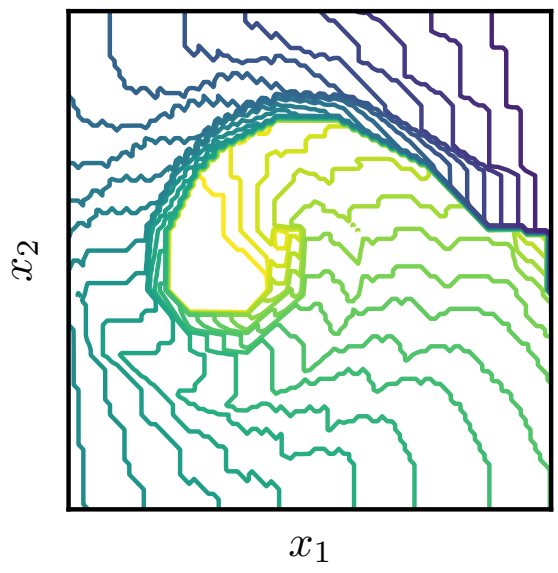
Approximate EP for Deep Gaussian Processes

José Miguel Hernández–Lobato
Department of Engineering
University of Cambridge

Joint work with
Thang D. Bui, Daniel Hernández-Lobato,
Yingzhen Li and Rich E. Turner.

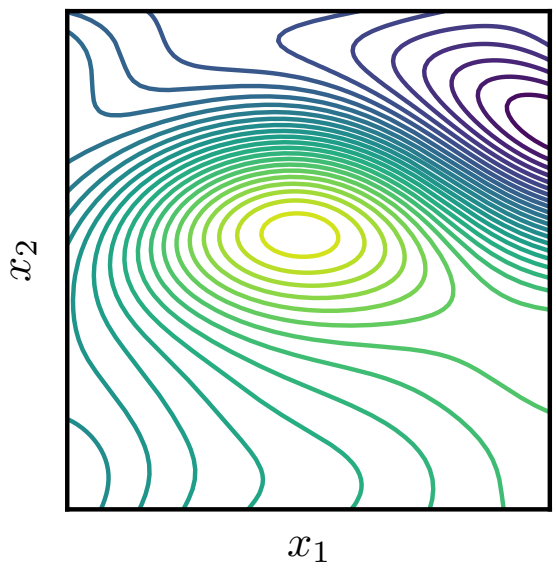
Motivation

Target function

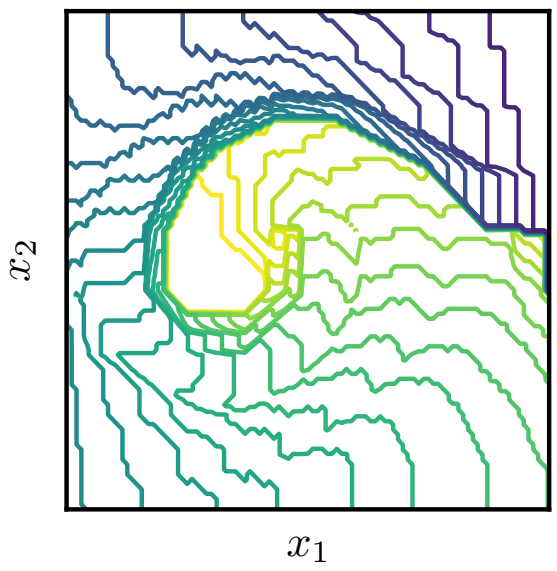


Motivation

GP fit

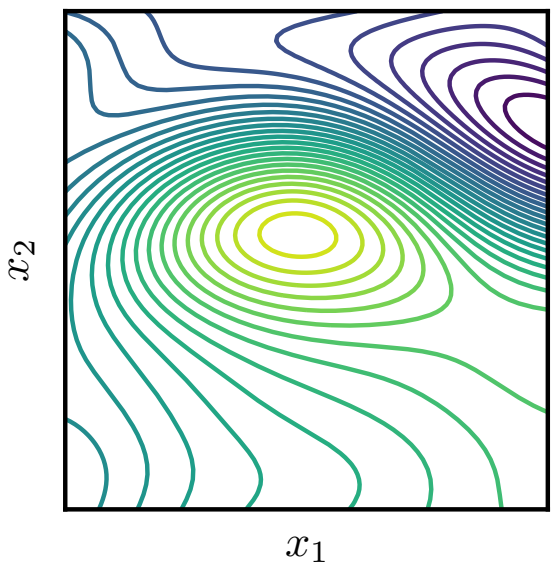


Target function

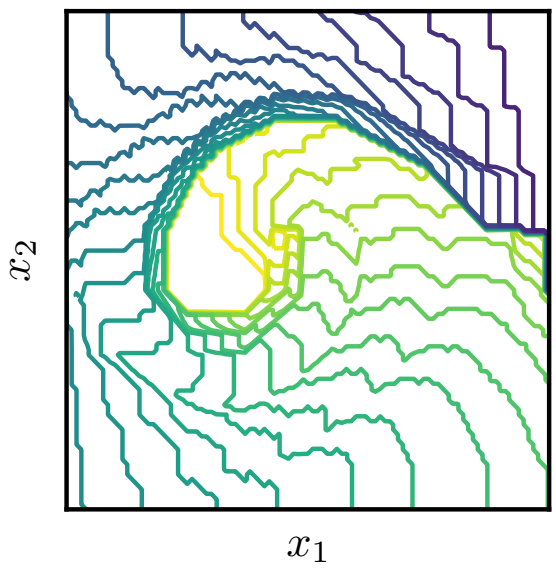


Motivation

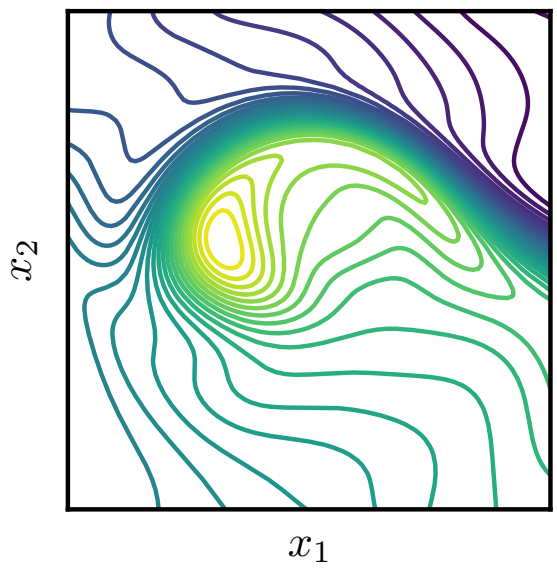
GP fit



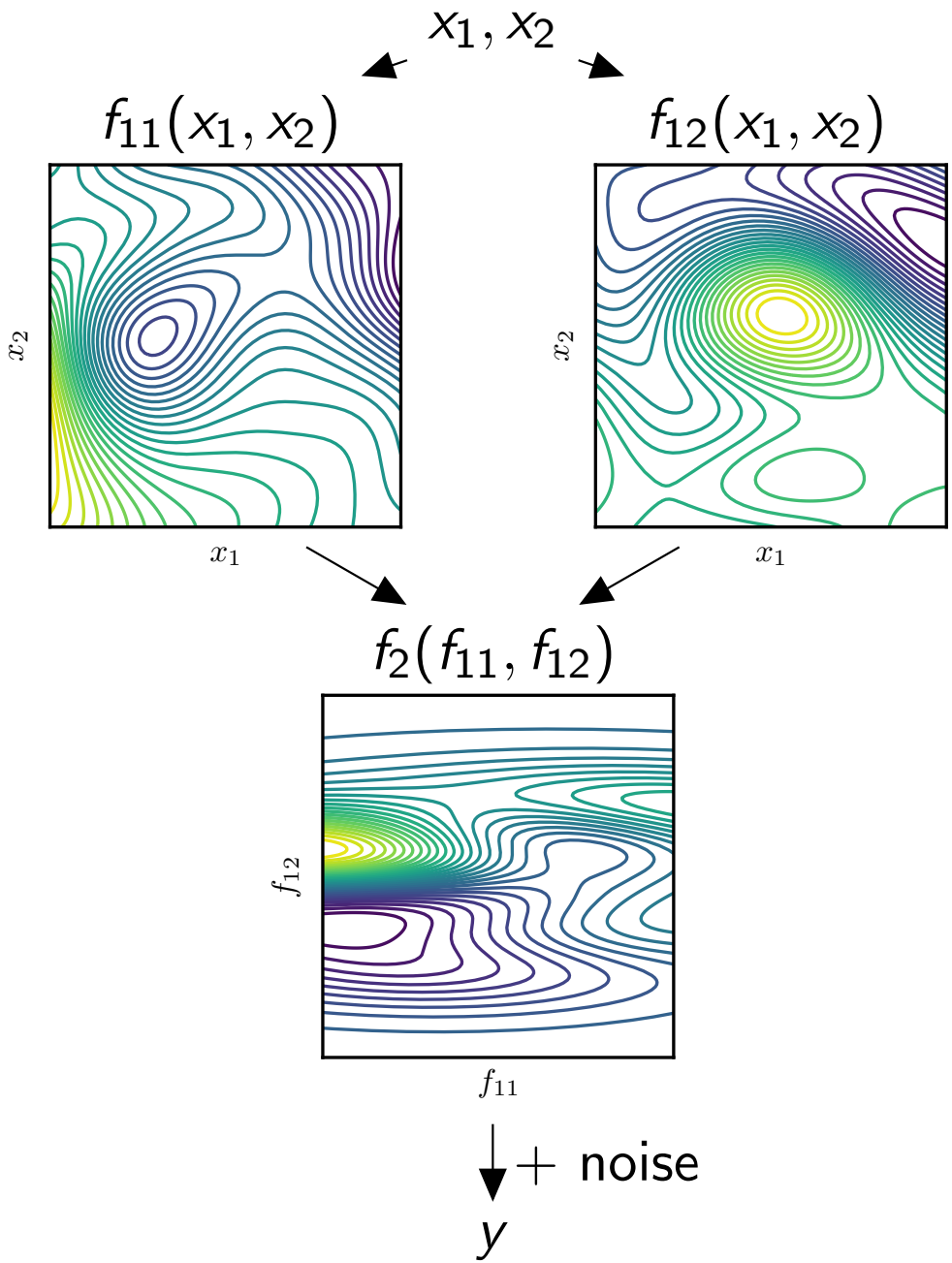
Target function



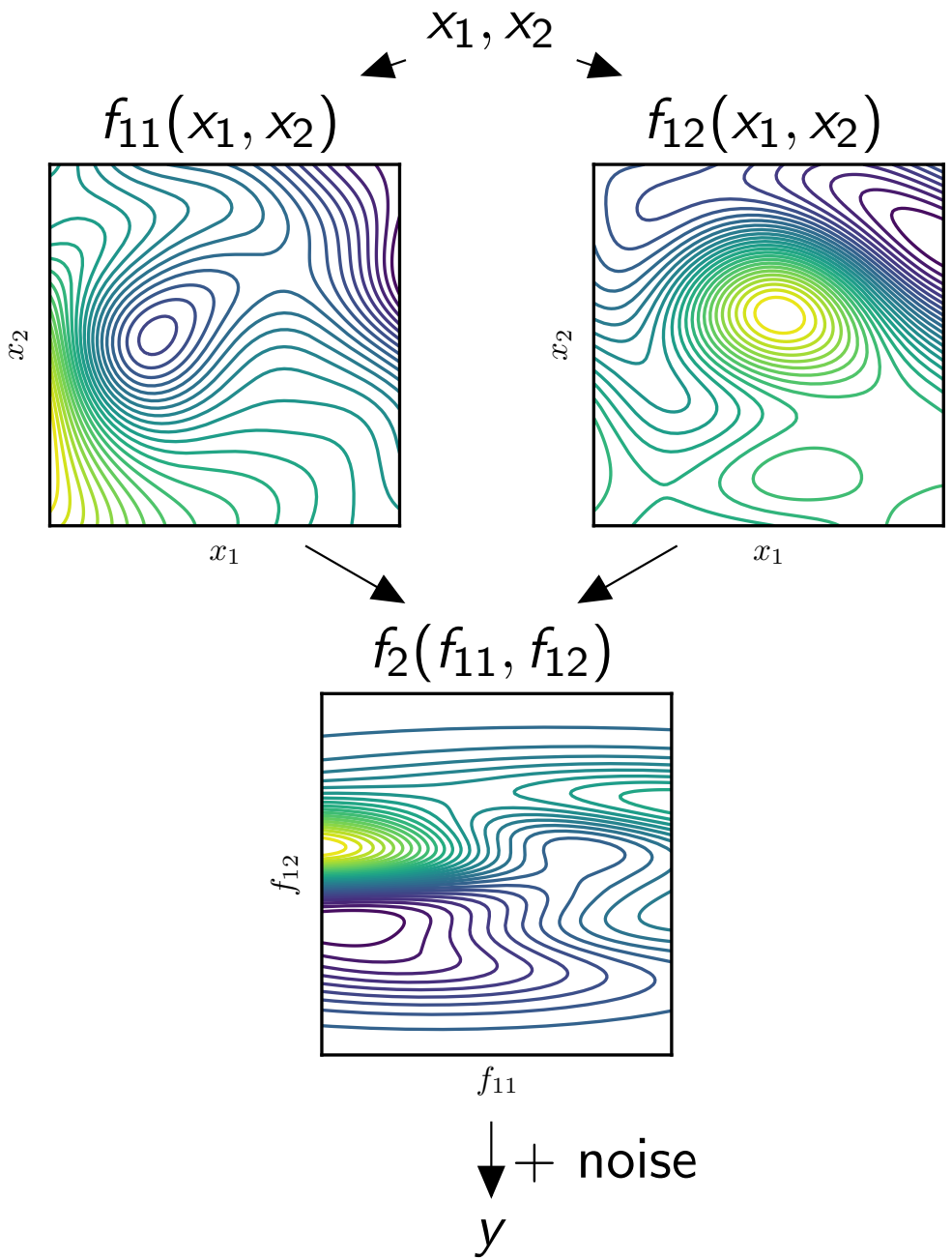
DGP fit



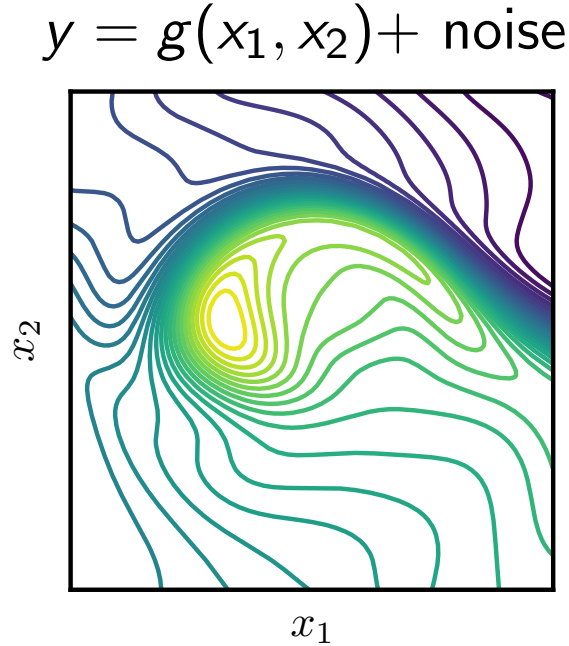
How do deep GPs work?



How do deep GPs work?



$\equiv$



Why deep GPs?

Advantages:

- useful input warping: automatic, nonparametric kernel design
- repair damage done by sparse approximations to GPs
- inference exact for layer's output when layer's input is deterministic

Why deep GPs?

Advantages:

- useful input warping: automatic, nonparametric kernel design
- repair damage done by sparse approximations to GPs
- inference exact for layer's output when layer's input is deterministic

Drawbacks:

- require complicated approximate inference methods
- high computational cost

Bayesian inference

Posterior over latent functions:

$$p(f_1, f_2, f_3|\mathbf{Y}) = \frac{p(f_1)p(f_2)p(f_3)\; p(\mathbf{Y}|f_1, f_2, f_3, \mathbf{X})}{p(\mathbf{Y})}$$

- GP priors
- Likelihood function
- Marginal likelihood

But the posterior $p(f_1, f_2, f_3|\mathbf{Y})$ is **intractable**.

Inducing inputs and outputs

Distribution on f given by GP with inducing inputs $\bar{\mathbf{X}}$ and outputs $\bar{\mathbf{f}}$.

$f(\mathbf{x}) \sim \mathcal{N}(m_{\mathbf{x}}, v_{\mathbf{x}})$, where

$$m_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \bar{\mathbf{f}} \,,$$

$$v_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\mathbf{x}} - \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} \,.$$

Inducing inputs and outputs

Distribution on f given by GP with inducing inputs $\bar{\mathbf{X}}$ and outputs $\bar{\mathbf{f}}$.

$f(\mathbf{x}) \sim \mathcal{N}(m_{\mathbf{x}}, v_{\mathbf{x}})$, where

$$m_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \bar{\mathbf{f}} \,,$$
$$v_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\mathbf{x}} - \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} \,.$$

$\bar{\mathbf{f}} \sim \mathcal{N}(\mathbf{m}_f, \mathbf{V}_f)$:

$$m_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{m}_f \,,$$
$$v_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\mathbf{x}} - \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} + \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{V}_f \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} \,.$$

Inducing inputs and outputs

Distribution on f given by GP with inducing inputs $\bar{\mathbf{X}}$ and outputs $\bar{\mathbf{f}}$.

$f(\mathbf{x}) \sim \mathcal{N}(m_{\mathbf{x}}, v_{\mathbf{x}})$, where

$$m_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \bar{\mathbf{f}} \,,$$
$$v_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\mathbf{x}} - \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} \,.$$

$\bar{\mathbf{f}} \sim \mathcal{N}(\mathbf{m}_f, \mathbf{V}_f)$:

$$m_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{m}_f \,,$$
$$v_{\mathbf{x}} = \mathbf{k}_{\mathbf{x},\mathbf{x}} - \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} + \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}} \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{V}_f \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} \,.$$

$\mathbf{x} \sim \mathcal{N}(\mathbf{m}_{\mathbf{x}}, \mathbf{V}_{\mathbf{x}})$, $f(\mathbf{x})$ no longer Gaussian, but

$$m_{\mathbf{x}} = \mathbb{E}_{\mathbf{x}}[\mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}}] \mathbf{K}_{\bar{\mathbf{x}},\bar{\mathbf{x}}}^{-1} \mathbf{m}_f \,,$$
$$v_{\mathbf{x}} = \text{complicated but doable, requires } \mathbb{E}_{\mathbf{x}}[\mathbf{k}_{\bar{\mathbf{x}},\mathbf{x}} \mathbf{k}_{\mathbf{x},\bar{\mathbf{x}}}] \,.$$

Expectation propagation

Approximates $\boxed{p(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta})}$ with $\boxed{q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta})}$

Expectation propagation

Approximates $\boxed{p(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta})}$ with $\boxed{q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta})}$

$p(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad f_1(\boldsymbol{\theta}) \; f_2(\boldsymbol{\theta}) \; f_3(\boldsymbol{\theta}) \quad \approx \quad q(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad \tilde{f}_1(\boldsymbol{\theta}) \; \tilde{f}_2(\boldsymbol{\theta}) \; \tilde{f}_3(\boldsymbol{\theta})$

Expectation propagation

Approximates $\boxed{p(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta})}$ with $\boxed{q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta})}$

$$p(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad f_1(\boldsymbol{\theta}) \; f_2(\boldsymbol{\theta}) \; f_3(\boldsymbol{\theta}) \quad \approx \quad q(\boldsymbol{\theta}) \propto \quad p_0(\boldsymbol{\theta}) \quad \tilde{f}_1(\boldsymbol{\theta}) \; \tilde{f}_2(\boldsymbol{\theta}) \; \tilde{f}_3(\boldsymbol{\theta})$$

The $\tilde{f}_n$ are tuned by minimizing the KL divergence

$$D_{\mathrm{KL}}[p_n||q] \quad \text{for } n = 1, \ldots, N, \quad \text{where} \quad \begin{array}{rcl} p_n(\boldsymbol{\theta}) & \propto & f_n(\boldsymbol{\theta}) \prod_{j \neq n} \tilde{f}_j(\boldsymbol{\theta}) \\ q(\boldsymbol{\theta}) & \propto & \tilde{f}_n(\boldsymbol{\theta}) \prod_{j \neq n} \tilde{f}_j(\boldsymbol{\theta}) \end{array}.$$

The EP approximation to the **evidence** $p(\mathbf{Y})$ is given by

$$\log Z_{\text{EP}} = \log Z_q + \sum_{n=1}^{N} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right) \right] ,$$

The EP approximation to the **evidence** $p(\mathbf{Y})$ is given by

$$\log Z_{\text{EP}} = \log Z_q + \sum_{n=1}^{N} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right) \right] ,$$

The EP solution for q can be obtained by solving

$$\max_{q} \min_{\tilde{f}_1,\ldots,\tilde{f}_N} \log Z_{\text{EP}} \quad \text{subject to} \quad q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta}) .$$

The EP approximation to the **evidence** $p(\mathbf{Y})$ is given by

$$\log Z_{\text{EP}} = \log Z_q + \sum_{n=1}^{N} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right) \right] ,$$

The EP solution for q can be obtained by solving

$$\max_{q} \min_{\tilde{f}_1,\ldots,\tilde{f}_N} \log Z_{\text{EP}} \quad \text{subject to} \quad q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta}) .$$

Can be solved with **double-loop** algorithm.

The EP approximation to the **evidence** $p(\mathbf{Y})$ is given by

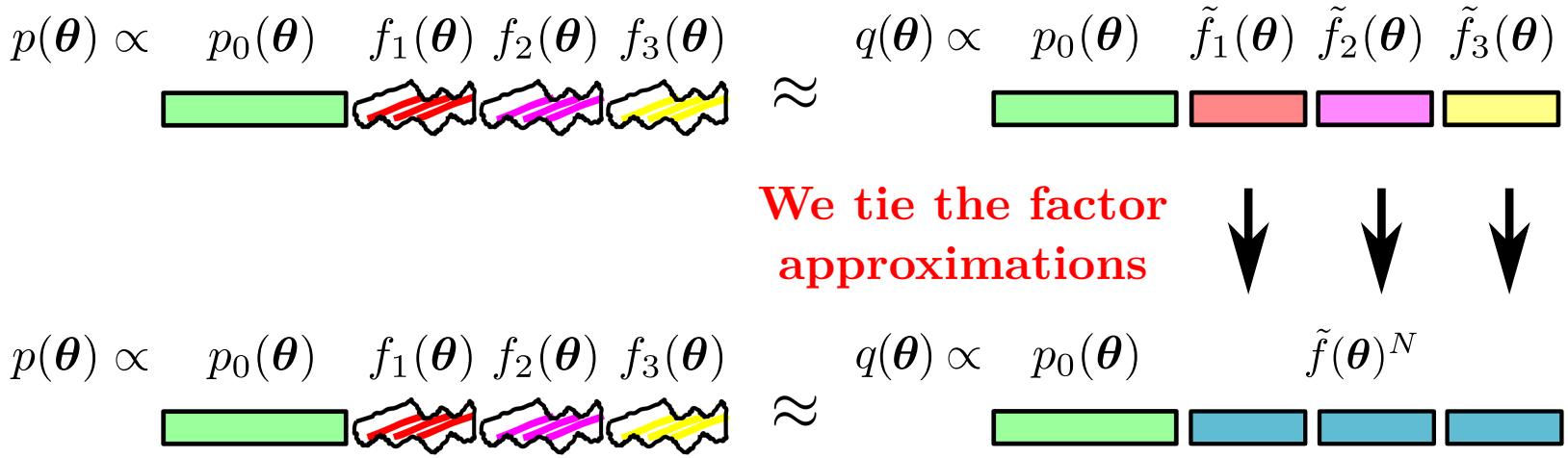
$$\log Z_{\text{EP}} = \log Z_q + \sum_{n=1}^{N} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}_n(\boldsymbol{\theta})} \right) \right] ,$$

The EP solution for q can be obtained by solving

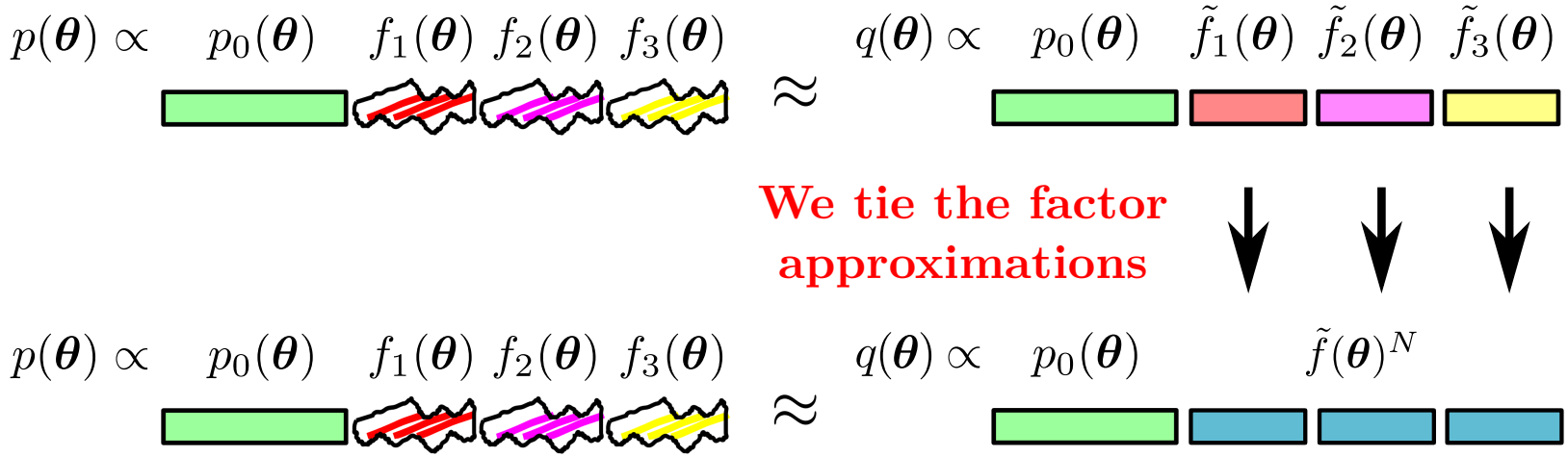
$$\max_{q} \min_{\tilde{f}_1,\ldots,\tilde{f}_N} \log Z_{\text{EP}} \quad \text{subject to} \quad q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} \tilde{f}_n(\boldsymbol{\theta}) .$$

Can be solved with **double-loop** algorithm. **Too slow in practice!**

Approximate Expectation Propagation



Approximate Expectation Propagation



- $\max_q \min_{\tilde{f}_1,\ldots,\tilde{f}_N}$ problem $\rightarrow$ $\max_q$ problem, **no double-loop needed!**

We only need to optimize

$$\log Z_{\mathrm{EP}} = \log Z_q + \sum_{n=1}^{N} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right) \right] .$$

We only need to optimize

$$\log Z_{\mathrm{EP}} = \log Z_q + \sum_{n=1}^{N} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right) \right] .$$

But this requires integrating the exact likelihood factors (**intractable**).

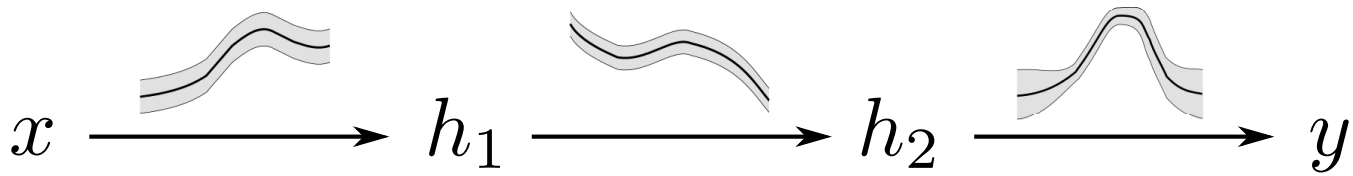
We only need to optimize

$$\log Z_{\text{EP}} = \log Z_q + \sum_{n=1}^{N} \log \mathbf{E}_q \left[\left(\frac{f_n(\boldsymbol{\theta})}{\tilde{f}(\boldsymbol{\theta})} \right) \right] .$$

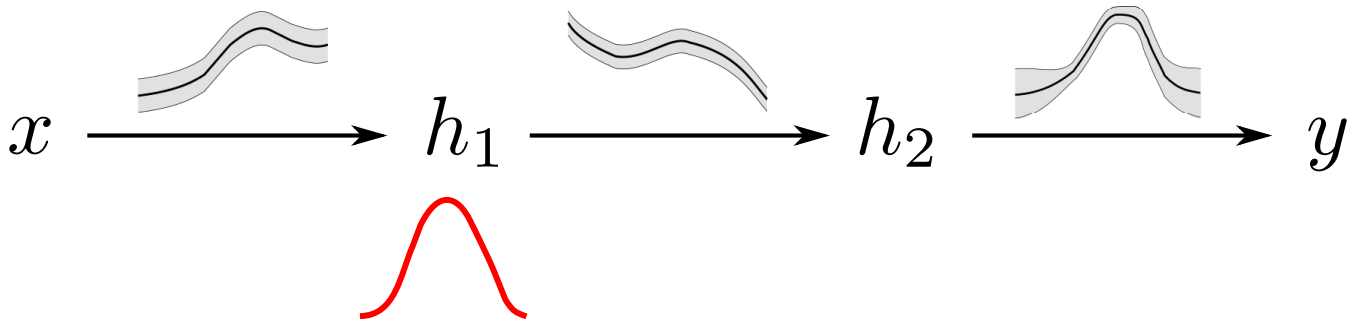
But this requires integrating the exact likelihood factors (**intractable**).

Solution: **moment match** each GP output to a Gaussian at each layer.

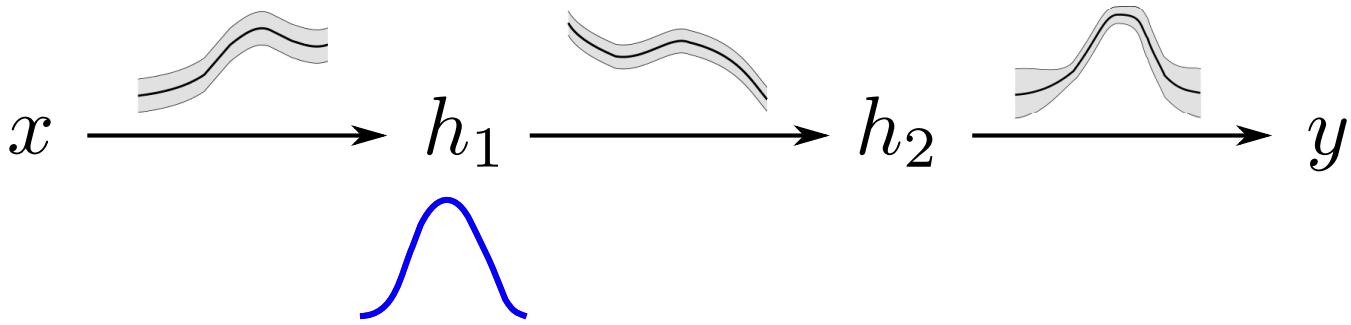
Approximate Gaussian message propagation



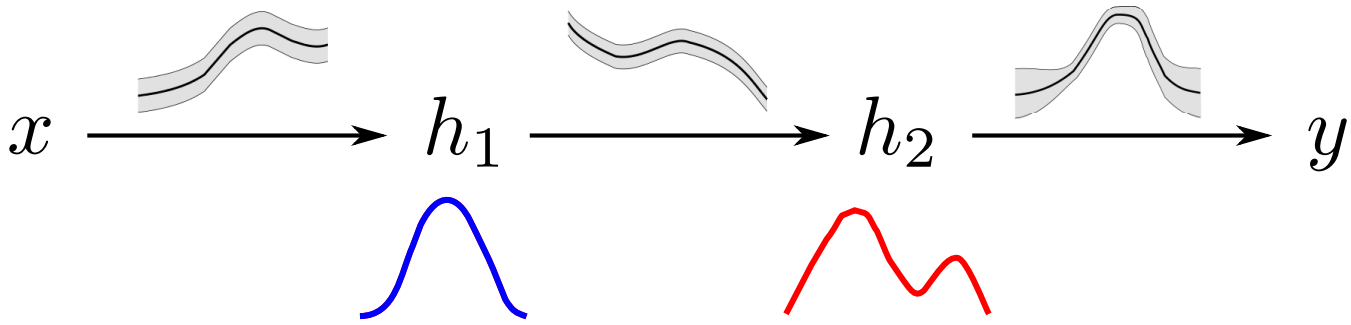
Approximate Gaussian message propagation



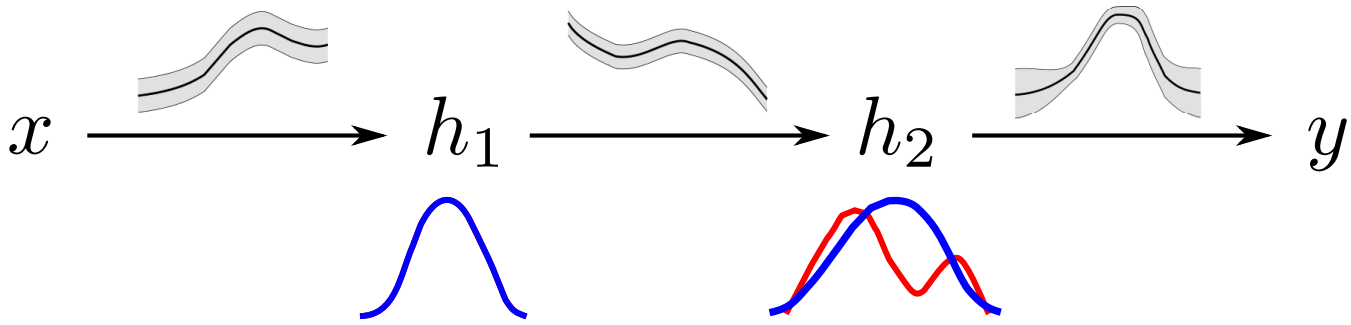
Approximate Gaussian message propagation



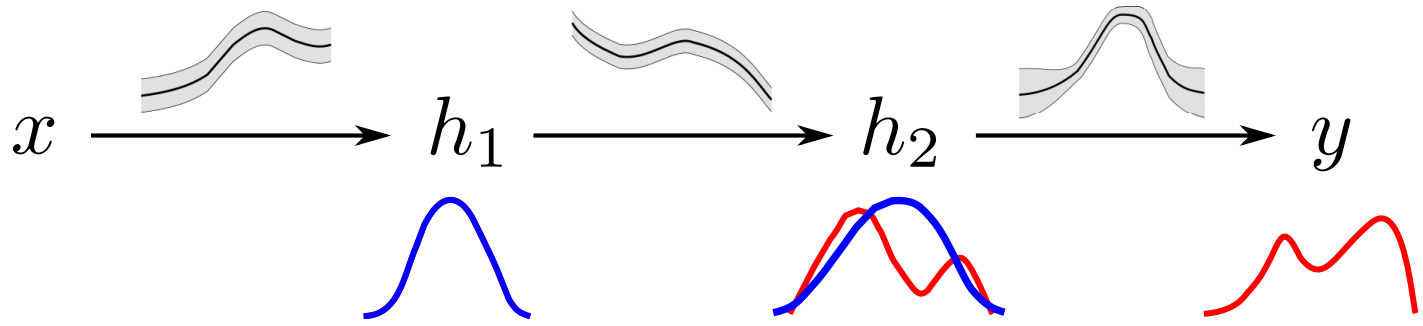
Approximate Gaussian message propagation



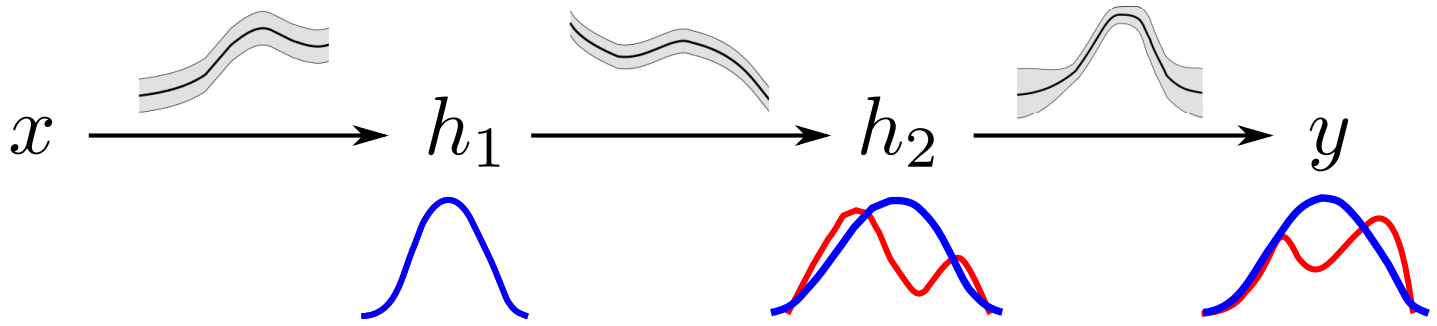
Approximate Gaussian message propagation



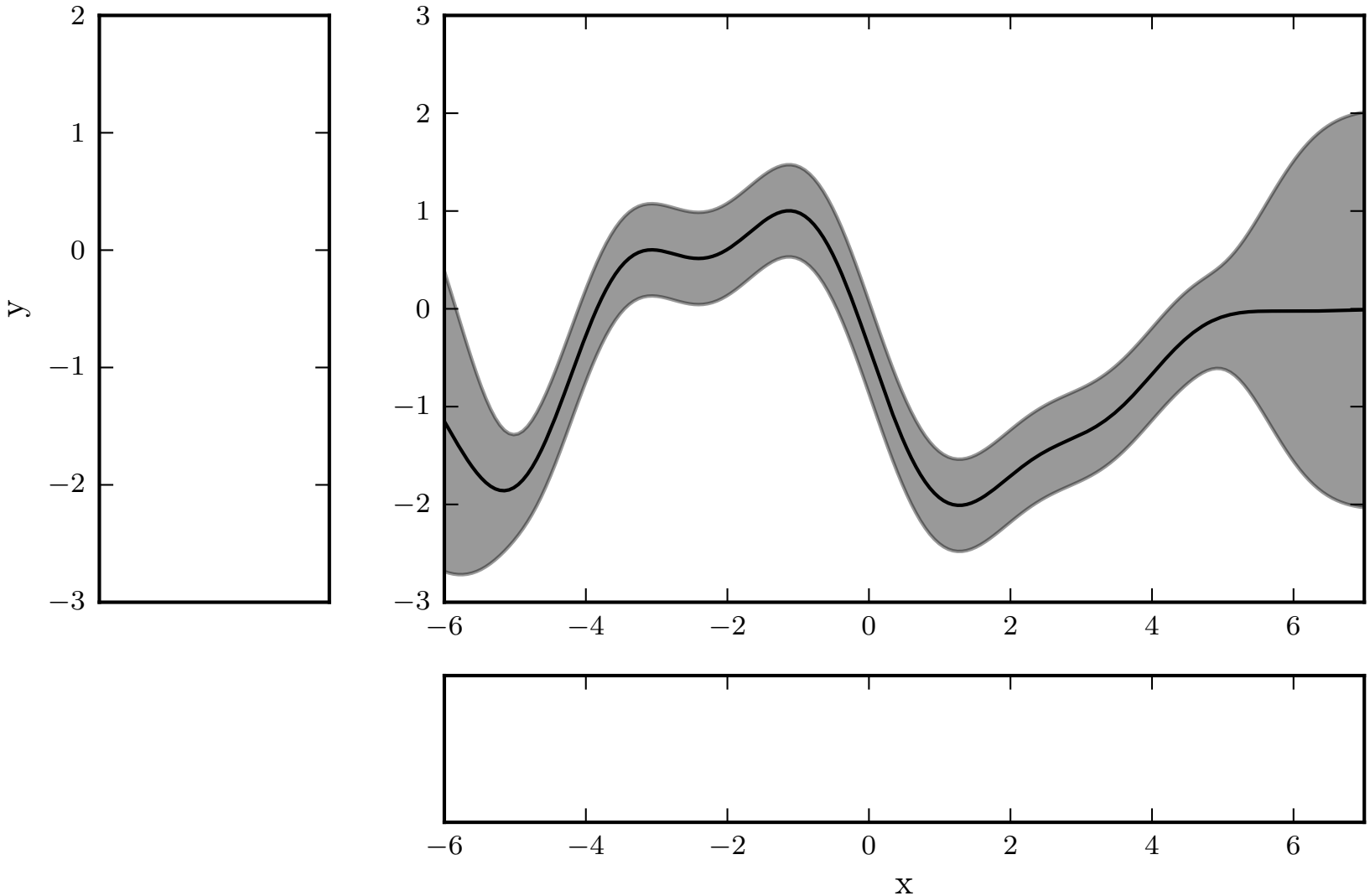
Approximate Gaussian message propagation



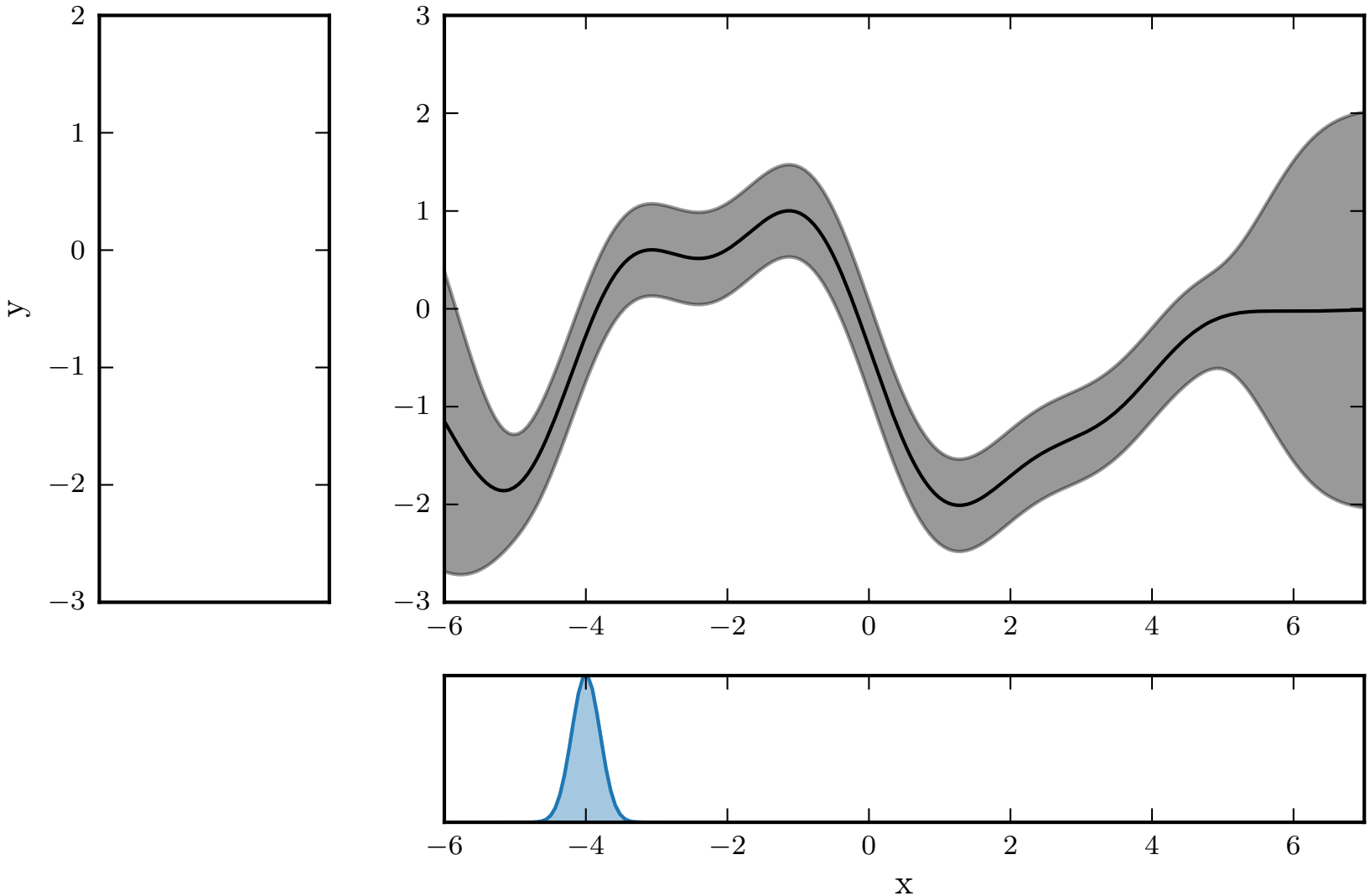
Approximate Gaussian message propagation



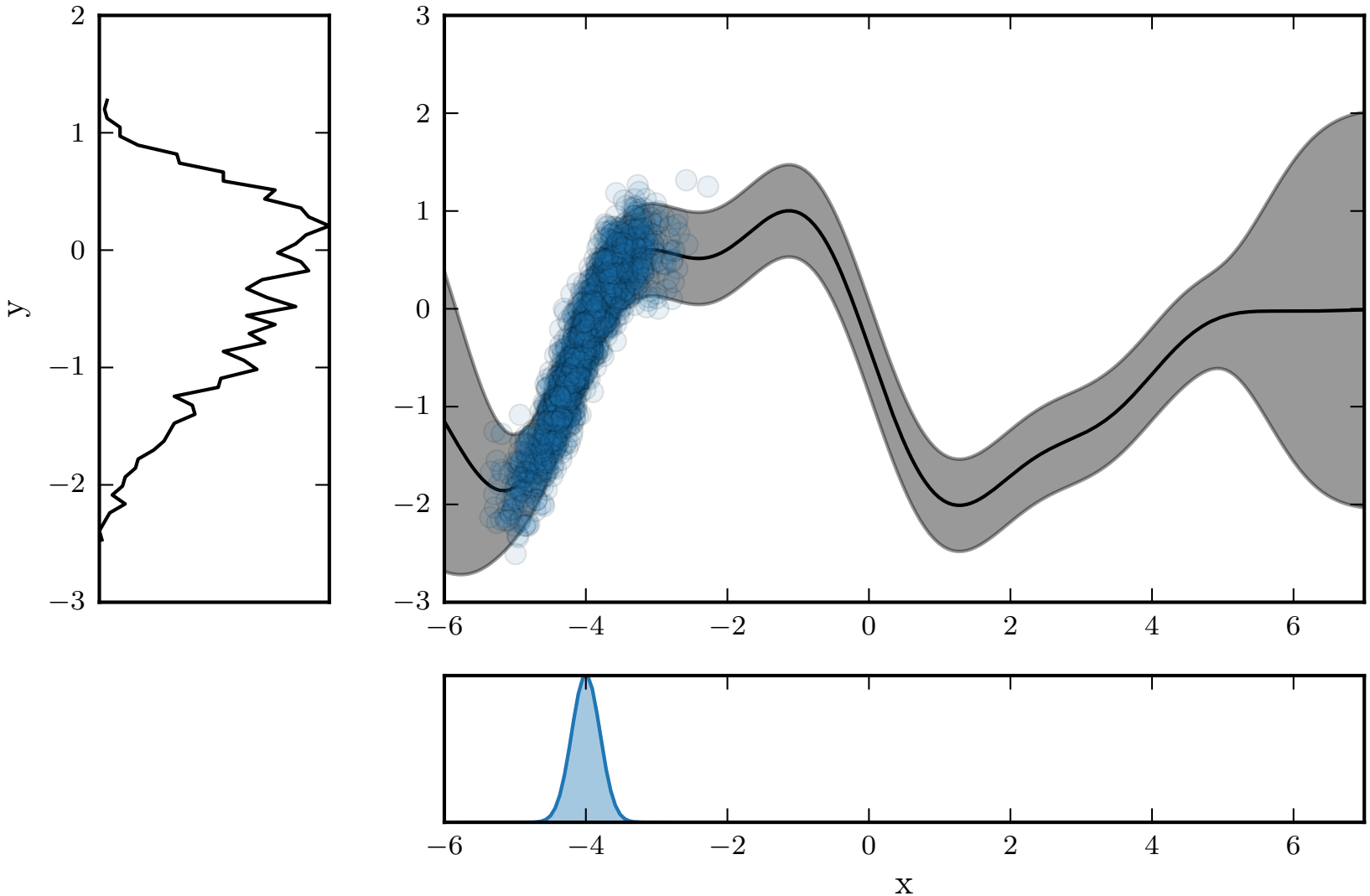
Gaussian projection



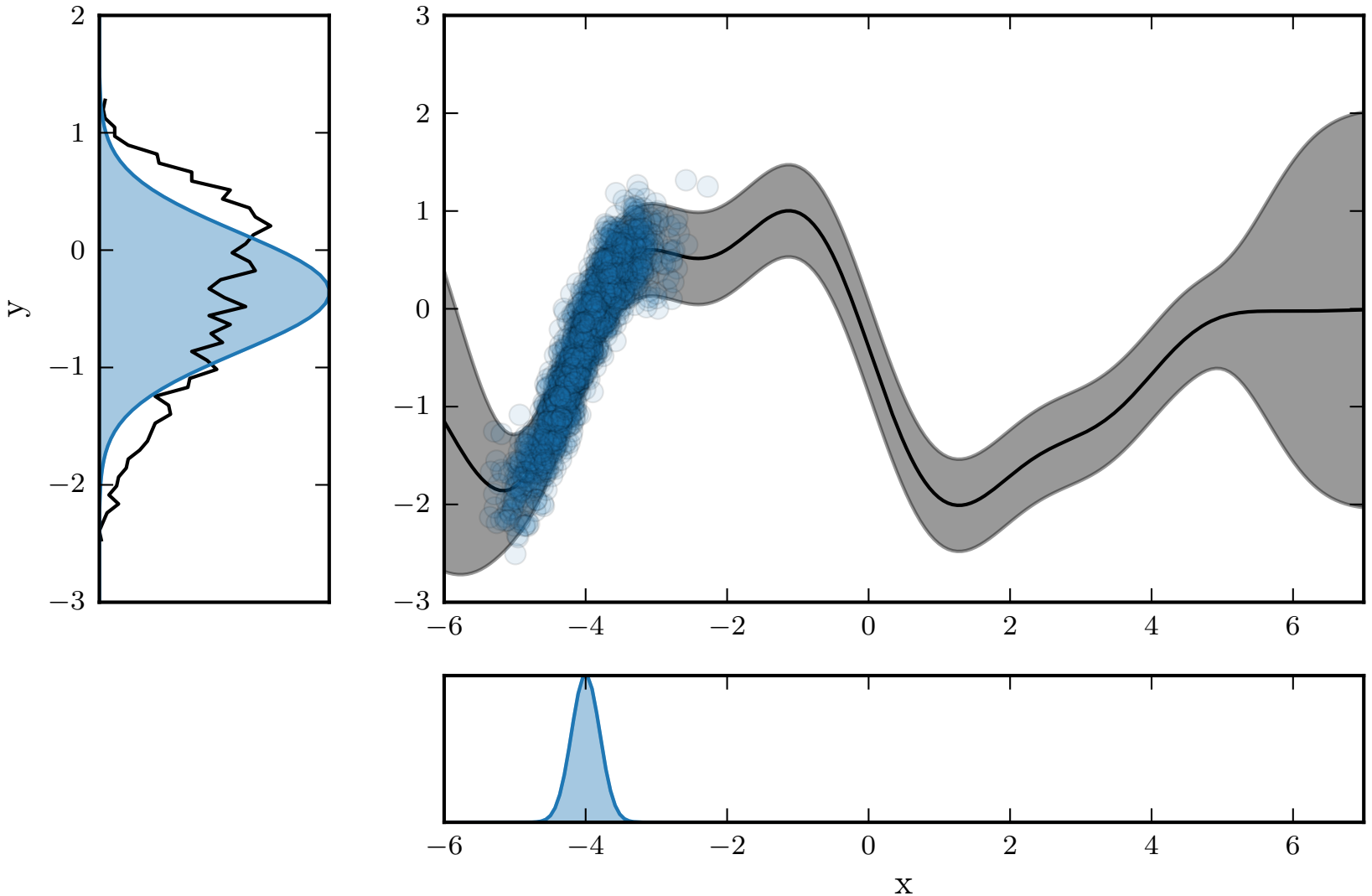
Gaussian projection



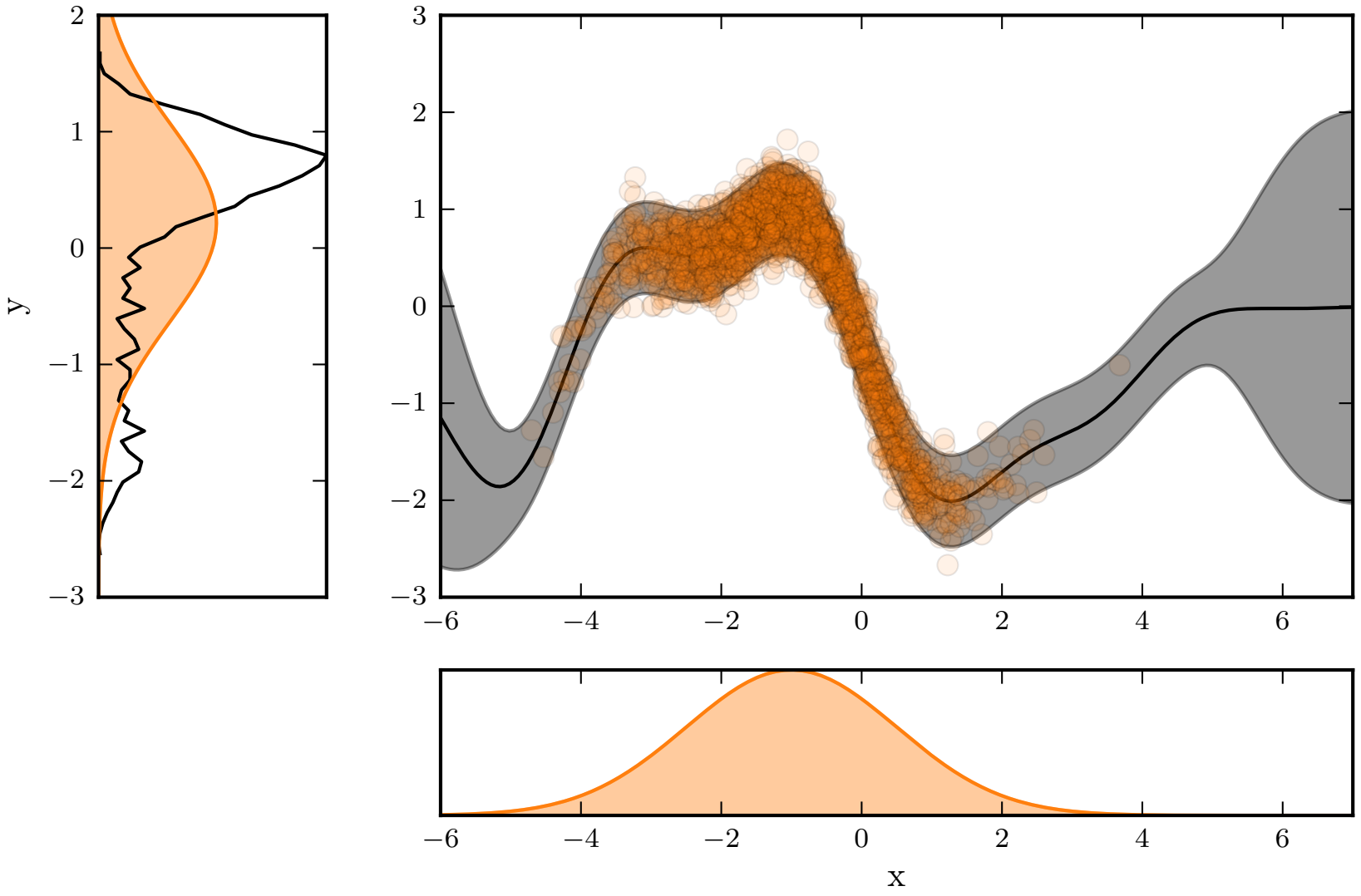
Gaussian projection



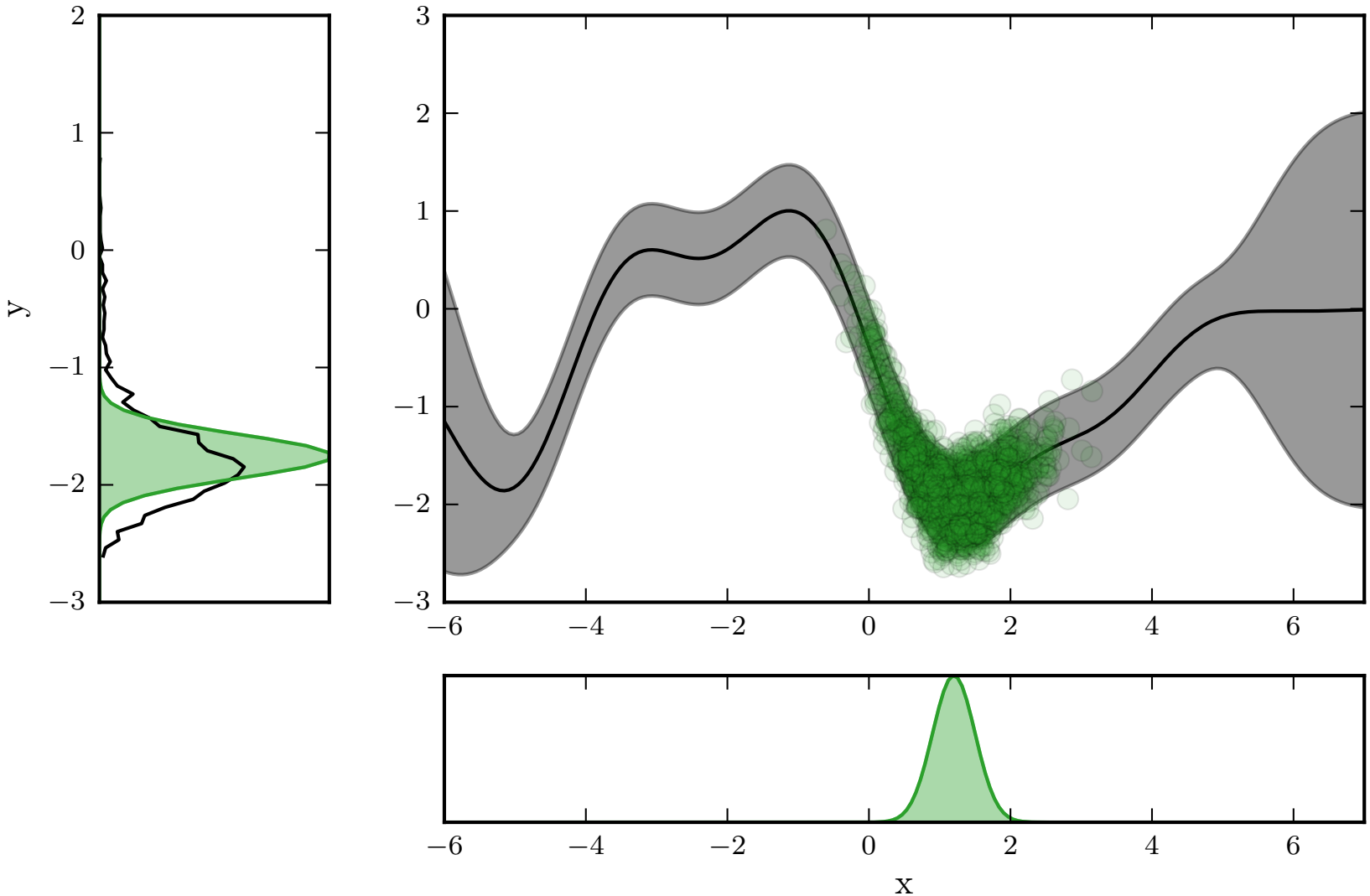
Gaussian projection



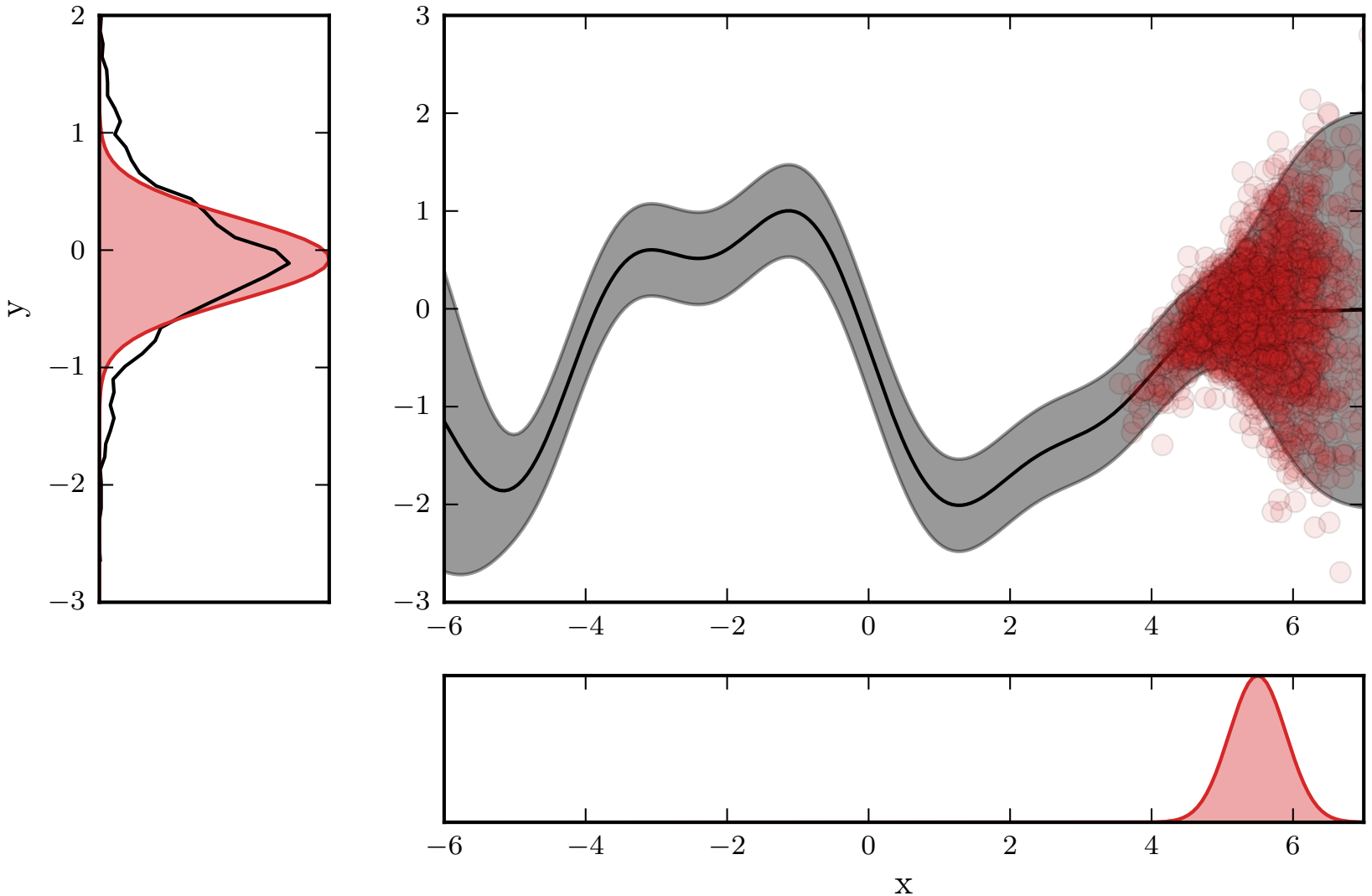
Gaussian projection



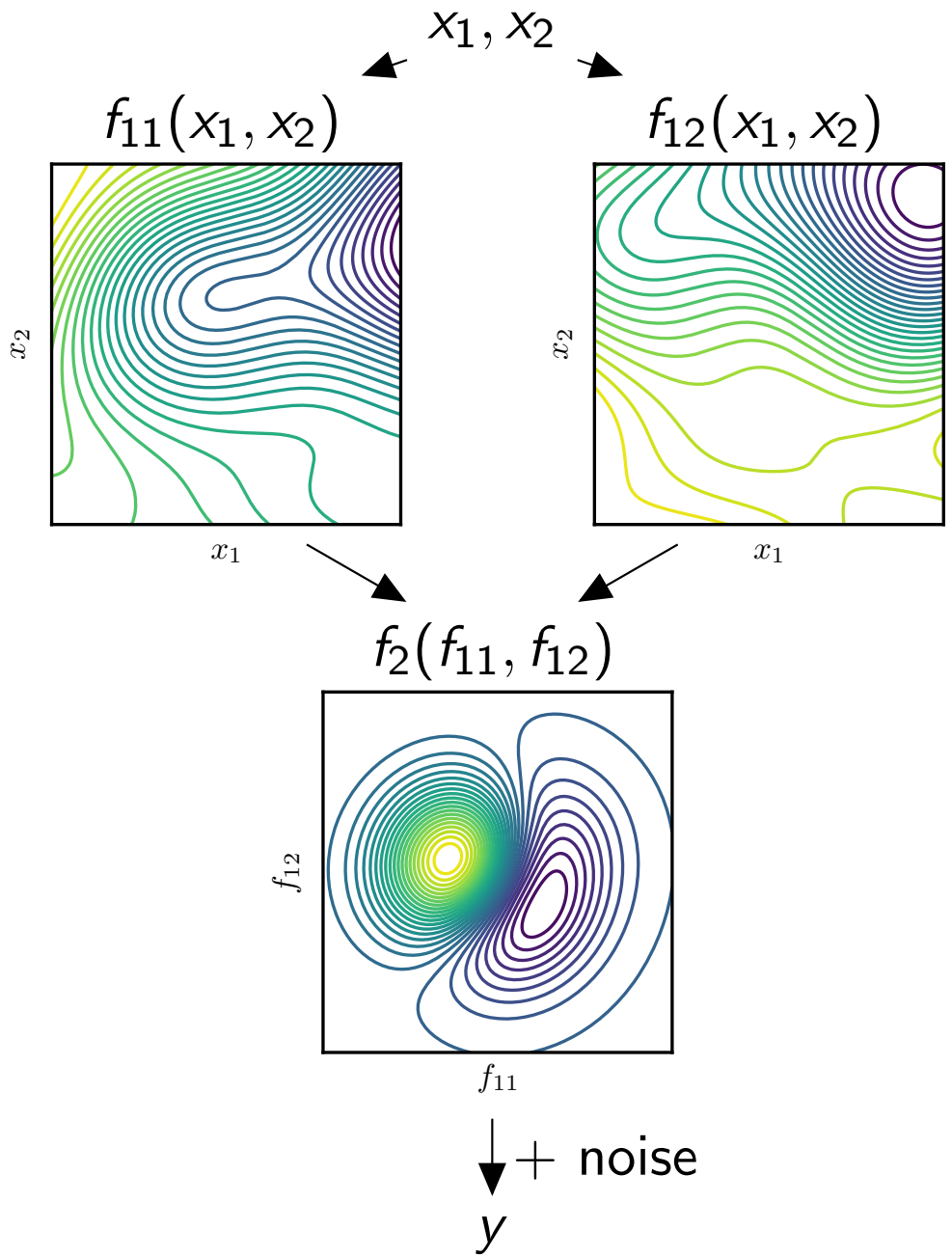
Gaussian projection



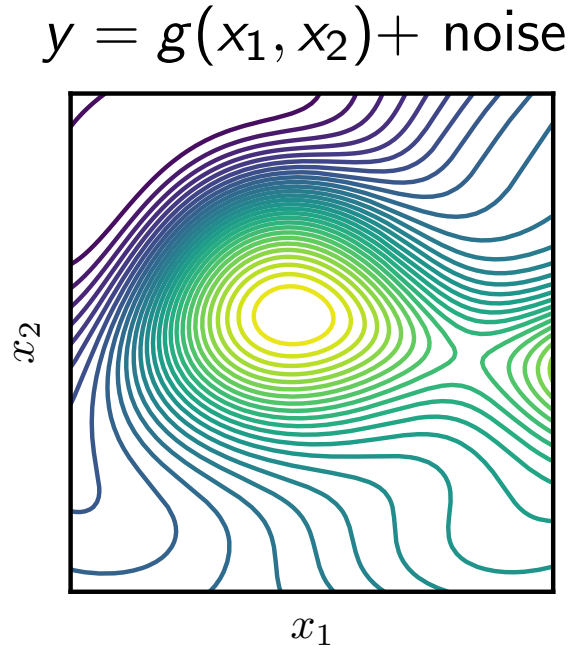
Gaussian projection



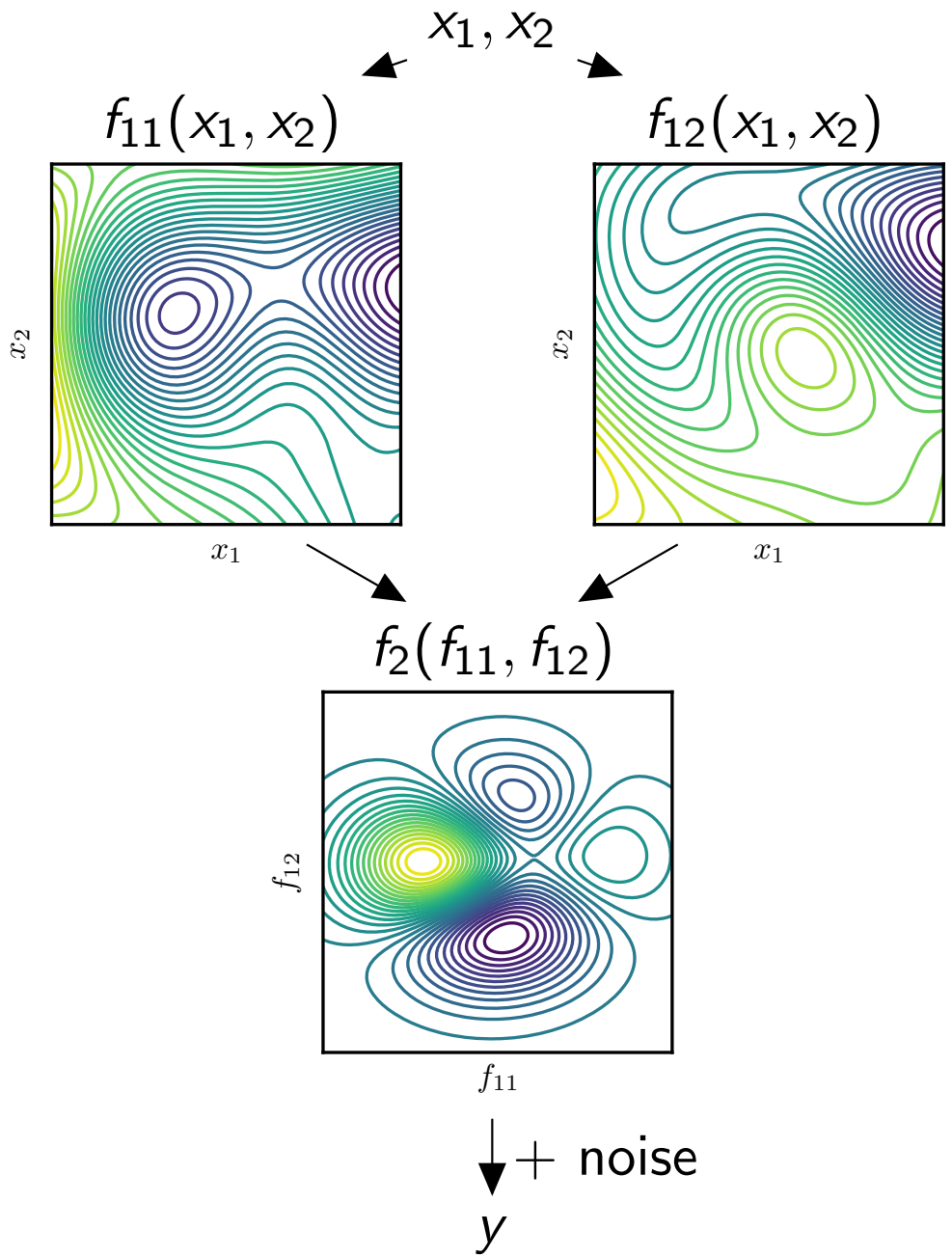
Training deep GPs: 1 epochs



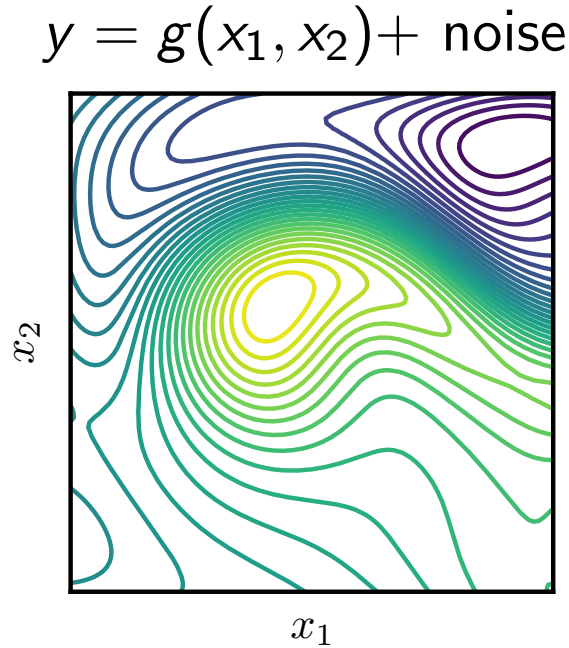
$\equiv$



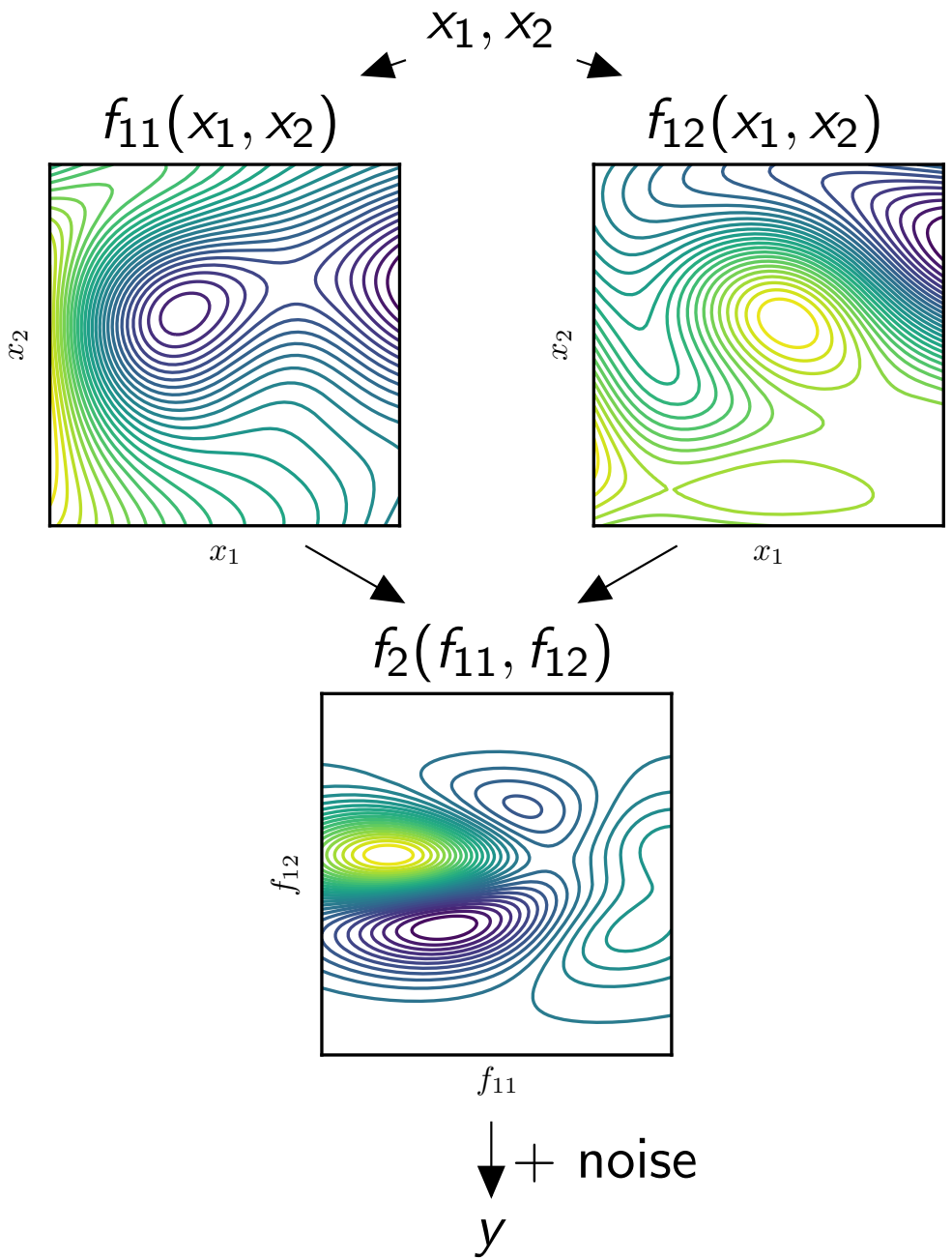
Training deep GPs: 2 epochs



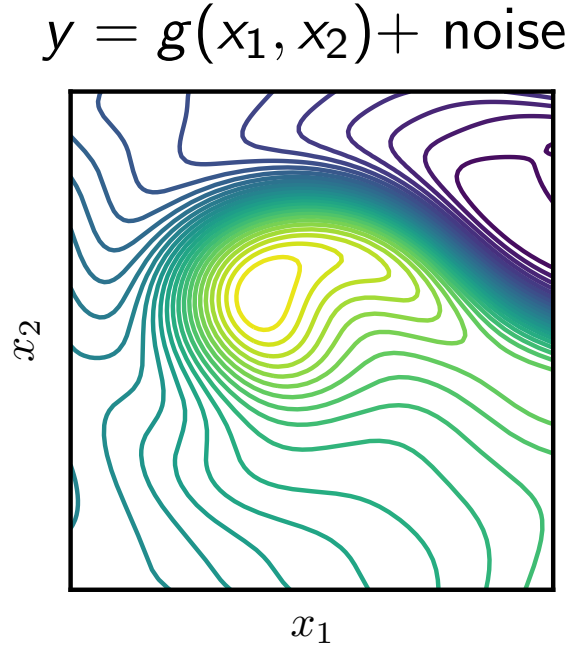
$\equiv$



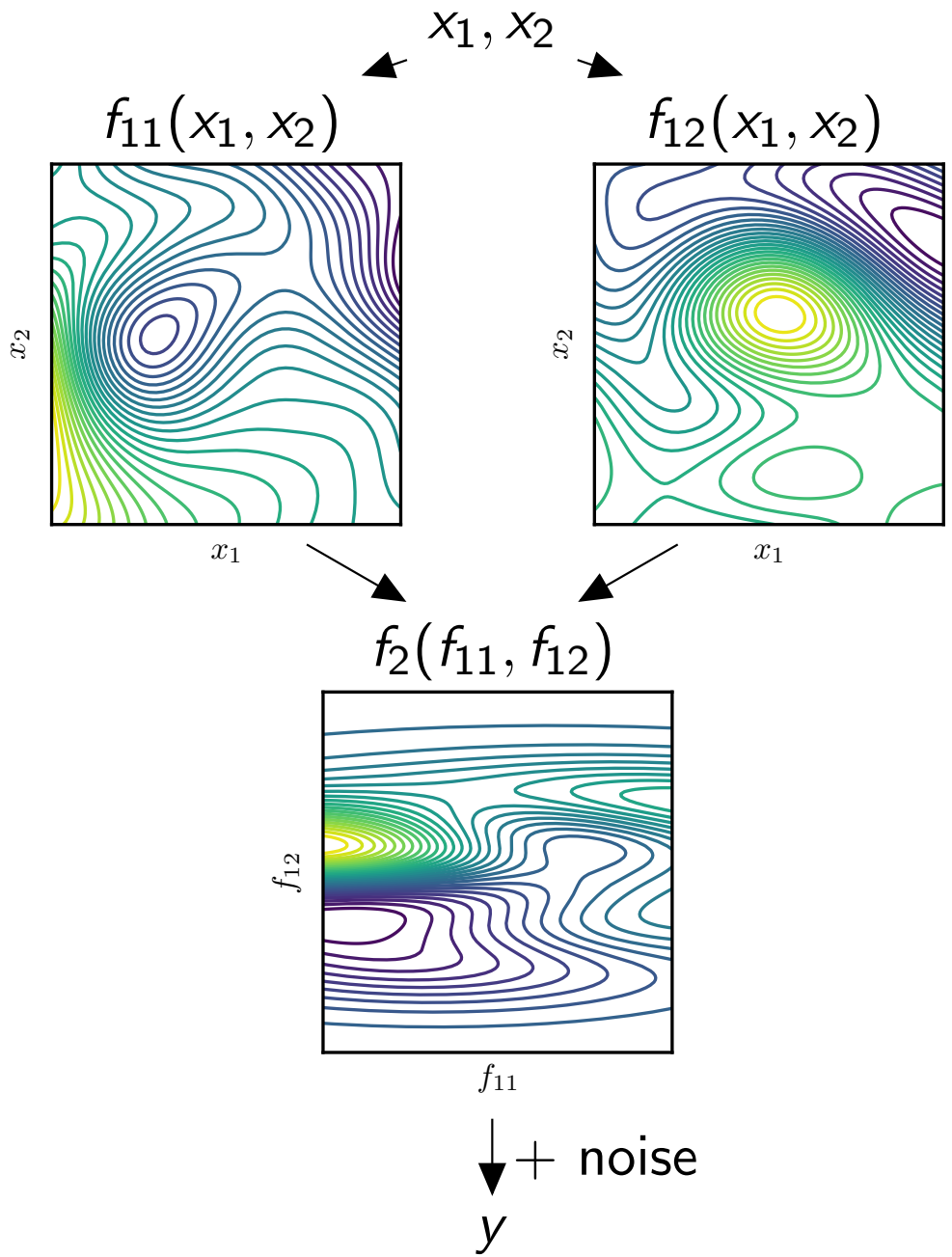
Training deep GPs: 6 epochs



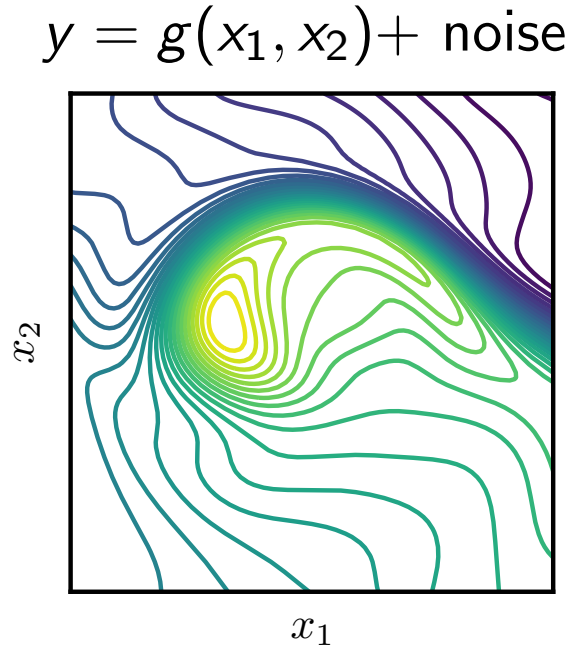
$\equiv$



Training deep GPs: 25 epochs

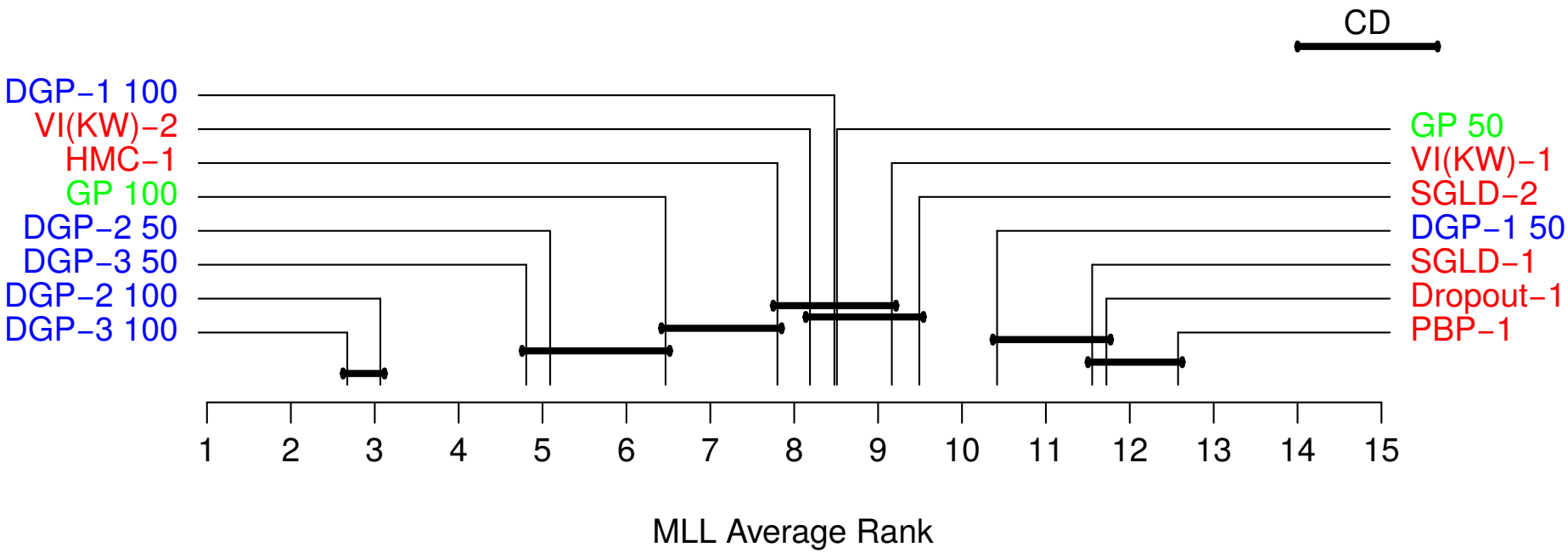


$\equiv$



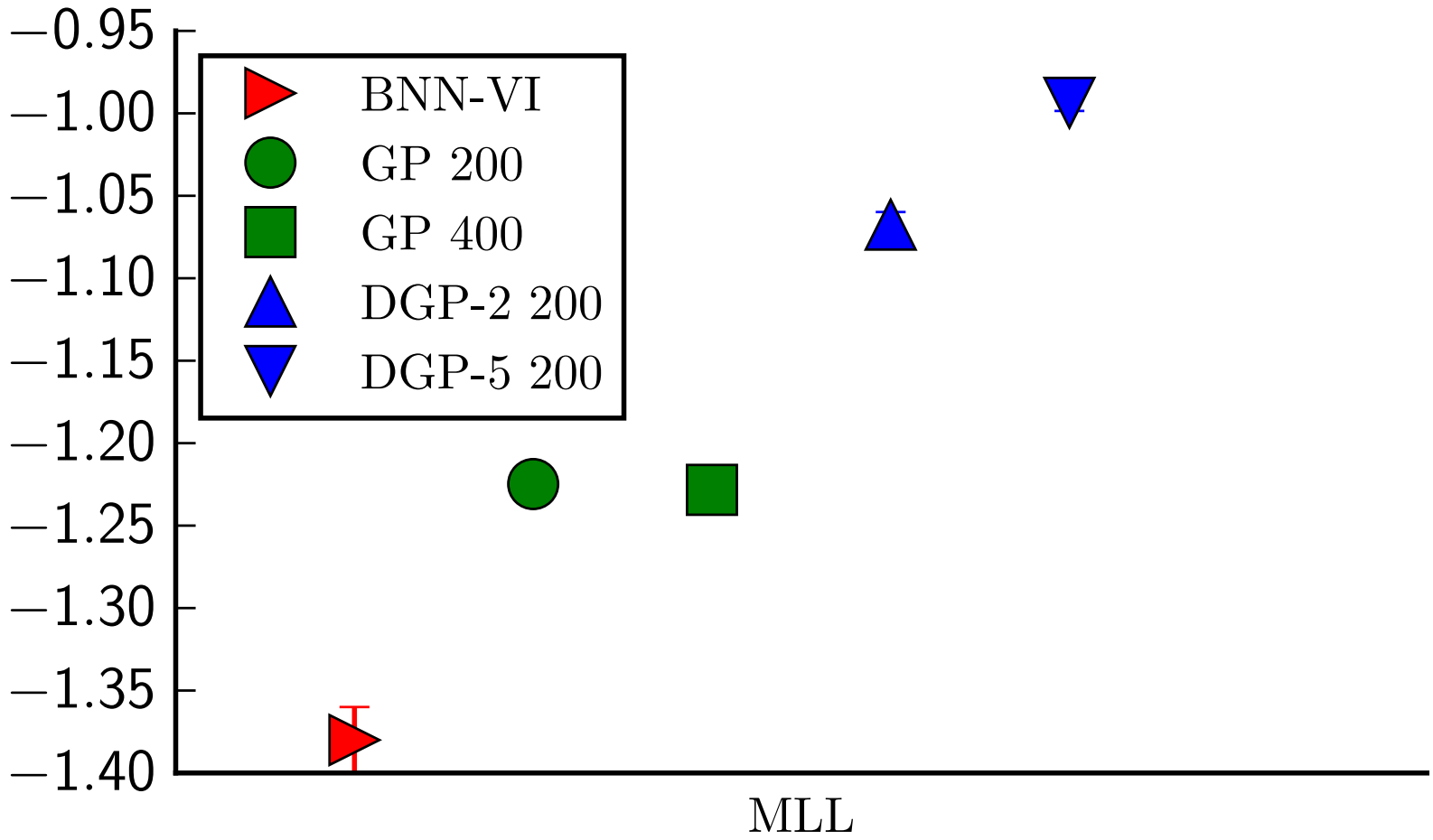
Comparison to Bayesian neural networks

Rankings of all methods across ten different regression datasets



Efficiency of organic photovoltaic molecules

Dataset: 50k/10k training/test points, 512-dim. binary input features



Summary

Novel approach for regression with Deep GPs.

- Inference based on approximate EP energy minimization.
- Extensive comparison to GPs and Bayesian neural networks.
- We obtain very good predictive performance.