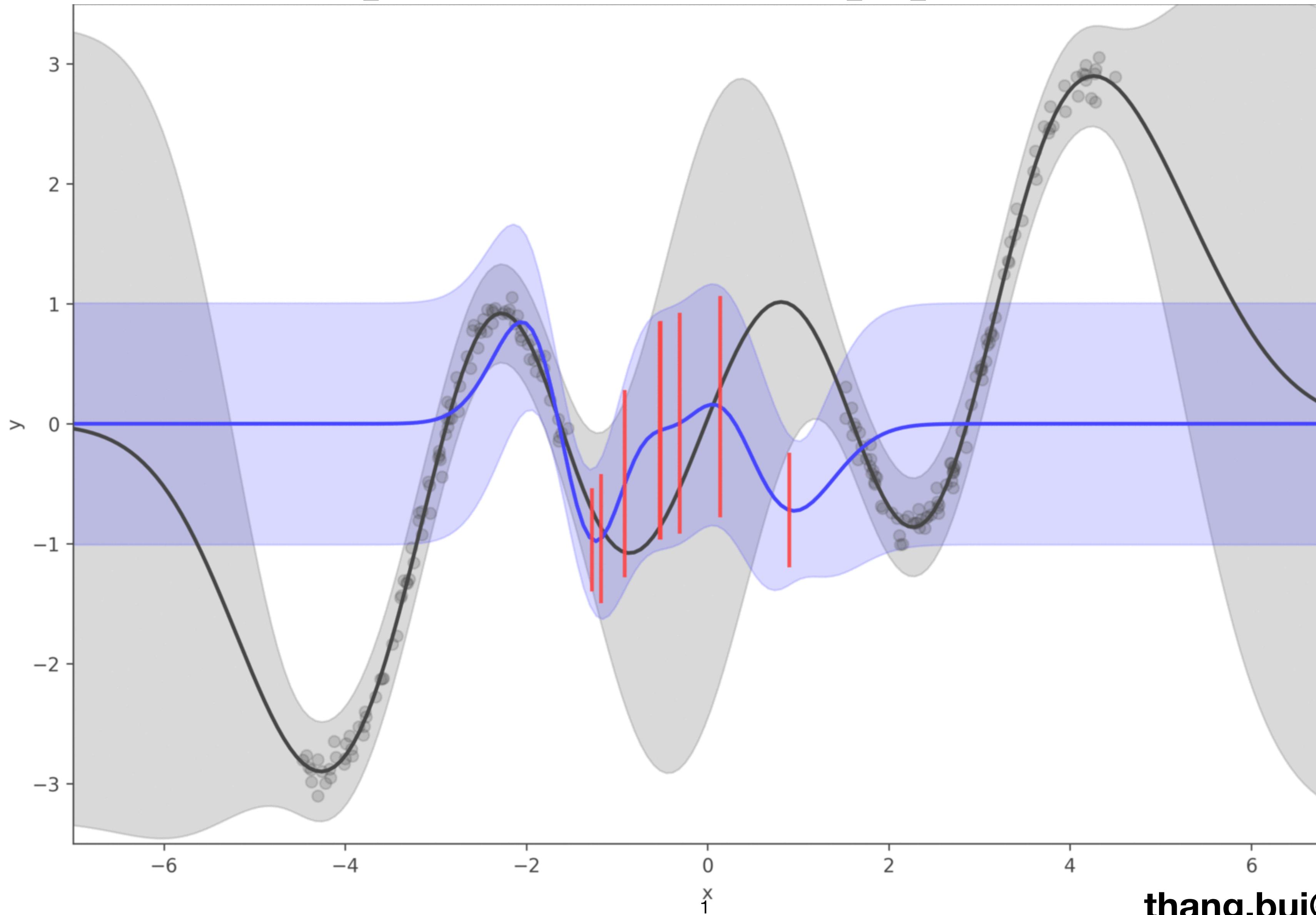
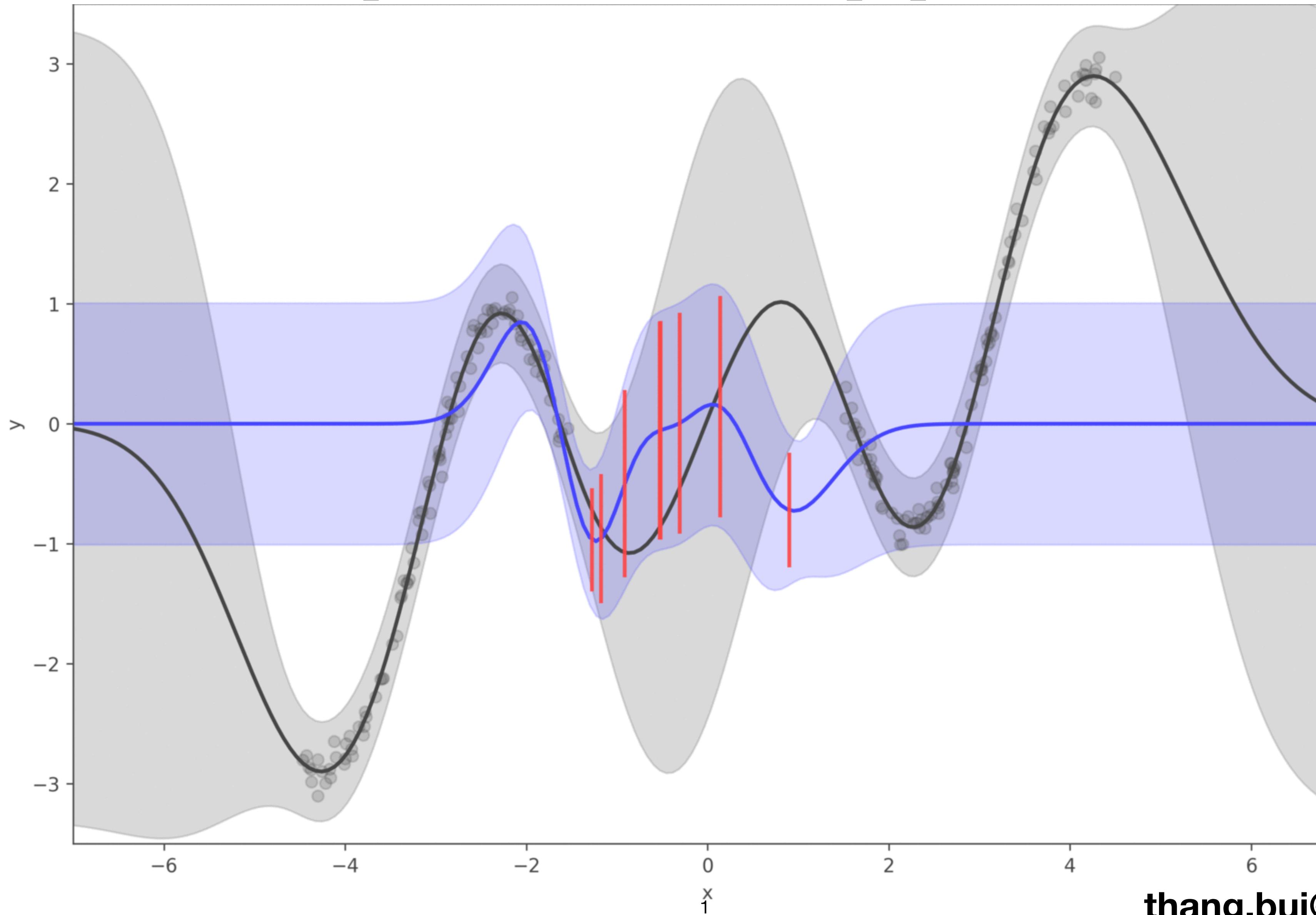


Gaussian process approximations



Gaussian process approximations



Big picture

Week 7 - lecture 1

Exact for GP regression

Week 7 - lecture 2

Laplace

Week 11,12

Variational inference

Expectation propagation

Week 9

Sampling

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

Exact or *full* approximations

Sparse approximations

Variational inference

(Power) Expectation propagation

$$\mathcal{O}(NM^2) \text{ or } \mathcal{O}(M^3)$$

Collapsed for GP regression or batch updates

Mini-batch stochastic updates

This lecture

$$\mathcal{O}(NM^2)$$

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

Gaussian process refresher

A *Gaussian process* is a collection of random variables, any finite number of which have a joint *Gaussian distribution*

Gaussian process refresher

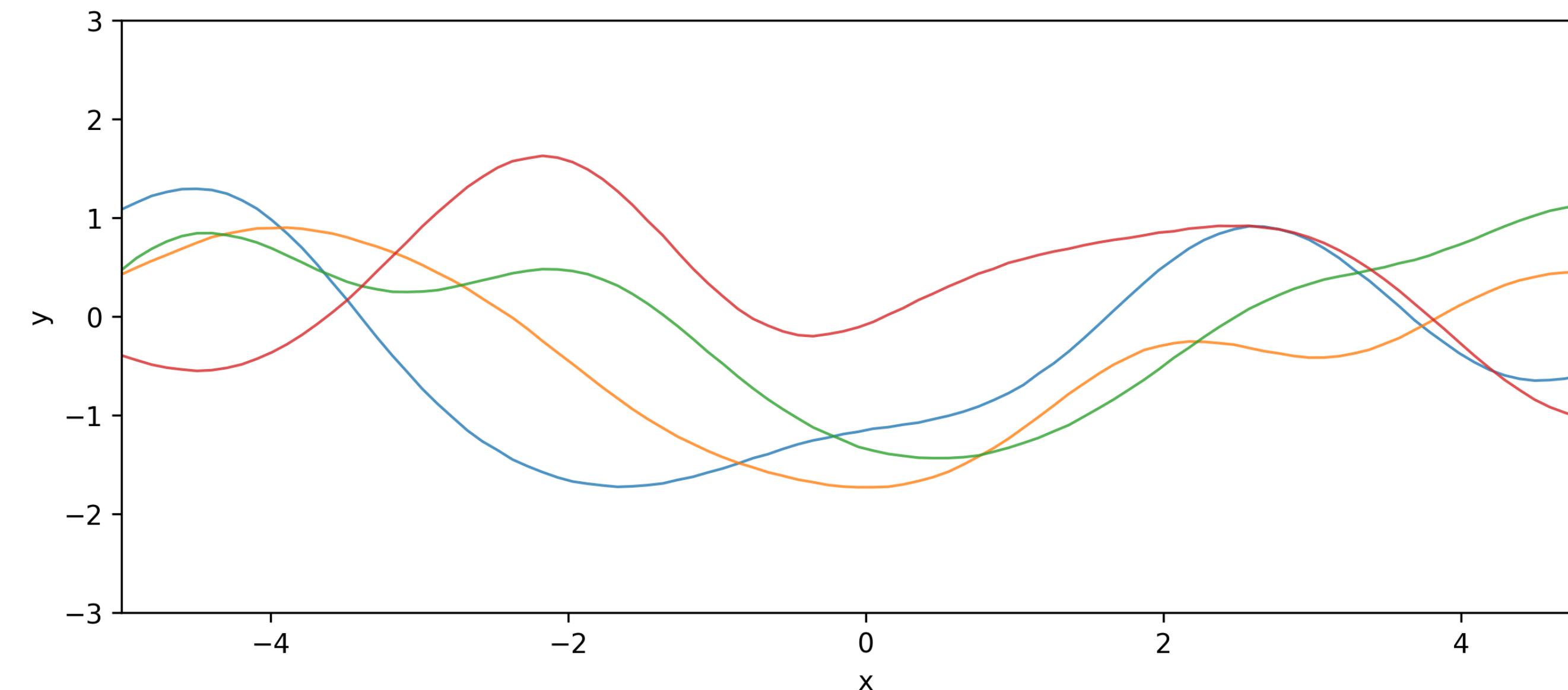
A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution

Multivariate Gaussian distribution

- $\mathbf{f} \sim \mathbf{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Mean $\boldsymbol{\mu}$
- Covariance matrix $\boldsymbol{\Sigma}$

Gaussian process

- $f(x) \sim \text{GP}(m(x), k(x, x'))$
- Mean function $m(x)$
- Covariance func. $k(x, x')$



Gaussian process refresher

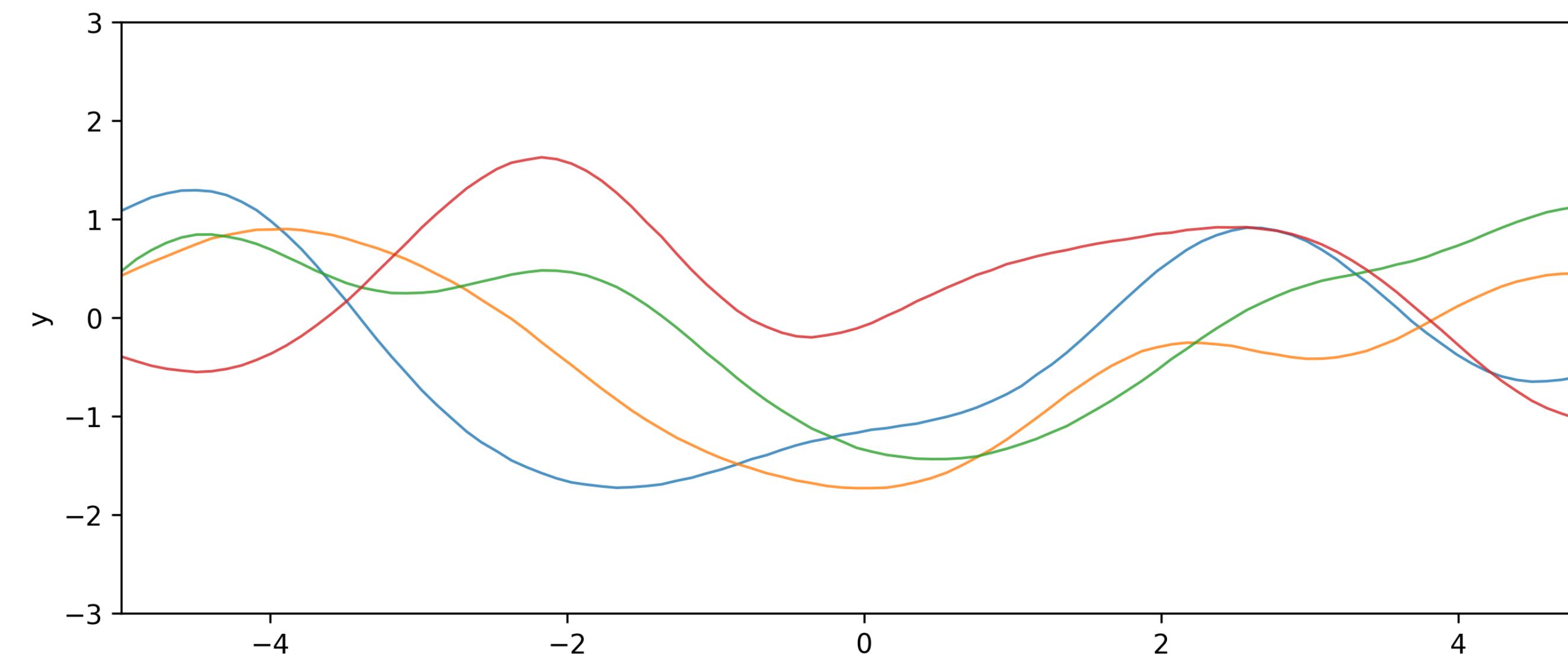
A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution

Multivariate Gaussian distribution

- $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
- Mean $\boldsymbol{\mu}$
- Covariance matrix $\boldsymbol{\Sigma}$

Gaussian process

- $f(x) \sim \text{GP}(m(x), k(x, x'))$
- Mean function $m(x)$
- Covariance func. $k(x, x')$



How do we go from a Gaussian distribution to a Gaussian process?

Gaussian process regression

Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim \mathcal{N}(0, 1)$$

Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim \mathcal{N}(0, 1)$$

Joint distribution $p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N}\left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix}\right)$

Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim \mathcal{N}(0, 1)$$

Joint distribution

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N} \left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right)$$

↑
observed

Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim N(0, 1)$$

Joint distribution

$$p(\mathbf{y}_1, \mathbf{y}_2) = N \left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right)$$



Conditioning on the observed variable

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)} = N \left(\mathbf{y}_1; \mathbf{m}_1 + \mathbf{K}_{12} \mathbf{K}_{22}^{-1} (\mathbf{y}_2 - \mathbf{m}_2), \mathbf{K}_{11} - \mathbf{K}_{12} \mathbf{K}_{22}^{-1} \mathbf{K}_{21} \right)$$

Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim N(0, 1)$$

Joint distribution

$$p(\mathbf{y}_1, \mathbf{y}_2) = N \left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right)$$

↑
observed

Conditioning on the observed variable

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)} = N \left(\mathbf{y}_1; \mathbf{m}_1 + \mathbf{K}_{12}\mathbf{K}_{22}^{-1}(\mathbf{y}_2 - \mathbf{m}_2), \mathbf{K}_{11} - \mathbf{K}_{12}\mathbf{K}_{22}^{-1}\mathbf{K}_{21} \right)$$

↑
prior ↑
 uncertainty reduction

Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim N(0, 1)$$

Joint distribution

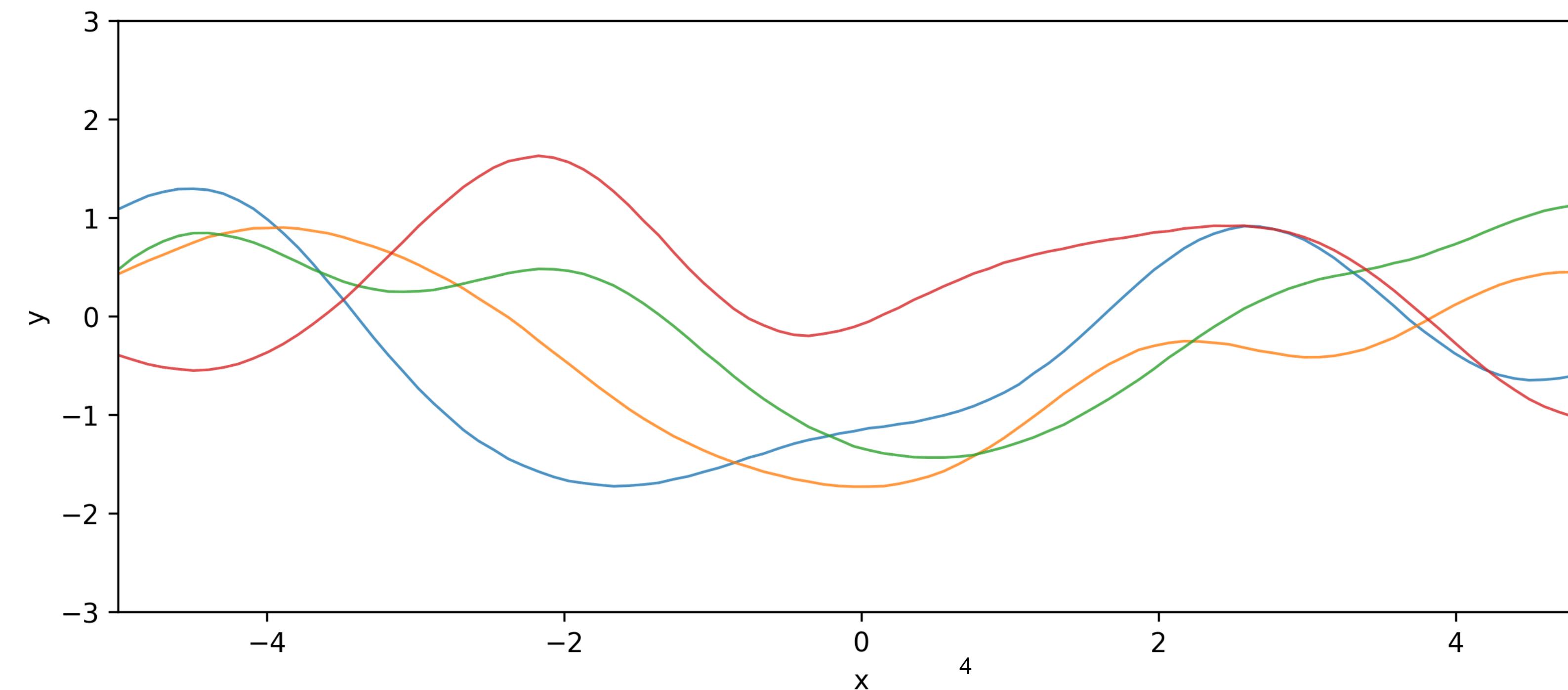
$$p(\mathbf{y}_1, \mathbf{y}_2) = N \left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right)$$

↑
observed

Conditioning on the observed variable

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)} = N \left(\mathbf{y}_1; \mathbf{m}_1 + \mathbf{K}_{12}\mathbf{K}_{22}^{-1}(\mathbf{y}_2 - \mathbf{m}_2), \mathbf{K}_{11} - \mathbf{K}_{12}\mathbf{K}_{22}^{-1}\mathbf{K}_{21} \right)$$

↑
prior ↑
uncertainty reduction



Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim N(0, 1)$$

Joint distribution

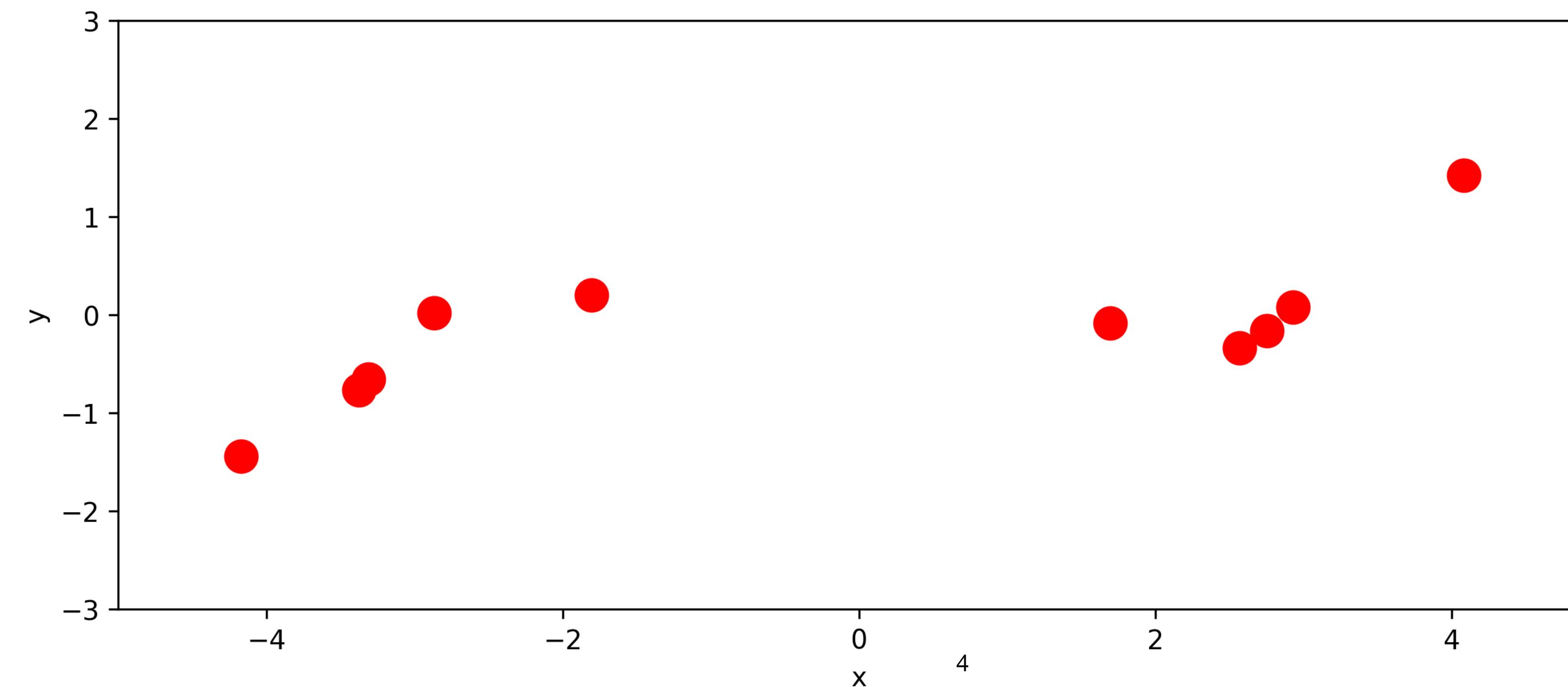
$$p(\mathbf{y}_1, \mathbf{y}_2) = N \left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right)$$

observed

Conditioning on the observed variable

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)} = N \left(\mathbf{y}_1; \mathbf{m}_1 + \mathbf{K}_{12} \mathbf{K}_{22}^{-1} (\mathbf{y}_2 - \mathbf{m}_2), \mathbf{K}_{11} - \mathbf{K}_{12} \mathbf{K}_{22}^{-1} \mathbf{K}_{21} \right)$$

prior
uncertainty reduction



Gaussian process regression

$$y = f(x) + \sigma_y \epsilon$$

$$f \sim \text{GP}(m(x), k(x, x'))$$

$$\epsilon \sim N(0, 1)$$

Joint distribution

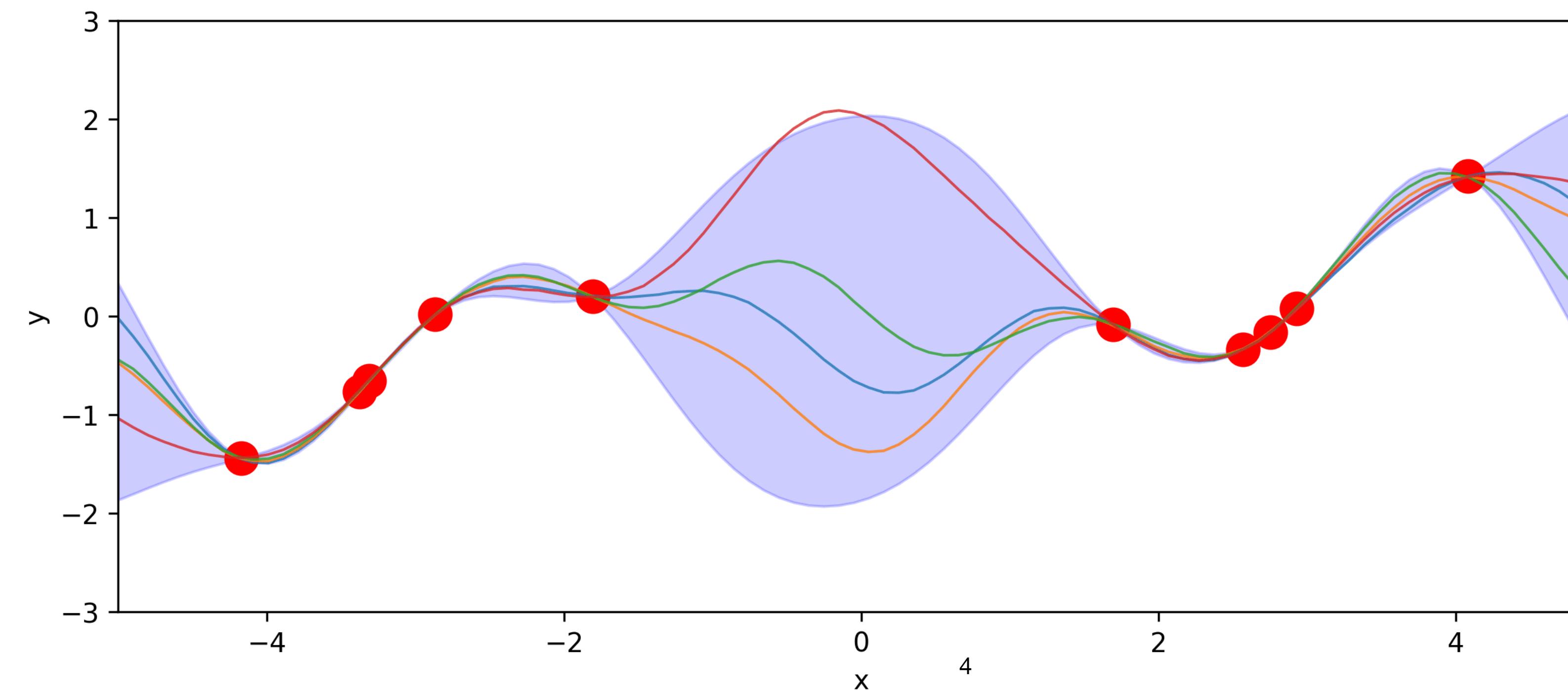
$$p(\mathbf{y}_1, \mathbf{y}_2) = N \left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}; \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \right)$$

observed

Conditioning on the observed variable

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)} = N \left(\mathbf{y}_1; \mathbf{m}_1 + \mathbf{K}_{12}\mathbf{K}_{22}^{-1}(\mathbf{y}_2 - \mathbf{m}_2), \mathbf{K}_{11} - \mathbf{K}_{12}\mathbf{K}_{22}^{-1}\mathbf{K}_{21} \right)$$

prior
uncertainty reduction



Algorithm

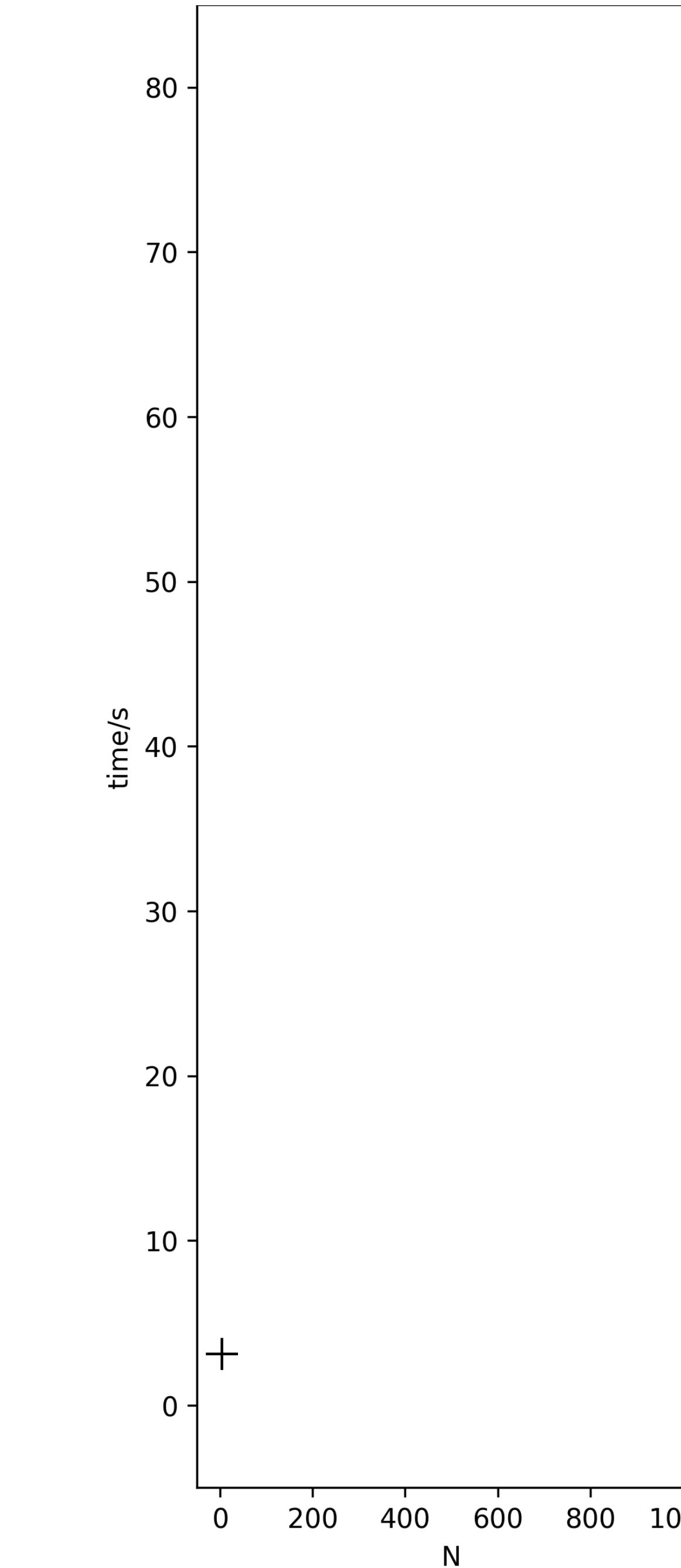
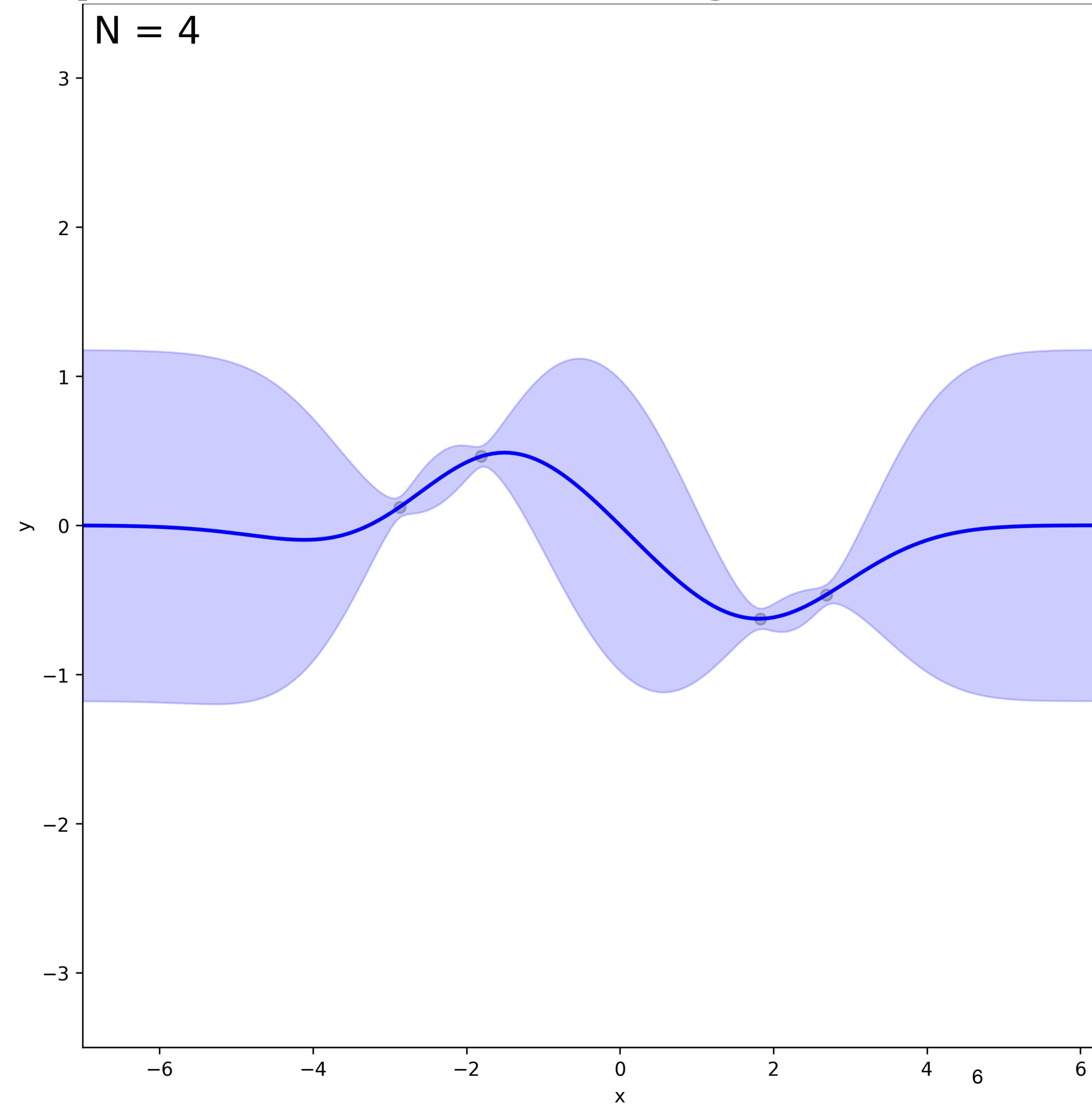
$N^3/3$ operations



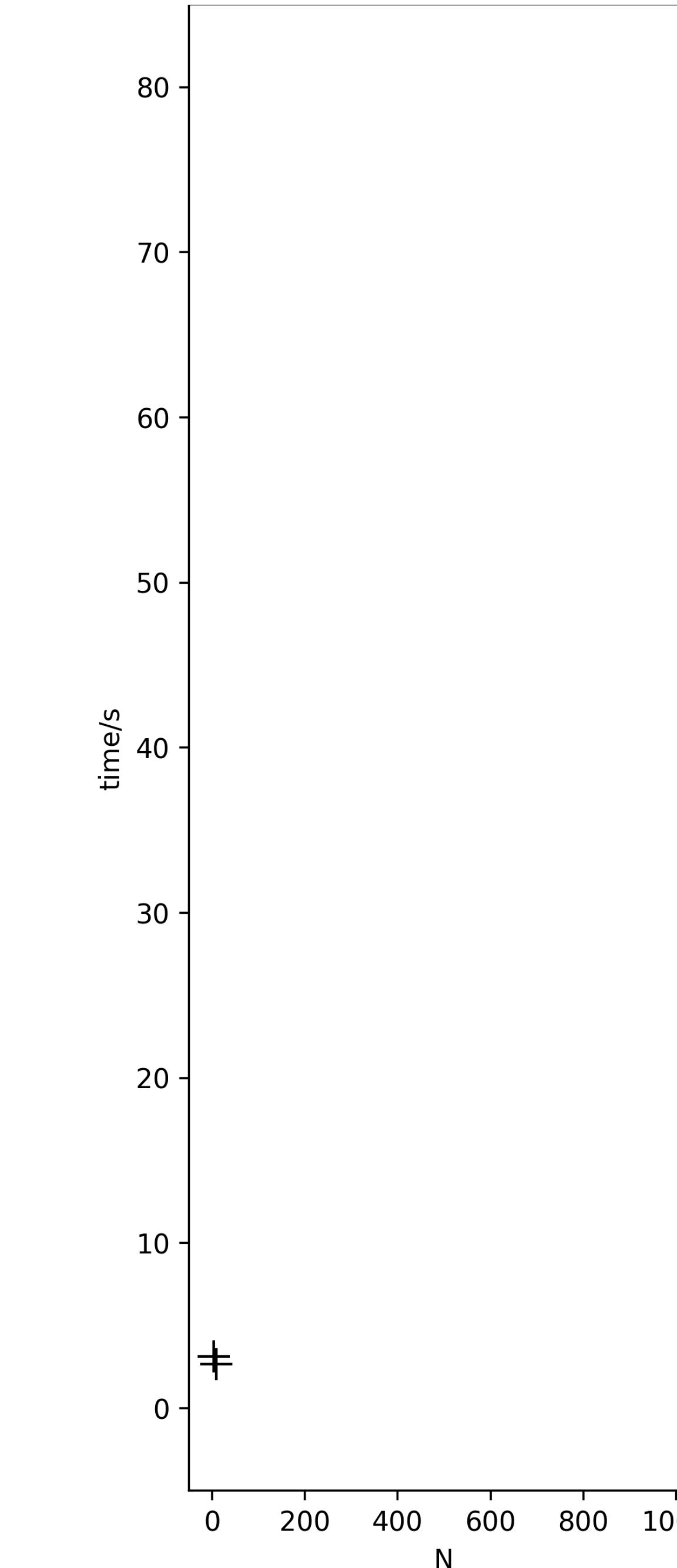
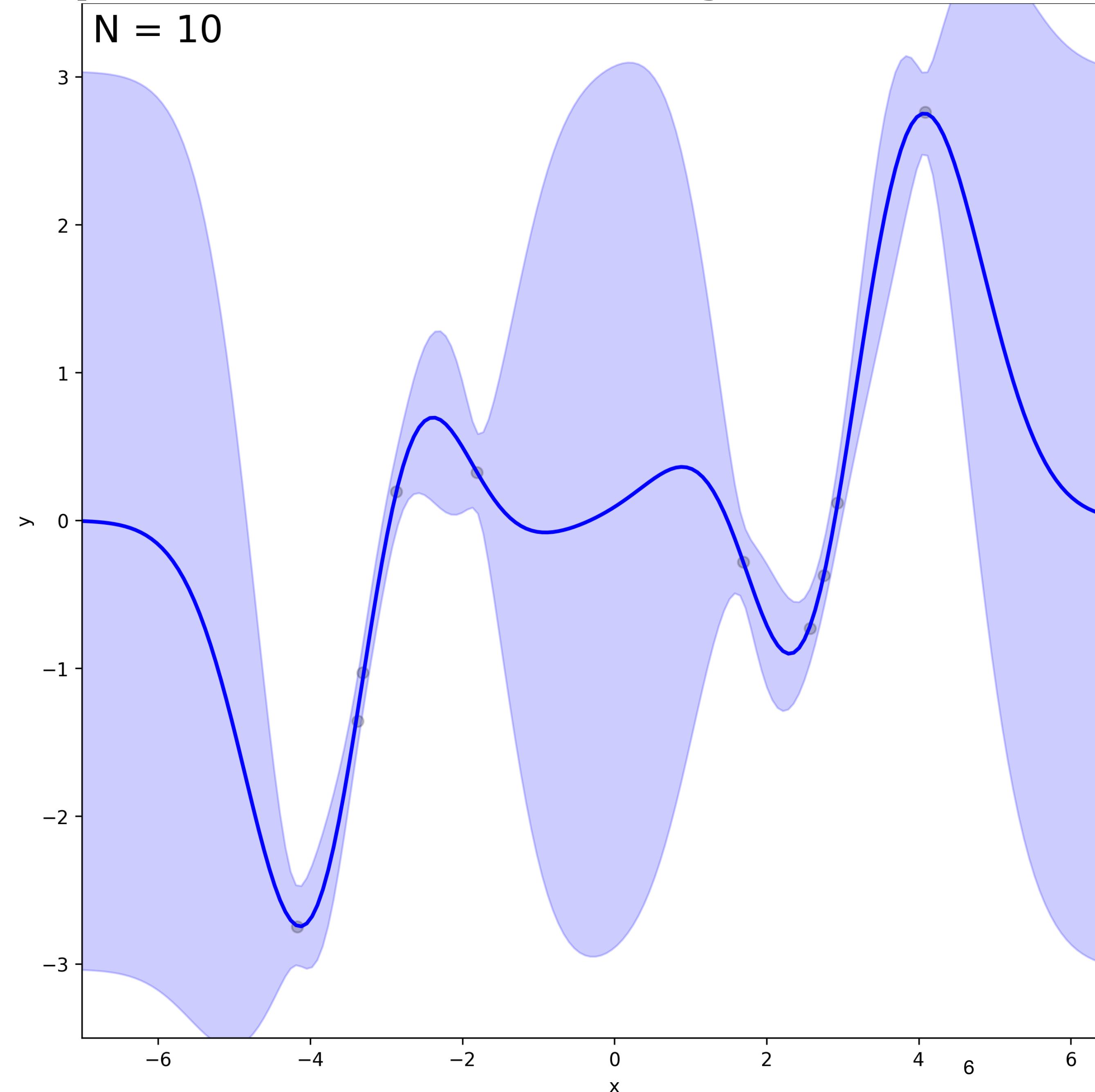
- $L \equiv$ Cholesky $k(\mathbf{X}, \mathbf{X})$ $k(\mathbf{X}, \mathbf{X}) = \mathbf{L} \cdot \mathbf{L}^T$, and \mathbf{L} lower-triangular
- $\alpha \equiv \mathbf{L}^T \backslash (\mathbf{L} \backslash \mathbf{t})$ $\mathbf{A}\mathbf{x} = \mathbf{b} \Rightarrow \mathbf{x} = \mathbf{A} \backslash \mathbf{b}$
- $m(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{X}) \cdot \alpha$
- $\mathbf{v} = \mathbf{L} \backslash k(\mathbf{X}, \mathbf{x}^*)$
- $\sigma(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{v}^T \mathbf{v}$

$$p(y^* | \mathbf{x}^*, \mathbf{X}, \mathbf{t}, \beta^{-1}) = \mathcal{N}(m(x^*), \sigma^2(x^*))$$
$$m(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{X}) k(\mathbf{X}, \mathbf{X})^{-1} \mathbf{t} \quad \mathcal{O}(N^3)$$
$$\sigma(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - k(\mathbf{x}^*, \mathbf{X}) k(\mathbf{X}, \mathbf{X})^{-1} k(\mathbf{X}, \mathbf{x}^*)$$

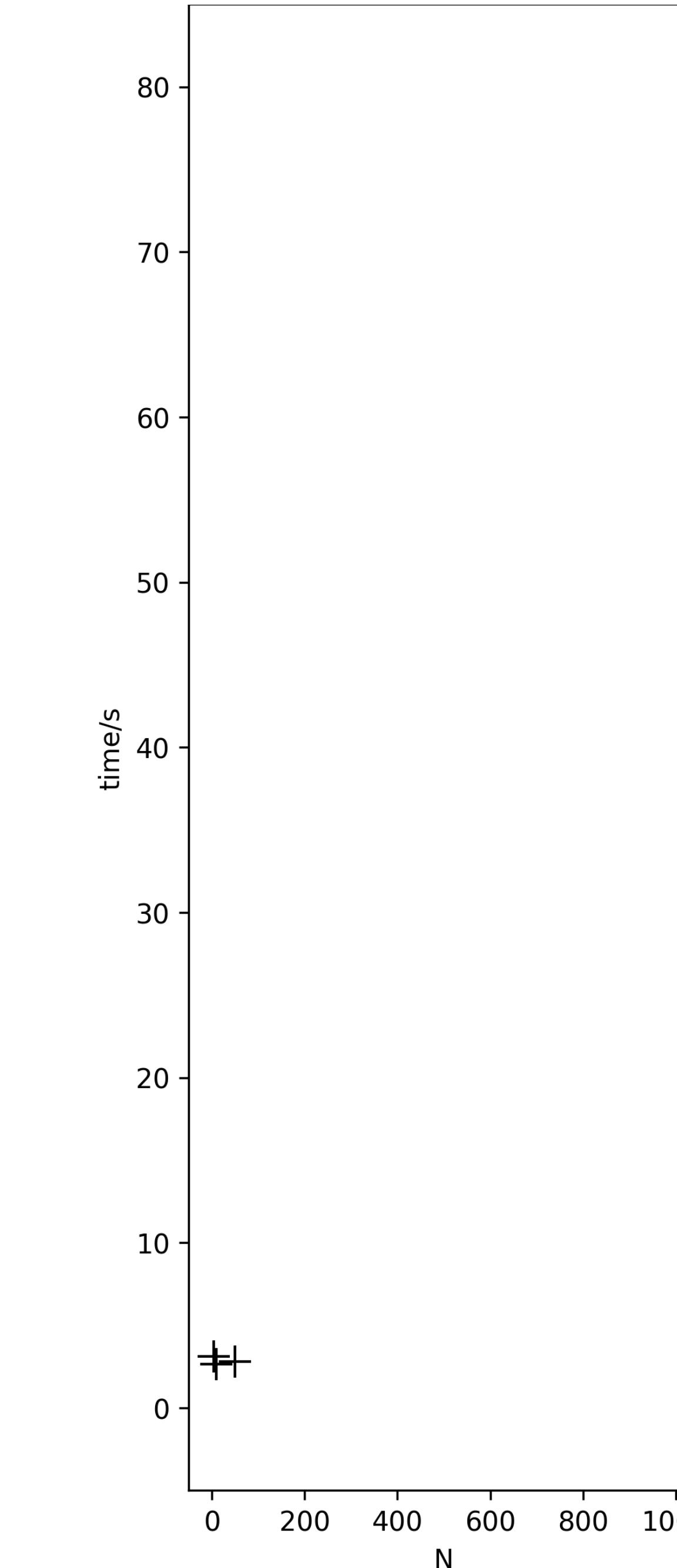
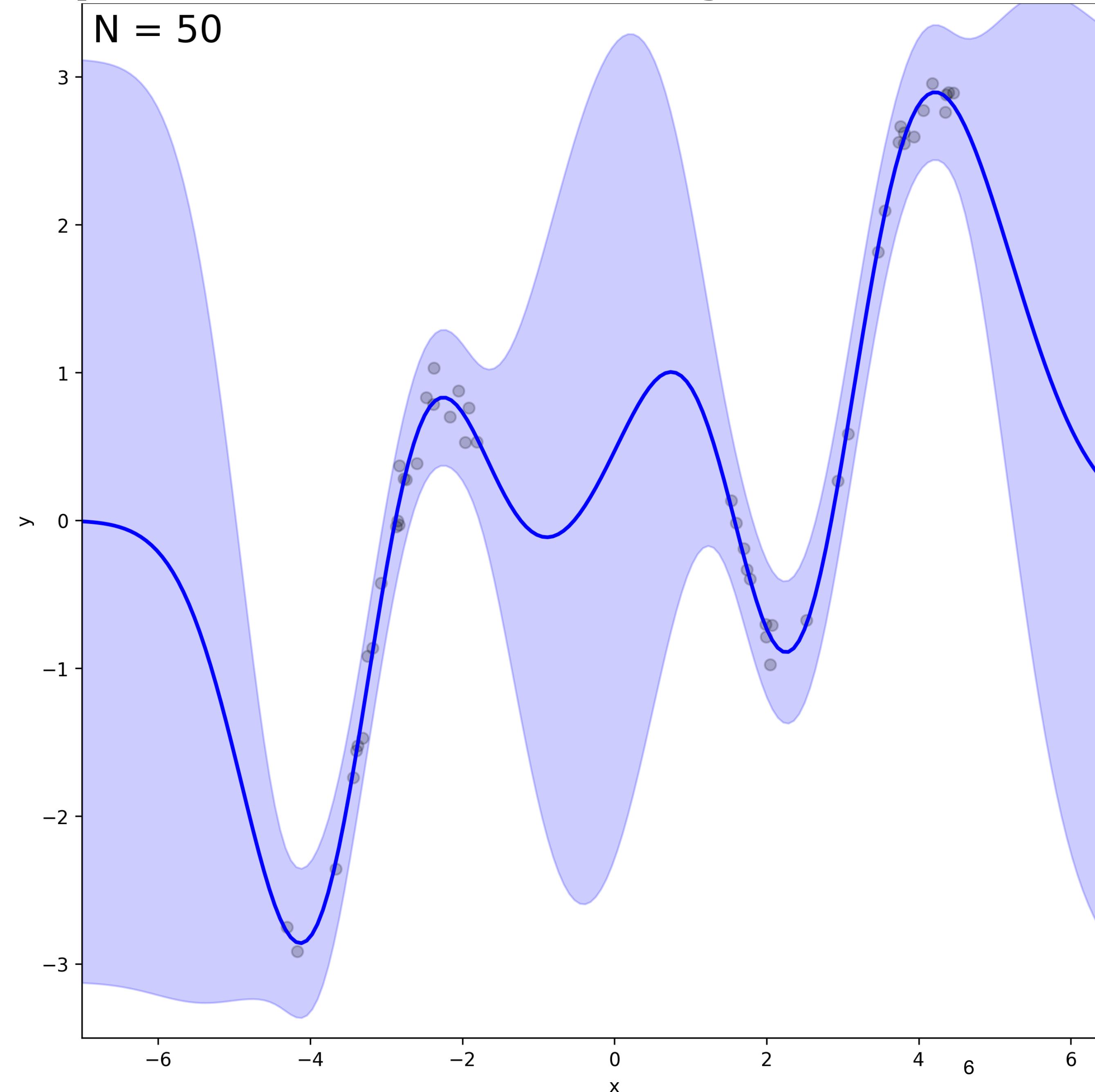
Computational intractability



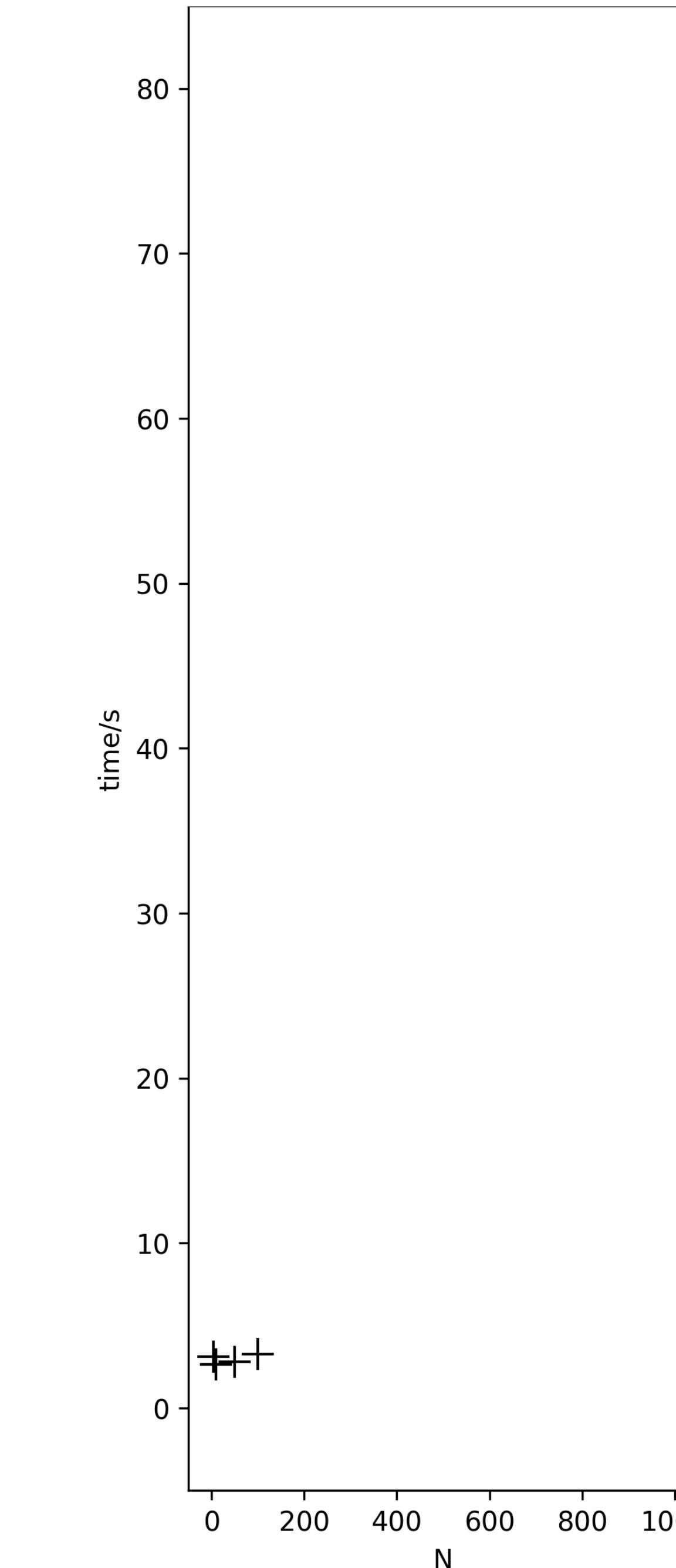
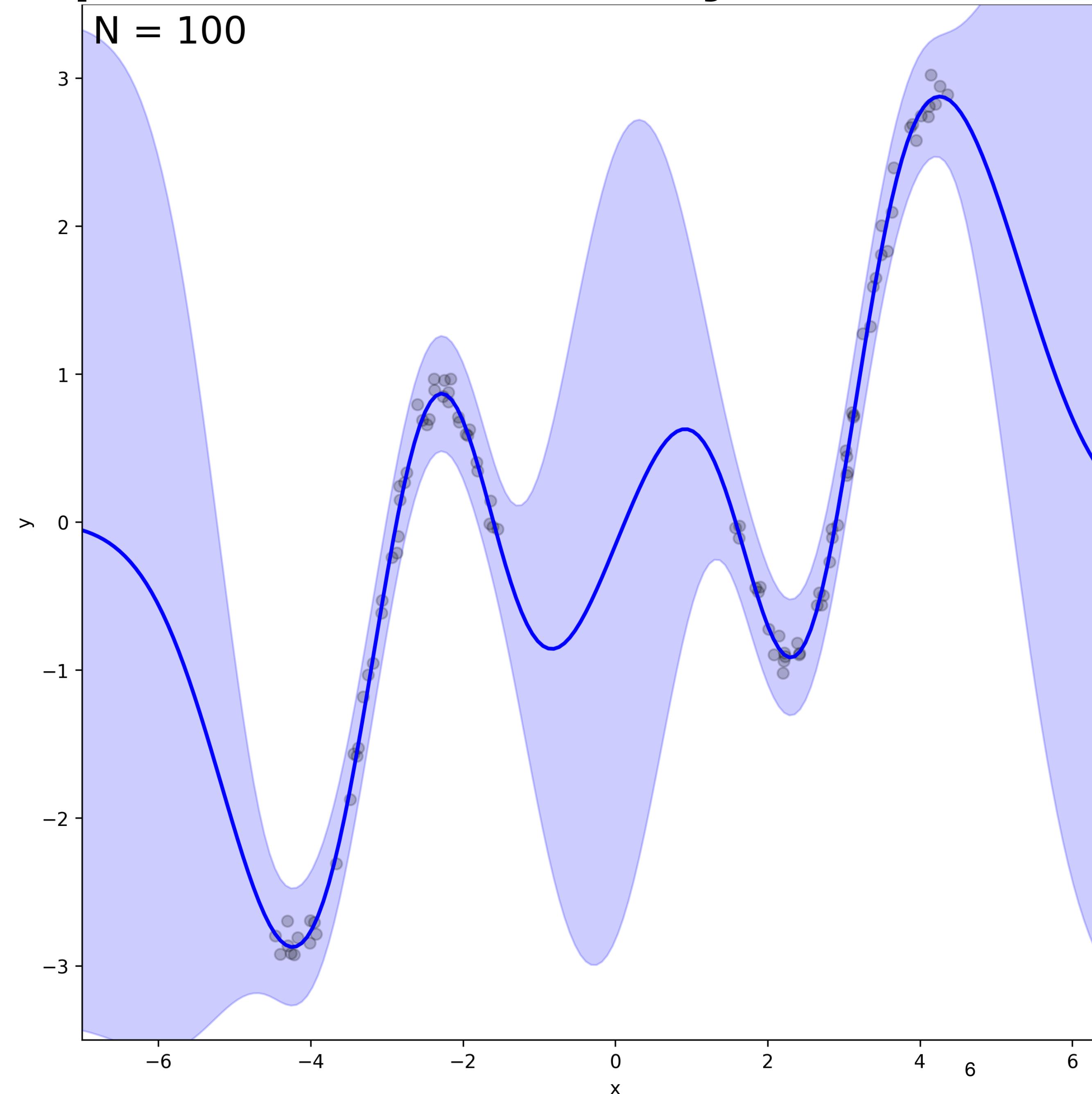
Computational intractability



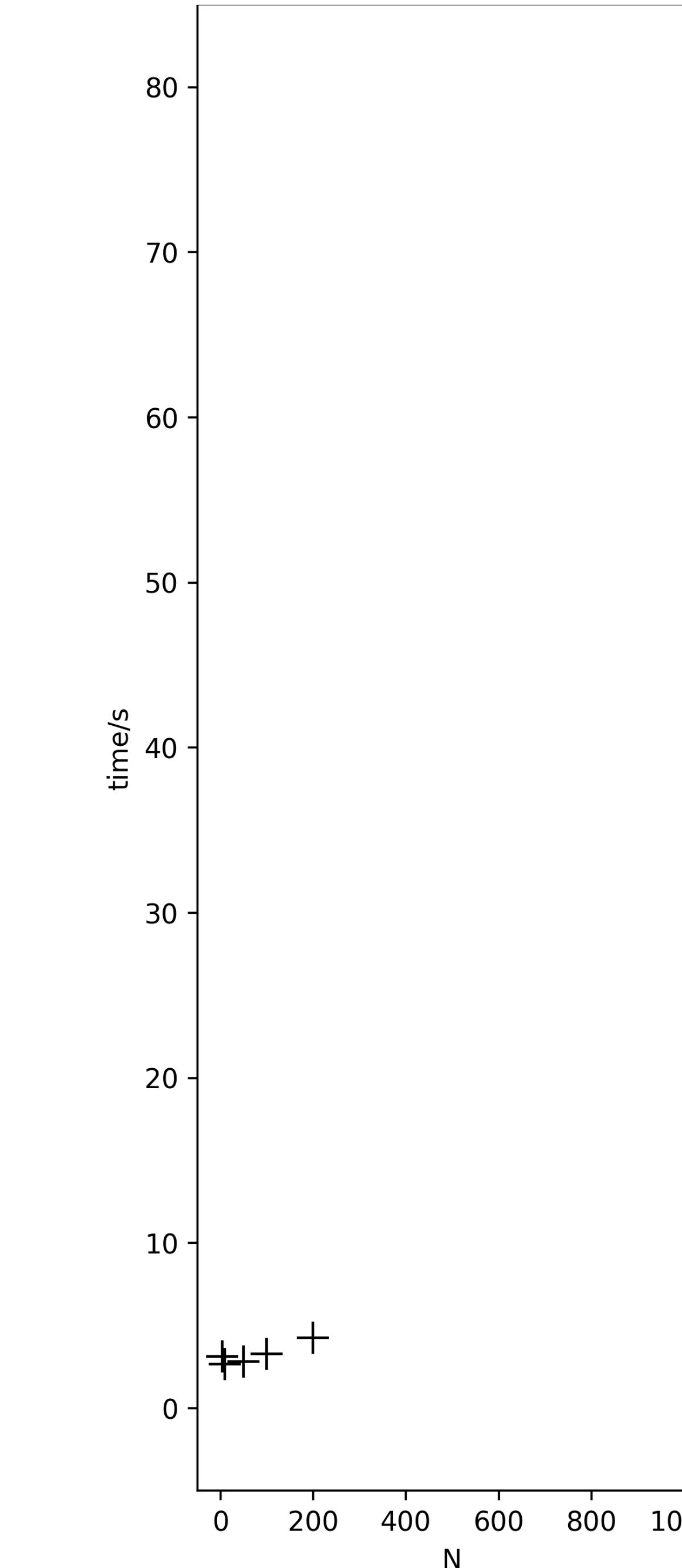
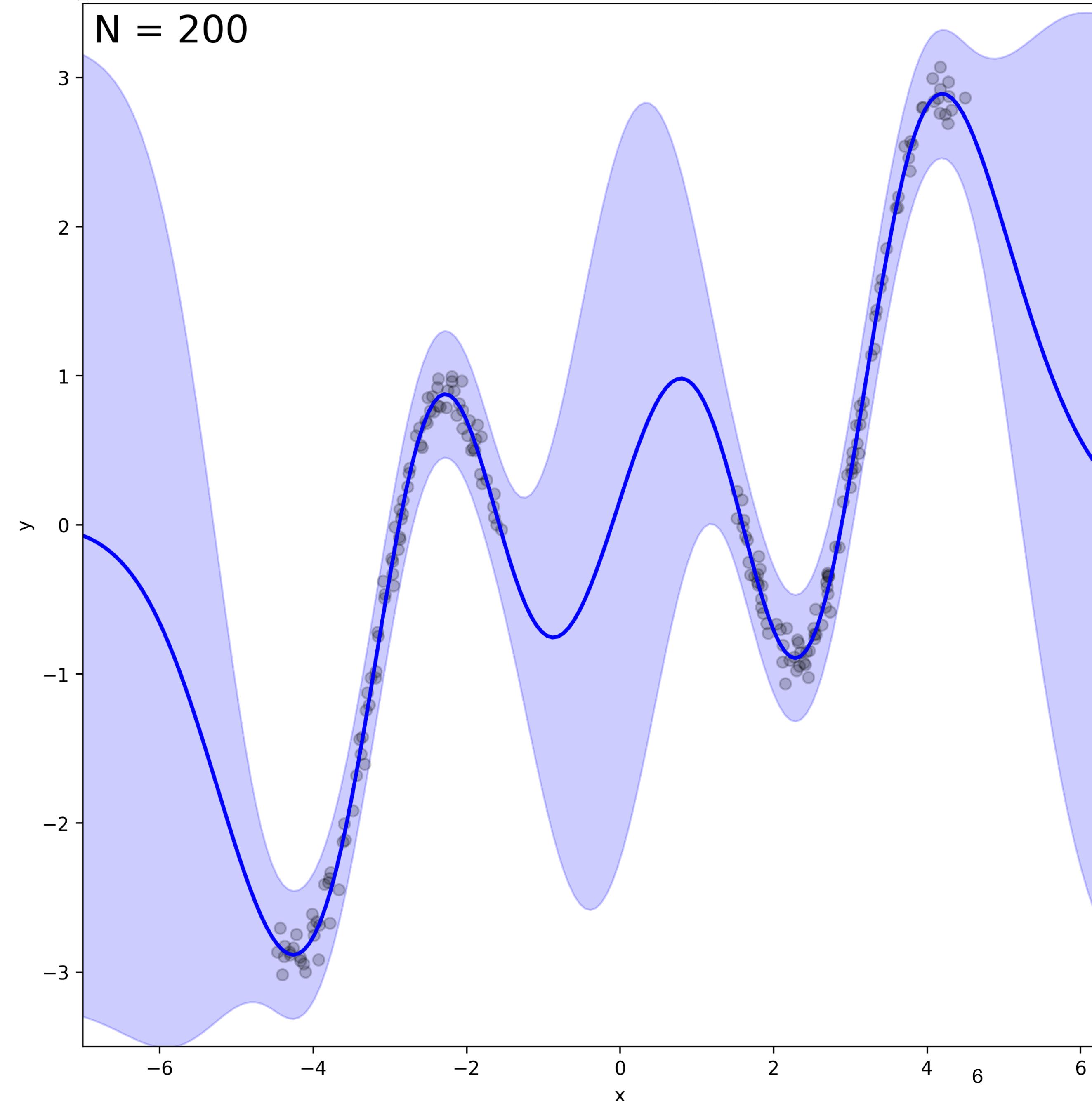
Computational intractability



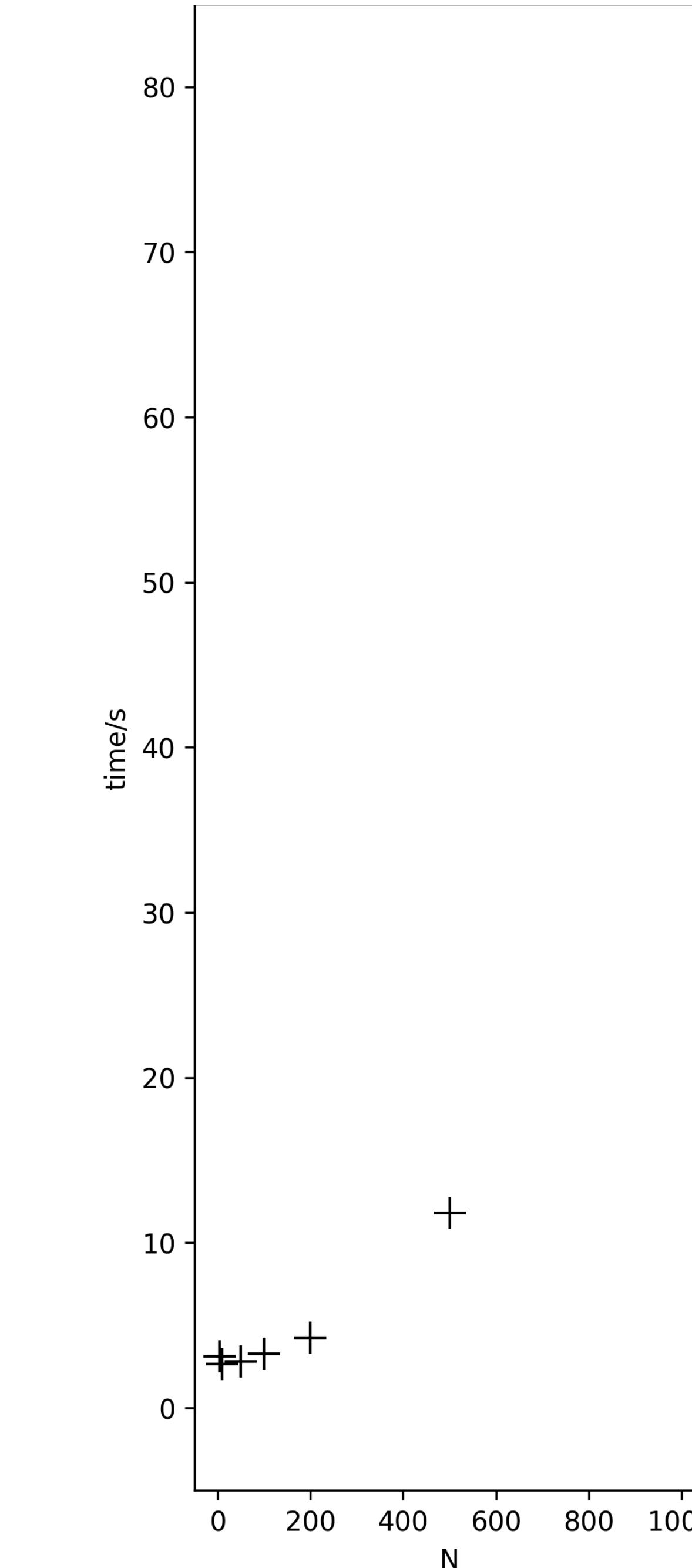
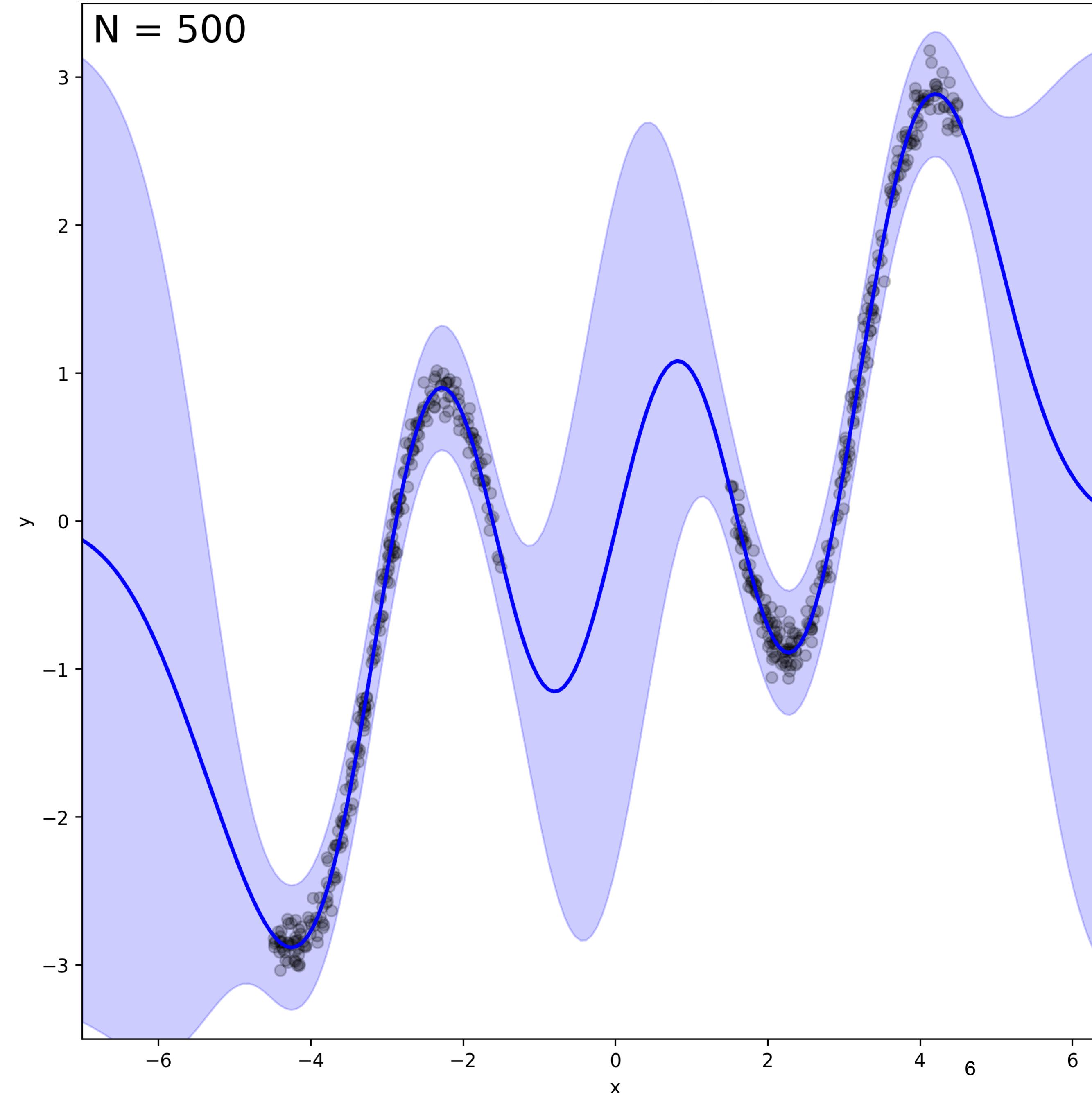
Computational intractability



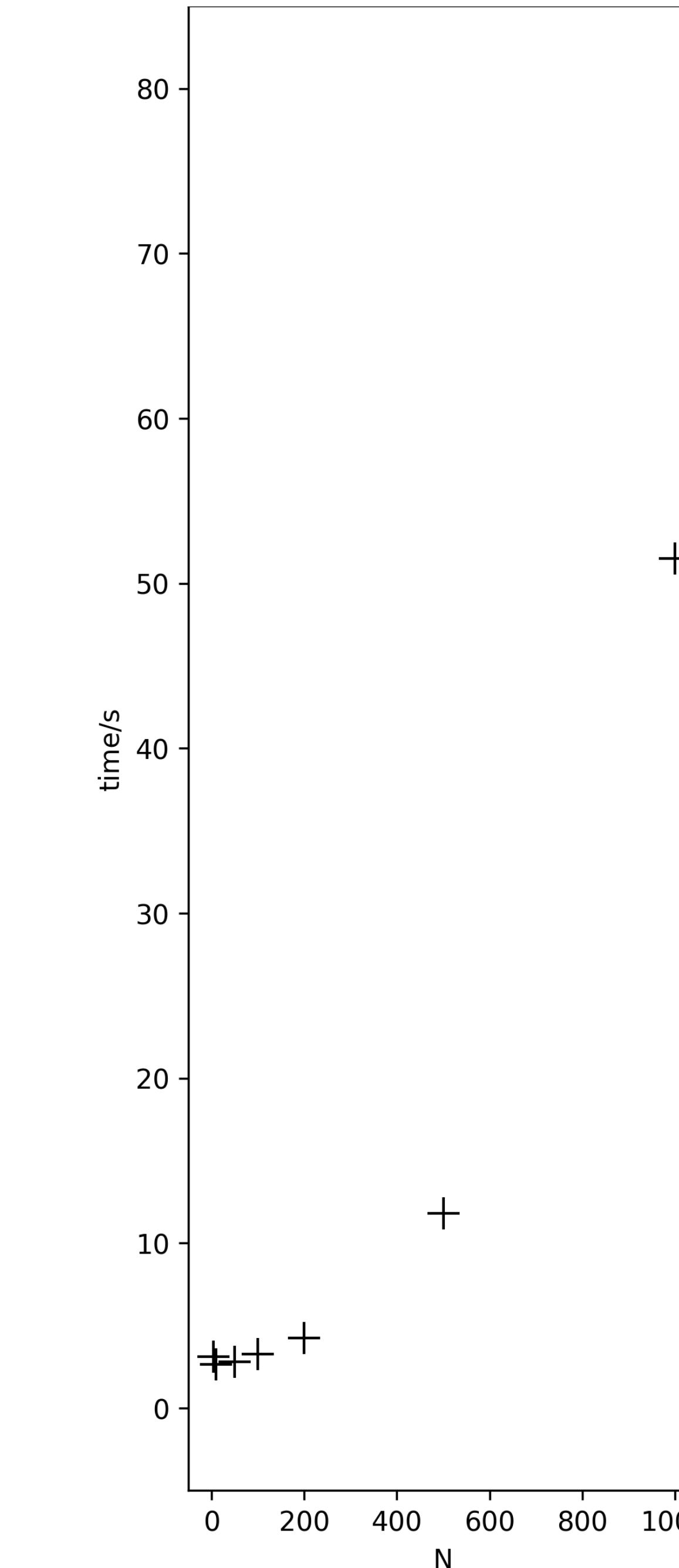
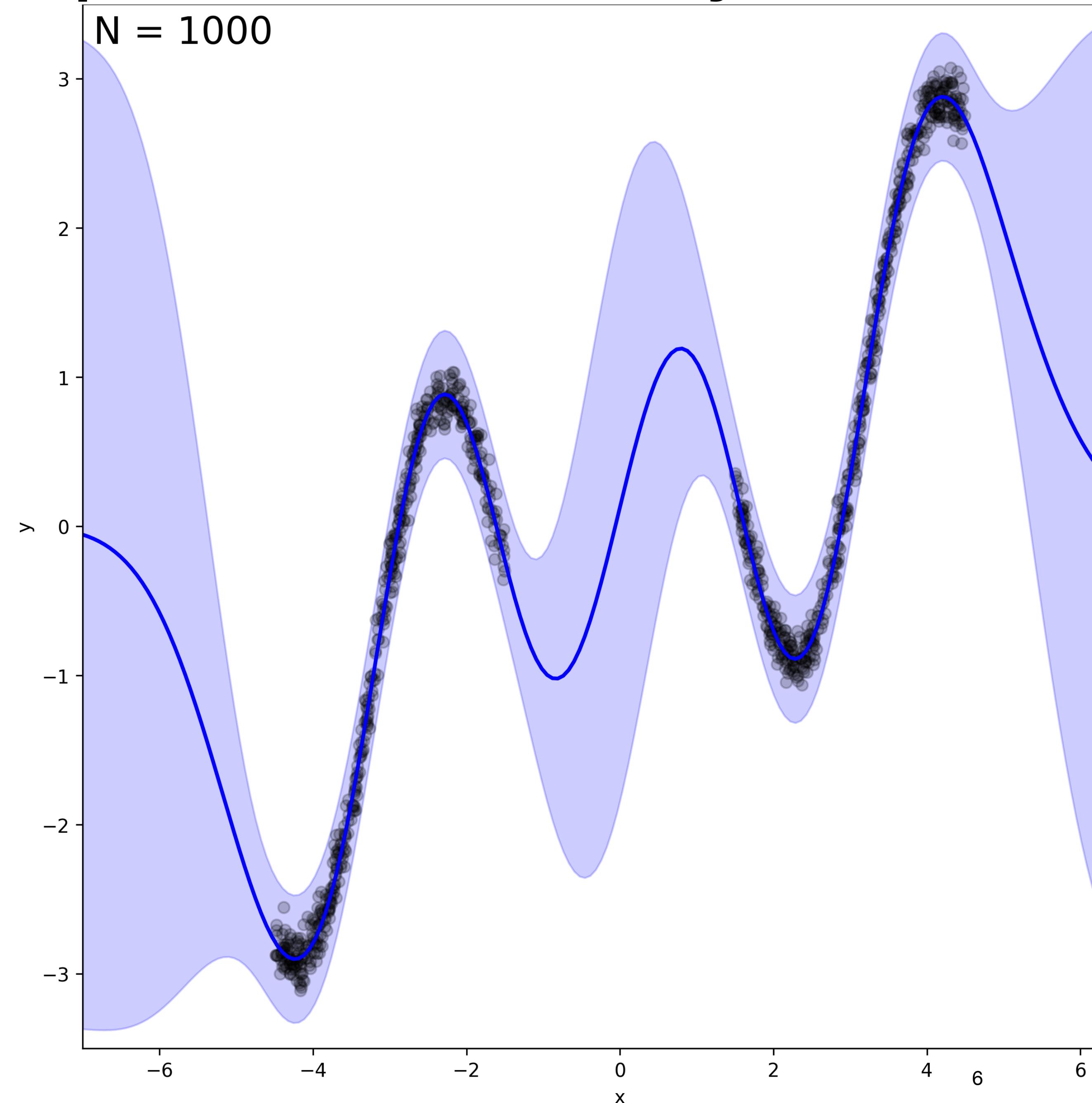
Computational intractability



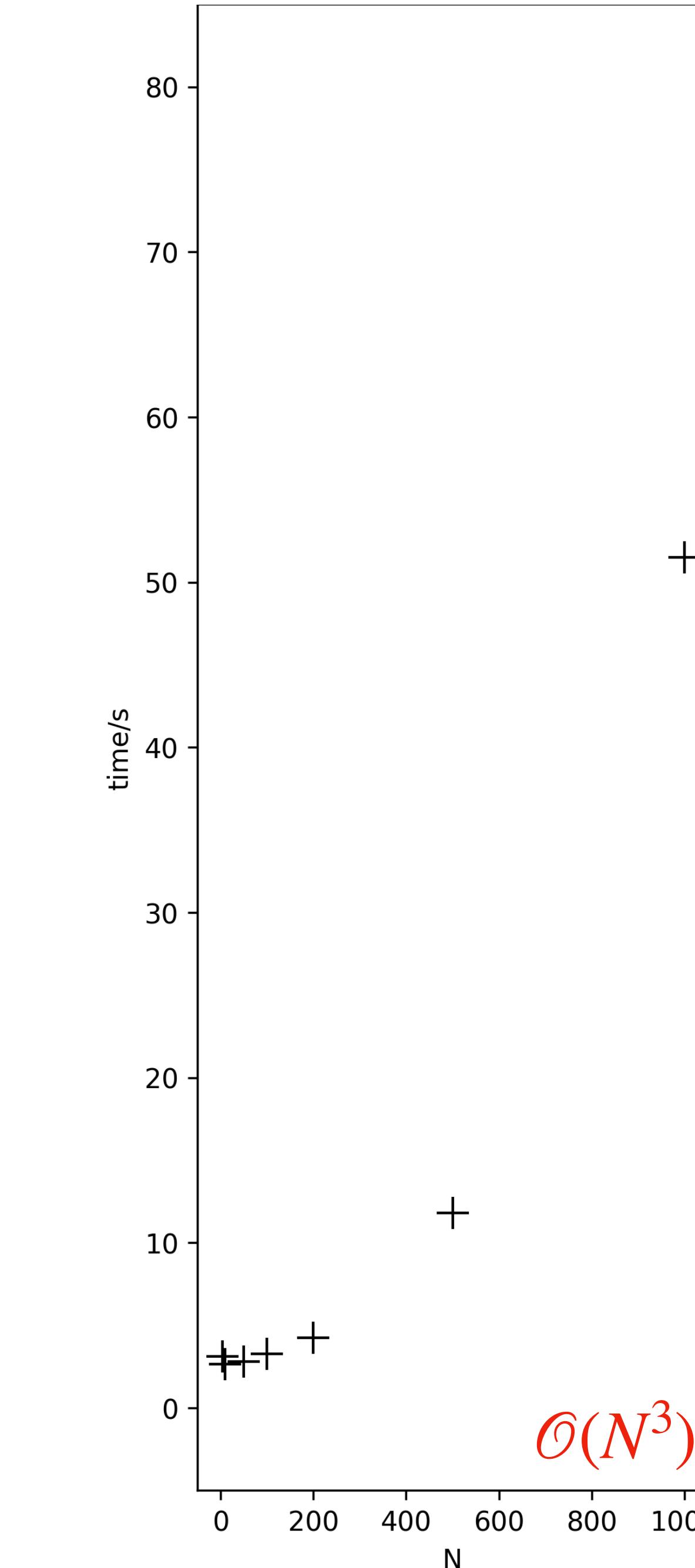
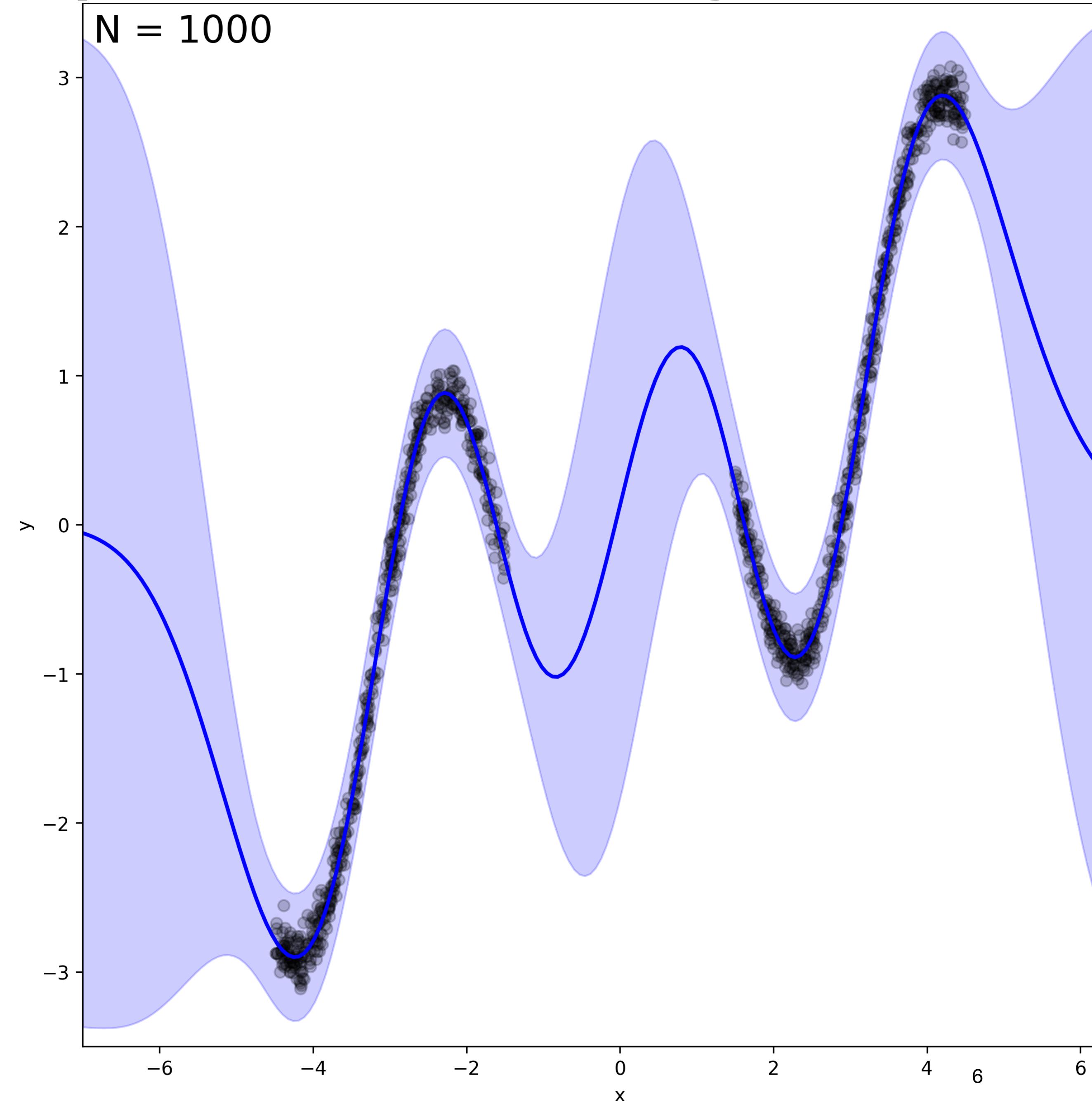
Computational intractability



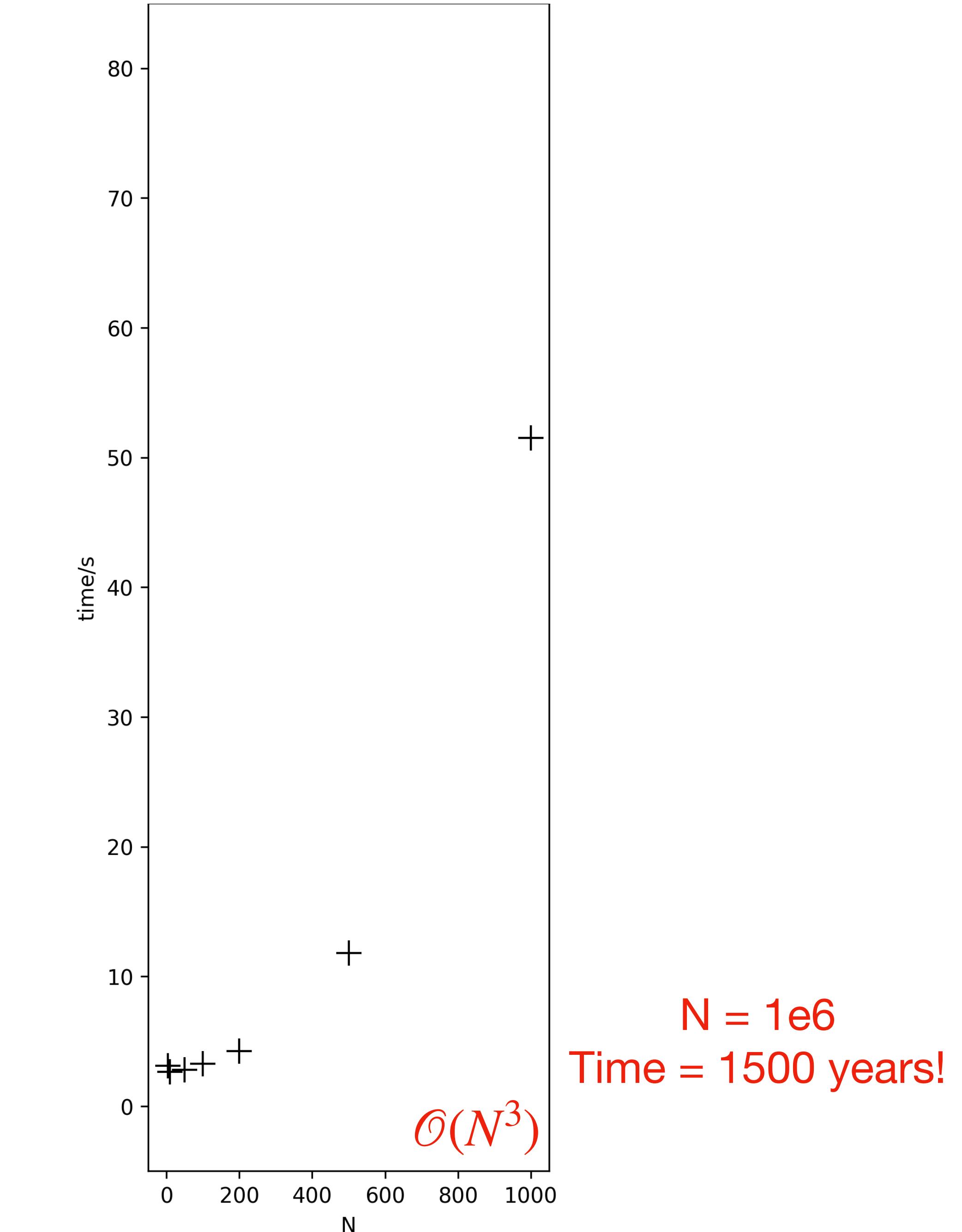
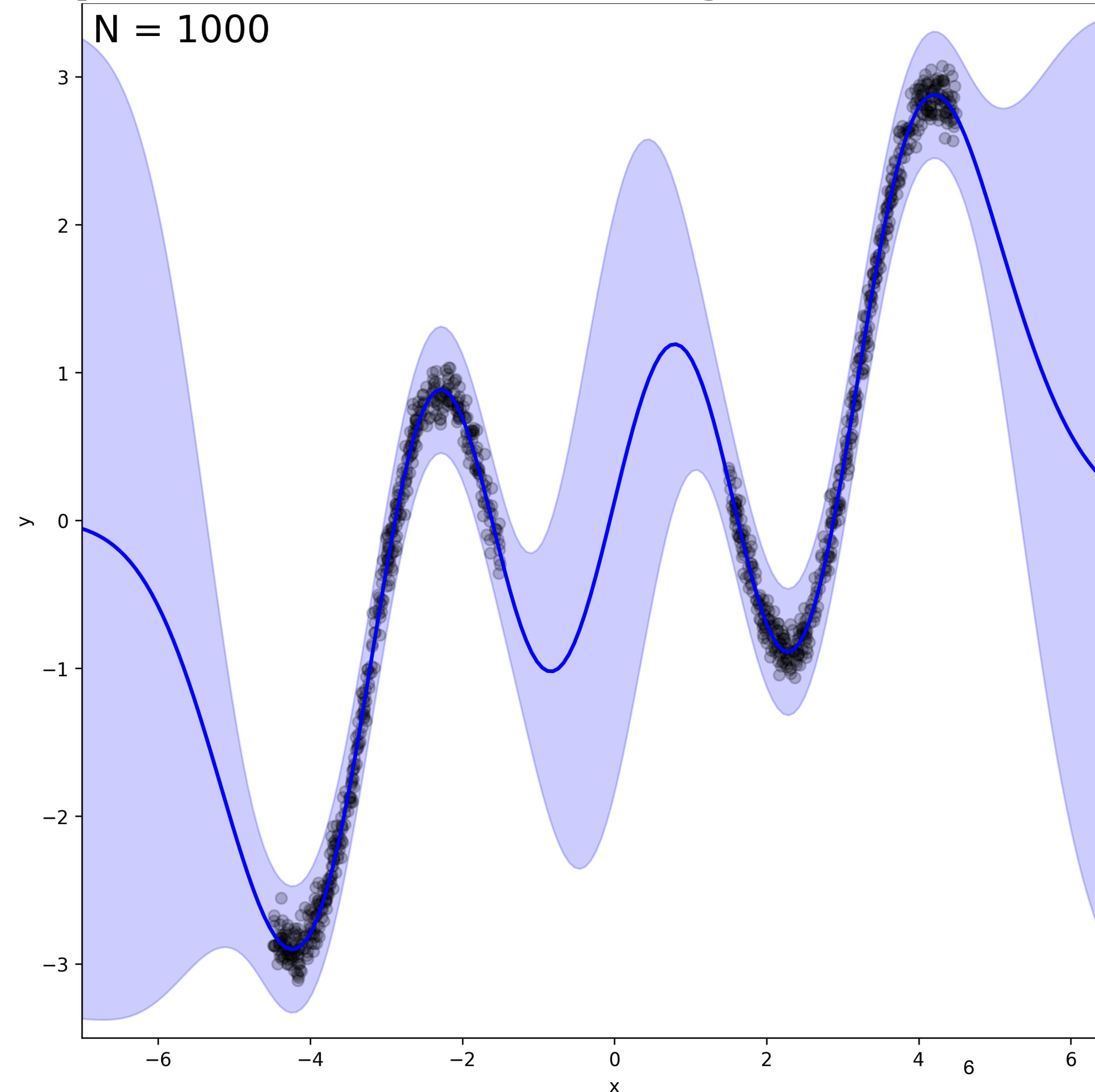
Computational intractability



Computational intractability



Computational intractability



Approximation using pseudo-data or inducing points

Approximation using pseudo-data or inducing points

Prior Likelihood

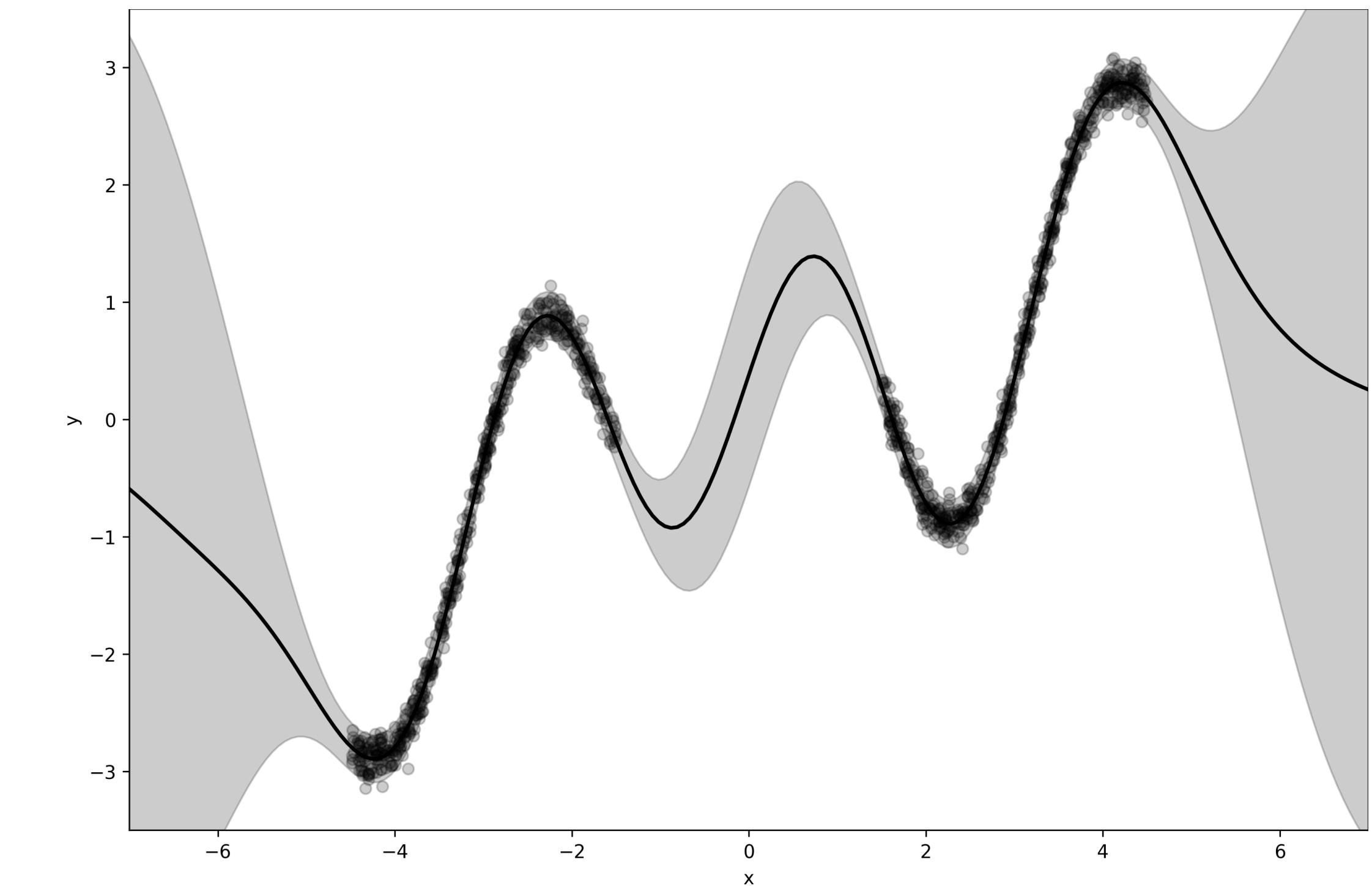
Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

Marginal likelihood

for hyperparameter optimisation aka finding theta

$\mathcal{O}(N^3)$ where $N = \dim(\mathbf{x}, \mathbf{y})$



Approximation using pseudo-data or inducing points

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})$$
$$f = \{\mathbf{u}, f_{\neq \mathbf{u}}\} \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mu, \Sigma)$$

Exact posterior

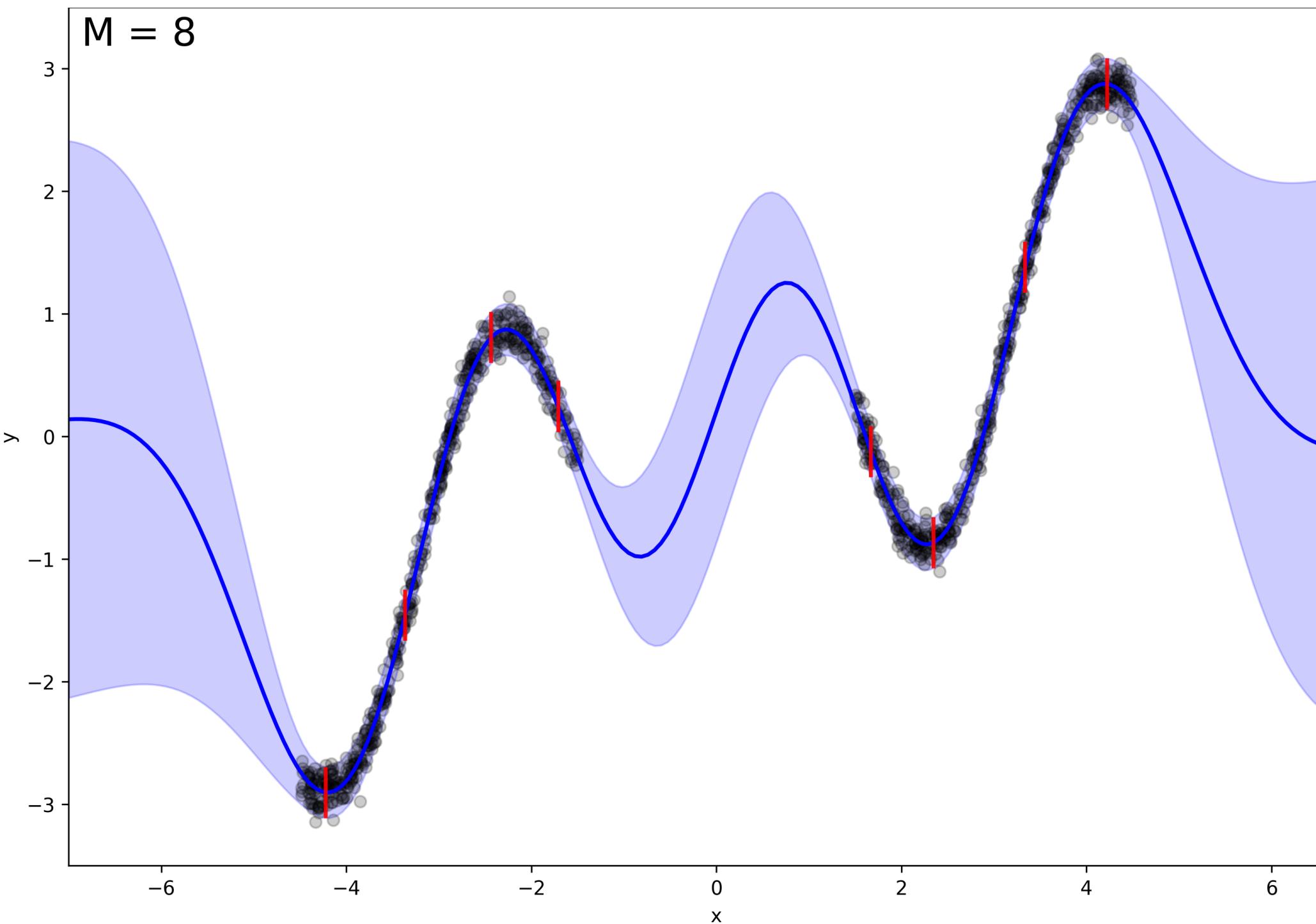
$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

Prior Likelihood

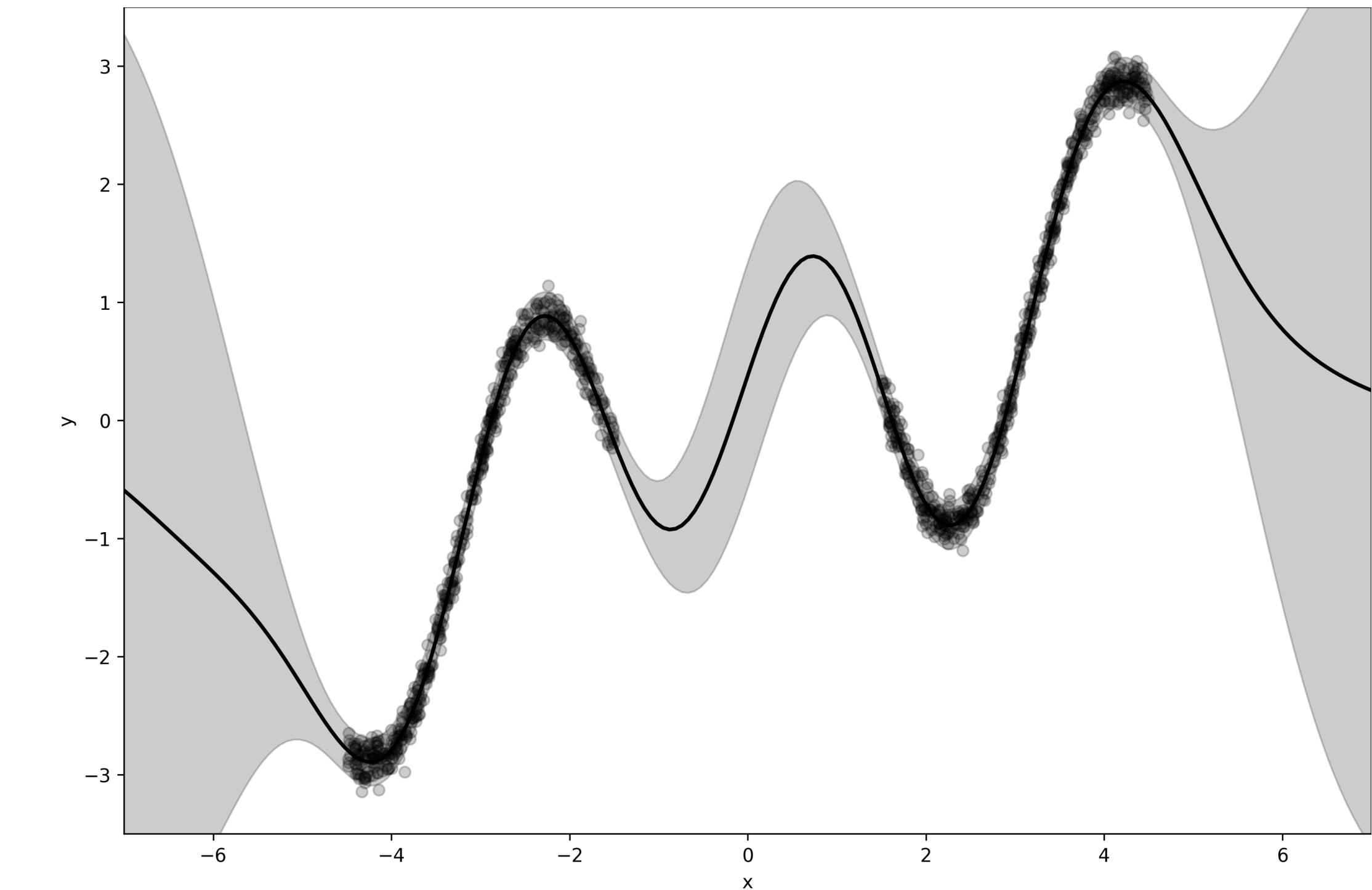
Marginal likelihood

for hyperparameter optimisation aka finding theta

$\mathcal{O}(NM^2)$ where $M = \dim(\mathbf{u}) \ll N$



$\mathcal{O}(N^3)$ where $N = \dim(\mathbf{x}, \mathbf{y})$



Approximation using pseudo-data or inducing points

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})$$
$$f = \{\mathbf{u}, f_{\neq \mathbf{u}}\} \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mu, \Sigma)$$

Exact posterior

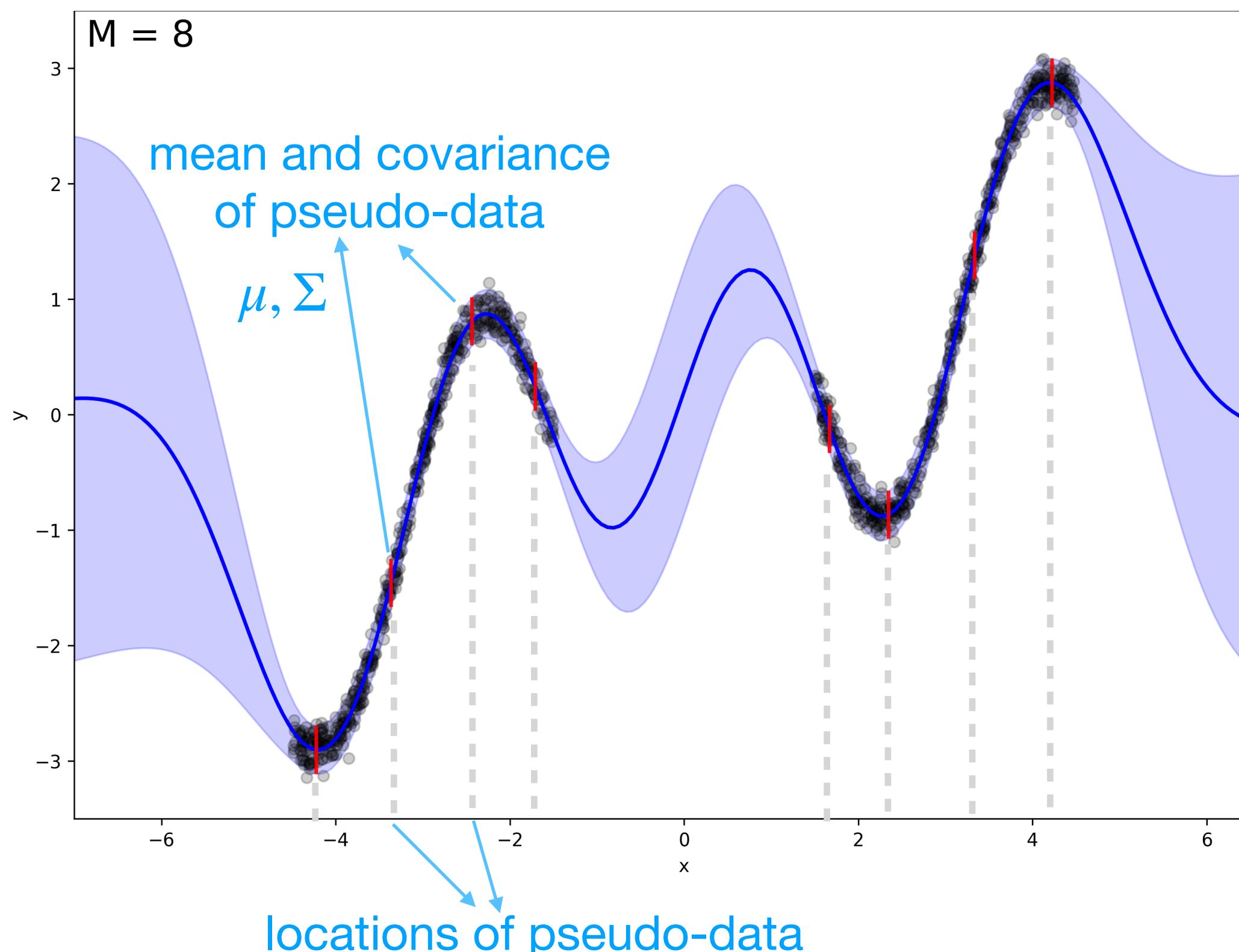
$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

Marginal likelihood

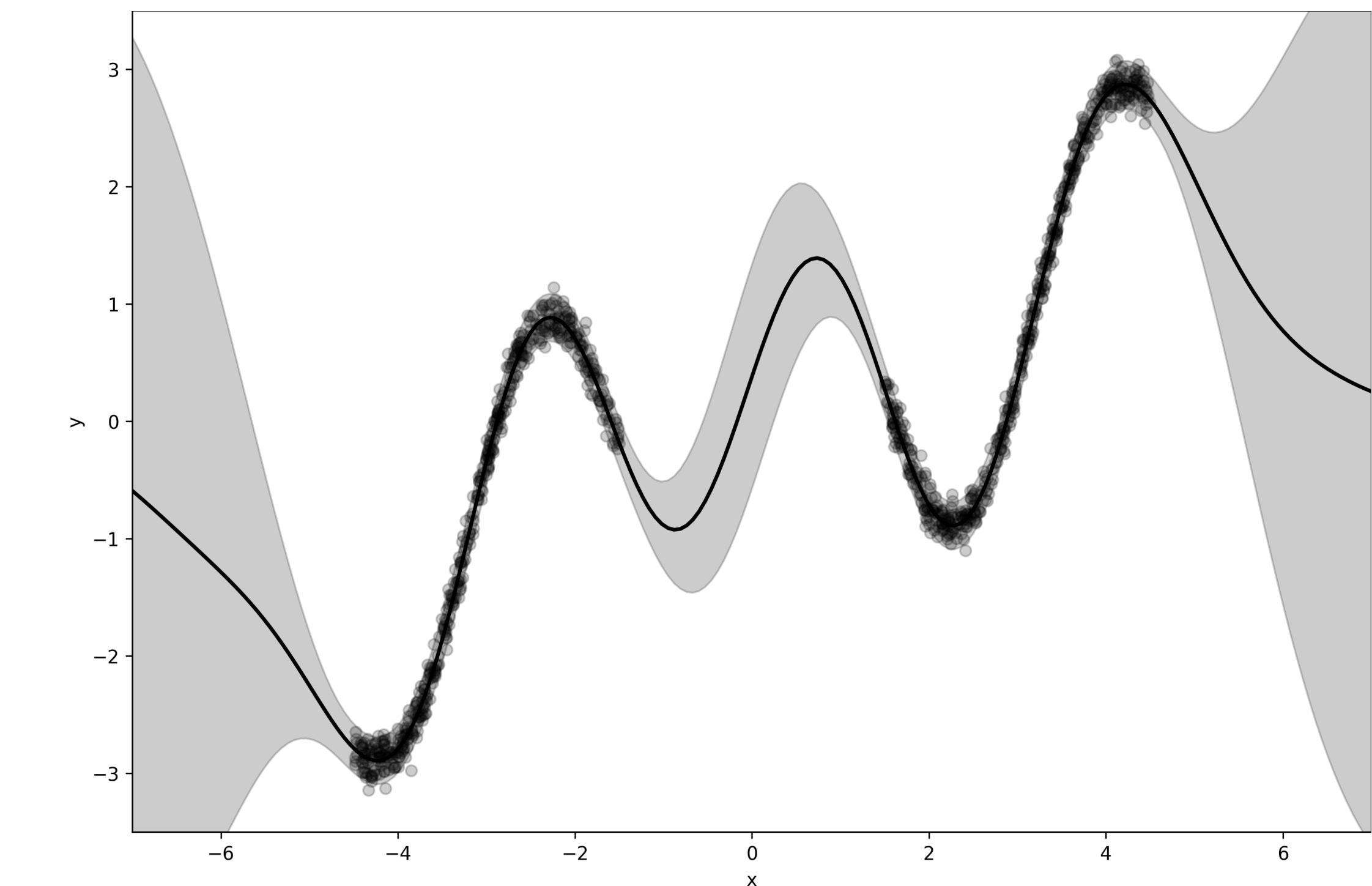
Prior Likelihood

for hyperparameter optimisation aka finding theta

$\mathcal{O}(NM^2)$ where $M = \dim(\mathbf{u}) \ll N$



$\mathcal{O}(N^3)$ where $N = \dim(\mathbf{x}, \mathbf{y})$



Approximation using pseudo-data or inducing points

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})$$
$$f = \{\mathbf{u}, f_{\neq \mathbf{u}}\} \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mu, \Sigma)$$

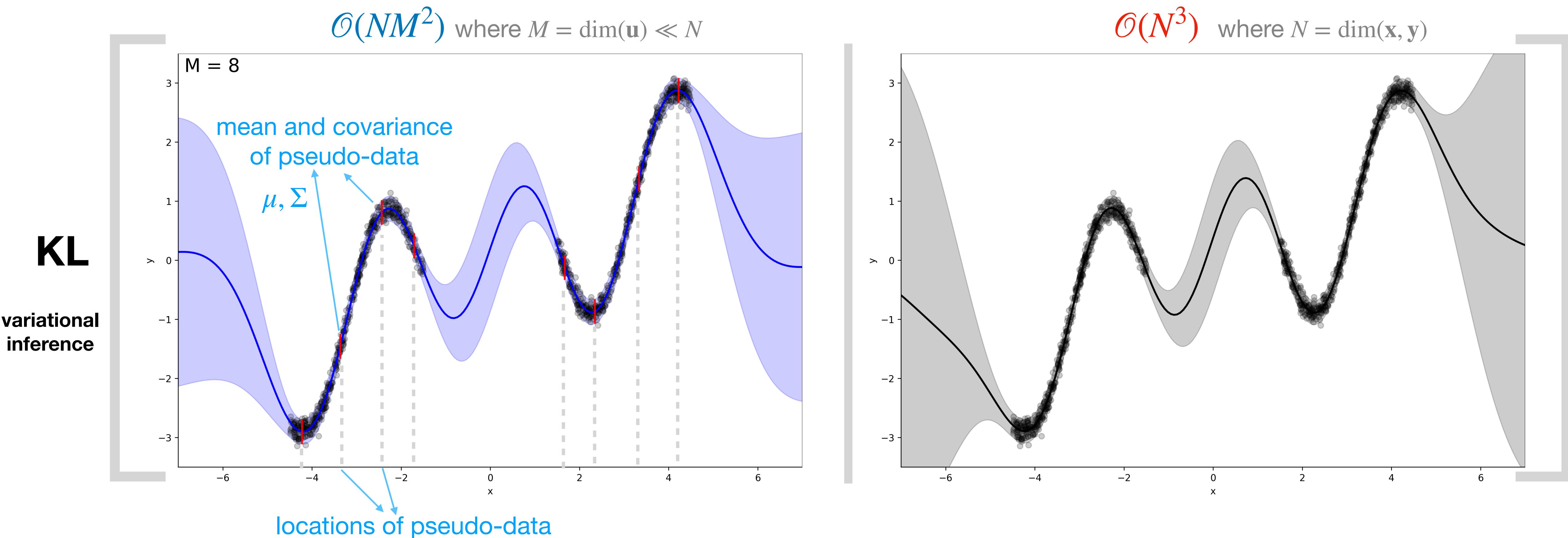
Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

Prior Likelihood

Marginal likelihood

for hyperparameter optimisation aka finding theta



Variational inference

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})$$

Exact posterior

$$p(f | \mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y} | f, \mathbf{x}, \theta)}{p(\mathbf{y} | \mathbf{x}, \theta)}$$

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})$$

Exact posterior

$$p(f | \mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y} | f, \mathbf{x}, \theta)}{p(\mathbf{y} | \mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \| p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})$$

Exact posterior

$$p(f | \mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y} | f, \mathbf{x}, \theta)}{p(\mathbf{y} | \mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \| p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

$$= \log p(\mathbf{y} | \mathbf{x}, \theta) + \int_f q(f|\theta) \log \frac{p(f_{\neq \mathbf{u}} | \mathbf{u}, \theta) q(\mathbf{u})}{p(f|\theta) p(\mathbf{y} | f, \mathbf{x}, \theta)}$$

sub in q and p

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$$

Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \parallel p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) + \int_f q(f|\theta) \log \frac{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)p(\mathbf{u}|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}$$

sub in q and p
noting $p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})$

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$$

Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \parallel p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) + \int_f q(f|\theta) \log \frac{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)p(\mathbf{u}|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}$$

sub in q and p
noting $p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})$

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$$

Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \parallel p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) + \int_f q(f|\theta) \log \frac{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)p(\mathbf{u}|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}$$

sub in q and p
noting $p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})$

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) - \mathcal{F}(q(\mathbf{u}), \theta)$$

\mathcal{F} = negative of the integral term

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$$

Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \parallel p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) + \int_f q(f|\theta) \log \frac{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)p(\mathbf{u}|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}$$

sub in q and p
noting $p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})$

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) - \mathcal{F}(q(\mathbf{u}), \theta)$$

\mathcal{F} = negative of the integral term

minimising $\text{KL}(\text{approx post} \parallel \text{exact post}) \equiv \text{maximising } \mathcal{F}$

as $\log p(\mathbf{y}|\mathbf{x}, \theta)$ does not depend on q

\mathcal{F} : variational lower bound or negative variational free-energy

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$$

Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \parallel p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) + \int_f q(f|\theta) \log \frac{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)p(\mathbf{u}|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}$$

sub in q and p
noting $p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})$

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) - \mathcal{F}(q(\mathbf{u}), \theta)$$

\mathcal{F} = negative of the integral term

minimising $\text{KL}(\text{approx post} \parallel \text{exact post}) \equiv \text{maximising } \mathcal{F}$

as $\log p(\mathbf{y}|\mathbf{x}, \theta)$ does not depend on q

\mathcal{F} : variational lower bound or negative variational free-energy

$$\mathcal{F}(q(\mathbf{u}), \theta) = \int_f q(f|\theta) \log \frac{p(\mathbf{u}|\theta)}{q(\mathbf{u})} + \int_f q(f|\theta) \log p(\mathbf{y}|f, \mathbf{x}, \theta)$$

conditional term cancelled out, split the log term

Variational inference

Approximate posterior

$$q(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$$

Exact posterior

$$p(f|\mathbf{x}, \mathbf{y}, \theta) = \frac{p(f|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}{p(\mathbf{y}|\mathbf{x}, \theta)}$$

$$\text{KL}(q(f|\theta) \parallel p(f|\mathbf{x}, \mathbf{y}, \theta)) = \int_f q(f|\theta) \log \frac{q(f|\theta)}{p(f|\mathbf{x}, \mathbf{y}, \theta)}$$

Kullback-Leibler divergence definition

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) + \int_f q(f|\theta) \log \frac{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)p(\mathbf{u}|\theta)p(\mathbf{y}|f, \mathbf{x}, \theta)}$$

sub in q and p
noting $p(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})$

$$= \log p(\mathbf{y}|\mathbf{x}, \theta) - \mathcal{F}(q(\mathbf{u}), \theta)$$

\mathcal{F} = negative of the integral term

minimising $\text{KL}(\text{approx post} \parallel \text{exact post}) \equiv \text{maximising } \mathcal{F}$

as $\log p(\mathbf{y}|\mathbf{x}, \theta)$ does not depend on q

\mathcal{F} : variational lower bound or negative variational free-energy

$$\mathcal{F}(q(\mathbf{u}), \theta) = \int_f q(f|\theta) \log \frac{p(\mathbf{u}|\theta)}{q(\mathbf{u})} + \int_f q(f|\theta) \log p(\mathbf{y}|f, \mathbf{x}, \theta)$$

conditional term cancelled out, split the log term

$$= -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u}|\theta)) + \sum_{n=1}^N \int_{f_n} q(f(\mathbf{x}_n)|\theta) \log p(y_n|f(\mathbf{x}_n), \theta)$$

since $p(\mathbf{y}|f, \mathbf{x}, \theta) = \prod_{n=1}^N p(y_n|f(\mathbf{x}_n), \theta)$ as in GPR and GPC

Computational complexity

Computational complexity

$$\mathcal{F}(q(\mathbf{u}), \theta) = -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \theta)) + \sum_{n=1}^N \int_{f_n} q(f(\mathbf{x}_n) \mid \theta) \log p(y_n \mid f(\mathbf{x}_n), \theta)$$

Computational complexity

KL term **1** Expected log likelihood term **2**

$$\mathcal{F}(q(\mathbf{u}), \theta) = -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \theta)) + \sum_{n=1}^N \int_{f_n} q(f(\mathbf{x}_n) \mid \theta) \log p(y_n \mid f(\mathbf{x}_n), \theta)$$

Computational complexity

KL term **1**

$$\mathcal{F}(q(\mathbf{u}), \theta) = -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \theta)) + \sum_{n=1}^N \int_{f_n}$$

Expected log likelihood term **2**

For general likelihood

2 can be estimated by Monte Carlo with reparameterisation trick or quadrature $\int_{f_n} \approx \sum_{f_{n,k}}$, where $f_{n,k} \sim q(f_n)$

Cost: $\mathcal{O}(M^3)$ for **1**

$\mathcal{O}(M^2)$ per data point for **2**

Total cost: $\mathcal{O}(NM^2 + M^3)$

Computational complexity

KL term **1**

$$\mathcal{F}(q(\mathbf{u}), \theta) = -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \theta)) + \sum_{n=1}^N \int_{f_n}$$

Expected log likelihood term **2**

For general likelihood

2 can be estimated by Monte Carlo with reparameterisation trick or quadrature $\int_{f_n} \approx \sum_{f_{n,k}}$, where $f_{n,k} \sim q(f_n)$

Cost: $\mathcal{O}(M^3)$ for **1**

$\mathcal{O}(M^2)$ per data point for **2**

Total cost: $\mathcal{O}(NM^2)$ if $M \ll N$

Computational complexity

KL term **1**

$$\mathcal{F}(q(\mathbf{u}), \theta) = -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \theta)) + \sum_{n=1}^N \int_{f_n}$$

Expected log likelihood term **2**

For general likelihood

2 can be estimated by Monte Carlo with reparameterisation trick or quadrature $\int_{f_n} \approx \sum_{f_{n,k}}$, where $f_{n,k} \sim q(f_n)$

Cost: $\mathcal{O}(M^3)$ for **1**

$\mathcal{O}(M^2)$ per data point for **2**

Total cost: $\mathcal{O}(NM^2)$ if $M \ll N$

Minibatch stochastic approximation

$$\sum_{n=1}^N \approx \frac{N}{B} \sum_{b=1}^B \quad \text{with batch size } B \ll N$$

Total cost: $\mathcal{O}(BM^2 + M^3)$

Hensman et al 2013

Computational complexity

KL term **1**

$$\mathcal{F}(q(\mathbf{u}), \theta) = -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \theta)) + \sum_{n=1}^N \int_{f_n}$$

Expected log likelihood term **2**

For general likelihood

2 can be estimated by Monte Carlo with reparameterisation trick or quadrature $\int_{f_n} \approx \sum_{f_{n,k}}$, where $f_{n,k} \sim q(f_n)$

Cost: $\mathcal{O}(M^3)$ for **1**

$\mathcal{O}(M^2)$ per data point for **2**

Total cost: $\mathcal{O}(NM^2)$ if $M \ll N$

Minibatch stochastic approximation

$$\sum_{n=1}^N \approx \frac{N}{B} \sum_{b=1}^B \quad \text{with batch size } B \ll N$$

Total cost: $\mathcal{O}(M^3)$ if $B \approx M$

Hensman et al 2013

Computational complexity

KL term **1**

$$\mathcal{F}(q(\mathbf{u}), \theta) = -\text{KL}(q(\mathbf{u}) \parallel p(\mathbf{u} \mid \theta)) + \sum_{n=1}^N \int_{f_n}$$

Expected log likelihood term **2**

$$q(f(\mathbf{x}_n) \mid \theta) \log p(y_n \mid f(\mathbf{x}_n), \theta)$$

For general likelihood

2 can be estimated by Monte Carlo with reparameterisation trick or quadrature $\int_{f_n} \approx \sum_{f_{n,k}}$, where $f_{n,k} \sim q(f_n)$

Cost: $\mathcal{O}(M^3)$ for **1**

$\mathcal{O}(M^2)$ per data point for **2**

Total cost: $\mathcal{O}(NM^2)$ if $M \ll N$

Minibatch stochastic approximation

Hensman et al 2013

$$\sum_{n=1}^N \approx \frac{N}{B} \sum_{b=1}^B$$

with batch size $B \ll N$

Total cost: $\mathcal{O}(M^3)$ if $B \approx M$

Collapsed bound for regression with Gaussian likelihood

Titsias 2009

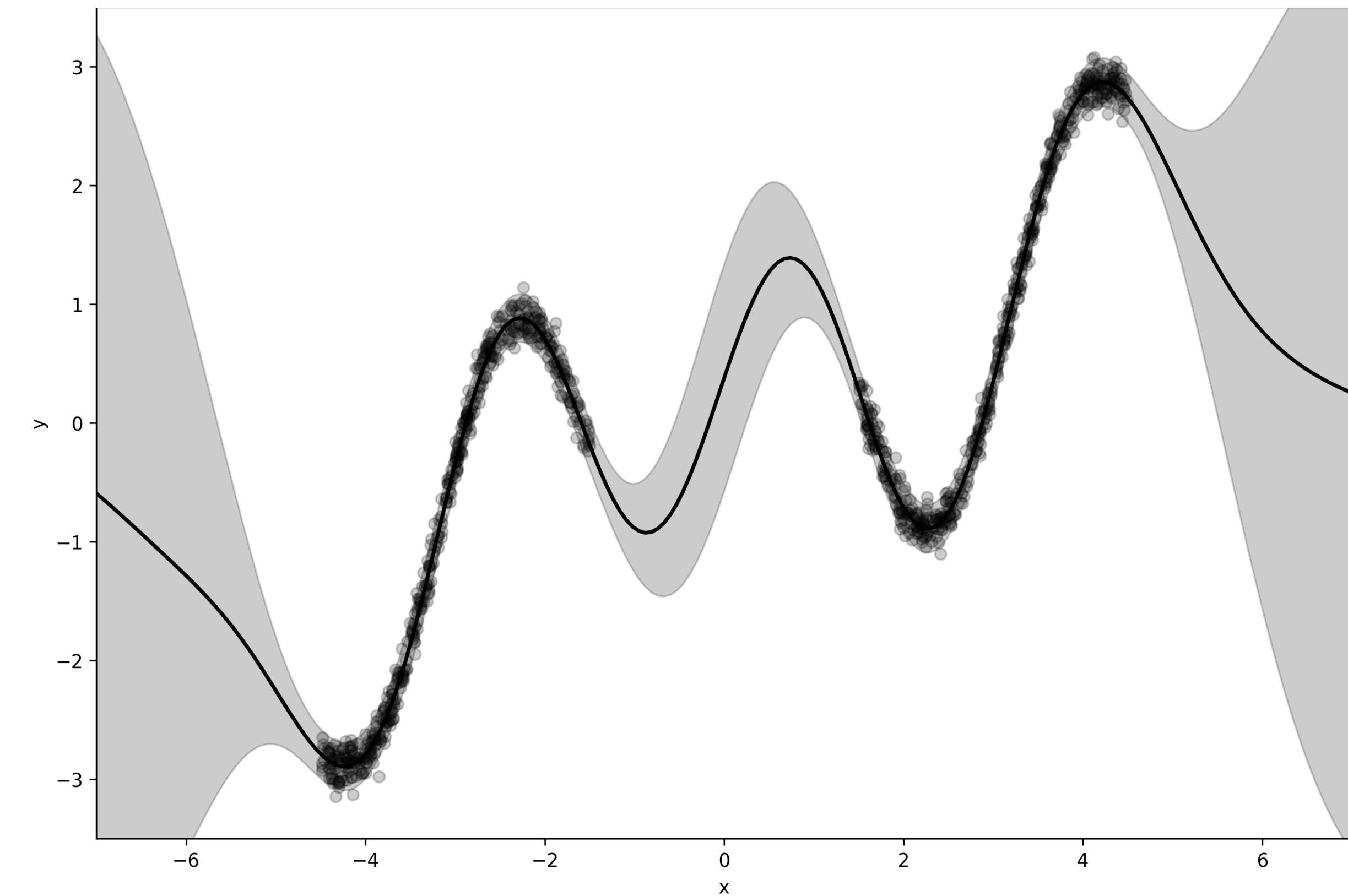
$$\text{optimal } q(\mathbf{u}) \propto p(\mathbf{u} \mid \theta) \mathcal{N}(\mathbf{y}; \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{u}, \sigma_y^2 \mathbf{I})$$

$$\mathcal{F}(\theta) = \log \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}} + \sigma_y^2 \mathbf{I}) - \frac{1}{2\sigma_y^2} \text{trace}(\mathbf{K}_{\mathbf{f}\mathbf{f}} - \mathbf{K}_{\mathbf{f}\mathbf{u}} \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \mathbf{K}_{\mathbf{u}\mathbf{f}})$$

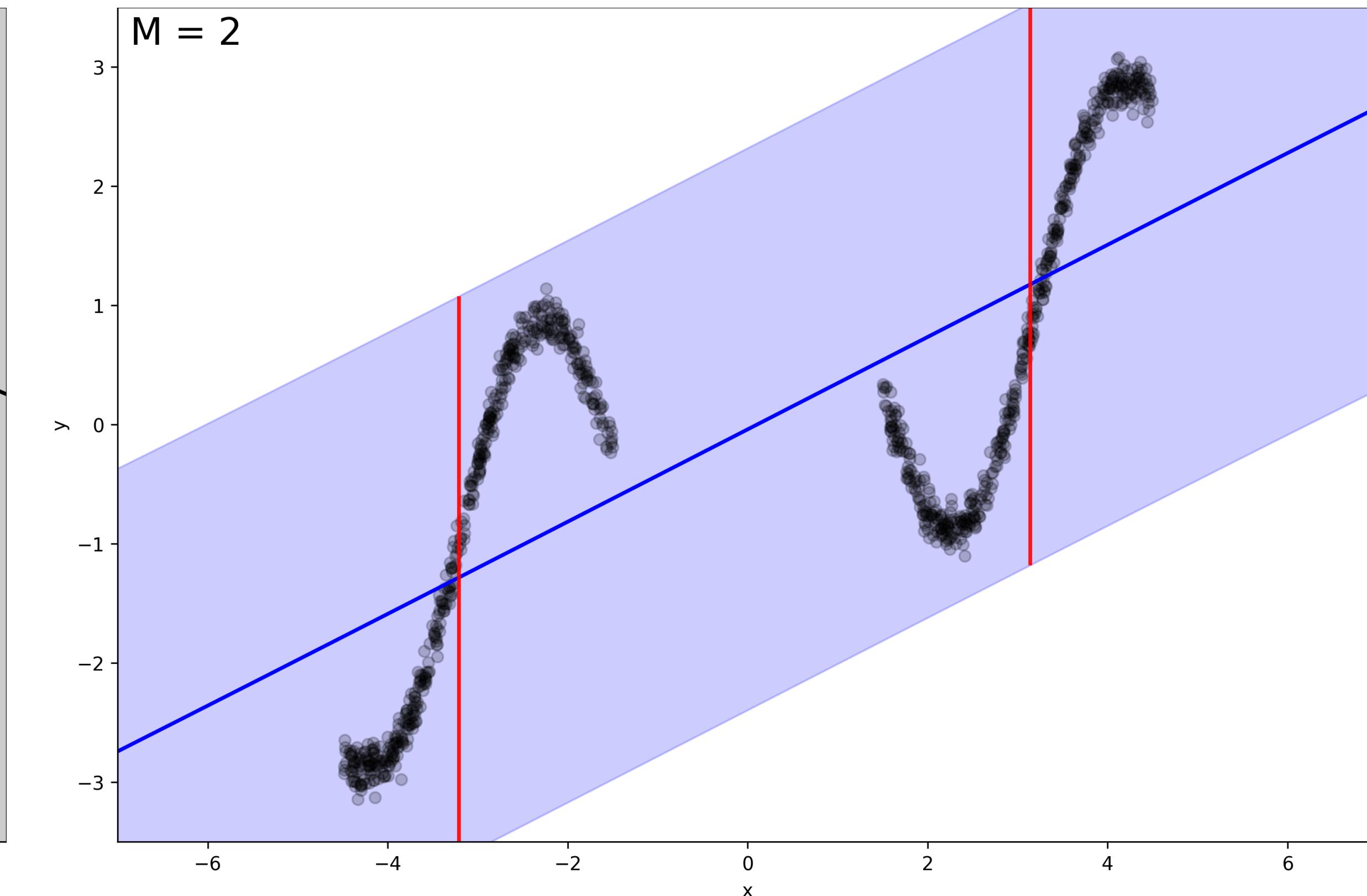
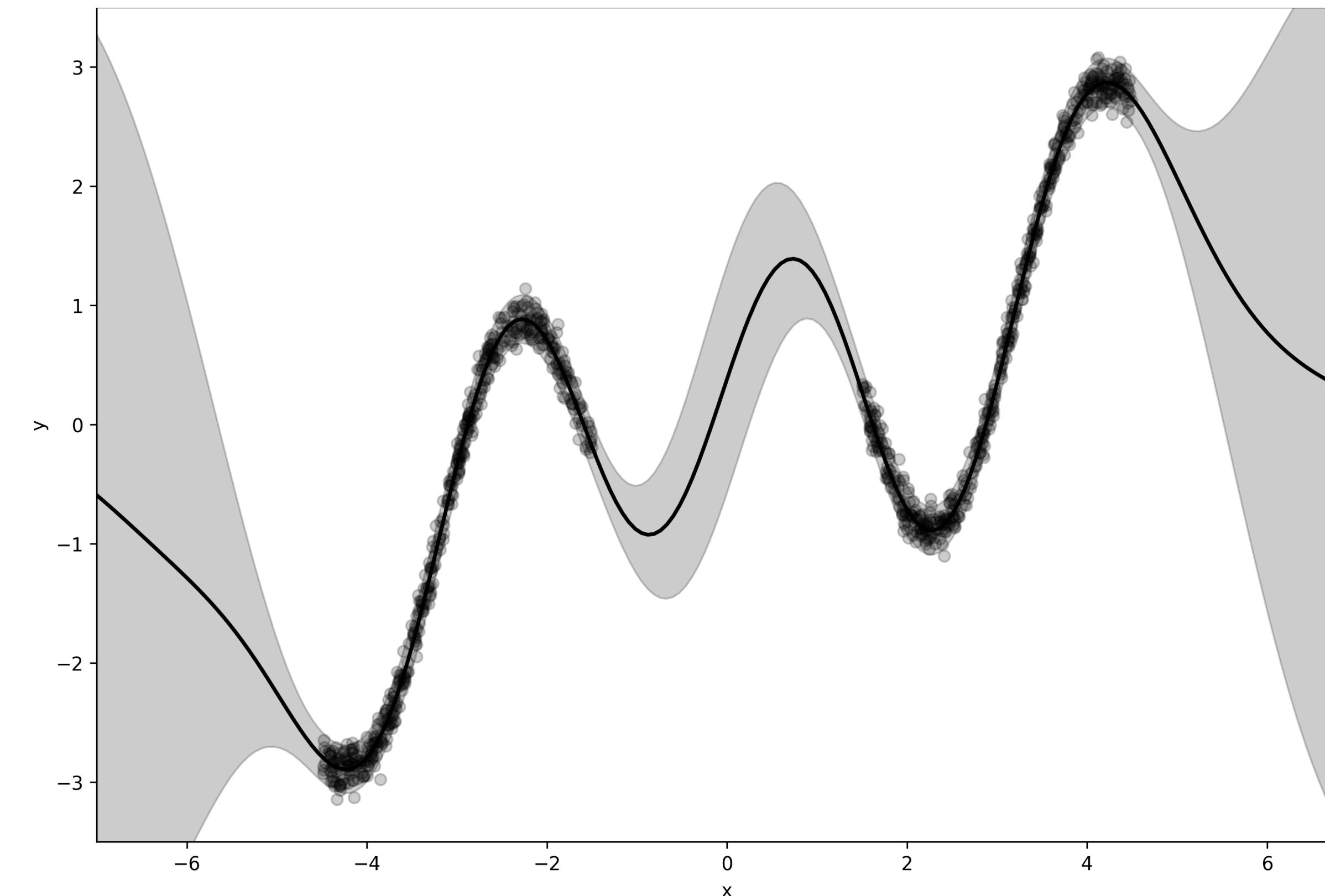
Total cost: $\mathcal{O}(NM^2)$

How many pseudo-data points is enough?

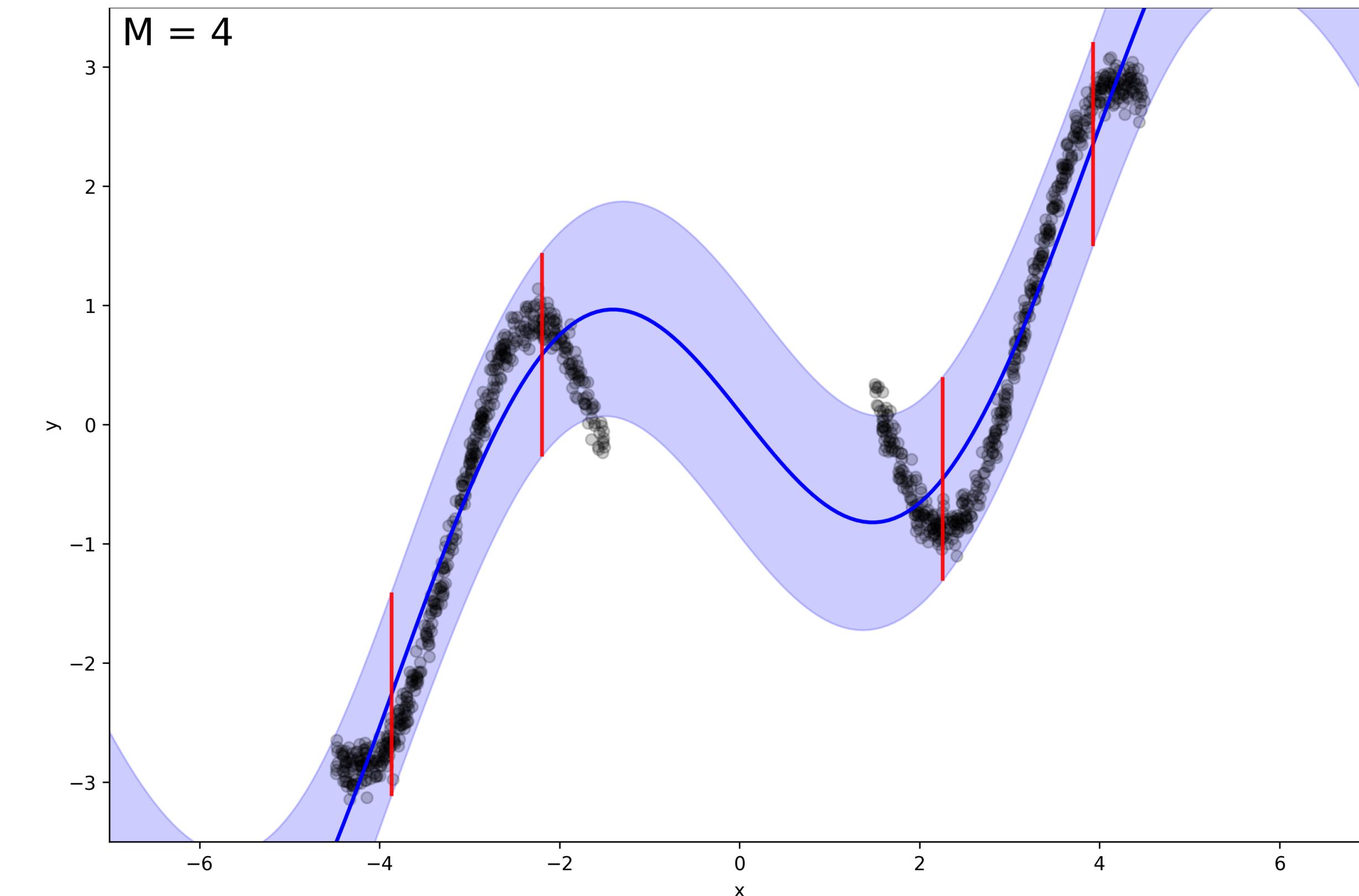
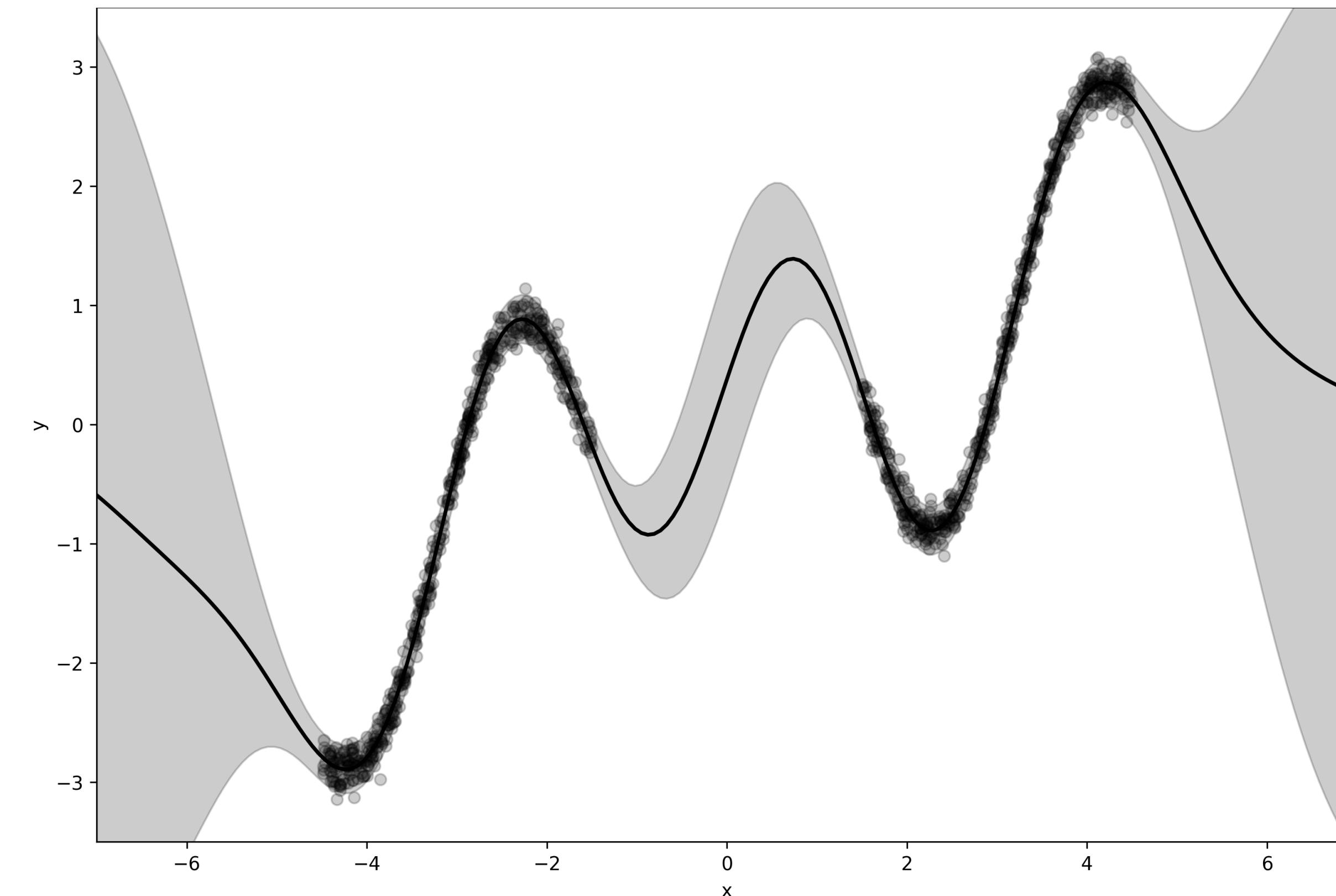
How many pseudo-data points is enough?



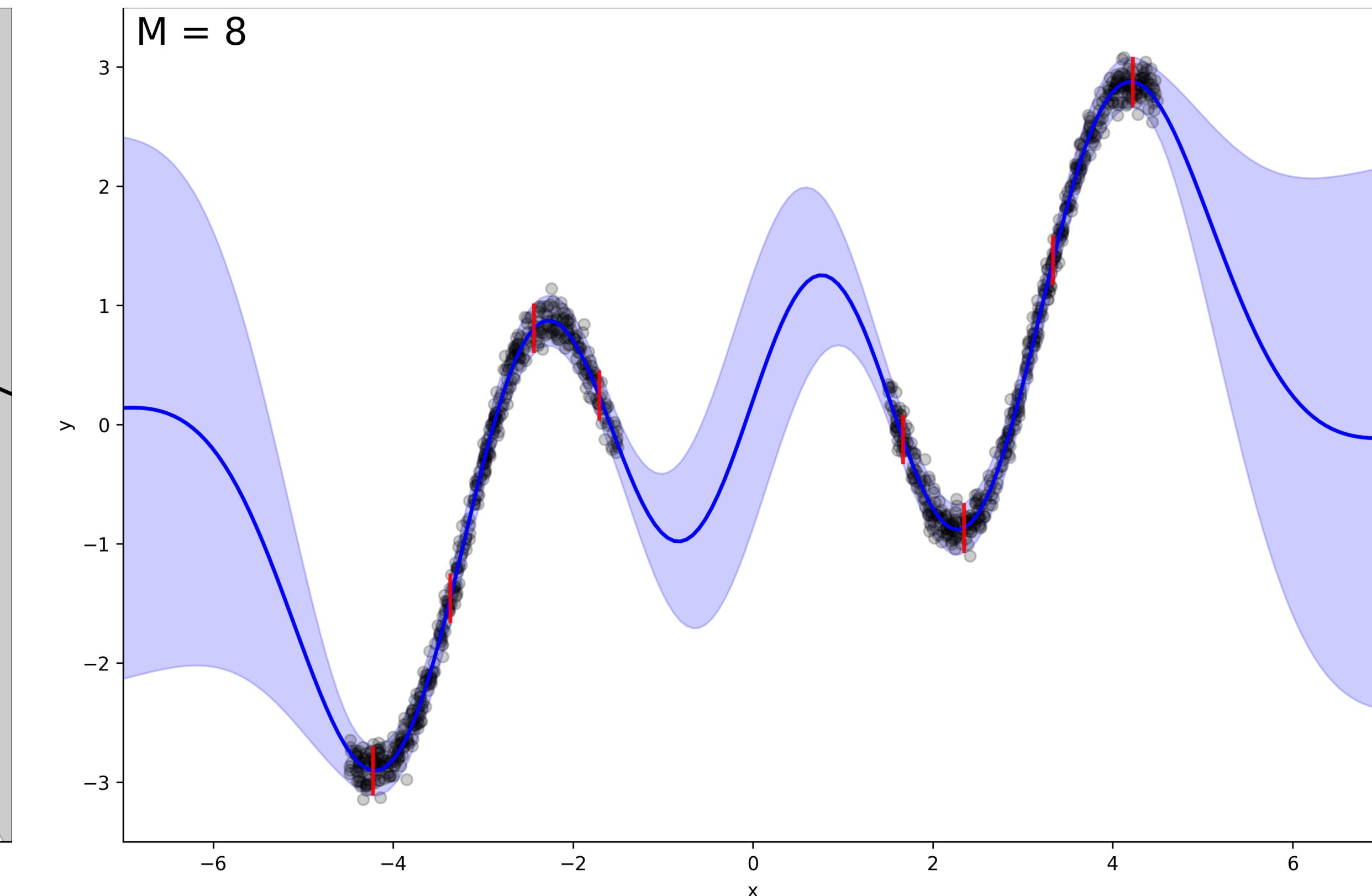
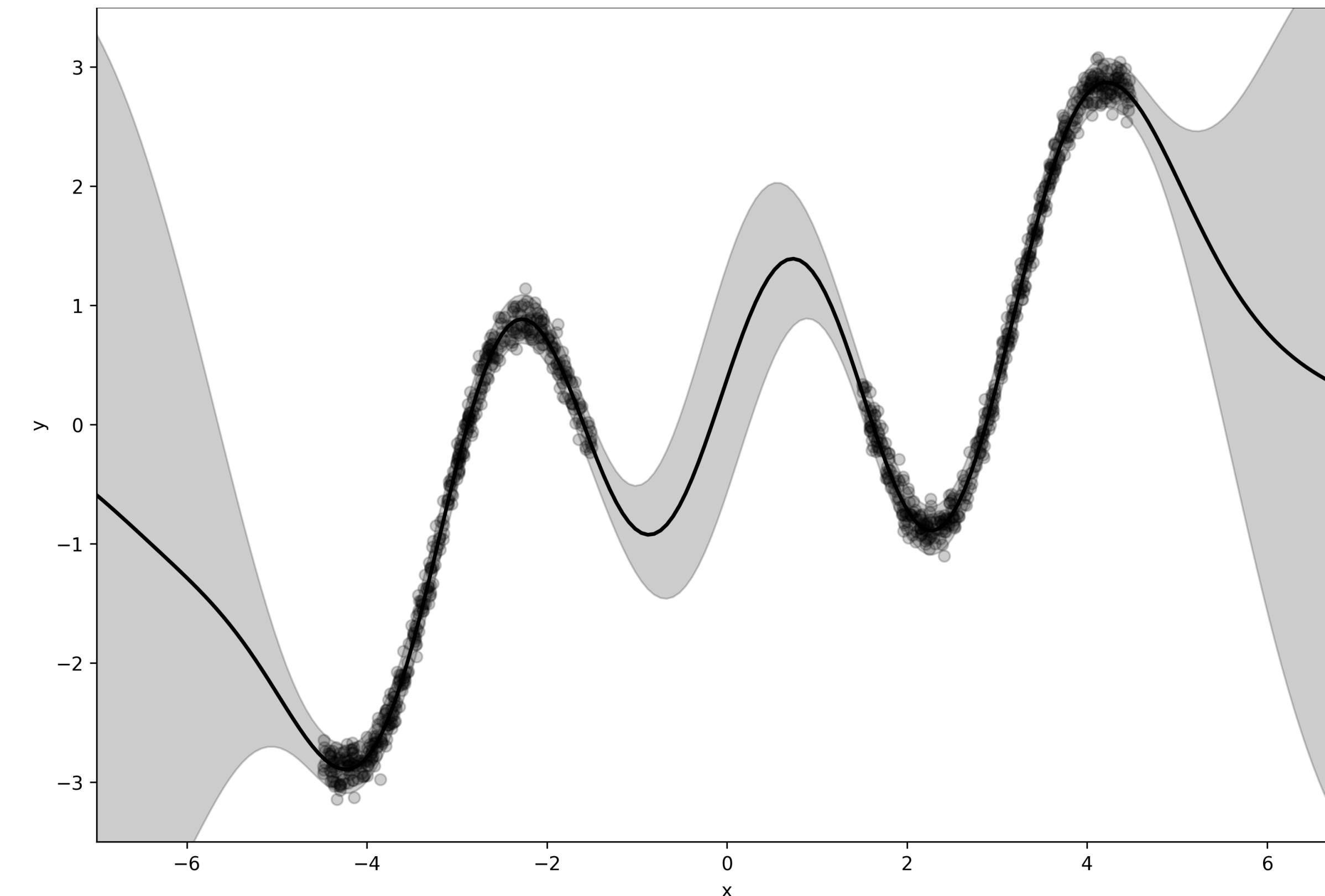
How many pseudo-data points is enough?



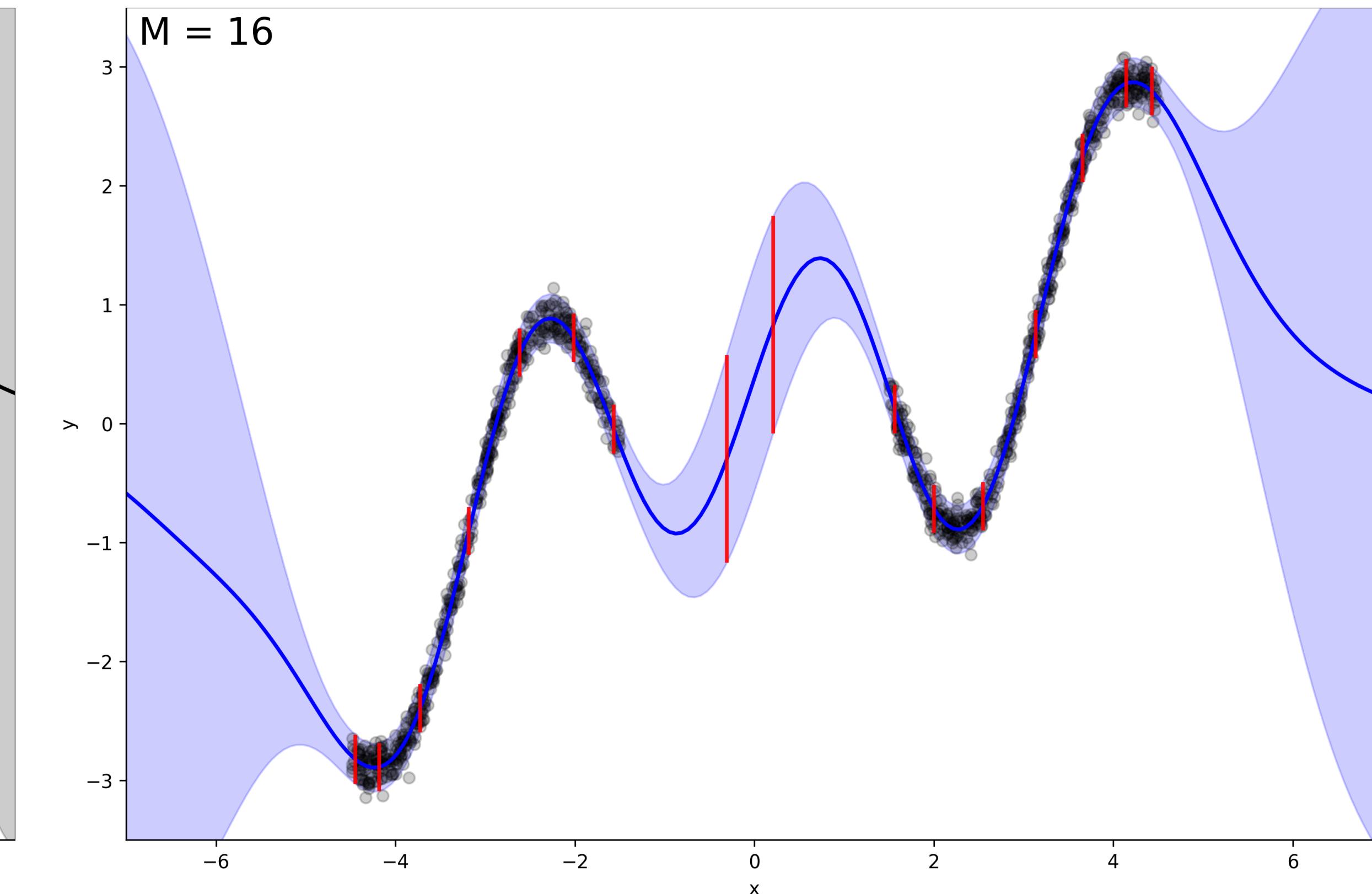
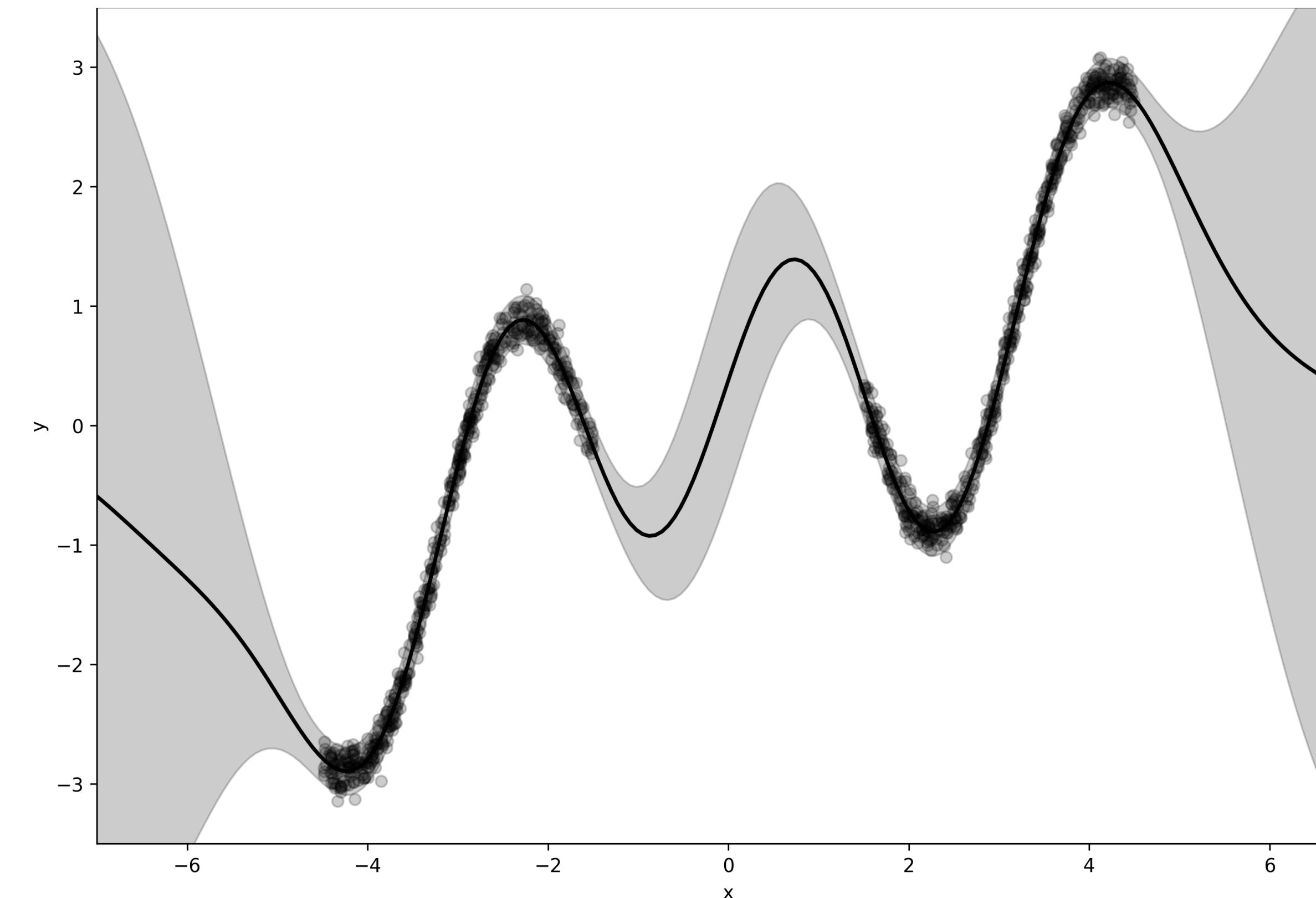
How many pseudo-data points is enough?



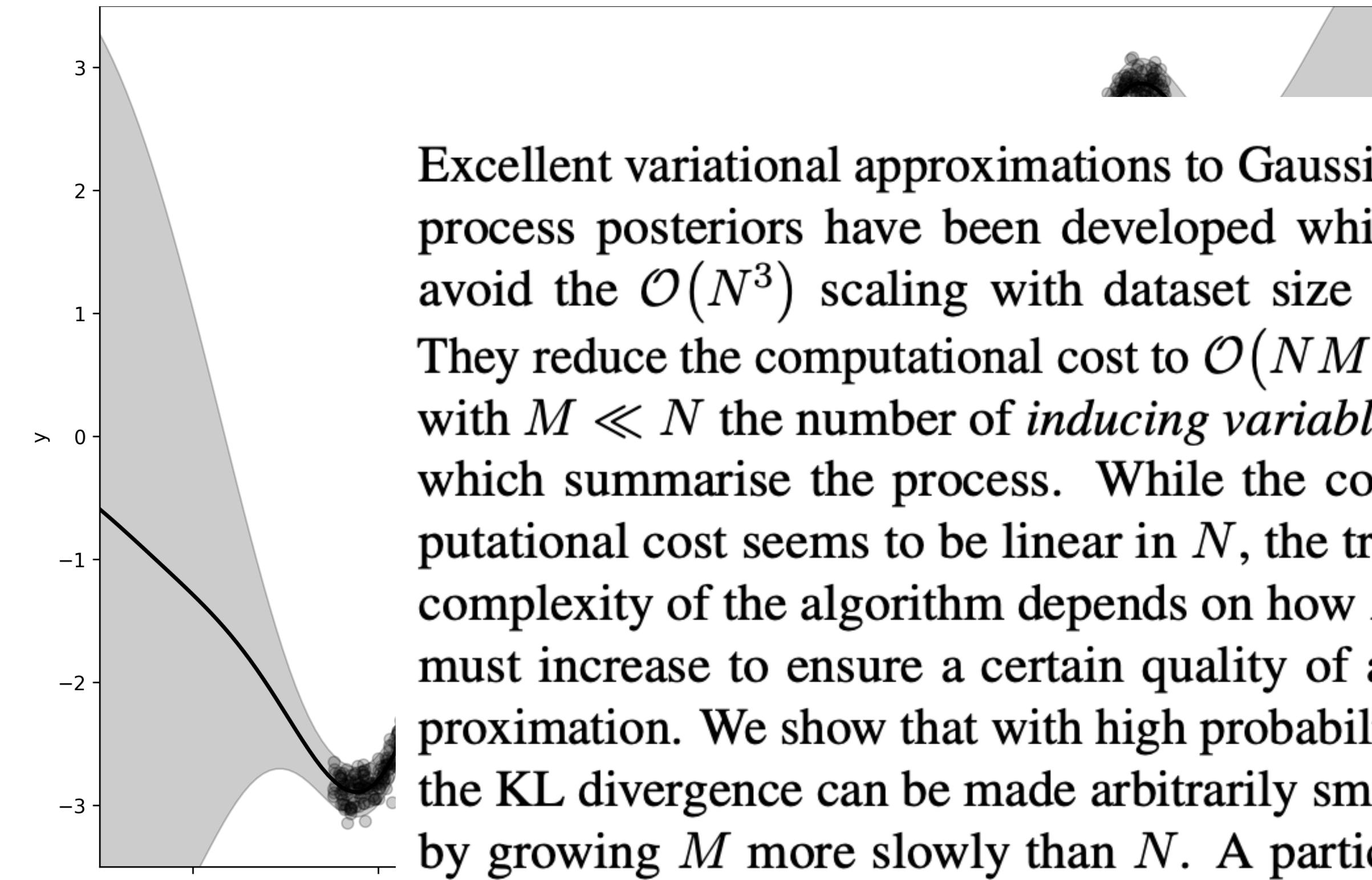
How many pseudo-data points is enough?



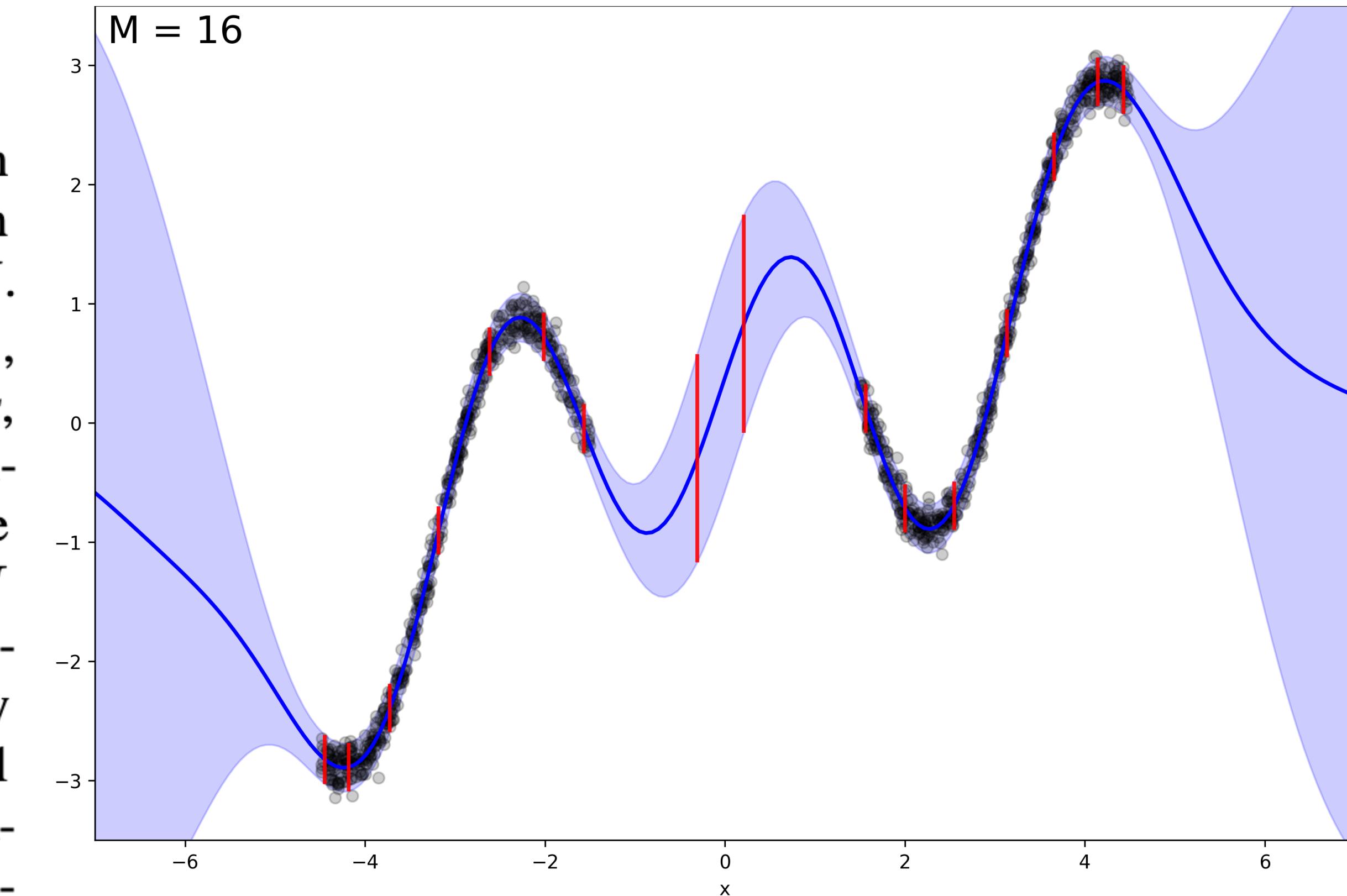
How many pseudo-data points is enough?



How many pseudo-data points is enough?



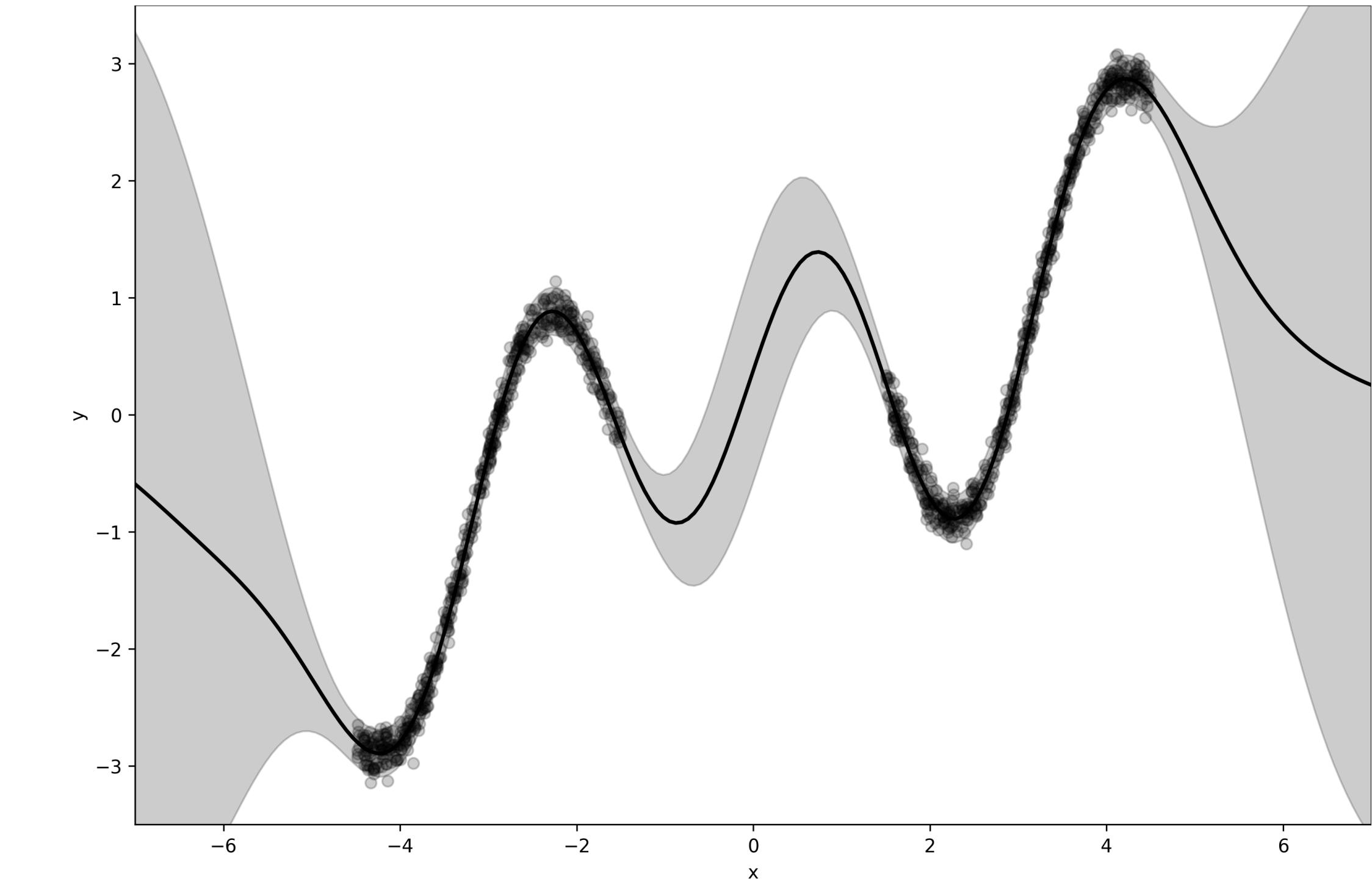
Our results show that as datasets grow, Gaussian process posteriors can be approximated cheaply, and provide a concrete rule for how to increase M in continual learning scenarios.



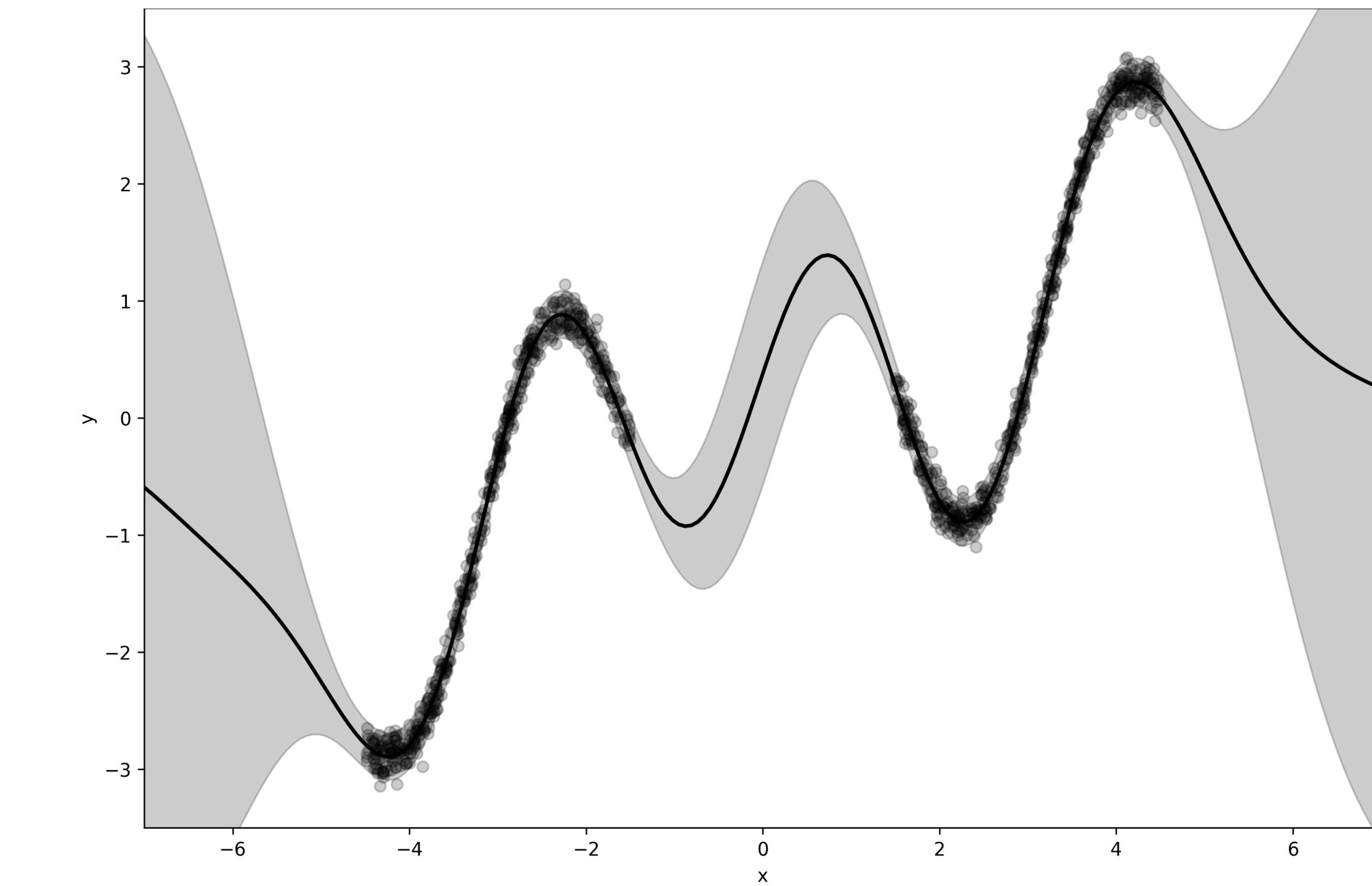
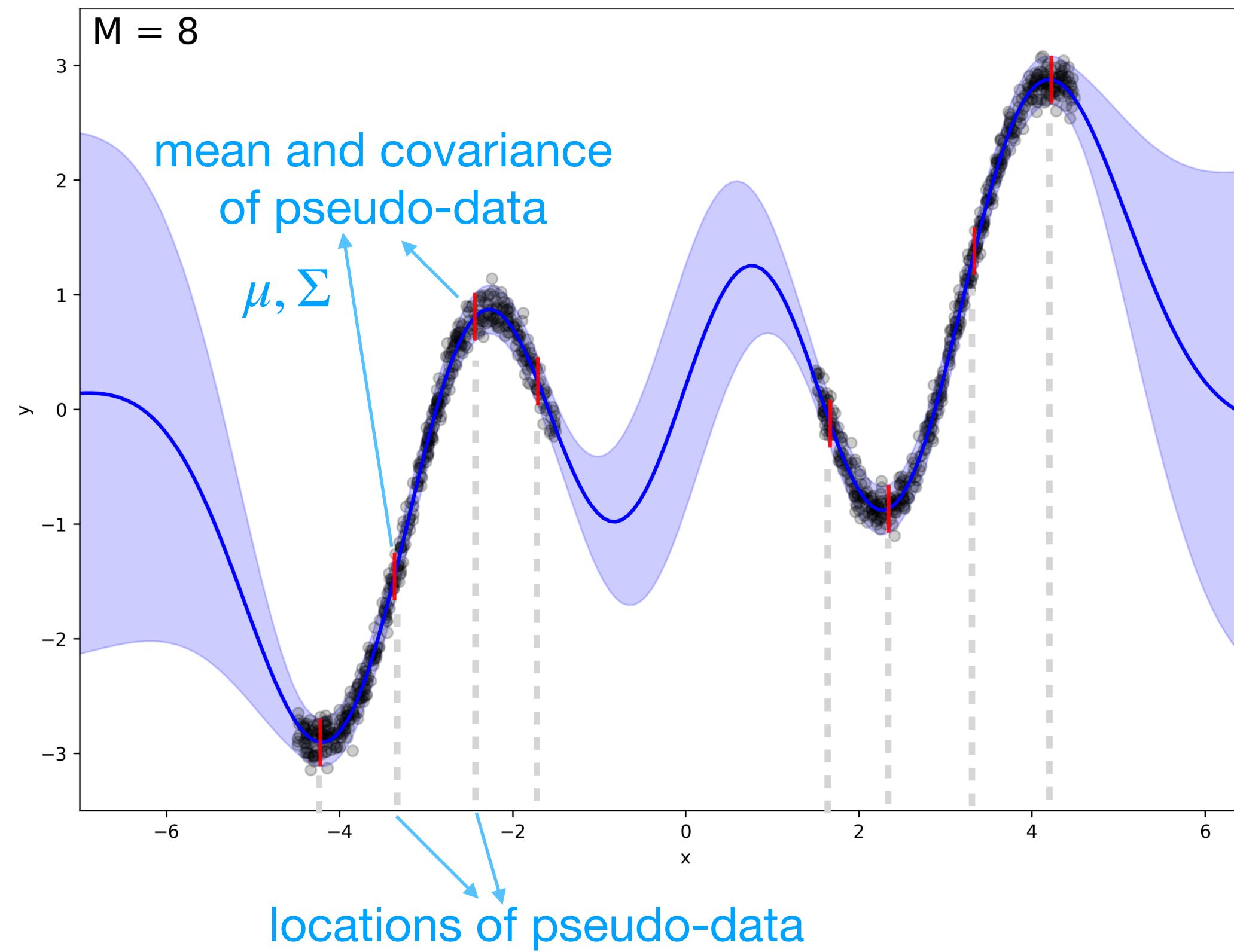
Rates of Convergence for Sparse Variational Gaussian Process Regression
Burt, Rasmussen and van der Wilk
ICML 2019

Research frontier 1: Power Expectation Propagation

Research frontier 1: Power Expectation Propagation

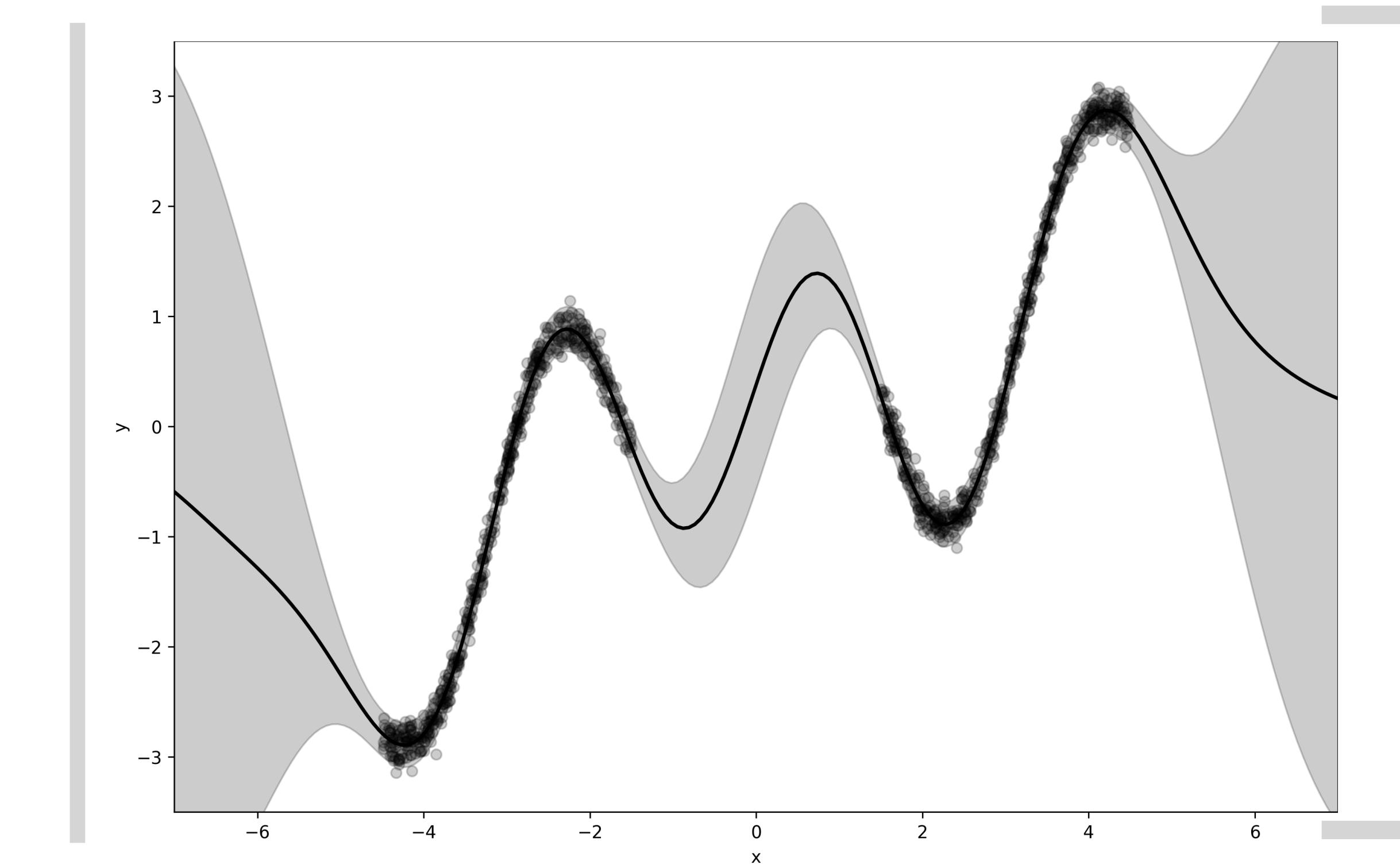
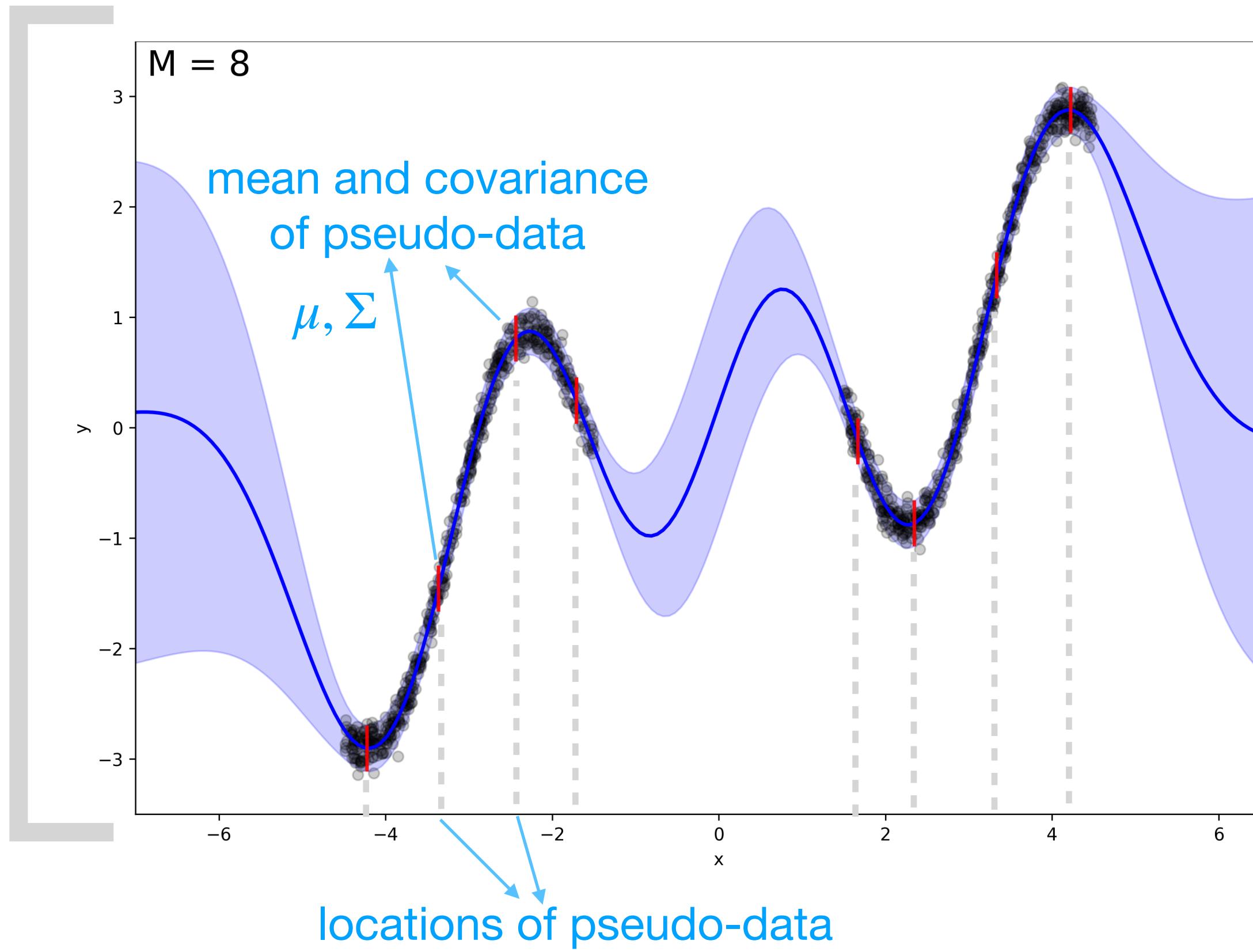


Research frontier 1: Power Expectation Propagation



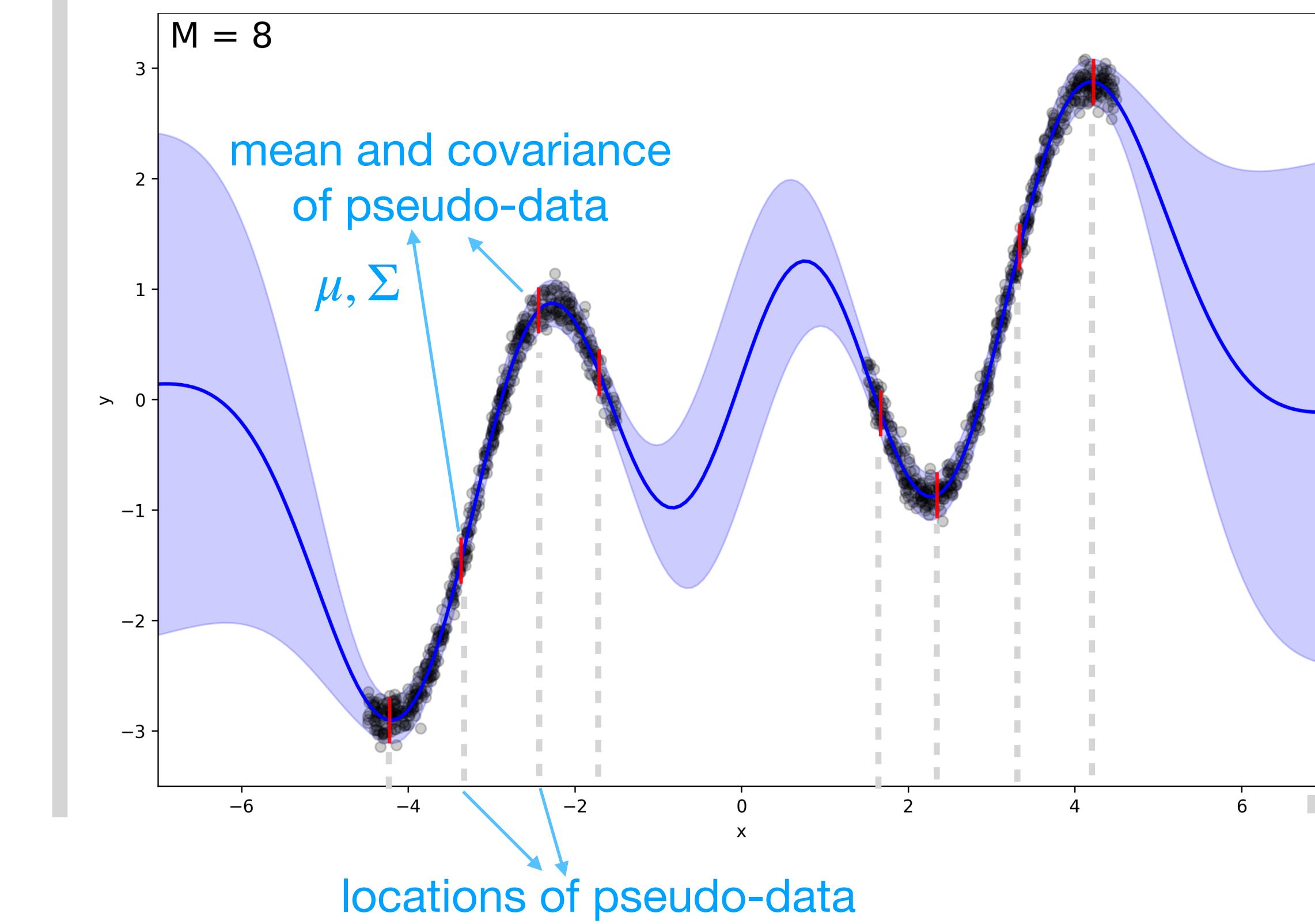
Research frontier 1: Power Expectation Propagation

KL
variational
inference



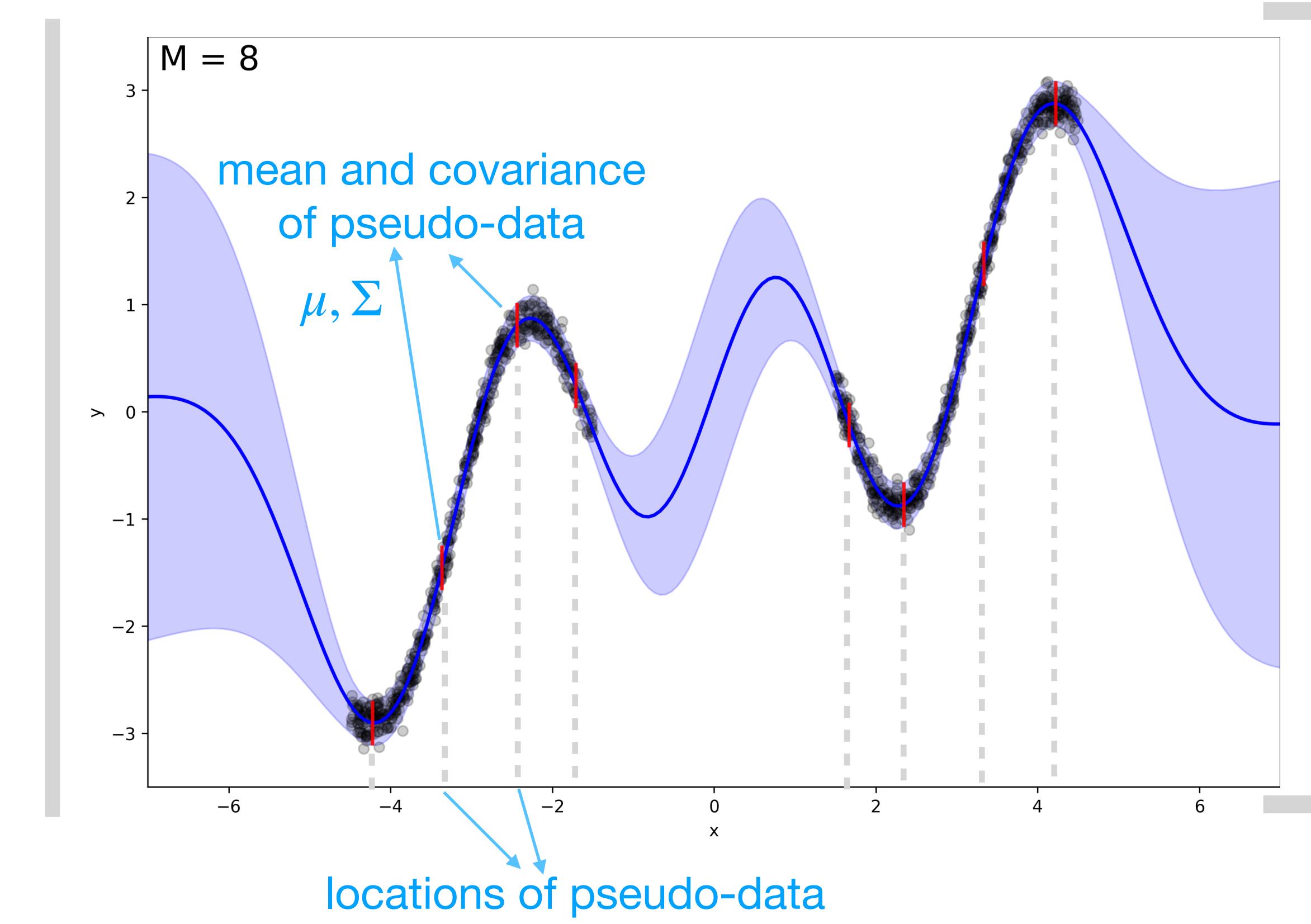
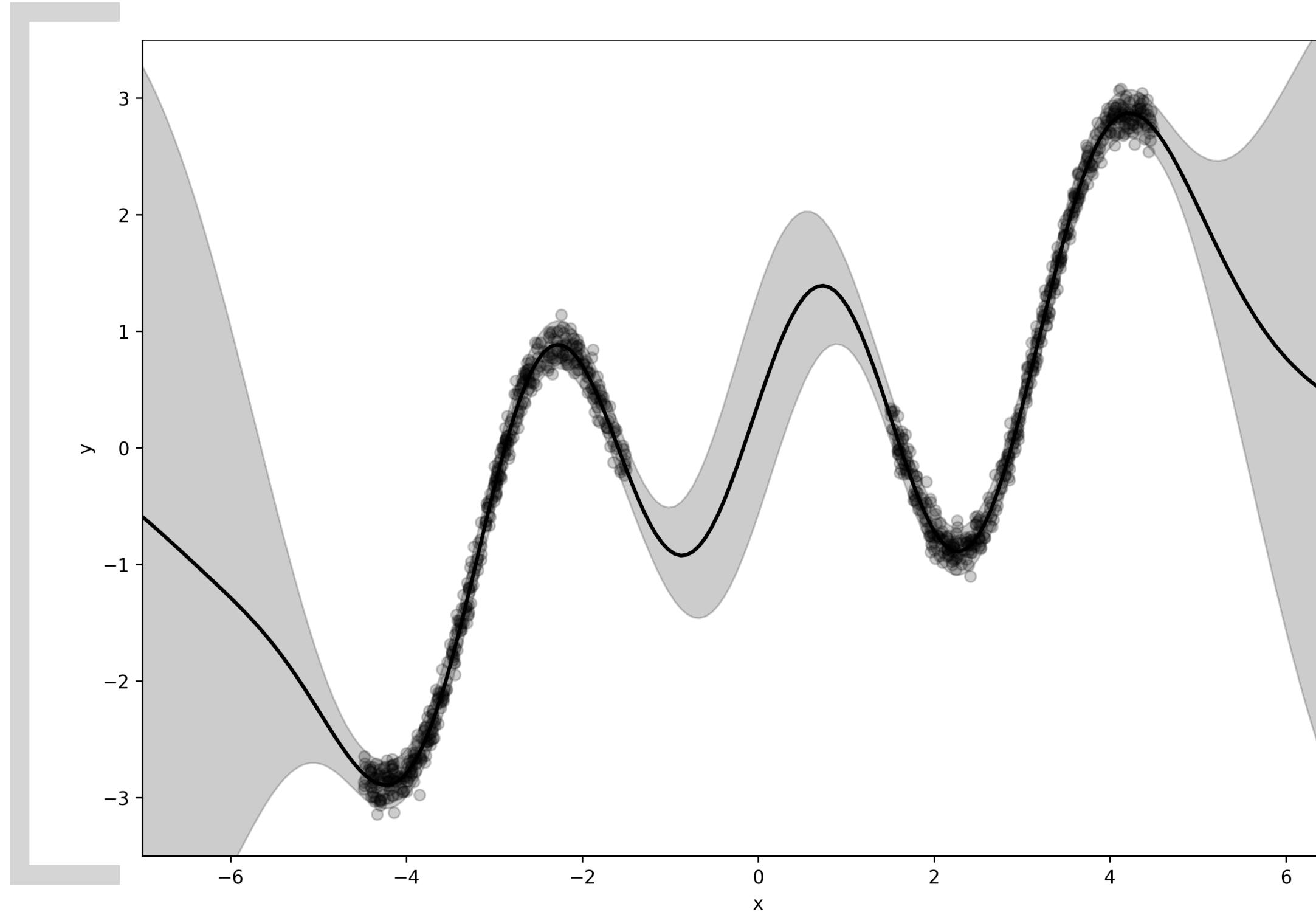
Research frontier 1: Power Expectation Propagation

KL
variational
inference

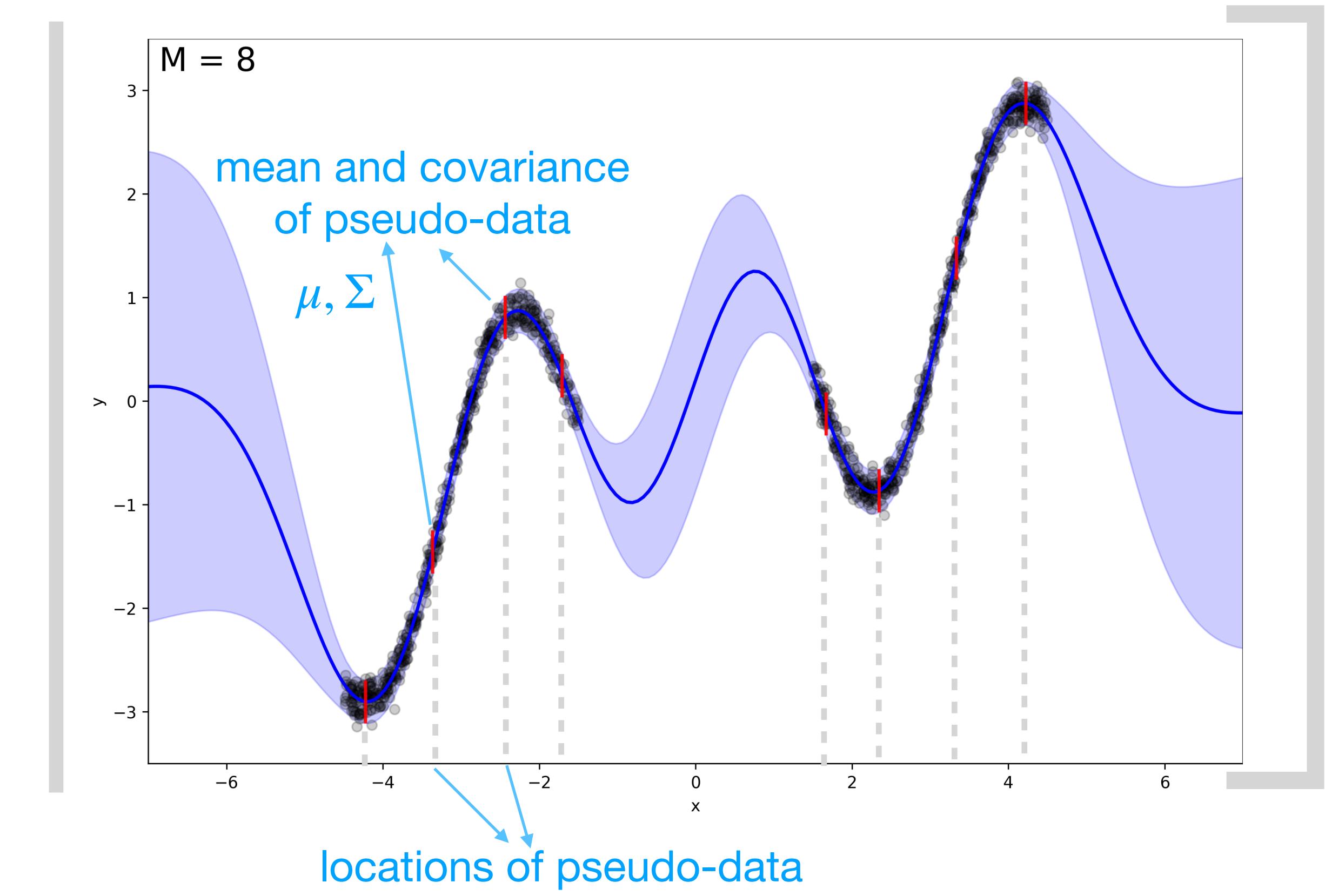
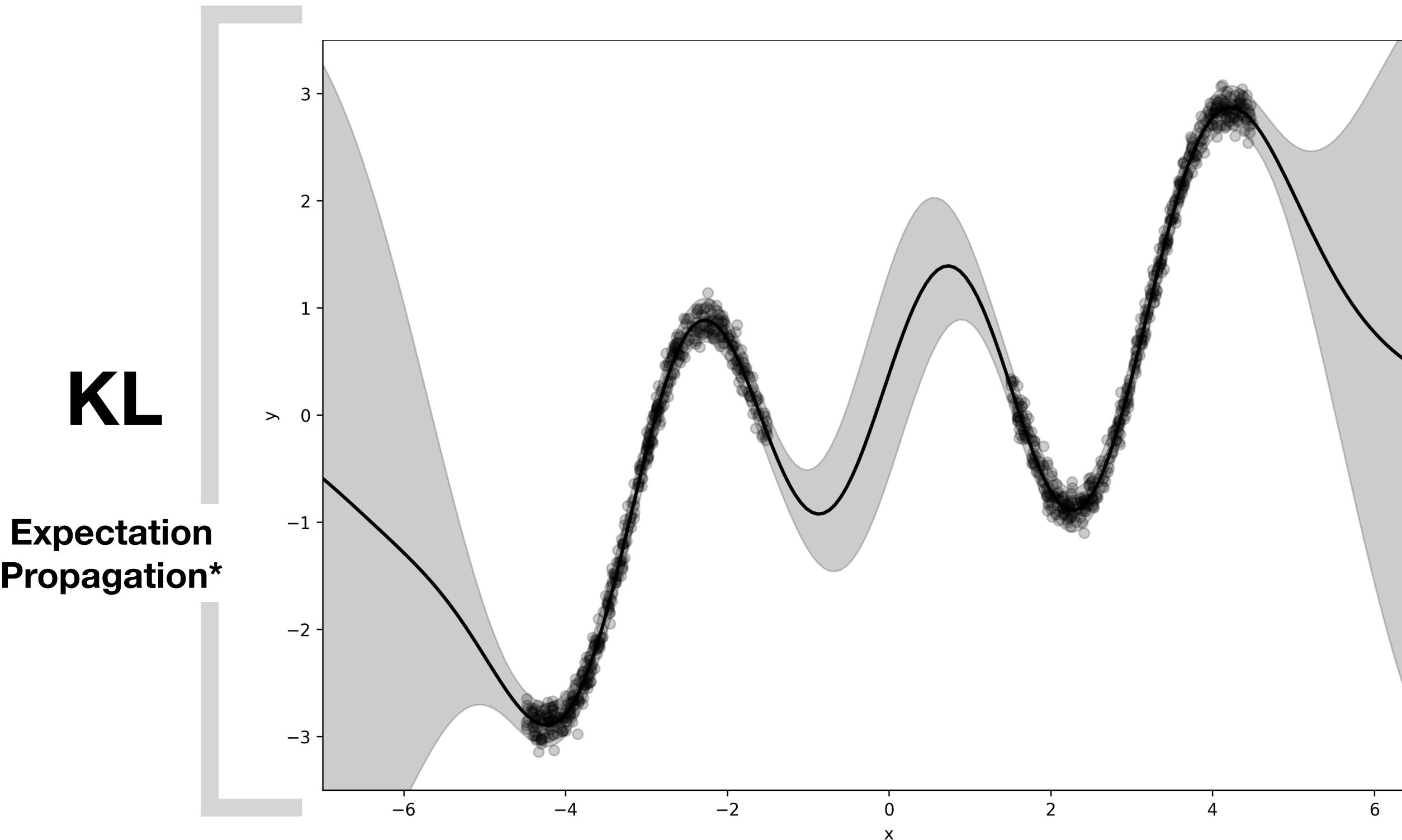


Research frontier 1: Power Expectation Propagation

KL
variational
inference

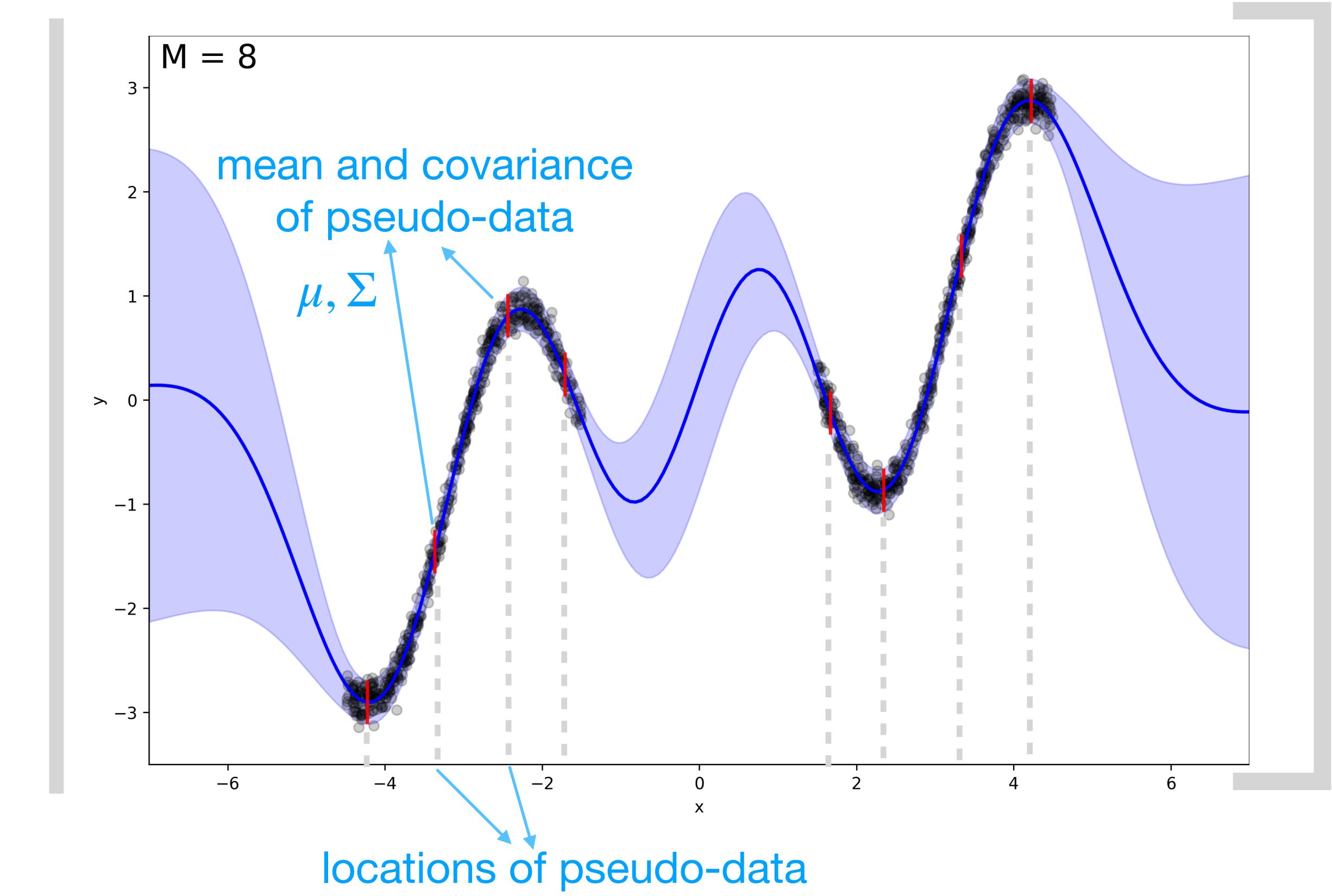
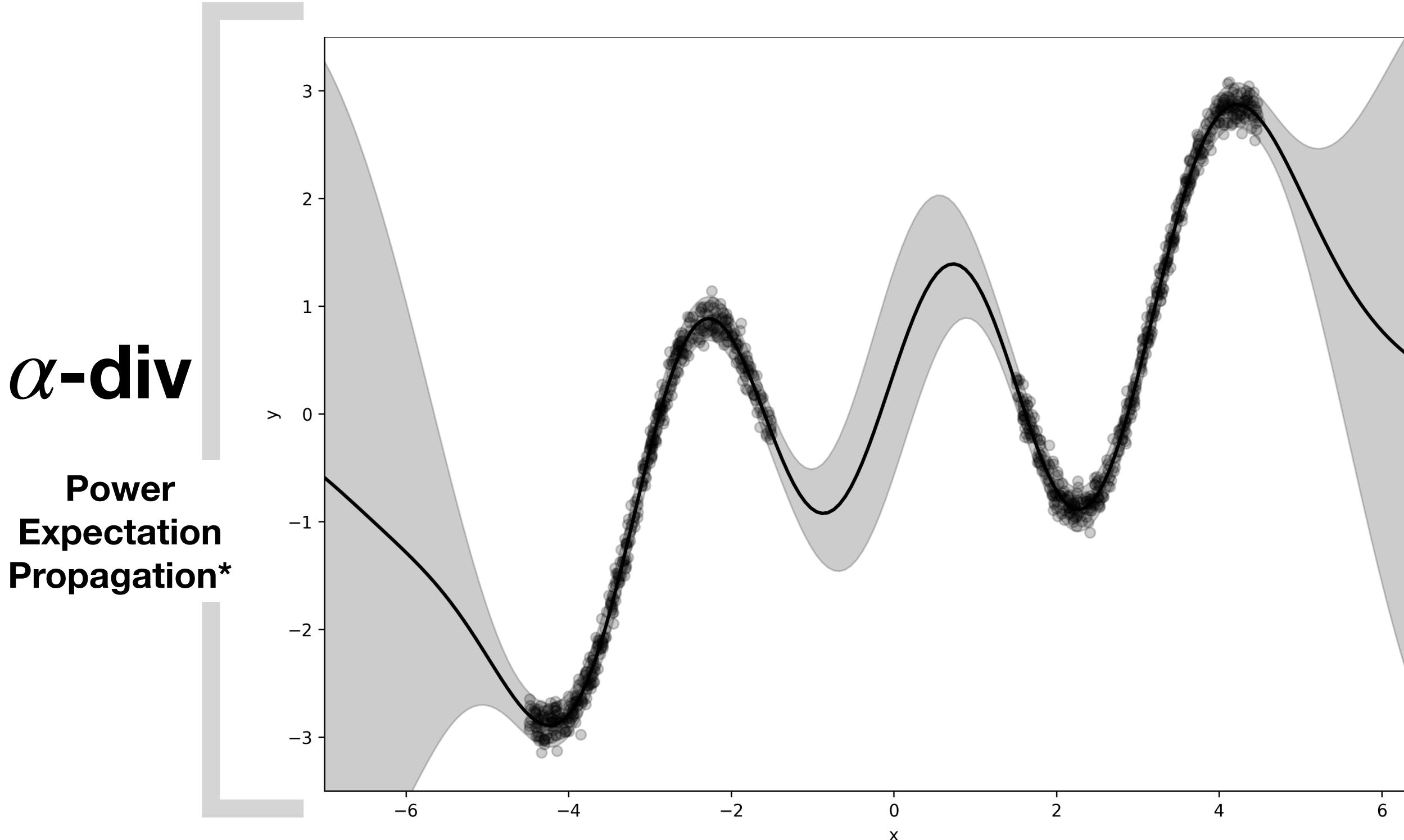


Research frontier 1: Power Expectation Propagation



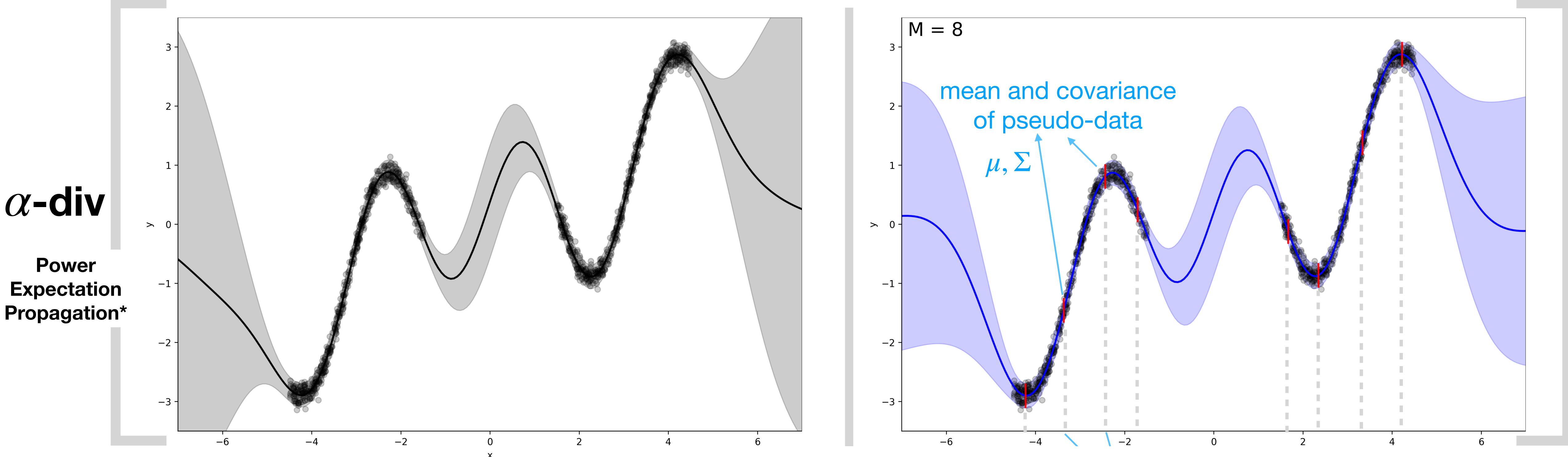
*Local KL or alpha-div, only deal with one likelihood factor at a time

Research frontier 1: Power Expectation Propagation

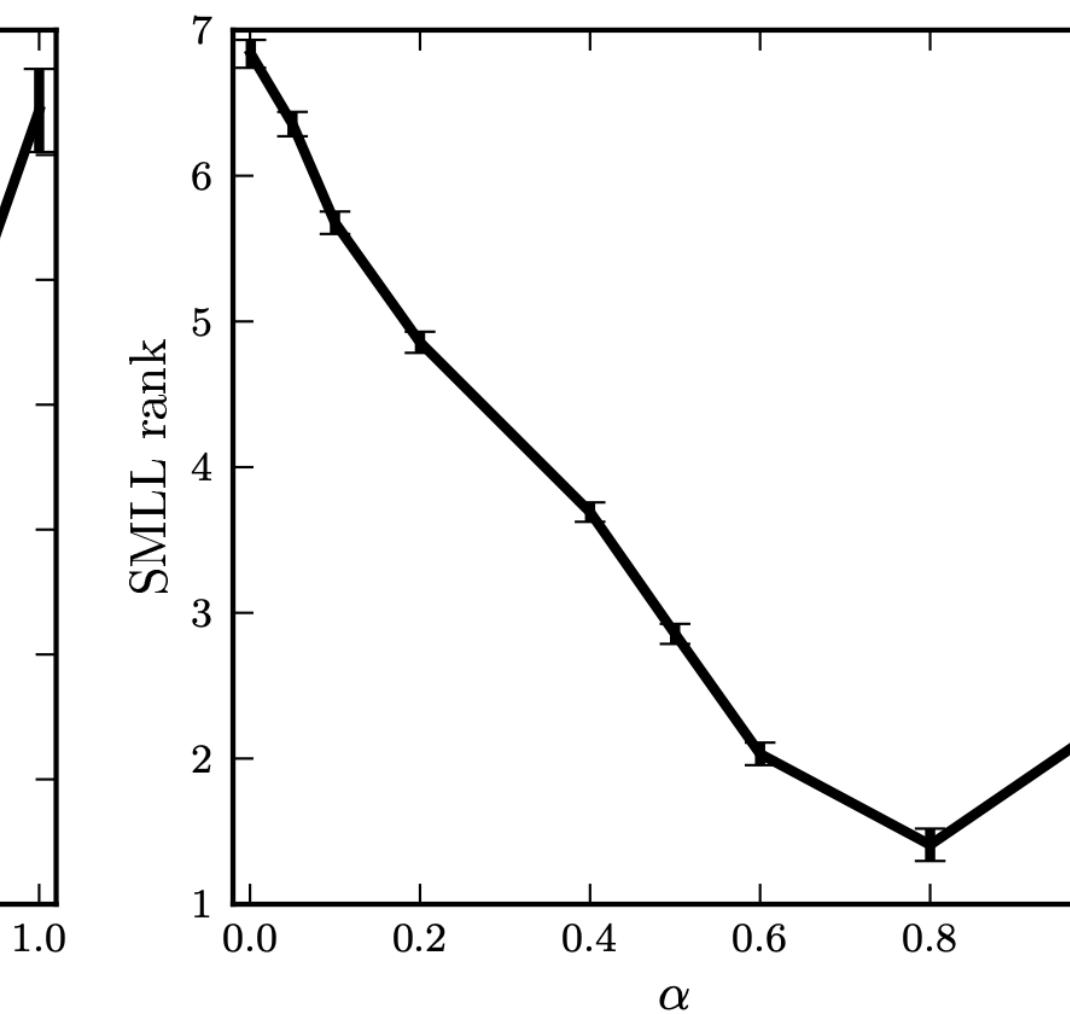
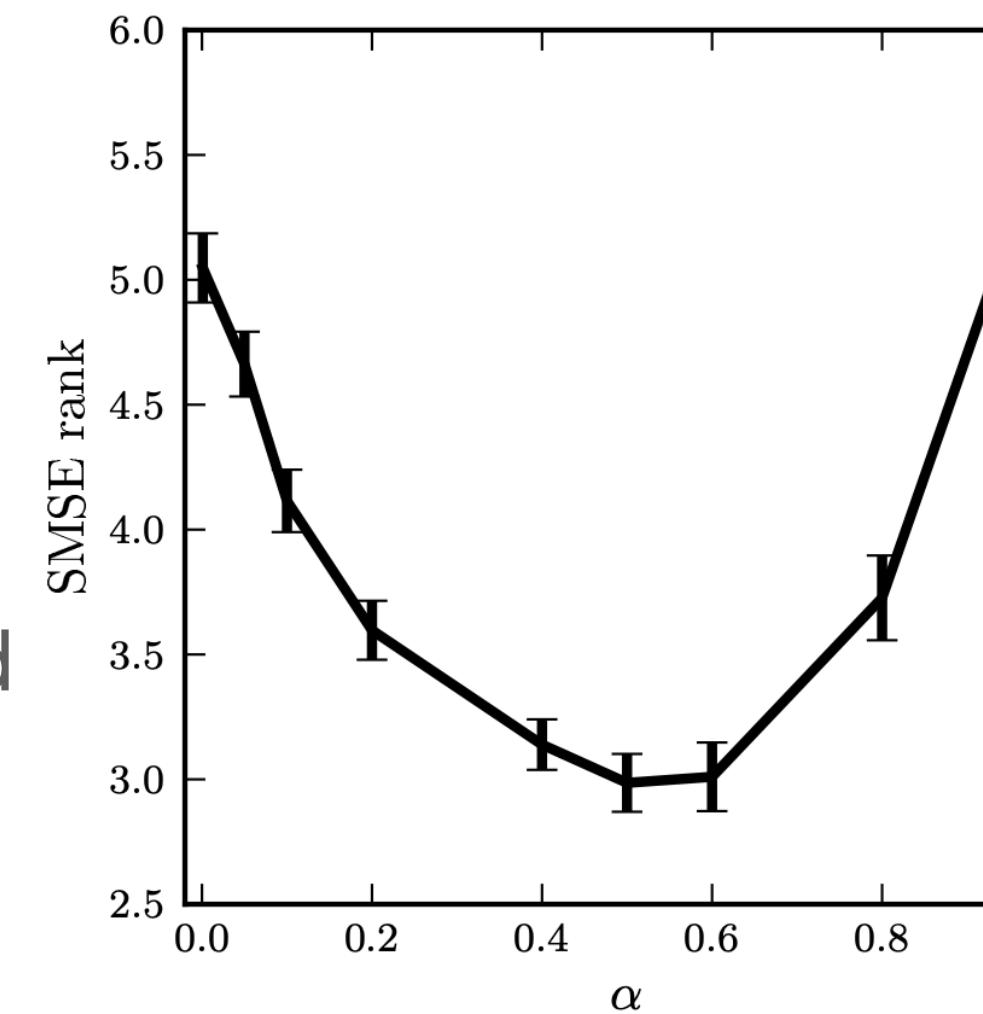


*Local KL or alpha-div, only deal with one likelihood factor at a time

Research frontier 1: Power Expectation Propagation



8 UCI regression datasets
20 random splits
 $M = 8 - 200$
Pseudo inputs and hypers optimised



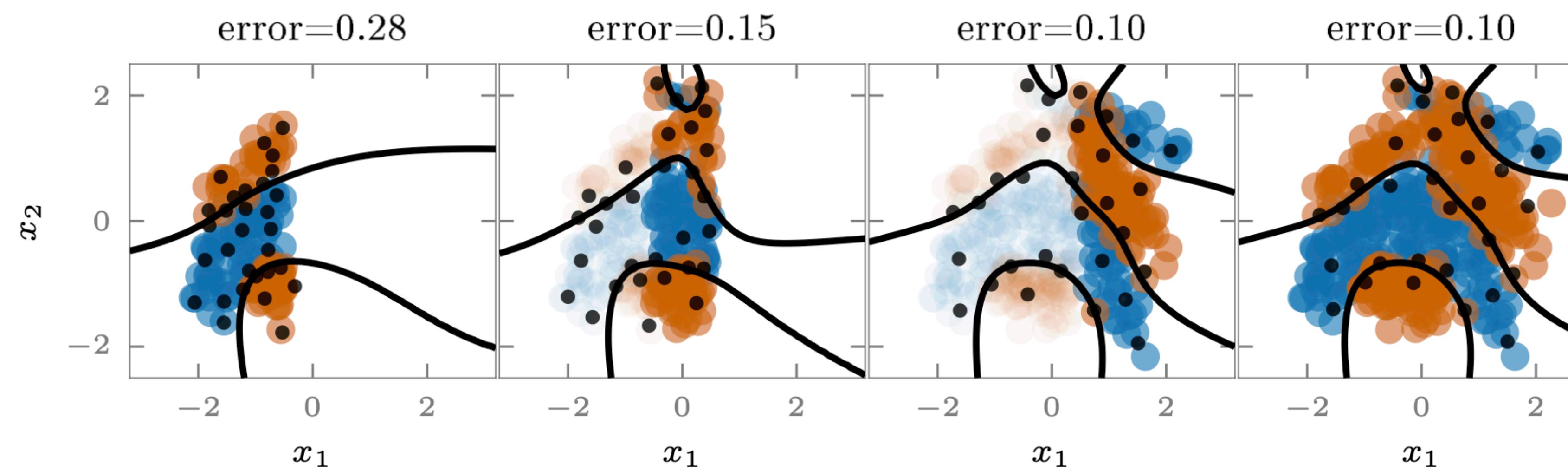
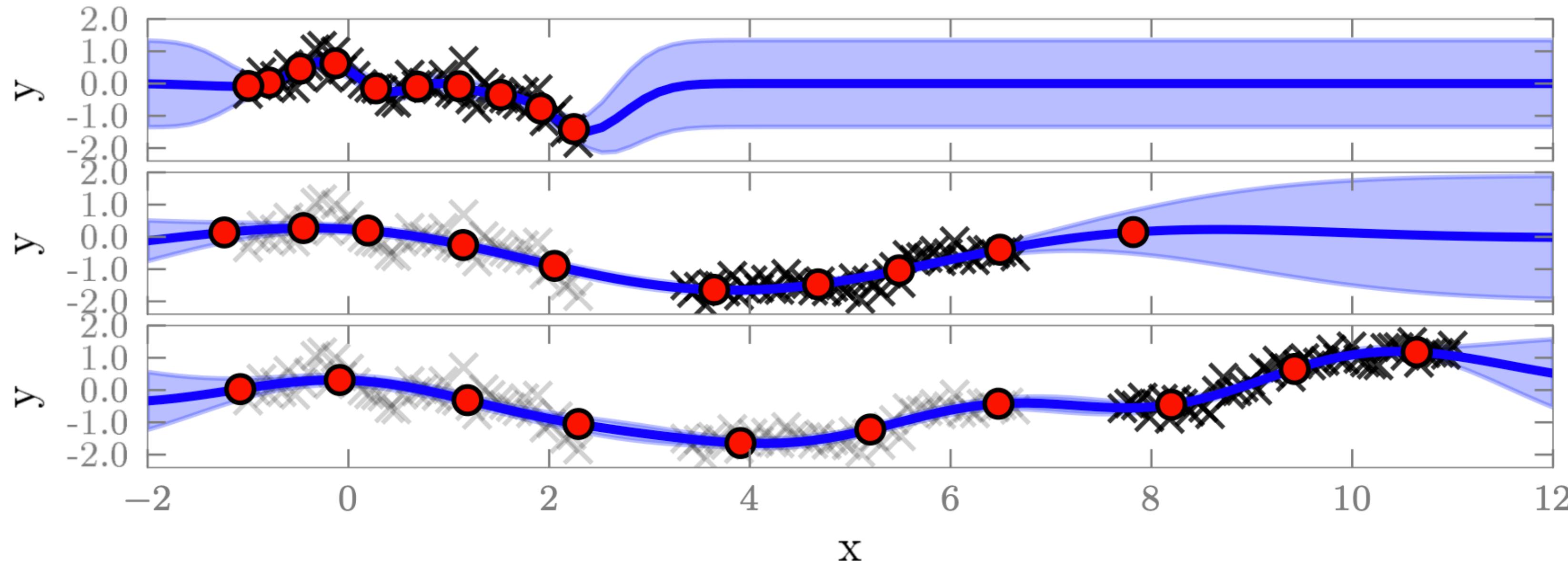
alpha = 0.5 performs
well on average

*Local KL or alpha-div, only deal with one likelihood factor at a time

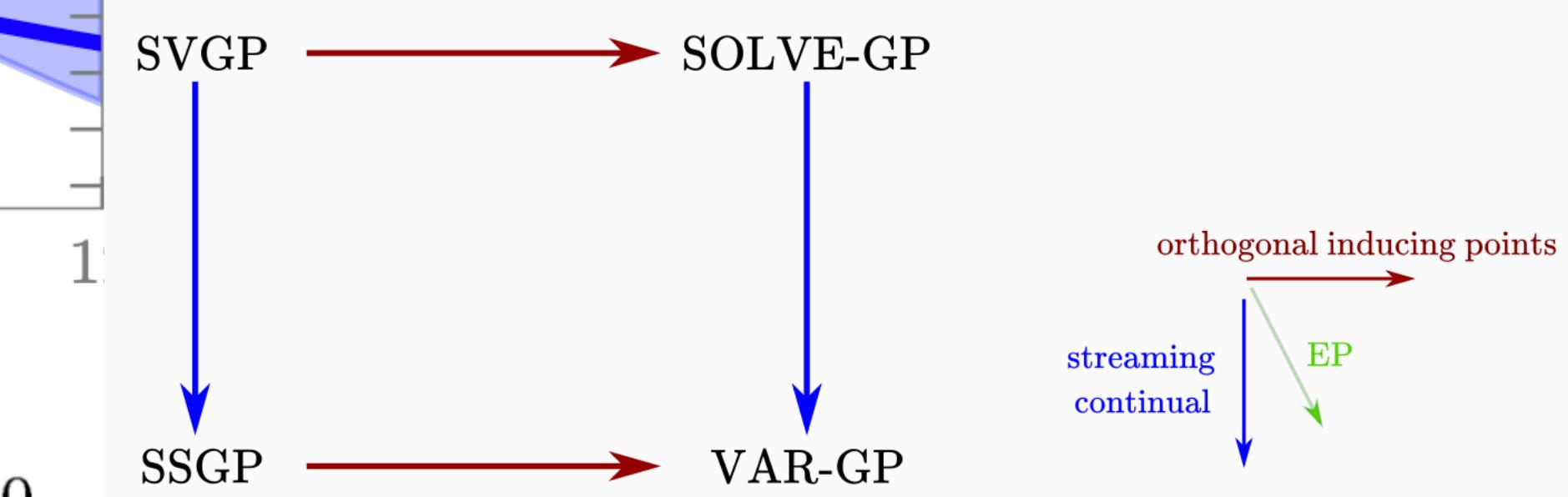
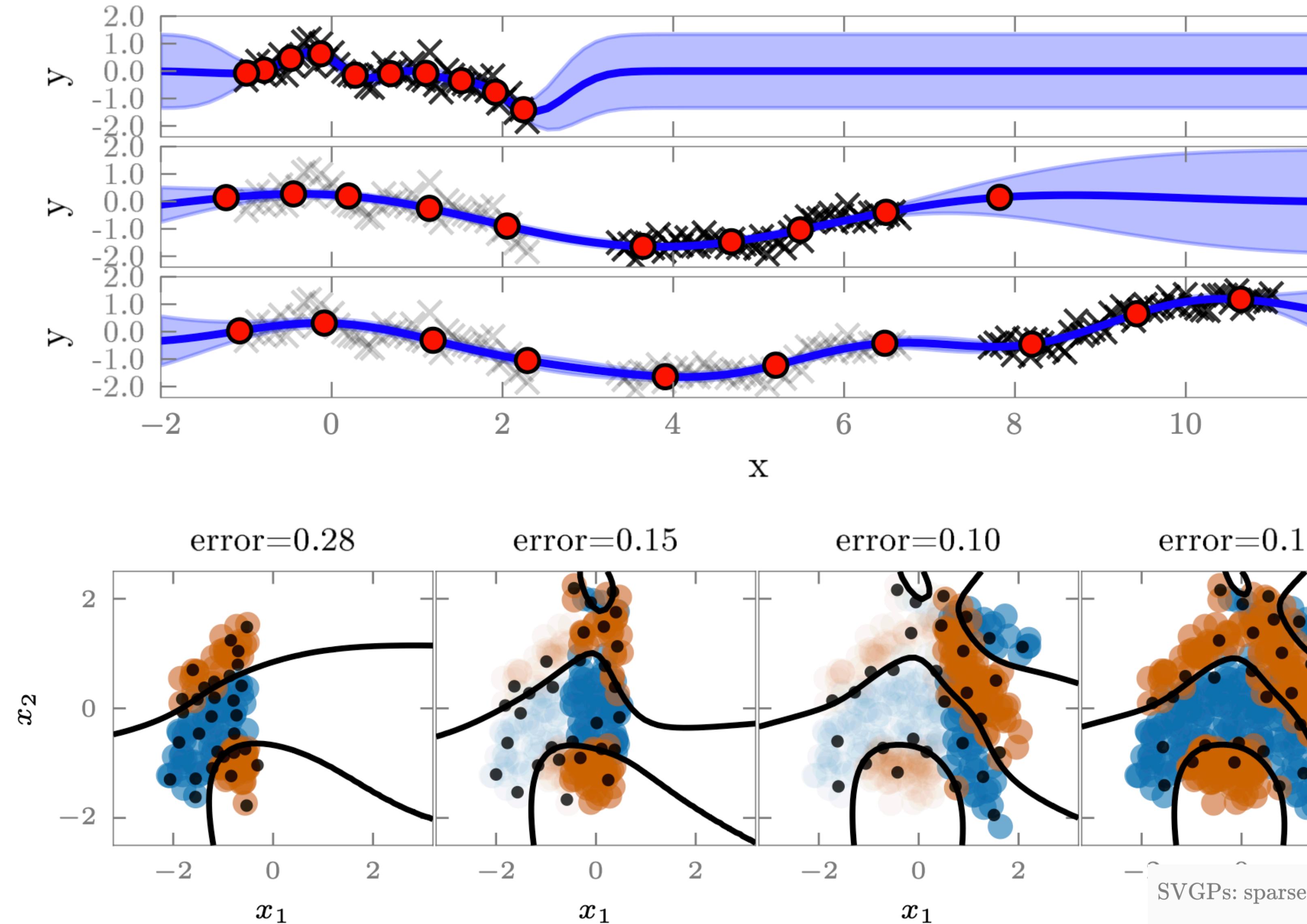
Research frontier 2: Orthogonal inducing points

	Sparse GP	Orthogonal Sparse GP
Inducing points	(\mathbf{z}, \mathbf{u})	$(\mathbf{z}_u, \mathbf{u}), (\mathbf{z}_v, \mathbf{v})$
Prior factorisation	$p(f) = p(f_{\neq u} \mathbf{u})p(\mathbf{u})$	$p(f) = p(f_{\neq u,v} \mathbf{u}, \mathbf{v})p(\mathbf{u}, \mathbf{v})$
	$p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{K}_{uu})$	$p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{K}_{uu})$
		$p(\mathbf{v} \mathbf{u}) = \mathcal{N}(\mathbf{v}; \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1}\mathbf{u}, \mathbf{K}_{vv} - \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uv})$
Approx. posterior	$q(f) = p(f_{\neq u} \mathbf{u})q(\mathbf{u})$	$q(f) = p(f_{\neq u,v} \mathbf{u}, \mathbf{v})q(\mathbf{u})q(\mathbf{v} \mathbf{u})$
	$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \Sigma)$	$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}_u, \Sigma_u)$
		$q(\mathbf{v} \mathbf{u}) = \mathcal{N}(\mathbf{v}; \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1}\mathbf{u} + \mathbf{m}_v, \Sigma_v)$
 This might seem poor compared to $q(\mathbf{u}, \mathbf{v}) = \mathcal{N}([\mathbf{u}, \mathbf{v}]^\top; \mathbf{m}_{uv}, \Sigma_{uv})$ but is more efficient		

Research frontier 3: Continual learning



Research frontier 3: Continual learning



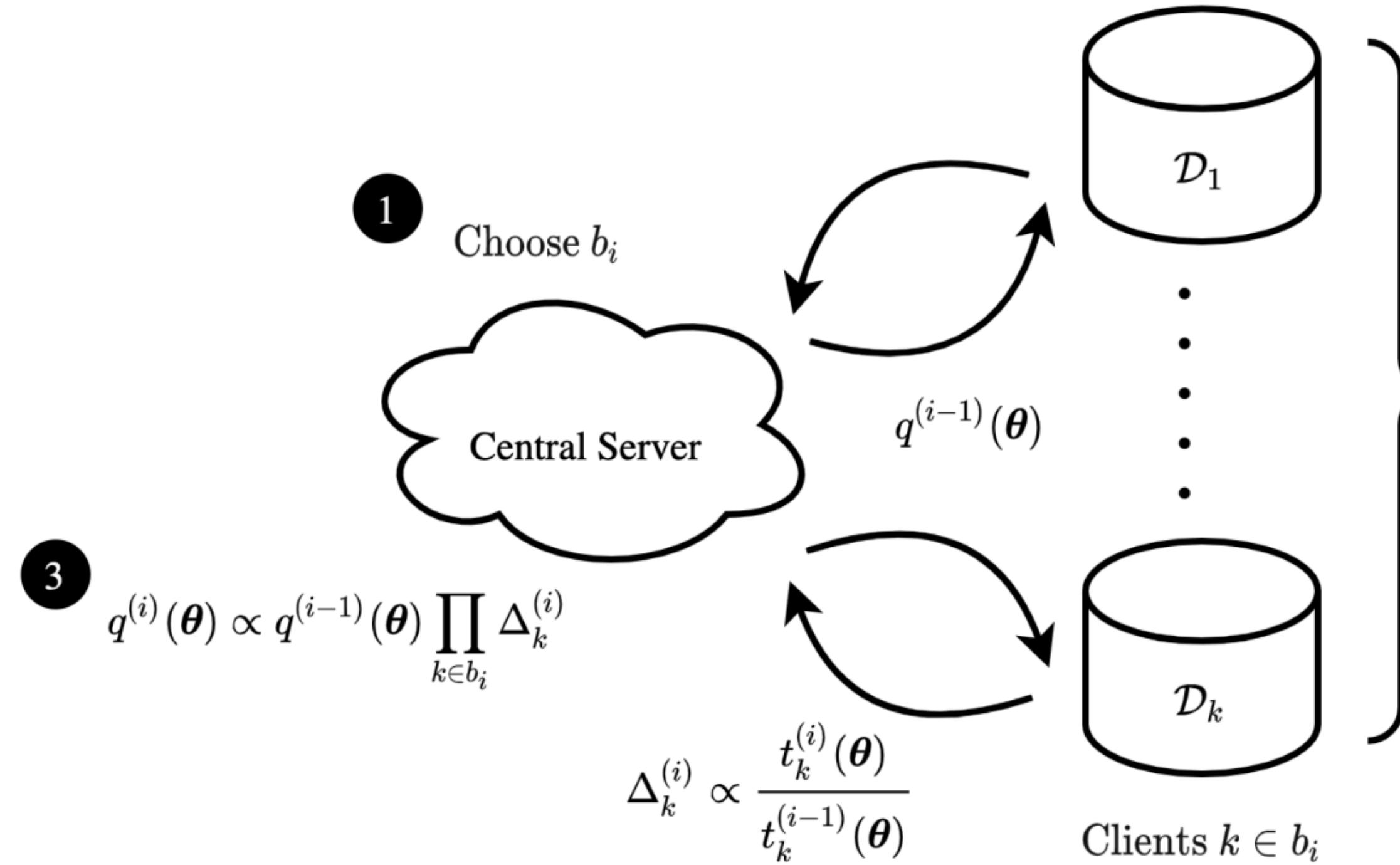
SVGPs: sparse variational inference for GPs (Titsias, 2009; Hensman et al, 2013)

SOLVE-GPs: sparse orthogonal variational inference for GPs (Shi et al, 2020)

SSGP: streaming sparse variational inference for GPs (B.*, Nguyen*, and Turner, 2017)

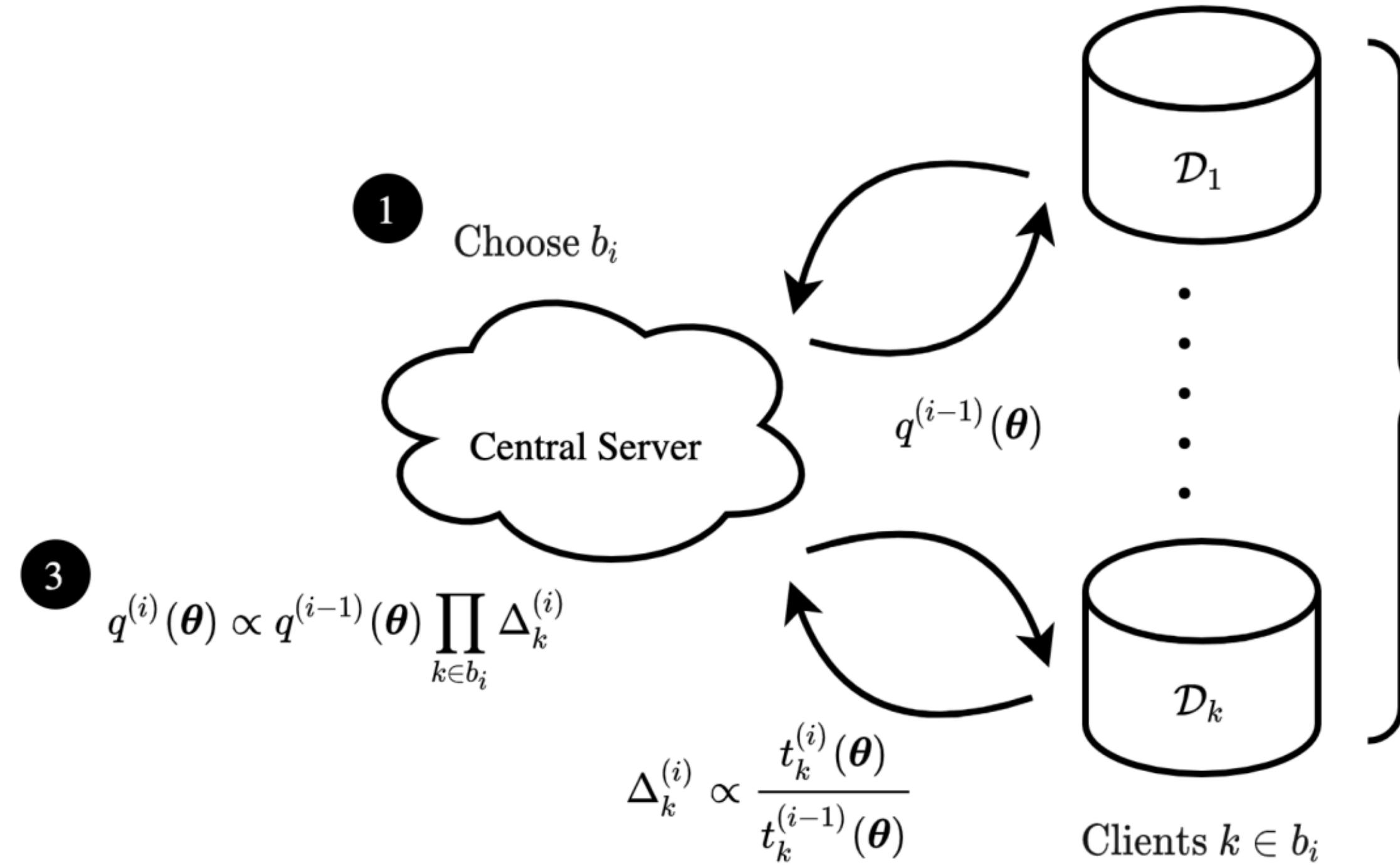
VAR-GP: variational autoregressive GPs (Kapoor, Karaletsos and B., 2020)

Research frontier 4: Federated learning



$$q_k^{(i)}(\boldsymbol{\theta}) = \arg \max_{q(\boldsymbol{\theta})} \int q(\boldsymbol{\theta}) \log \frac{q^{(i-1)}(\boldsymbol{\theta}) p(\mathbf{y}_k | \boldsymbol{\theta})}{q(\boldsymbol{\theta}) t_k^{(i-1)}(\boldsymbol{\theta})} d\boldsymbol{\theta}$$
$$t_k^{(i)}(\boldsymbol{\theta}) \propto \frac{q_k^{(i)}(\boldsymbol{\theta})}{q^{(i-1)}(\boldsymbol{\theta})} t_k^{(i-1)}(\boldsymbol{\theta})$$

Research frontier 4: Federated learning



$$2 \quad q_k^{(i)}(\theta) = \arg \max_{q(\theta)} \int q(\theta) \log \frac{q^{(i-1)}(\theta)p(\mathbf{y}_k|\theta)}{q(\theta)t_k^{(i-1)}(\theta)} d\theta$$

$$t_k^{(i)}(\theta) \propto \frac{q_k^{(i)}(\theta)}{q^{(i-1)}(\theta)} t_k^{(i-1)}(\theta)$$

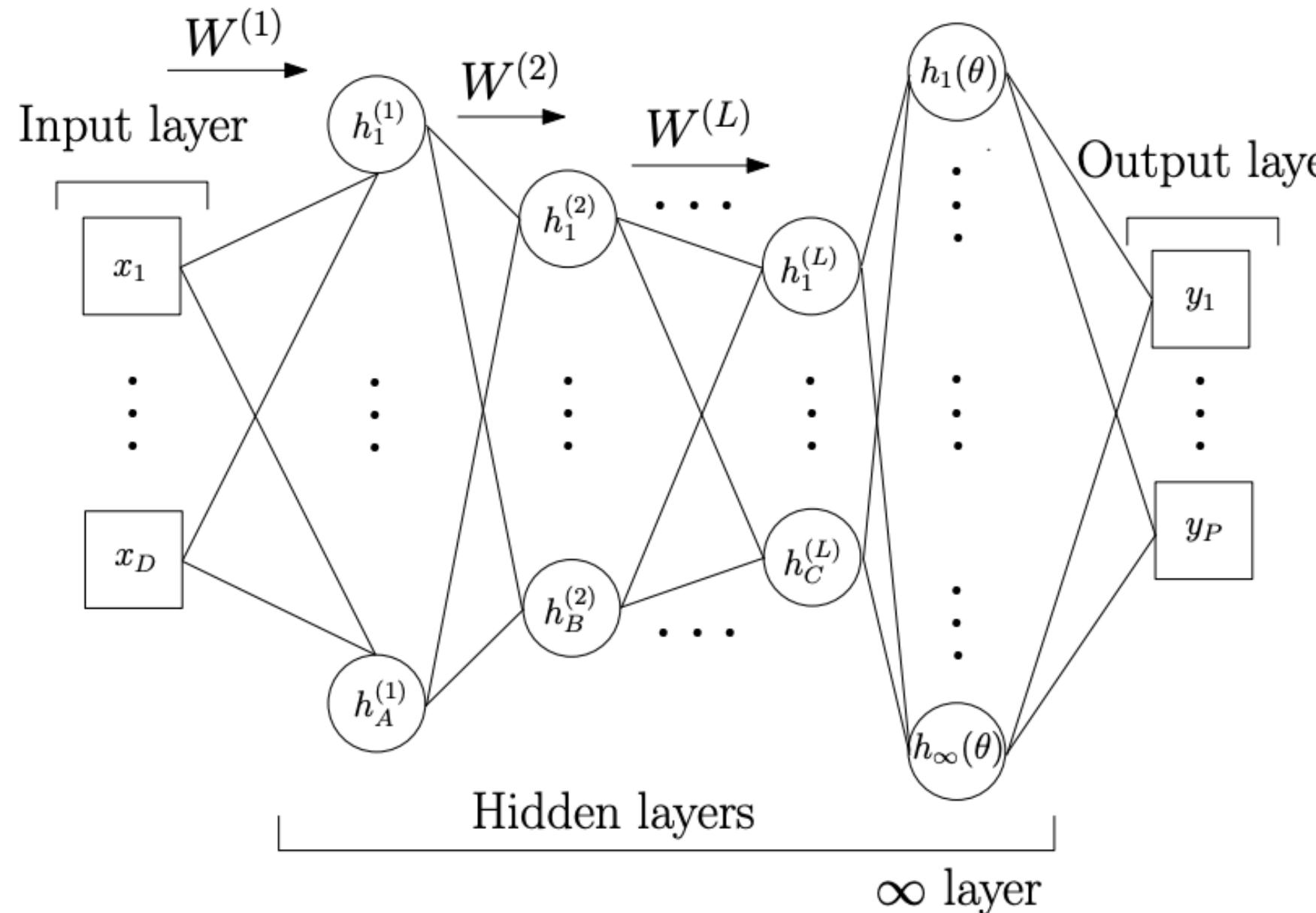
Federated posterior

$$p(\mathbf{y}, f, \theta) = p(\theta)p(f|\theta) \prod_{k=1}^K \prod_{n=1}^{N_k} p(y_{kn} | f(\mathbf{x}_{kn}), \theta)$$

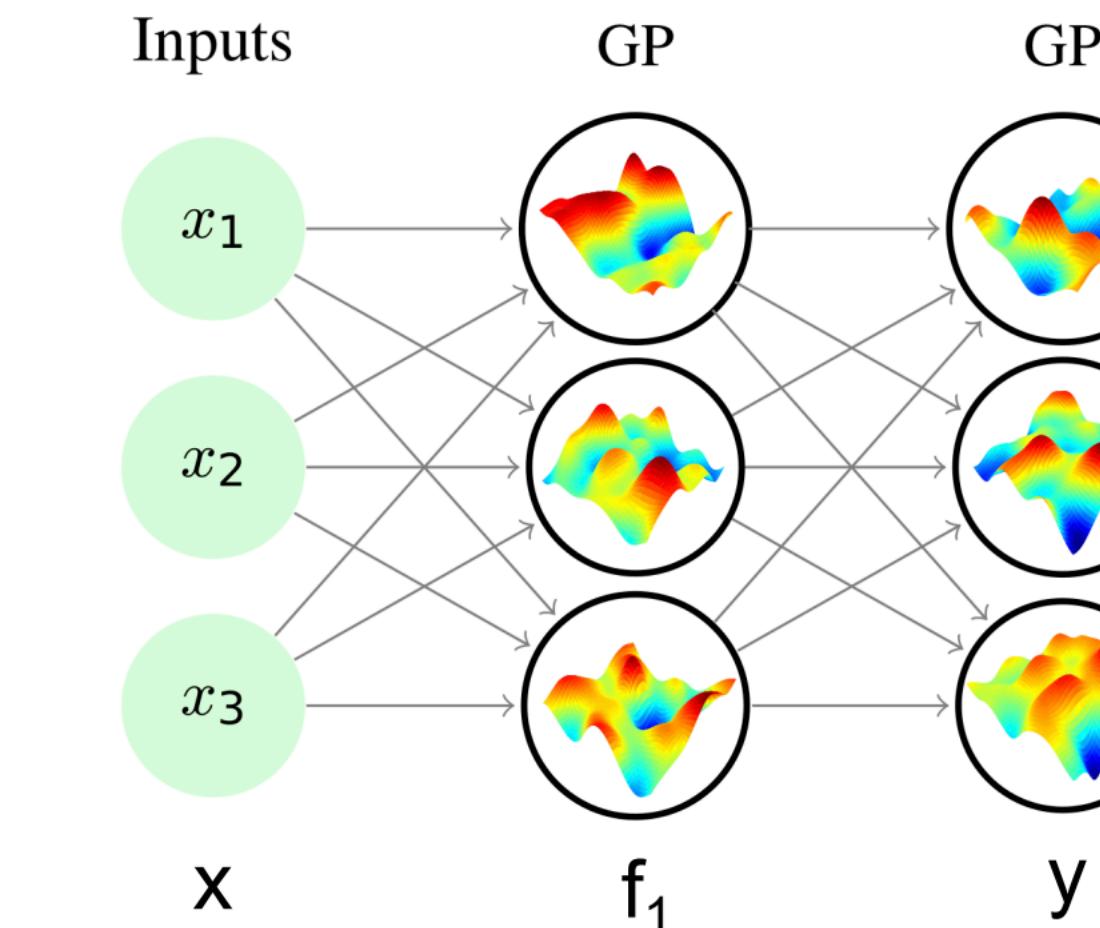
Introducing inducing points per client

$$q(f, \theta) = q(\theta)q(f|\theta) \propto p(\theta) \prod_k t_k(\theta) \int_z p(f|\theta) \prod_k t_k(\mathbf{u}|\mathbf{z}, \theta) \prod_k t_k(\mathbf{z}) dz$$

Research frontier 5: Merging deep nets and GPs



Deep Gaussian processes (Damianou and Lawrence, 2013)

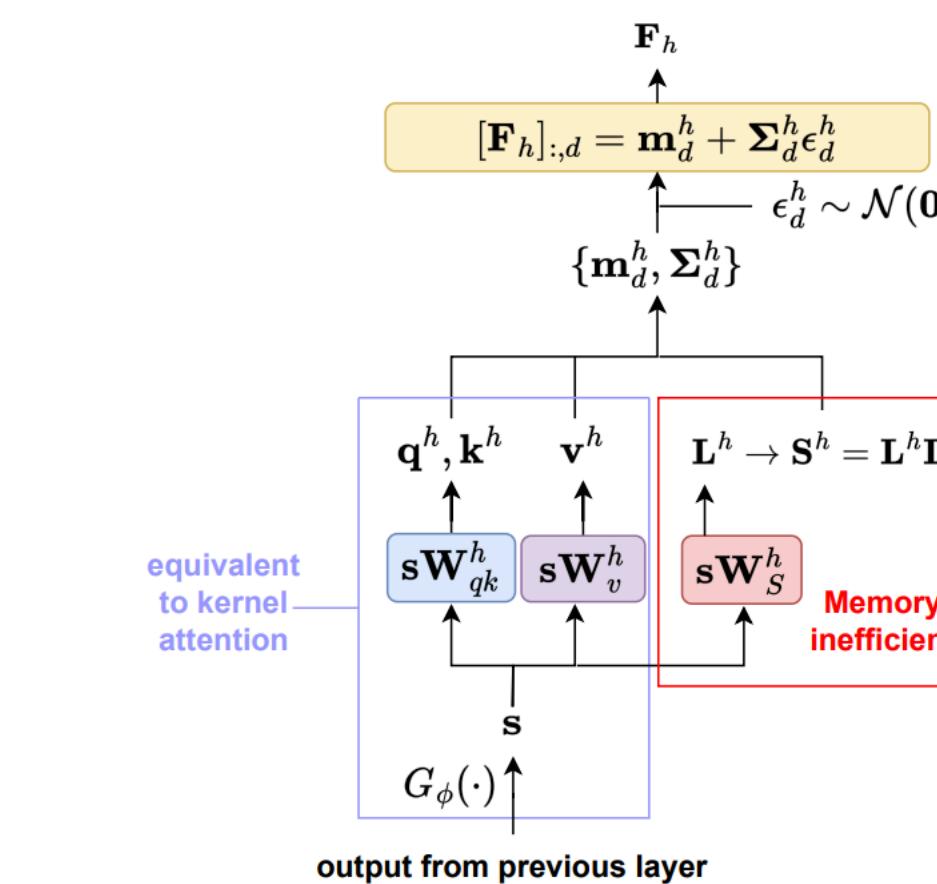


awarded “Test of time” at AISTATS 2023

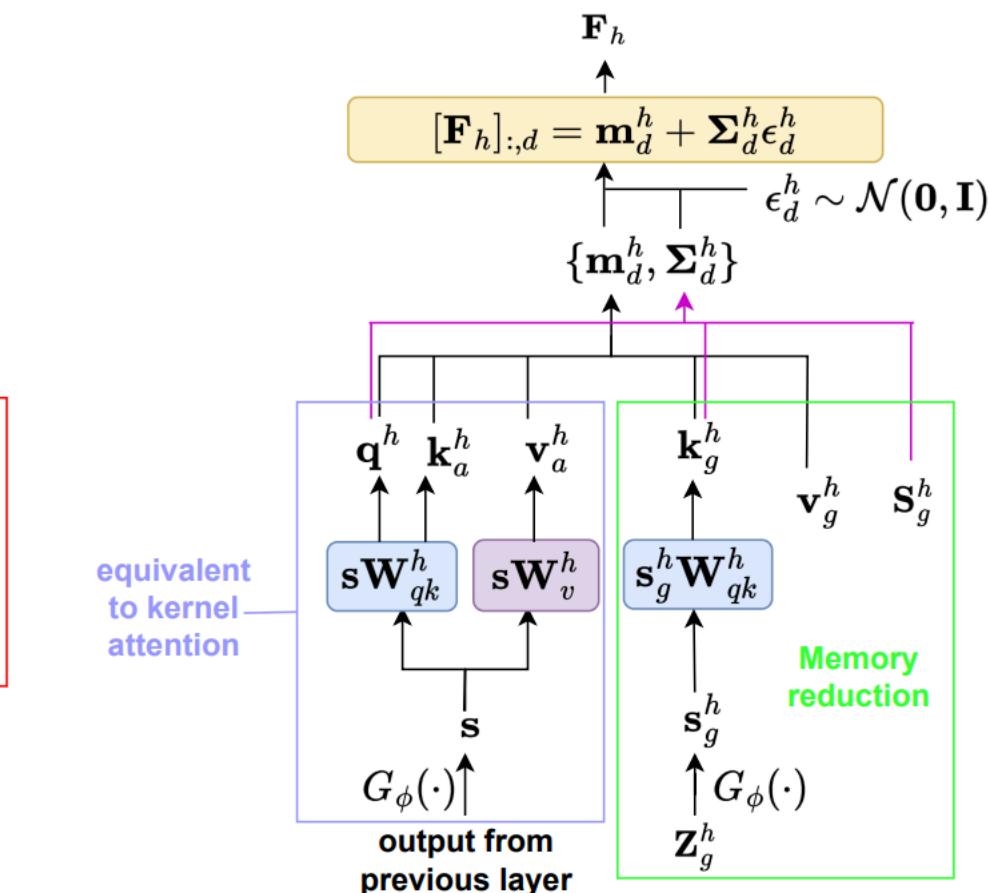
Deep kernel learning (Wilson et al, 2016)



(a) Vanilla Transform



(b) Standard SGPA (our



(c) Decoupled SGPA (ours)

Big picture

Week 7 - lecture 1

Exact for GP regression

Week 7 - lecture 2

Laplace

Week 11,12

Variational inference

Expectation propagation

Week 9

Sampling

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

Exact or *full* approximations

Sparse approximations

Variational inference

(Power) Expectation propagation

$$\mathcal{O}(NM^2) \text{ or } \mathcal{O}(M^3)$$

Collapsed for GP regression or batch updates

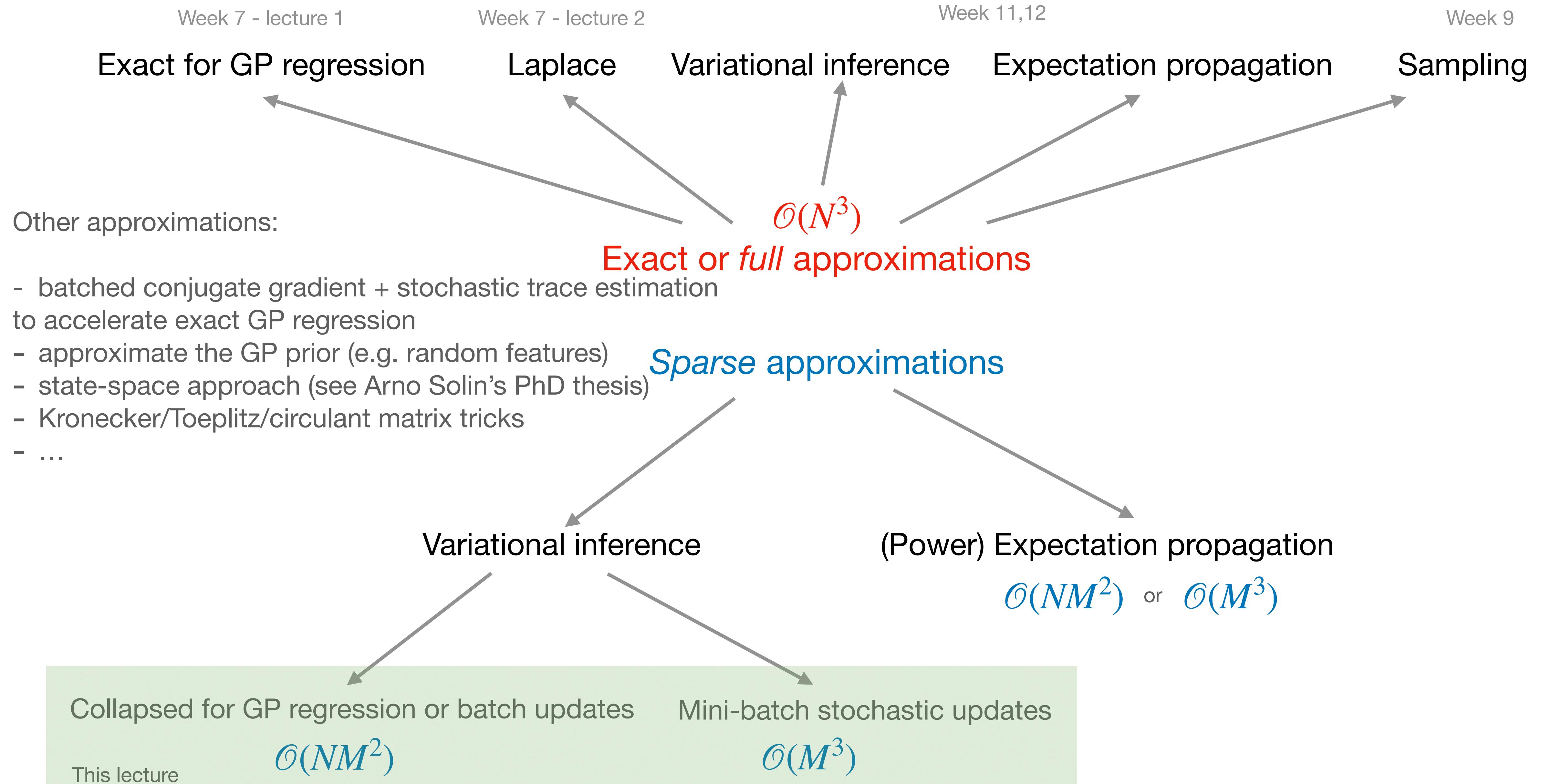
Mini-batch stochastic updates

This lecture

$$\mathcal{O}(NM^2)$$

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

Big picture



Thanks for your attention!

Questions?