanding Paper Award.**

J. T. Wilson,* V. Borovitskiy,* P. Mostowsky,* A. Terenin,* M. P. Deisenroth. Pathwise Conditioning of Gaussian Process. *Journal of Machine Learning Research*, 2021.

*Equal contribution

UNIVERSITY OF CAMBRIDGE