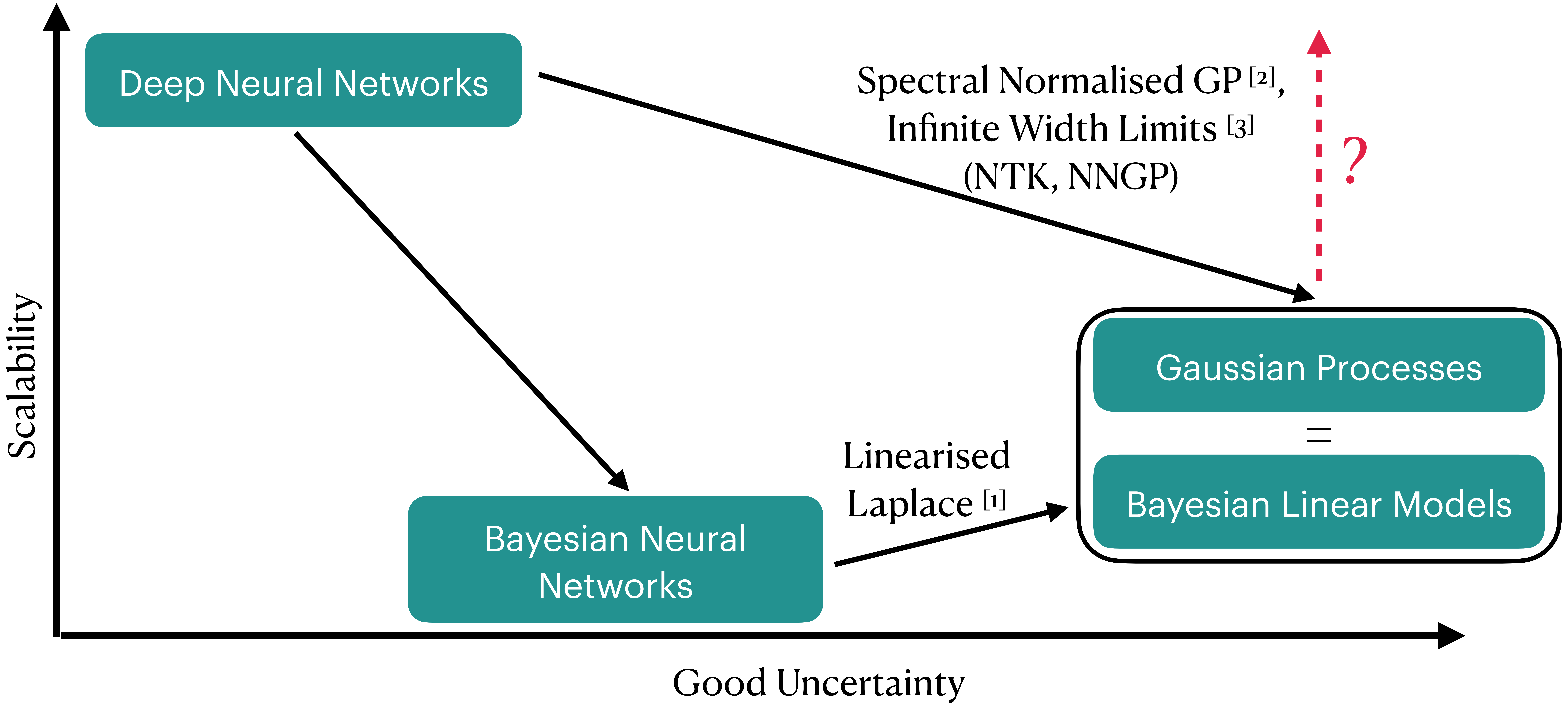
Stochastic Gradient Descent for GPs and Linearised NNs

Shreyas Padhy
Cambridge MLG Group
29 February 2024
SIAM UQ Conference

Computational and Biological Learning
DEPARTMENT OF ENGINEERING

UNIVERSITY OF CAMBRIDGE

The Bayesian Model Landscape



[1] Padhy, S.*, Antorán, J.,*, Barbano, R., Nalisnick, E., ... and Hernández-Lobato, J.M., 2022. Sampling-based inference for large linear models, with application to linearised Laplace. ***ICLR 2023***
[2] Padhy, S.*, Liu, J. Z.*, Ren, J.*, Lin, Z., Wen, Y., Jerfel, G., ... & Lakshminarayanan, B. A simple approach to improve single-model deep uncertainty via distance-awareness. ***JMLR 2023***
[3] Adlam, B., Lee, J., Padhy, S., Nado, Z. and Snoek, J., 2023. Kernel Regression with Infinite-Width Neural Networks on Millions of Examples. *arXiv preprint*

Computational Considerations

Gaussian Processes

$$f \sim \mathrm{GP}(\mu(\,.\,), K(\,.\,,\,.\,))$$

$$\left[\begin{pmatrix} f(X_*) \\ f(X) \end{pmatrix}\right] \sim \mathcal{N}\left(0, \left[\begin{pmatrix} K_{**} & K_{*n} \\ K_{*n}^\top & K_{nn} \end{pmatrix}\right]\right)$$

Posterior Distribution

$$p(f_* \mid f, X, y) = \mathcal{N}(\mu_{f|y}, \Sigma_{f|y})$$

Predictive Mean

$$\mu_{f|y} = K_{*n}(K_{nn} + \sigma^2 I)^{-1} y \qquad \mathcal{O}(n^3)$$

Uncertainty Estimate

$$\Sigma_{f|y} = K_{**} - K_{*n}^\top (K_{nn} + \sigma^2 I)^{-1} K_{n*} \qquad \mathcal{O}(n^3)$$

Can we SGD in the era of deep learning?

- Can we cross the $\mathcal{O}(n^3)$ hurdle using SGD?

- SGD needs -
  - Parametric view of model
  - Unbiased mini-batch objective
  - Linear scaling with n

I. Estimate the Mean of GPs

- We have

$$\mu_{f|y}(X^*) = K_{*n}\left(K_{nn} + \sigma^2 I\right)^{-1} y$$

Representer Weights $\in \mathbb{R}^n$

$$\mu_{f|y}(X^*) = K_{*n}\boldsymbol{v}^* = \sum_{i=1}^{N} K_{*i} v_i^*$$

- Where

$$\boldsymbol{v}^* = (K_{nn} + \sigma^2 I)^{-1} y$$

n Linear System of Equations

Conjugate Gradients

Stochastic Gradient Descent

$$\boldsymbol{v}^* = \underset{\boldsymbol{v}\in\mathbb{R}^N}{\arg\min} \sum_{i=1}^{N} \frac{\left(y_i - K_{x_i,n}\boldsymbol{v}\right)^2}{\sigma^2} + \|\boldsymbol{v}\|^2_{K_{nn}}$$

$$\frac{d\mathcal{L}(\mathbf{v})}{d\mathbf{v}}\bigg|_{\mathbf{v}=\mathbf{v}^*} = 0$$

[1] **Padhy, S.***, Lin, JA*, Antorán, J.,*, ... and Hernández-Lobato, J.M., 2022. Sampling from Gaussian Process Posteriors using Stochastic Gradient Descent. ***NeurIPS 2023***

I. Estimate the Mean of GPs

- We have $\boldsymbol{v}^* = \arg\min_{\boldsymbol{v}\in\mathbb{R}^N} \sum_{i=1}^{N} \frac{\left(y_i - K_{x_i,n}\boldsymbol{v}\right)^2}{\sigma^2} + \|\boldsymbol{v}\|^2_{K_{nn}}$

Easily minibatched

$$\frac{N}{B}\sum_{i}^{B} \frac{\left(y_i - K_{x_i,n}\boldsymbol{v}\right)^2}{\sigma^2}$$

$\mathcal{O}(n^2)$ space

$$\mathbf{v}^\top K_{nn} \mathbf{v}$$

$$K_{nn} \approx \Phi(x)\Phi(x)^T, \quad \Phi(x) \in \mathbb{R}^{n,L}$$

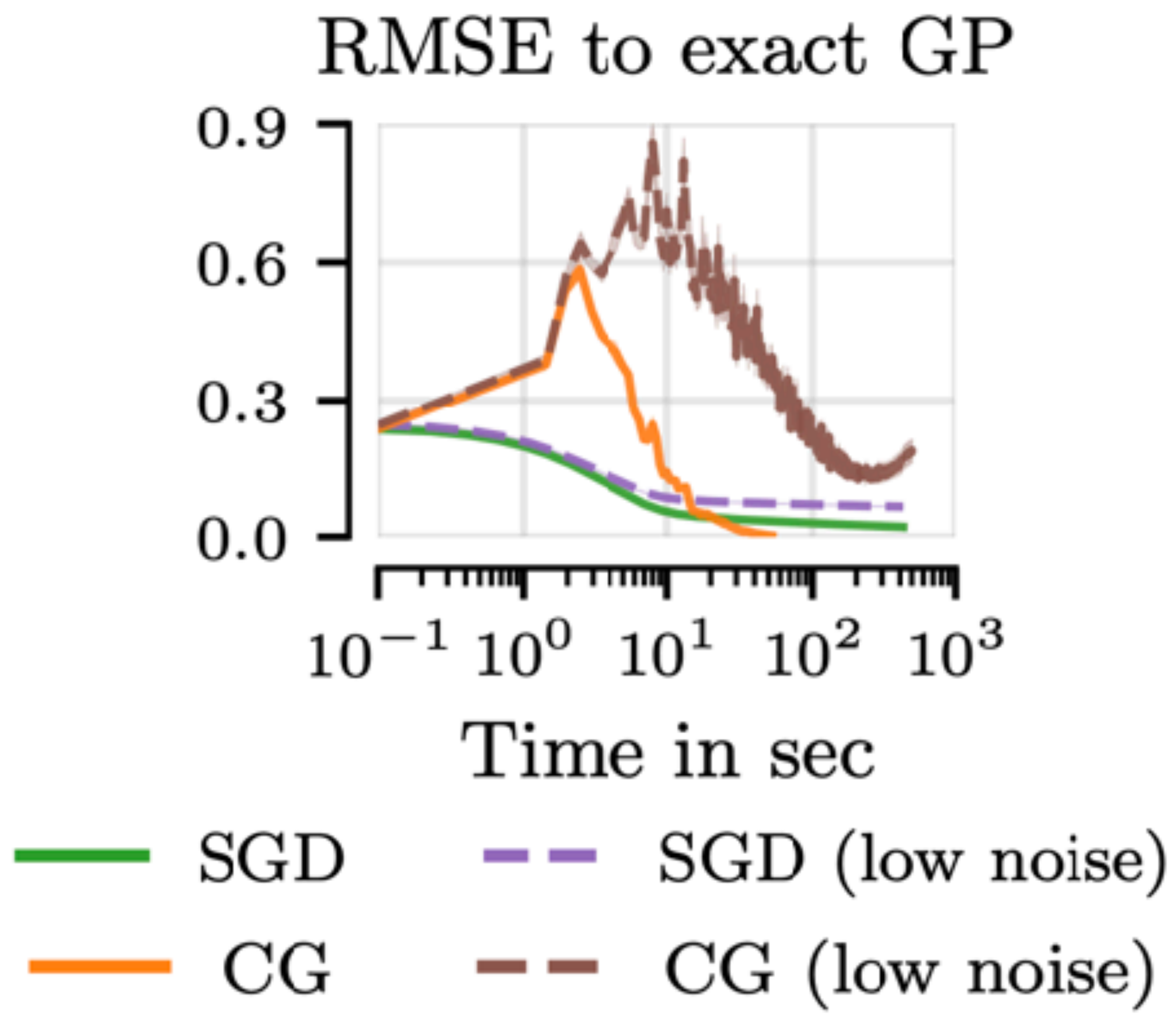
$$\mathbf{v}^\top K_{nn}\mathbf{v} \approx \sum_{\ell=1}^{L} \left(\boldsymbol{v}^T \phi_\ell(x)\right)^2$$

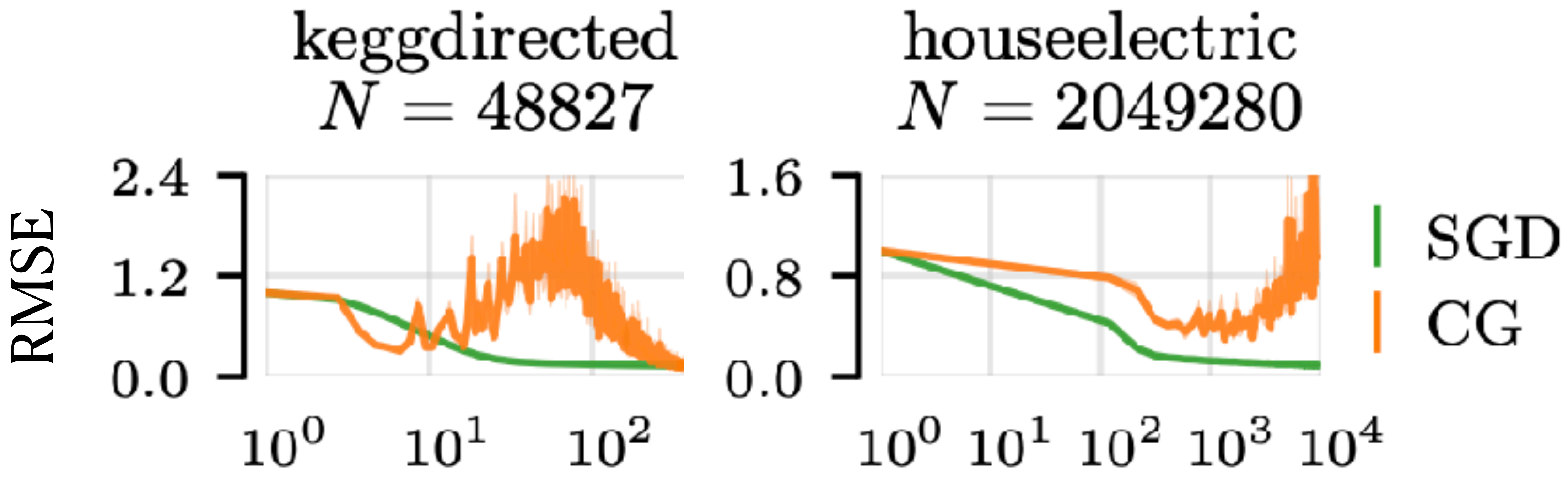
$\mathcal{O}(n)$

$$\frac{N}{B}\sum_{i}^{B} \frac{\left(y_i - K_{x_i,n}\boldsymbol{v}\right)^2}{\sigma^2} + \sum_{\ell=1}^{L} \left(\boldsymbol{v}^T \phi_\ell(x)\right)^2$$

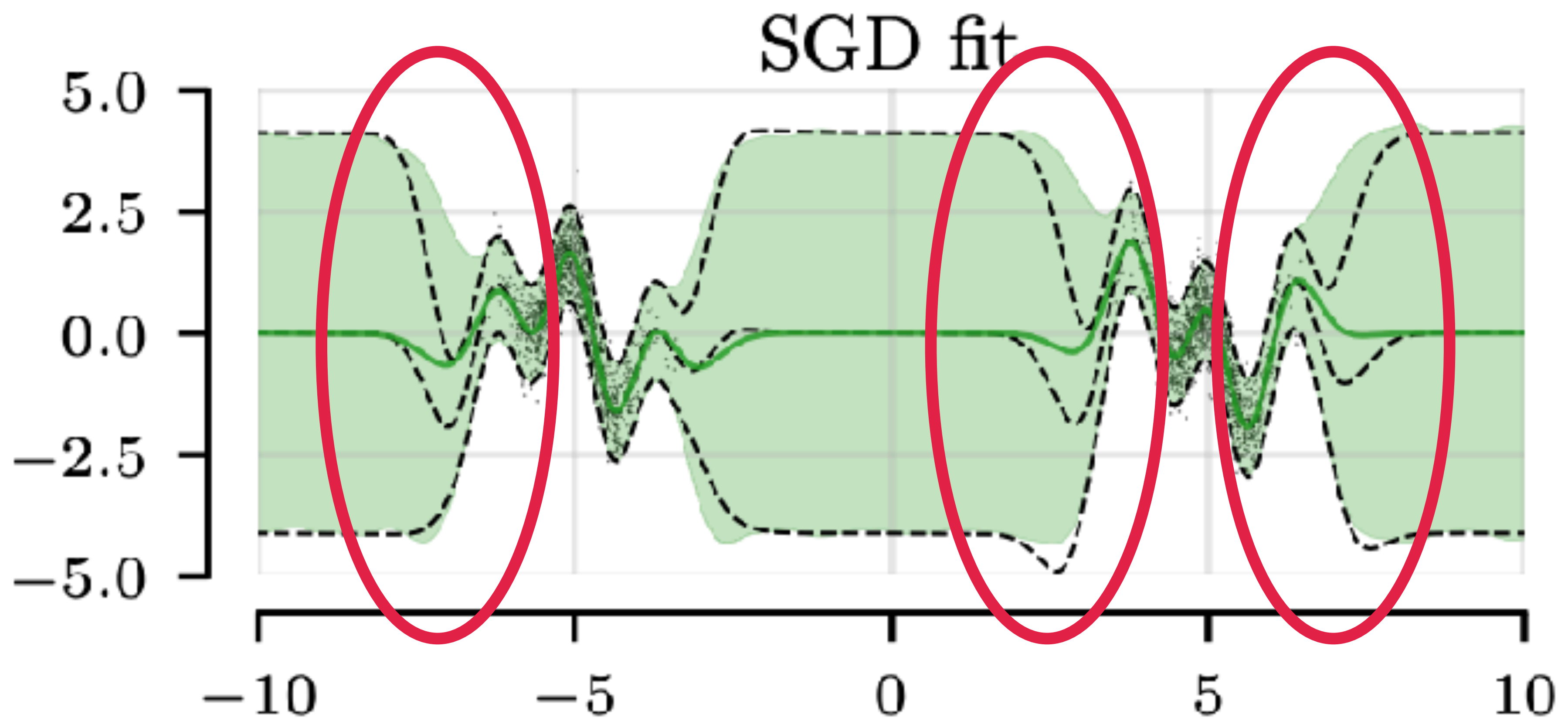
SGD scales much better than CG

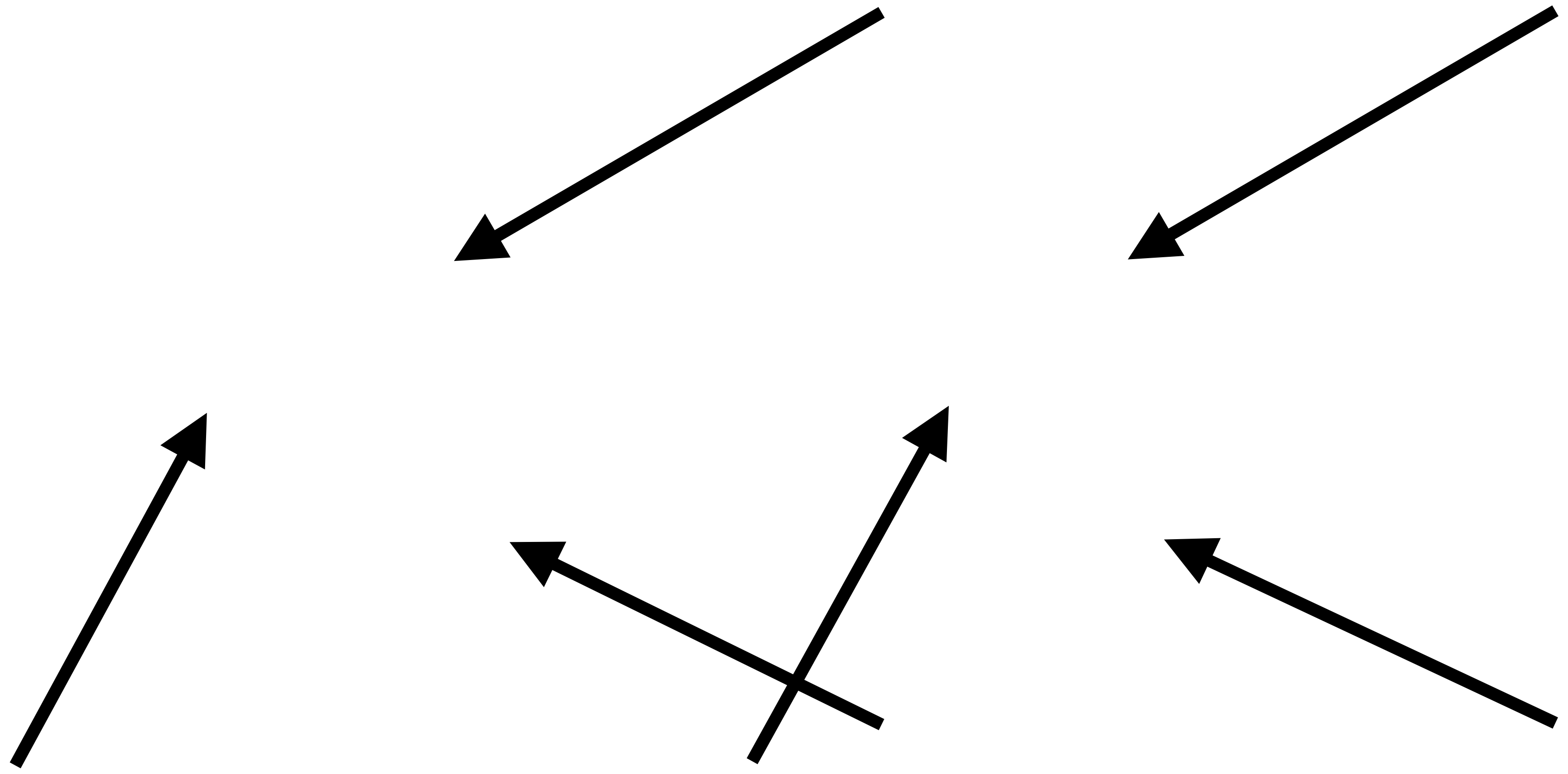
- CG has non-monotonic convergence guarantee in $\mathcal{O}\left(\sqrt{\mathrm{cond}(K_{nn} + \sigma^2 I)} \log \frac{\mathrm{cond}(K_{nn} + \sigma^2)\|y\|}{\varepsilon}\right)$ steps
- SGD monotonically converges (to approx. soln), has no dependence on conditioning!





Spectral Analysis of SGD Behaviour





$$\left\| \operatorname{proj}_{u_i} \mu_{f|\mathbf{y}} - \operatorname{proj}_{u_i} \mu_{\mathrm{SGD}} \right\|_{H_k} \leq \frac{4G+1}{\eta} \sqrt{\frac{\log \frac{N}{\delta}}{t\lambda_i}}$$

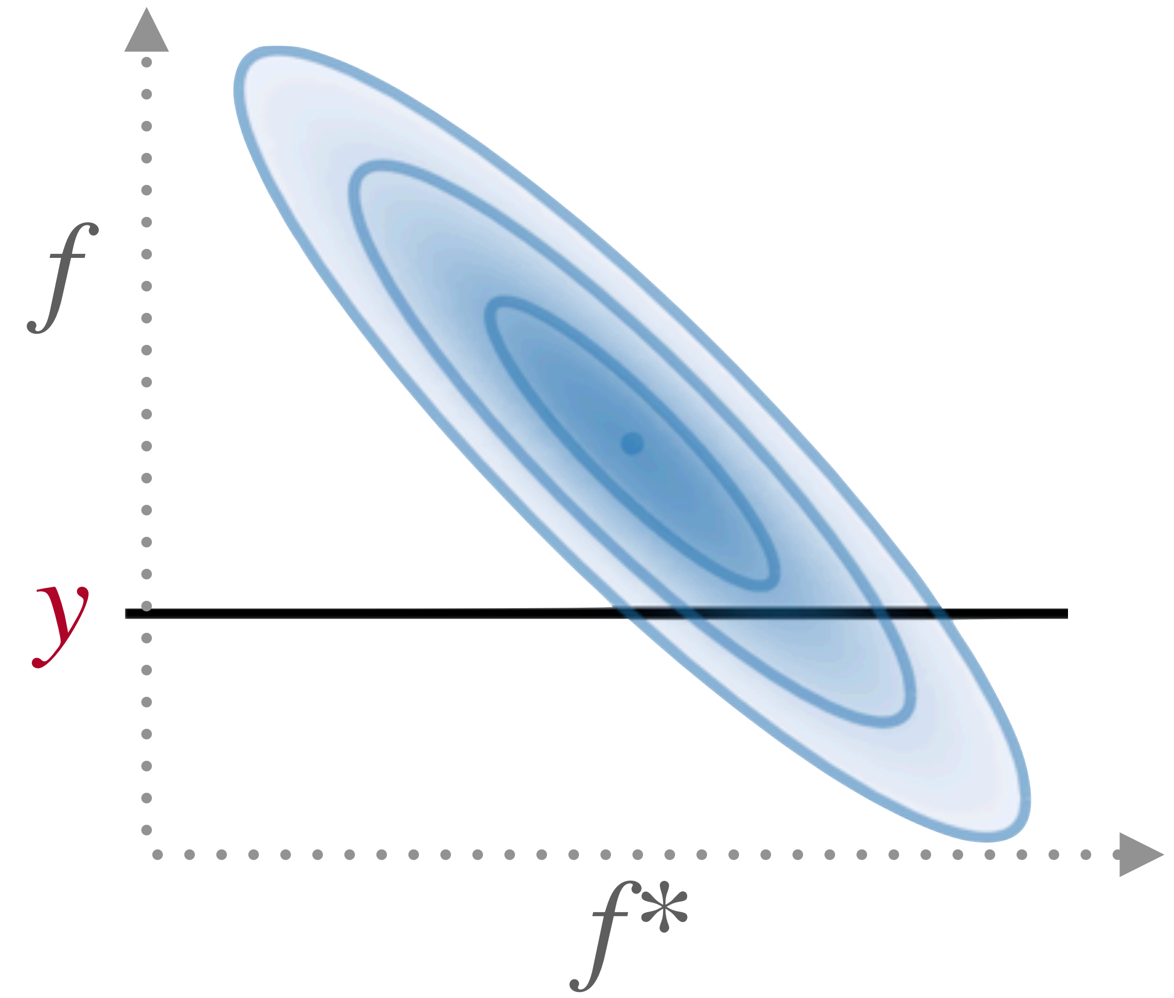
Can we estimate the uncertainties with SGD?

$$\Sigma_{f|y} = K_{**} - K_{*n}^{\top}(K_{nn} + \sigma^2 I)^{-1} K_{n*}$$

- No, because we can't solve one SGD optimisation per test datapoint…

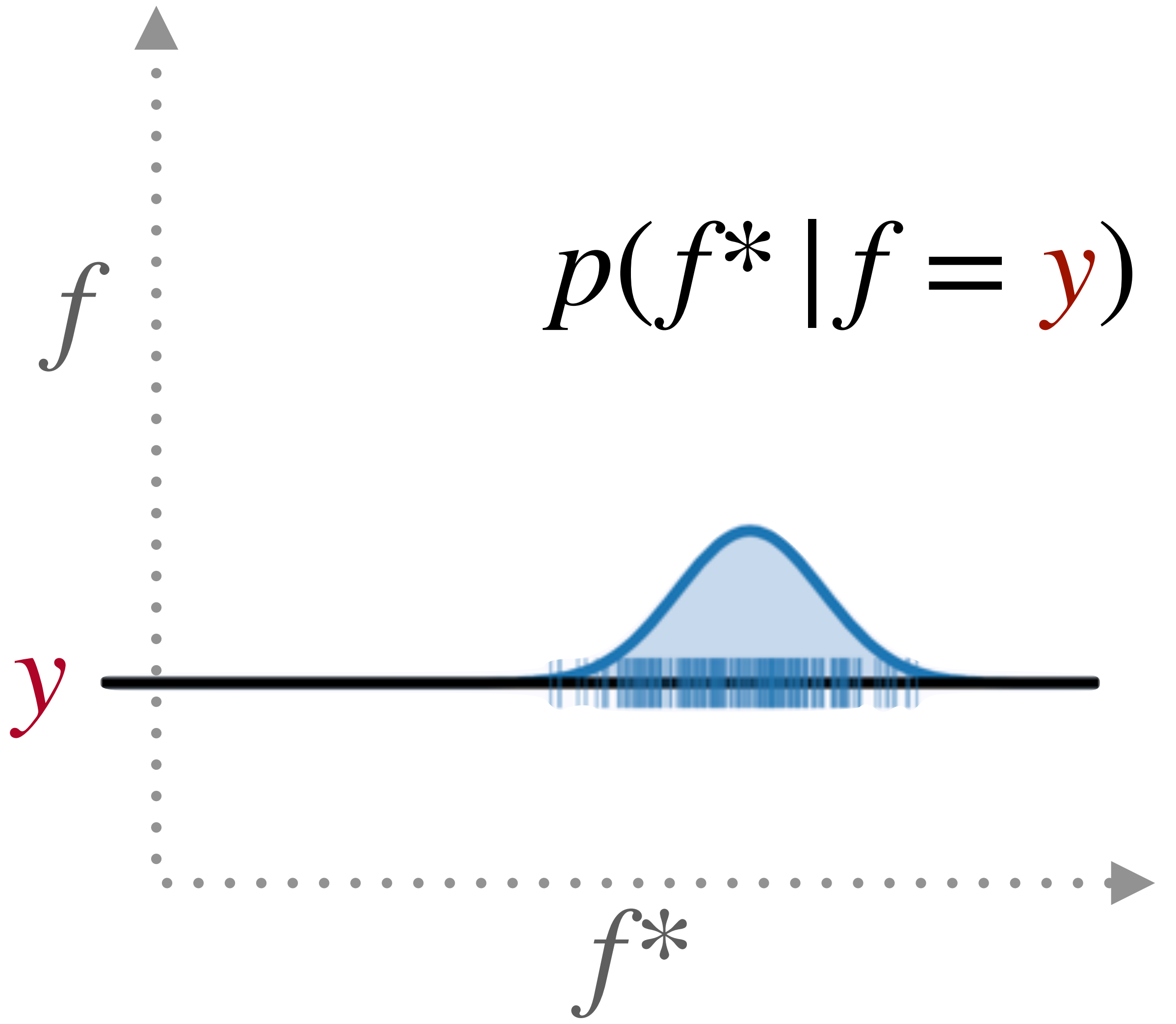
- Can we at least draw samples from the posterior $\mathcal{N}\left(\mu_{f|y}, \Sigma_{f|y}\right)$?
  - Option 1: **Cholesky decomposition**
    1. Decompose $\Sigma = LL^T$
    2. Draw sample from unit Gaussian, $\epsilon \sim \mathcal{N}(0,I)$
    3. Sample from posterior is $\mu_{f|y} + L\epsilon$
  - **Can we do better?**

A Path to More Efficient Sampling [1]



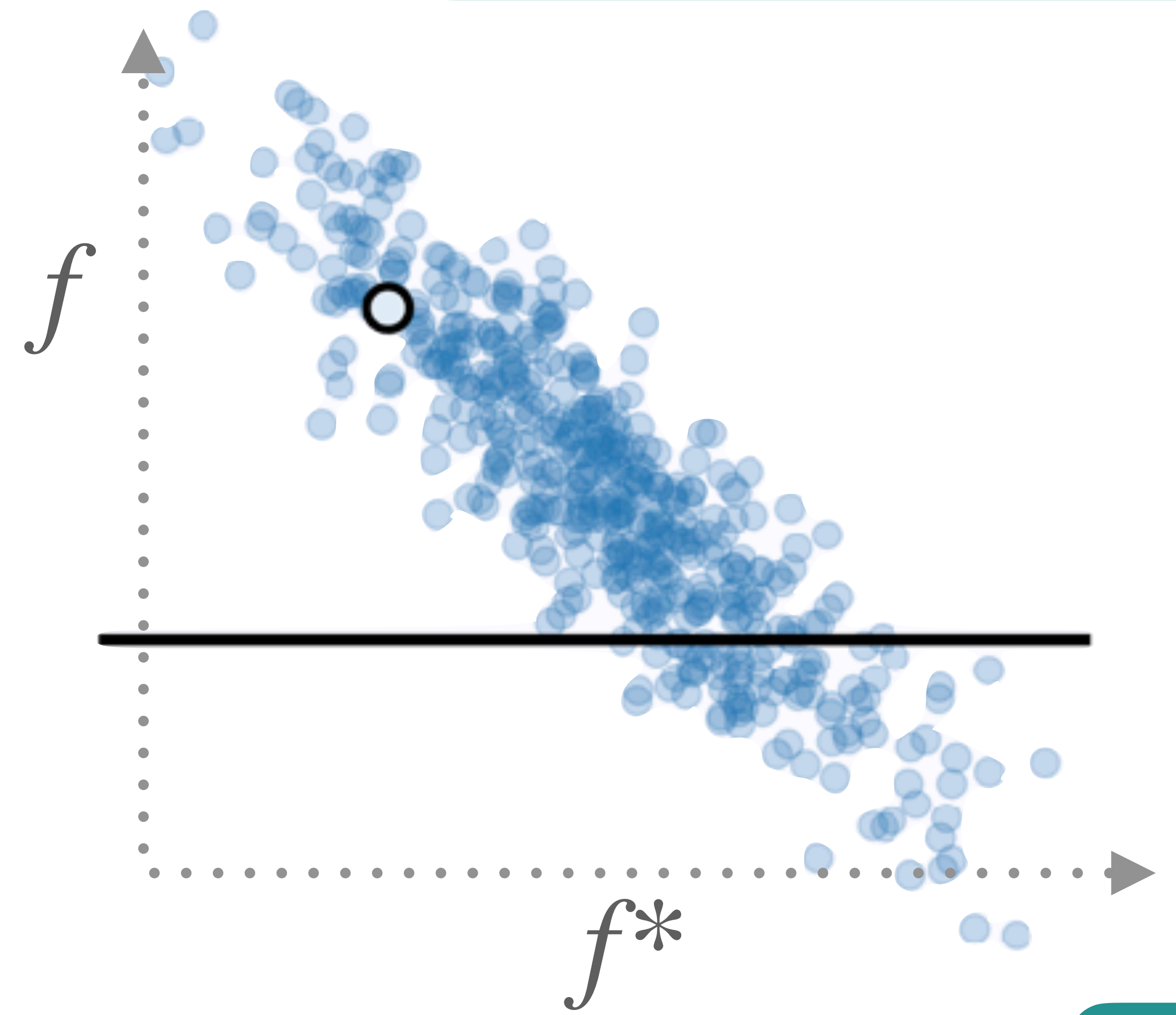
$\mathcal{O}(n^3) + \mathcal{O}(n^{*3})$

$$p(f^* \mid f = y) = \mathcal{N}(K_{*n}K_{nn}^{-1}y, K_{**} - K_{*n}K_{nn}^{-1}K_{n*})$$



Distributional View

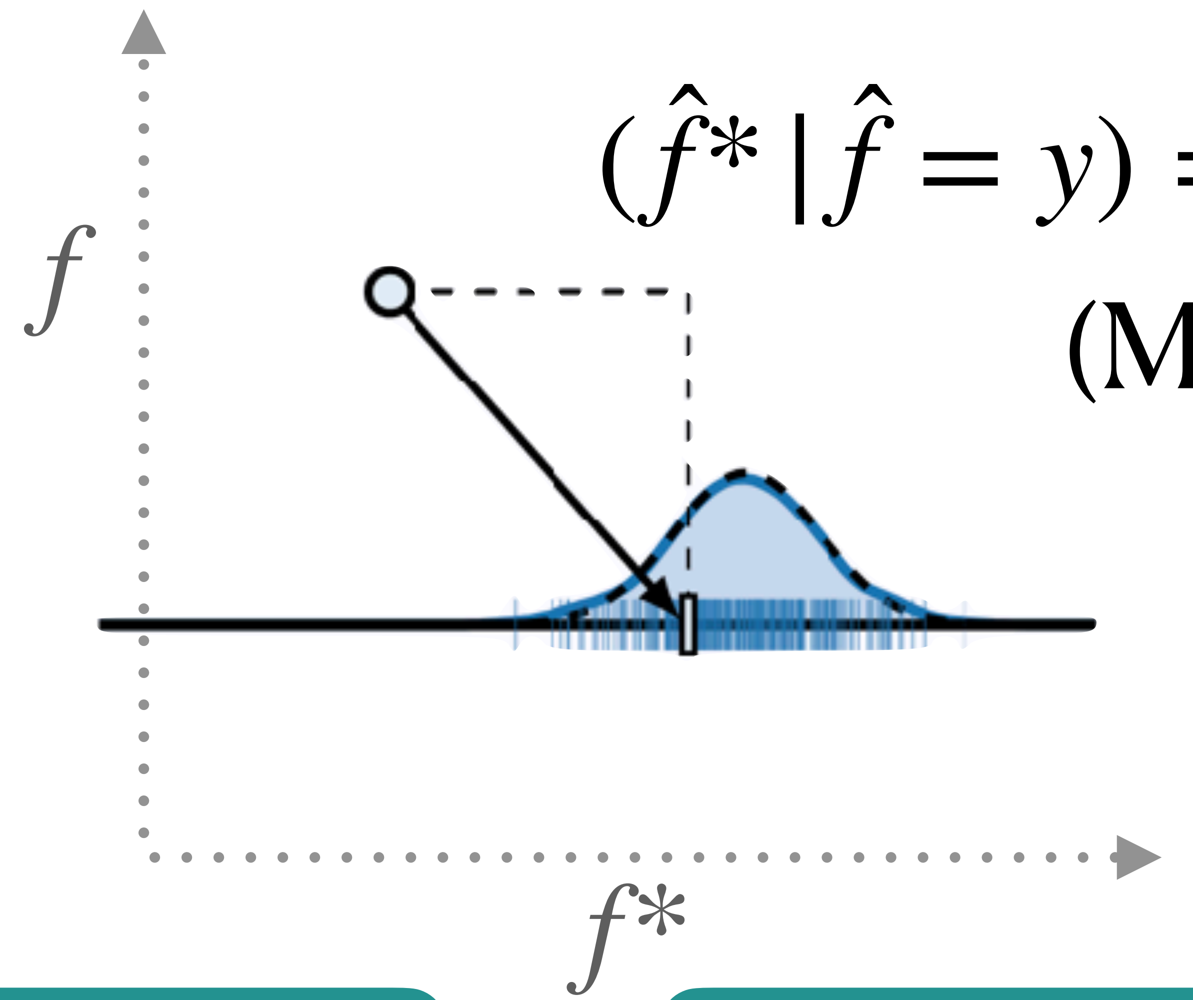
Joint Distribution → Conditional Distribution → Conditional Sample



$$(\hat{f}^* \mid \hat{f} = y) = \hat{f}^* + K_{*n}K_{nn}^{-1}(y + \epsilon - \hat{f})$$

(Matheron's Rule)

$\mathcal{O}(n^3)$



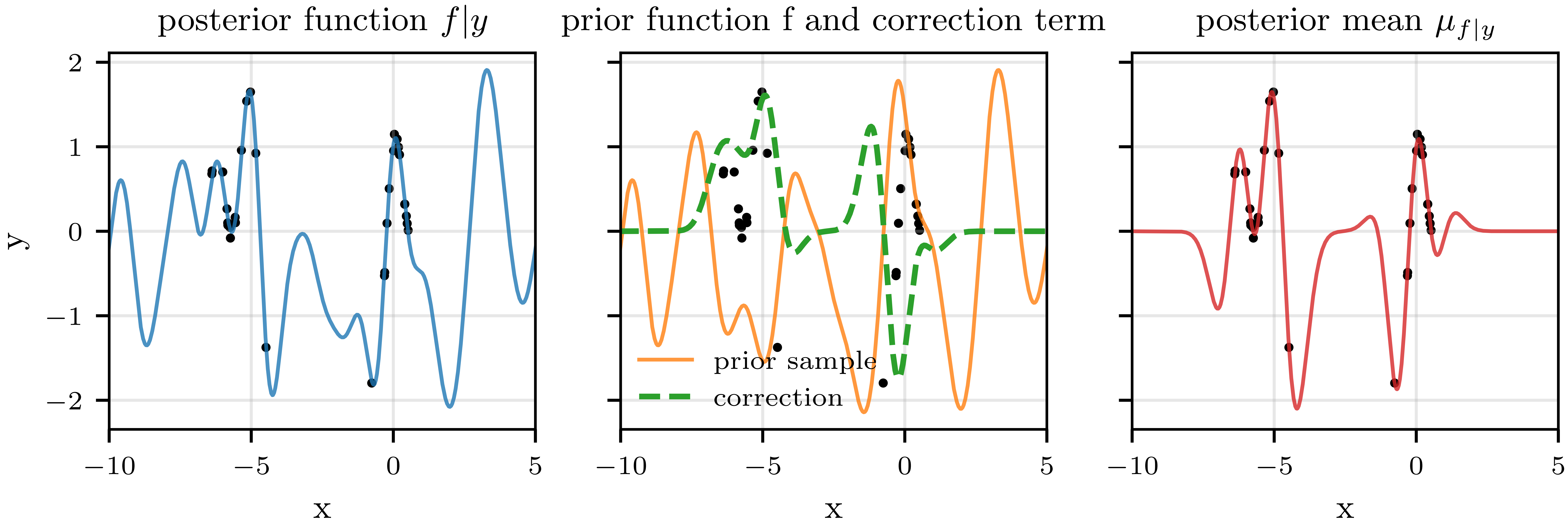
Individual Sample View

Joint Sample → Conditional Sample

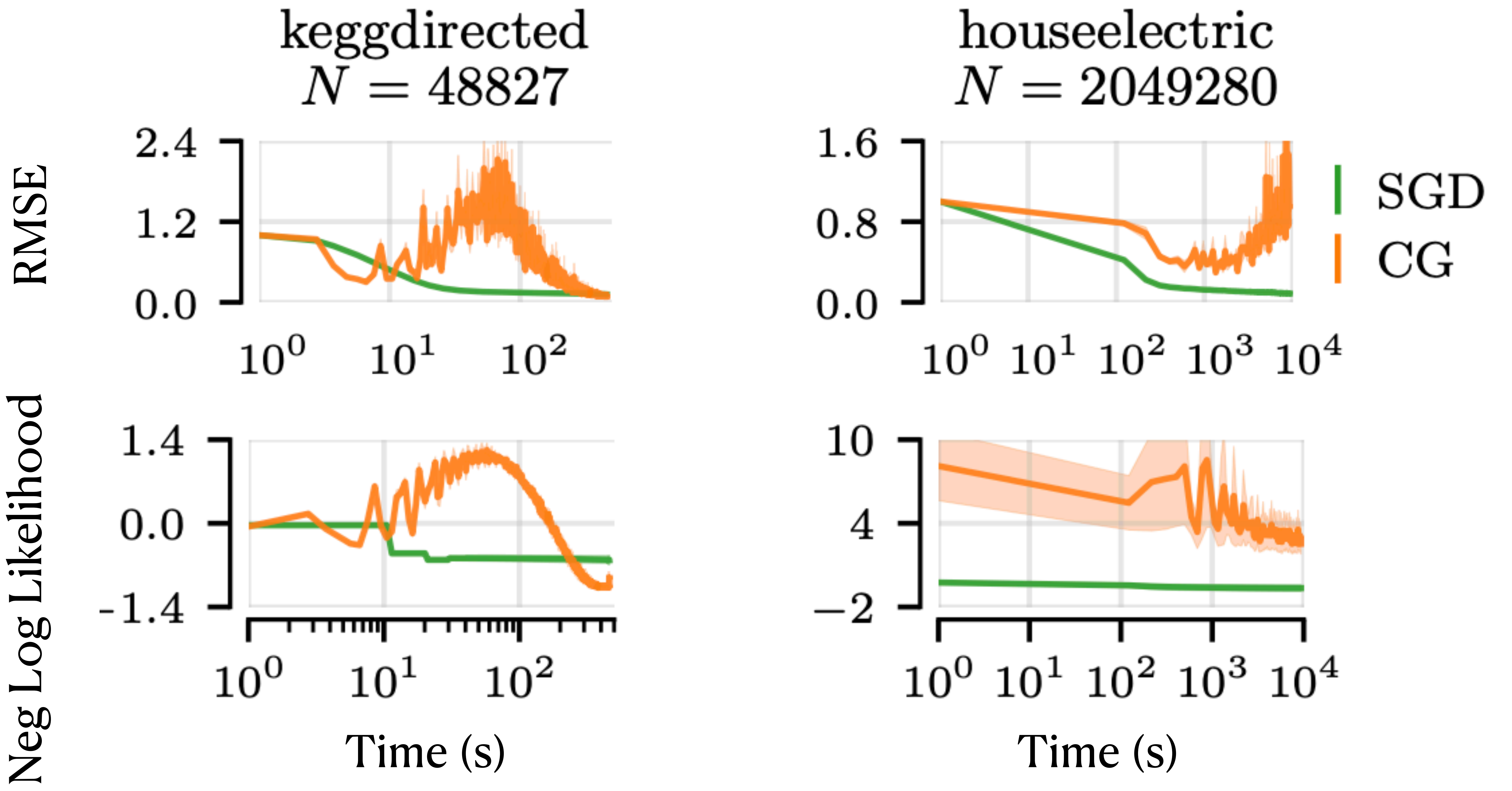
[1] Wilson, J.T., Borovitskiy, V., Terenin, A., Mostowsky, P. and Deisenroth, M.P., 2021. Pathwise conditioning of gaussian processes. *The Journal of Machine Learning Research*, 22(1), pp.4741-4787.

Sample from the Posterior

$$(f \mid \boldsymbol{y})(\cdot) = f(\cdot) \underbrace{+K_{(\cdot)n}\overbrace{\left(K_{nn}+\sigma^2 I\right)^{-1}\left(-f(x)+\epsilon\right)}^{v^*_{\text{sample}}}}_{\text{correction term}} \underbrace{+K_{(\cdot)n}\overbrace{\left(K_{nn}+\sigma^2 I\right)^{-1} y}^{v^*}}_{\text{mean } \mu_{f|\mathbf{y}}(\cdot)}$$



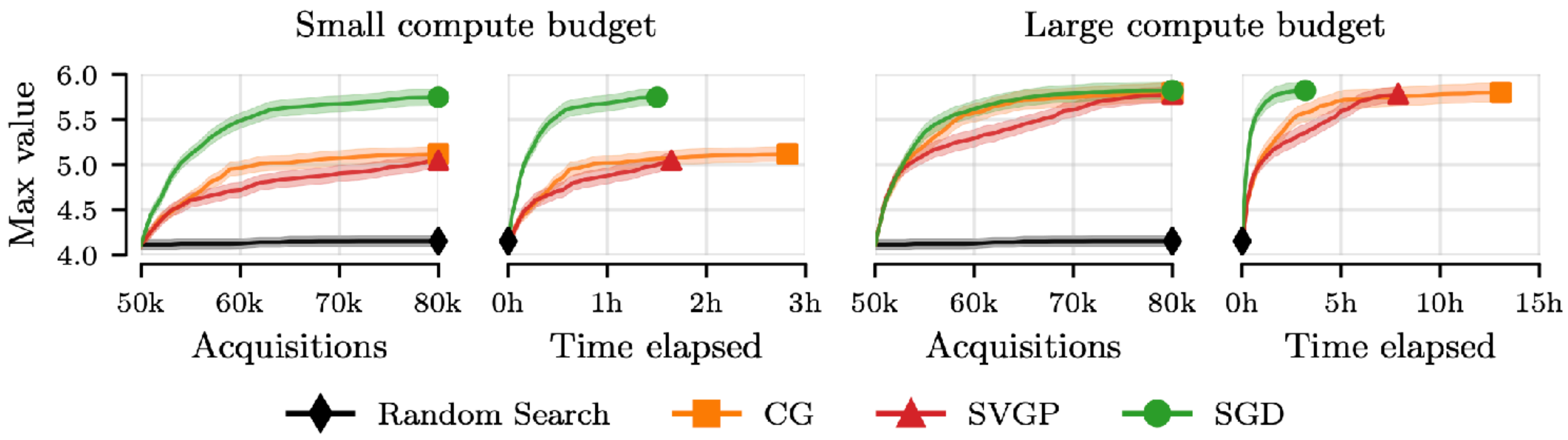
SGD scales much better in uncertainty estimates



[1] **Padhy, S.***, Lin, JA*, Antorán, J.,*, ... and Hernández-Lobato, J.M., 2022. Sampling from Gaussian Process Posteriors using Stochastic Gradient Descent. ***NeurIPS 2023***

Where can we apply this?

- Sequential Decision Making -> Bayesian Optimisation at a fixed compute budget



[1] **Padhy, S.***, Lin, JA*, Antorán, J.,*, ... and Hernández-Lobato, J.M., 2022. Sampling from Gaussian Process Posteriors using Stochastic Gradient Descent. ***NeurIPS 2023***

Uncertainty in Deep NNs: Linearised Laplace

- Given a neural network $f : \mathbb{R}^{d'} \to \mathbb{R}^m$ parameterised by $\theta \in \mathbb{R}^d$

- Augment $f(x)$ with uncertainty from the **linearised** model around MAP solution $\bar{w}$

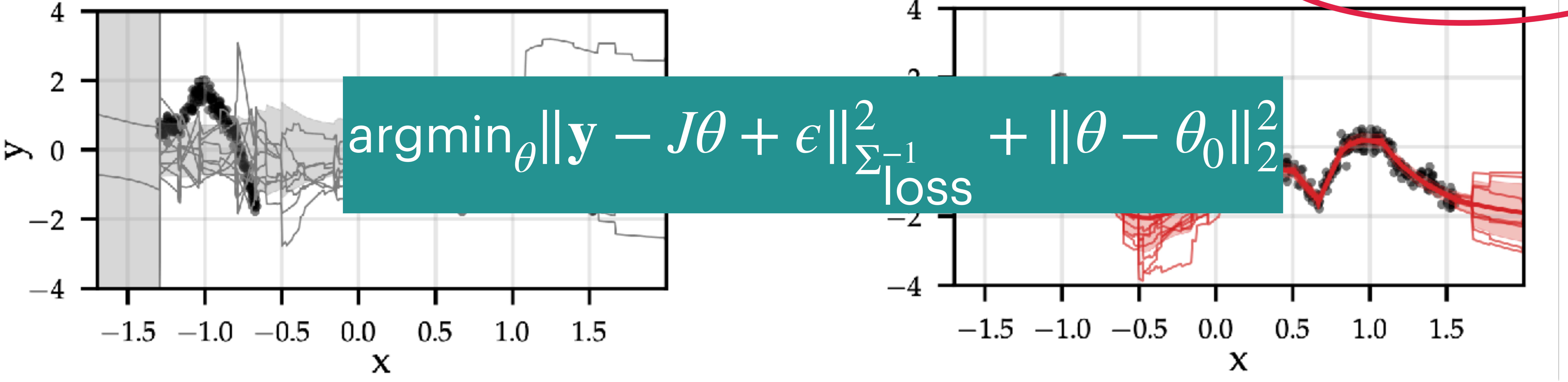
$$h(\theta, x) = f(\bar{w}, x) + \nabla_w f(\bar{w}, x)(\theta - \bar{w}), \qquad \theta \sim \mathcal{N}(0, A^{-1})$$

$$h(\theta, x) = \text{MAP solution} + J(x)(\theta - \bar{w})$$

- Turns out $h \sim \text{GP}(0,k)$ where $k(x_i, x_j) = J(x_i)^T A^{-1} J(x_j)$

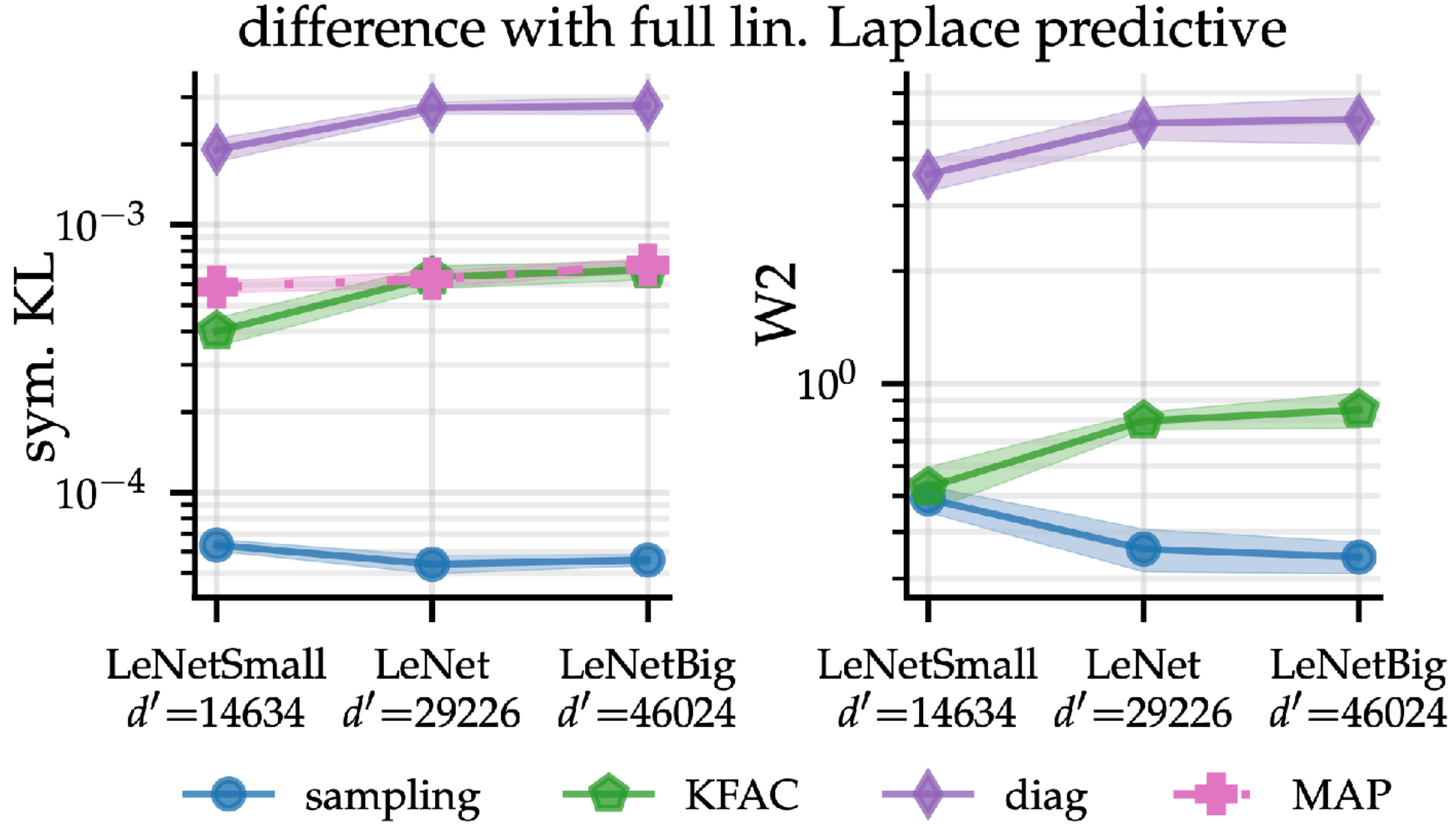
$\mathcal{O}(d^3) \to \mathcal{O}(d)$

Prior samples $h \sim GP(0, k)$ **Posterior samples** $h \sim GP(\mu_{h|y}, k_{h|y})$



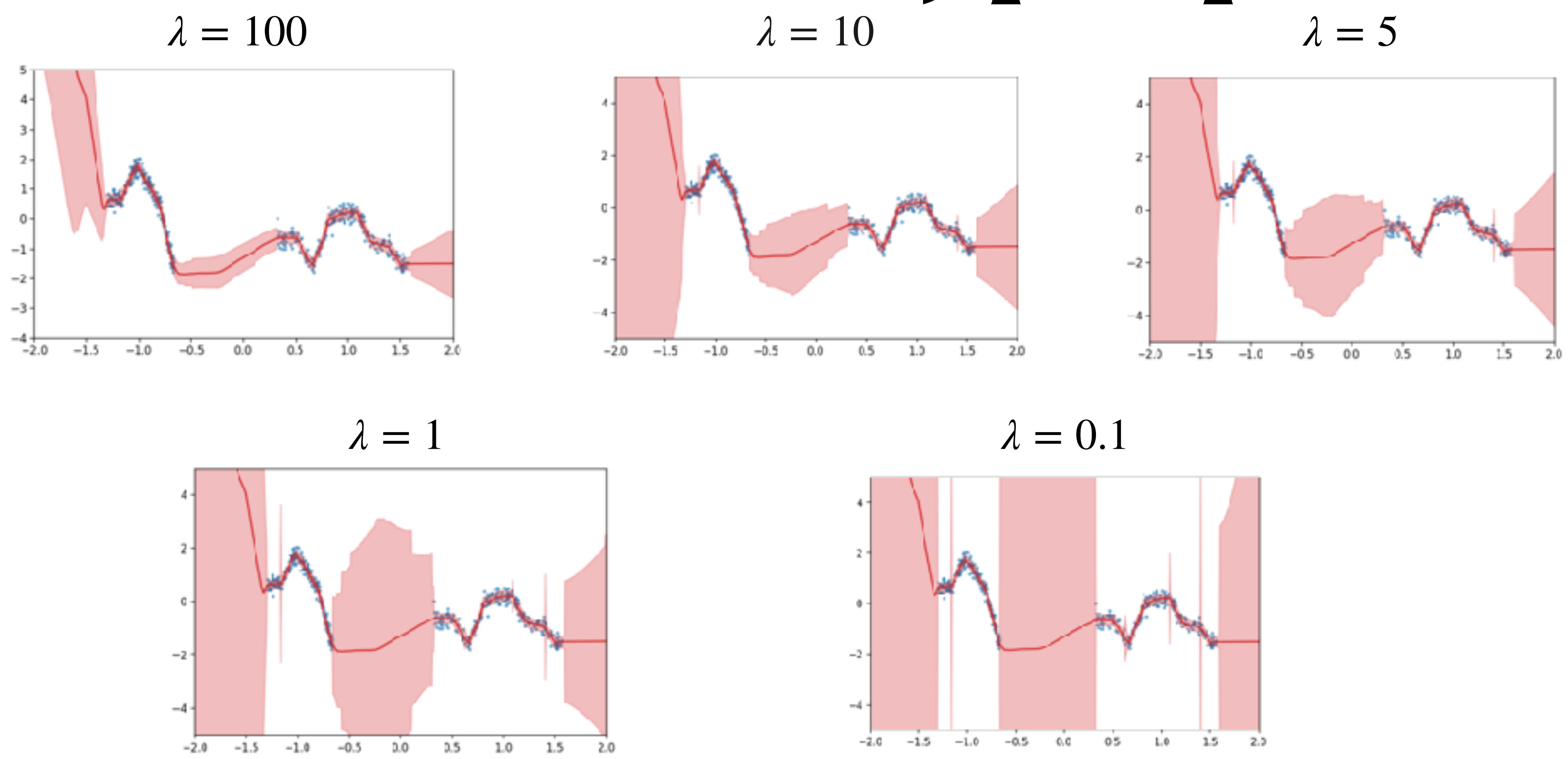
$$\text{argmin}_\theta \|\mathbf{y} - J\theta + \epsilon\|^2_{\Sigma^{-1}_{\text{loss}}} + \|\theta - \theta_0\|^2_2$$

How accurate are posterior samples?

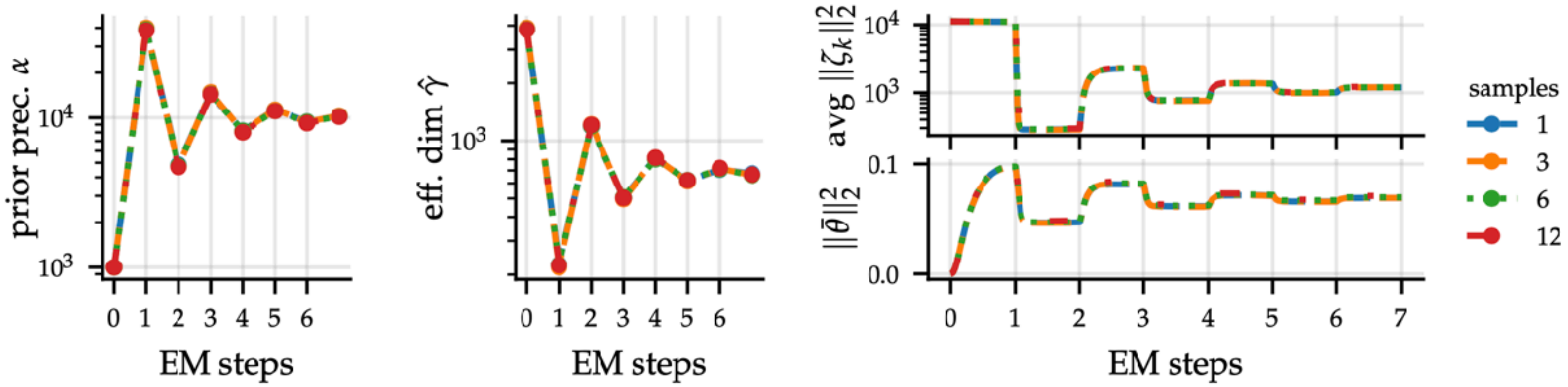


[1] **Padhy, S.***, Antorán, J.,*, Barbano, R., Nalisnick, E., ... and Hernández-Lobato, J.M., 2022. Sampling-based inference for large linear models, with application to linearised Laplace. ***ICLR 2023***

We can tune certain hyperparaemeters

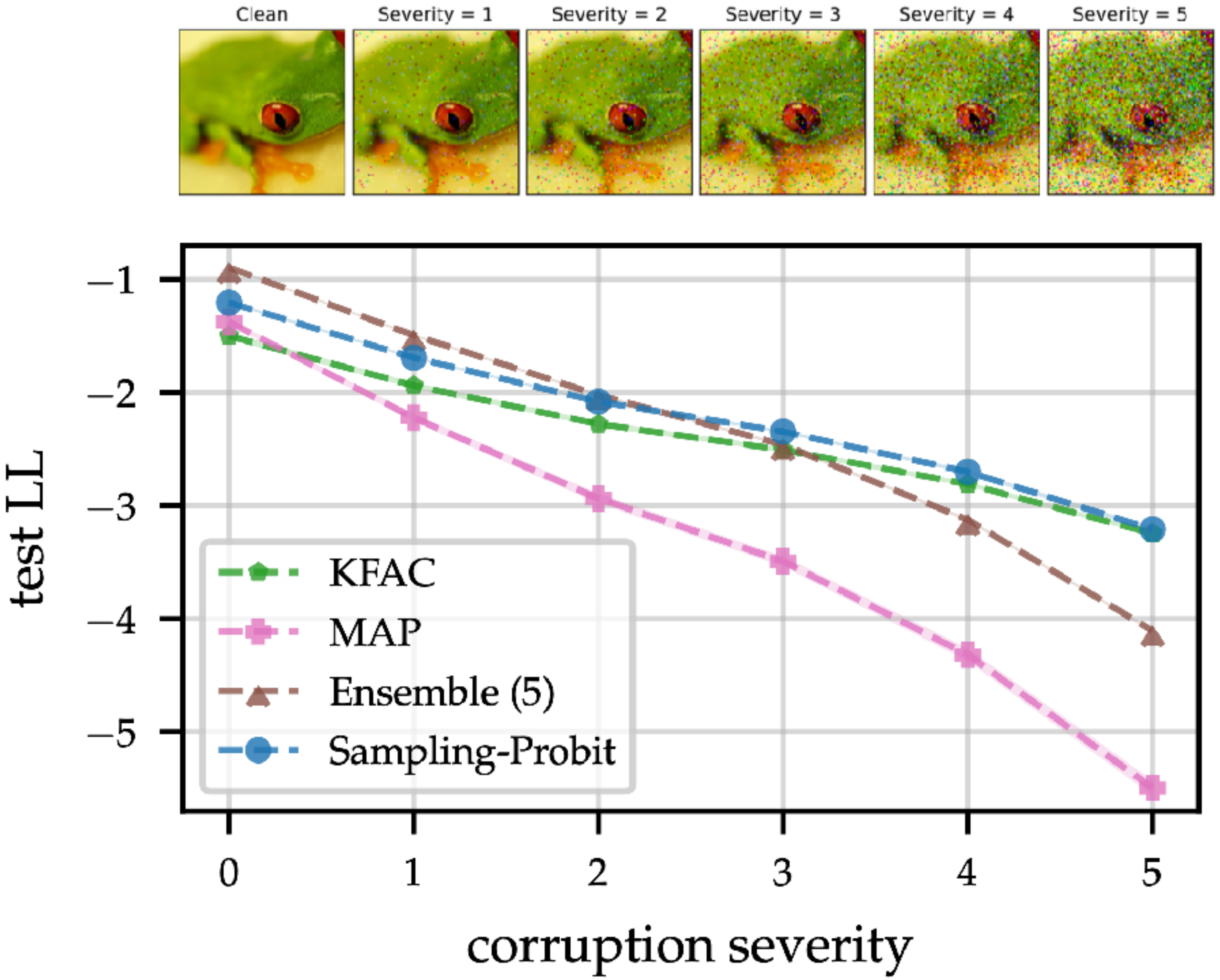


- We can estimate first-order updates for the prior precision $A = \lambda I$



Uncertainty on CIFAR100

ResNet-18 ($d = 11M$) on CIFAR-100 ($nm = 5M$)

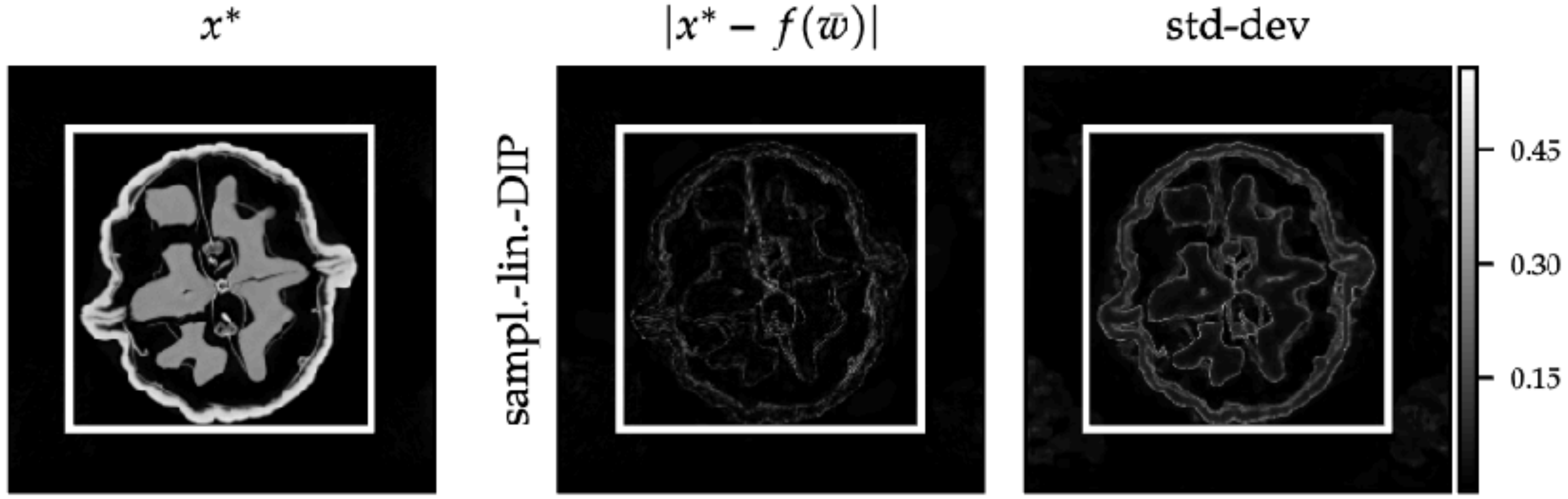


How far does this scale?

- ImageNet-scale [1] ($nm = 2B, d = 15M$)
- 2D Computed Tomography [1] ($m = 13k, d = 3M$)
- Large-scale/ill-conditioned regression ($n = 2M$)

	κ	MAP	Ensemble 5 NNs	KFAC	Sampling
marginal LL	1	-0.936	**-0.815**	-1.493	-0.917
joint LL	2	-9.347	-6.700	-6.286	**-5.611**
	3	-18.733	-13.268	-12.246	**-10.675**
	4	-28.093	-20.029	-20.493	**-16.154**
	5	-37.416	-26.938	-31.221	**-21.981**

	$m = 7680$			
	LL		wall-clock time (min.)	
Method	marginal	(10×10)	params optim.	prediction
MCDO-UNet	0.028	2.474	0	$3'$
lin.-UNet	2.214	2.601	$1260'$	$196'$
sampl.-lin.-UNet	**2.341**	**2.869**	$12'$	$14'$



	Dataset	HOUSEELEC
	N	2049280
RMSE	SGD	**0.09 ± 0.00**
	CG	0.87 ± 0.14
	SVGP	0.10 ± 0.02
RMSE †	SGD	**0.09 ± 0.00**
	CG	0.93 ± 0.19
	SVGP	—
Hours	SGD	2.69 ± 0.91
	CG	2.62 ± 0.01
	SVGP	**0.04 ± 0.00**

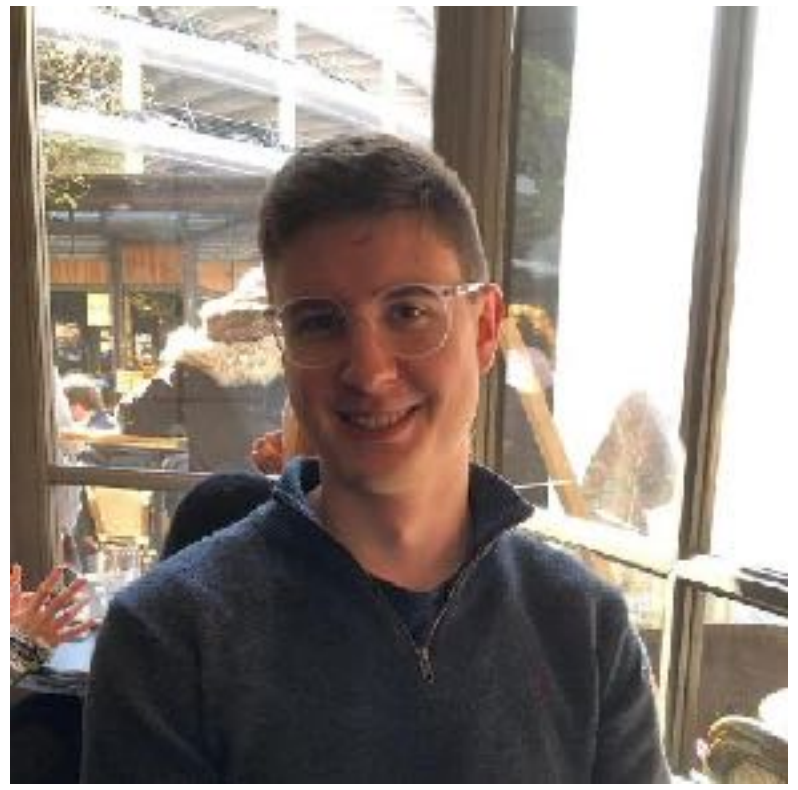
[1] **Padhy, S.***, Antorán, J.,*, Barbano, R., Nalisnick, E., ... and Hernández-Lobato, J.M., 2022. Sampling-based inference for large linear models, with application to linearised Laplace. ***ICLR 2023***

My Collaborators

Andy Lin

Javier Antoran

Riccardo Barbano



Dave Janz

Alex Terenin

Miguel Hernandez-Lobato

Appendix: Linear Models are GPs

$$y_i = \phi(x_i)\theta + \eta_i$$

$$y_i = GP(0, k(\,.\,,\,.\,)) + \eta_i$$

$$y_i \in \mathbb{R}^m$$

$$\theta \in \mathbb{R}^d$$

$$\phi(x_i) \in \mathbb{R}^{m \times d}$$

$$i \in \{1, \ldots, n\}$$

$$\theta \sim \mathcal{N}(0, A^{-1})$$

$$\eta_i \sim \mathcal{N}(0, B_i^{-1})$$

$\longrightarrow$

where $K_n n = \Phi^\top A^{-1} \Phi$

Appendix: Hparam Opt in Linear Models

- $\log \det H$ cannot be estimated from samples...

- MacKay proposed an alternative first order optimal update for $\mathscr{M}(\alpha)$(assume$A = \alpha I$)

$$\alpha = \frac{\mathrm{Tr}(H^{-1}\ \Phi^T \mathrm{B} \Phi)}{\|\bar{\theta}\|^2} = \frac{\mathrm{Tr}(H^{-1}\ M)}{\|\bar{\theta}\|^2}$$

- This *can be* estimated using only samples from the posterior

$\mathcal{O}(kdnm)$

$$\mathrm{Tr}\left\{H^{-1}M\right\} = \mathrm{Tr}\left\{H^{\frac{-1}{2}}MH^{\frac{-1}{2}}\right\} = \mathbb{E}\left[z_1^T M z_1\right] \approx \frac{1}{k}\sum_{j=1}^{k} z_j^T \Phi^T\ \mathrm{B}\Phi z_j$$