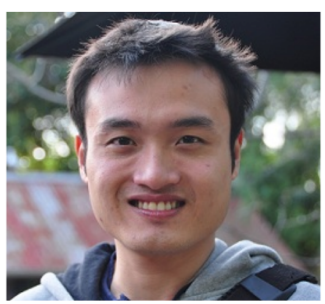
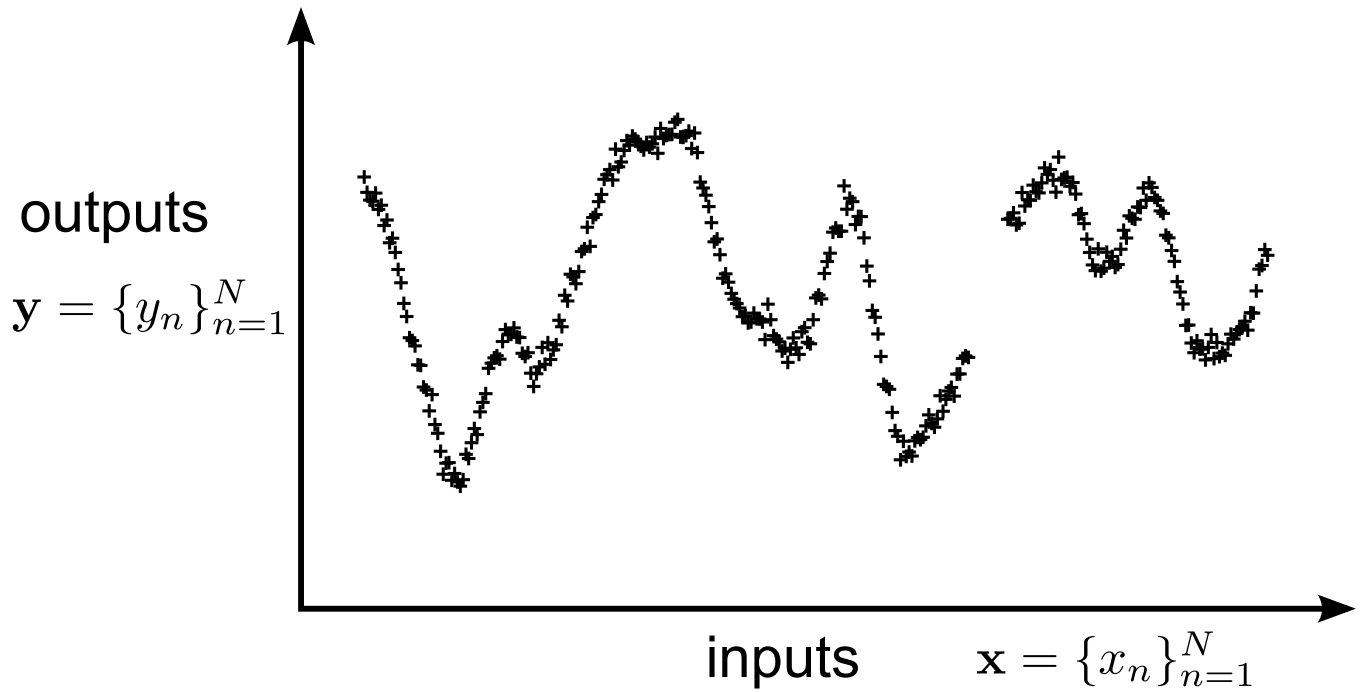
Power Expectation Propagation for Deep Gaussian Processes

Dr. Richard E. Turner (ret26@cam.ac.uk)
Computational and Biological Learning Lab, Department of Engineering, University of Cambridge

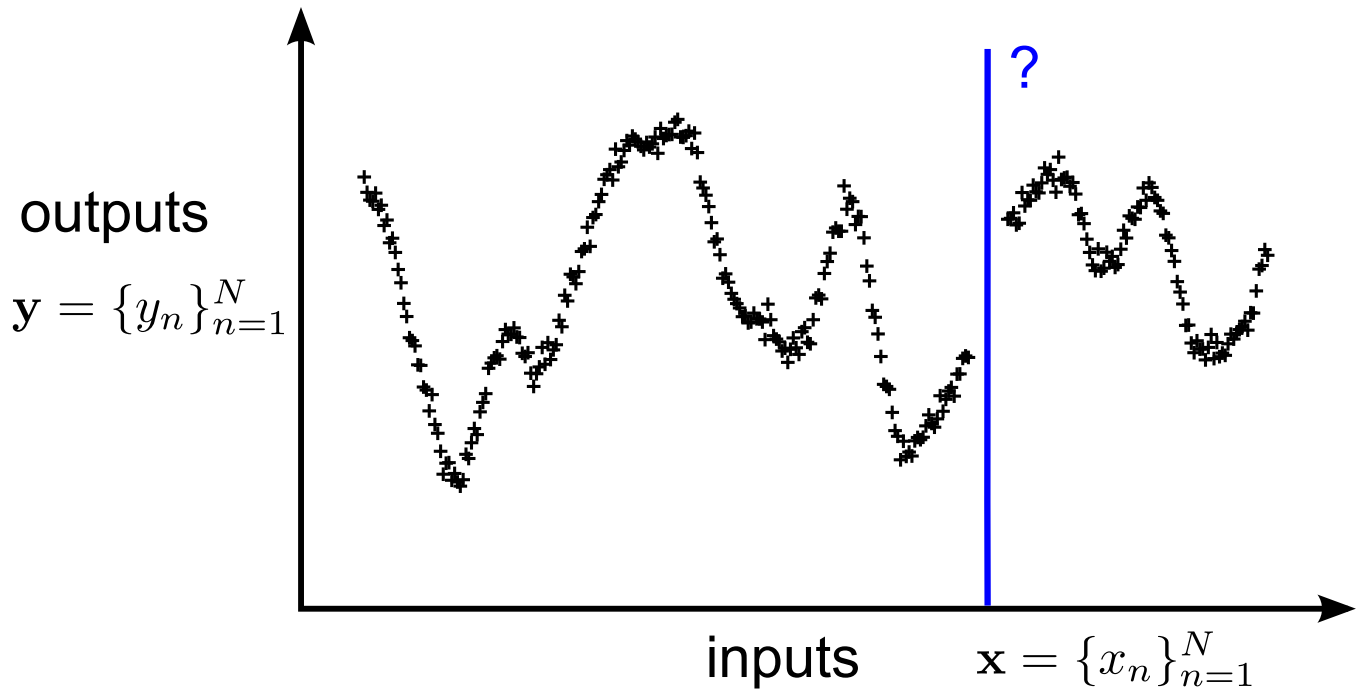
with Thang Bui, Yingzhen Li, José Miguel Hernández Lobato, Daniel Hernández Lobato, Josiah Jan




Motivation: Gaussian Process regression



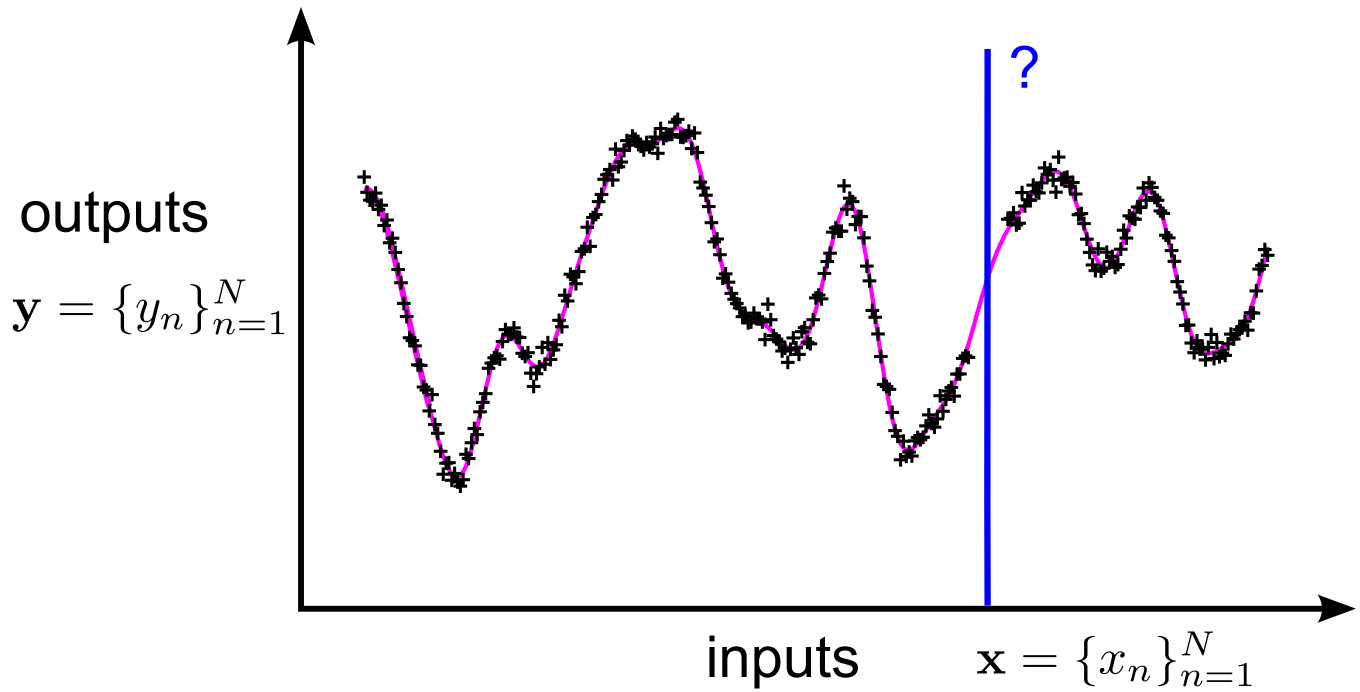
Motivation: Gaussian Process regression



Motivation: Gaussian Process regression

$$p(f|\theta) = \mathcal{GP}(f; 0, \mathrm{K}_\theta)$$

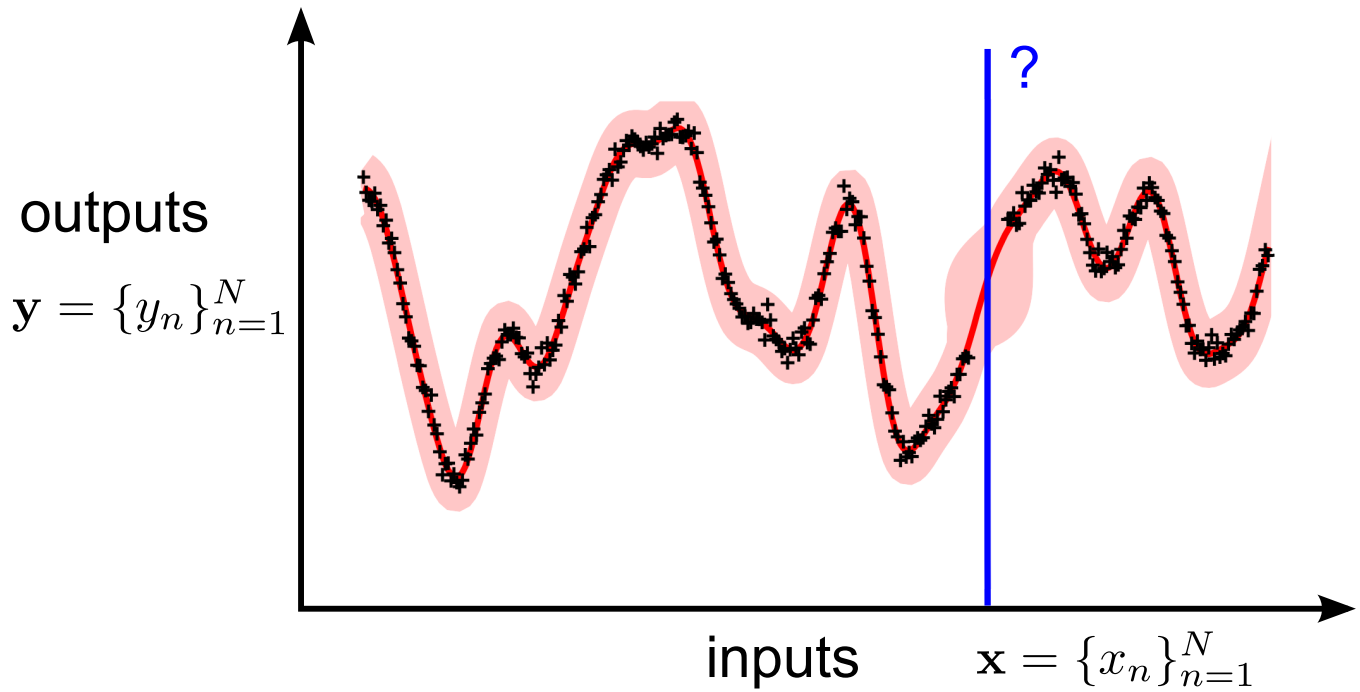
$$p(y_n|f, x_n, \theta)$$



Motivation: Gaussian Process regression

$$p(f|\theta) = \mathcal{GP}(f; 0, \mathrm{K}_\theta) \qquad \text{inference \& learning} \qquad p(f|\mathbf{y}, \mathbf{x}, \theta)$$

$$p(y_n|f, x_n, \theta) \qquad\qquad\qquad\qquad p(\mathbf{y}|\mathbf{x}, \theta)$$



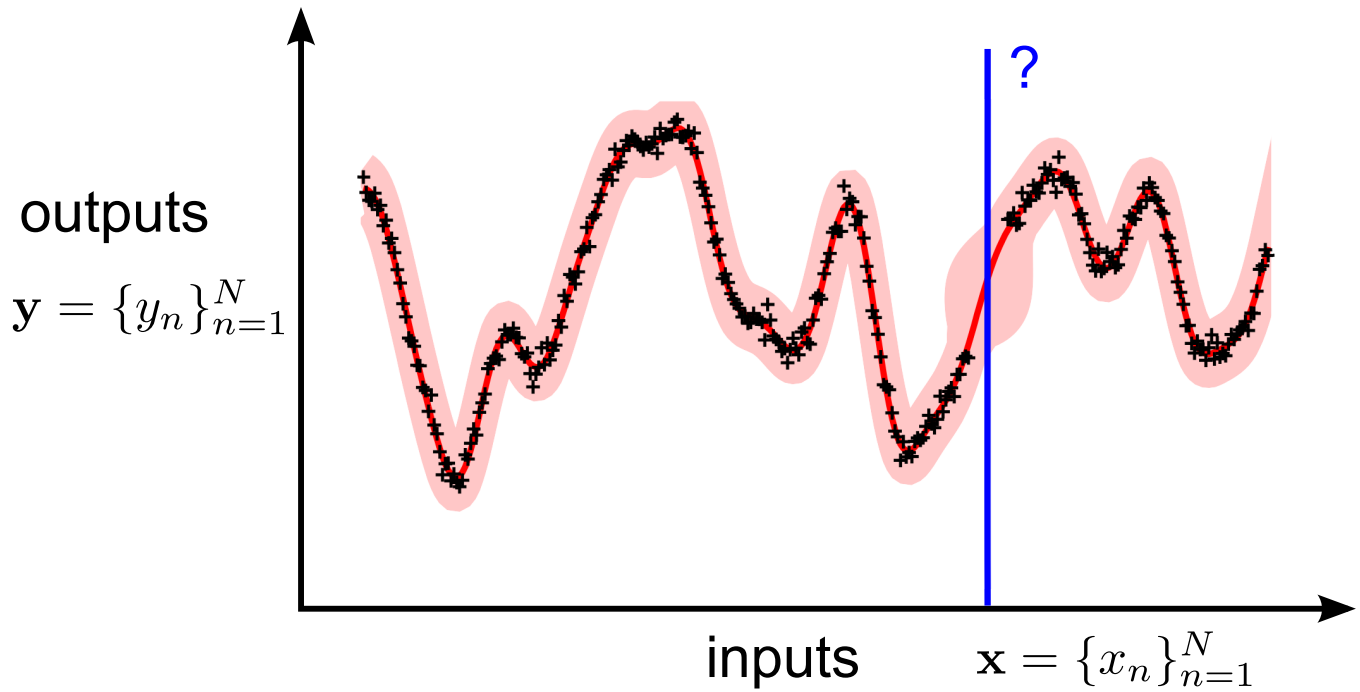
Motivation: Gaussian Process regression

$$p(f|\theta) = \mathcal{GP}(f; 0, \mathrm{K}_\theta) \qquad p(f|\mathbf{y}, \mathbf{x}, \theta)$$

$$p(y_n|f, x_n, \theta) \qquad p(\mathbf{y}|\mathbf{x}, \theta)$$

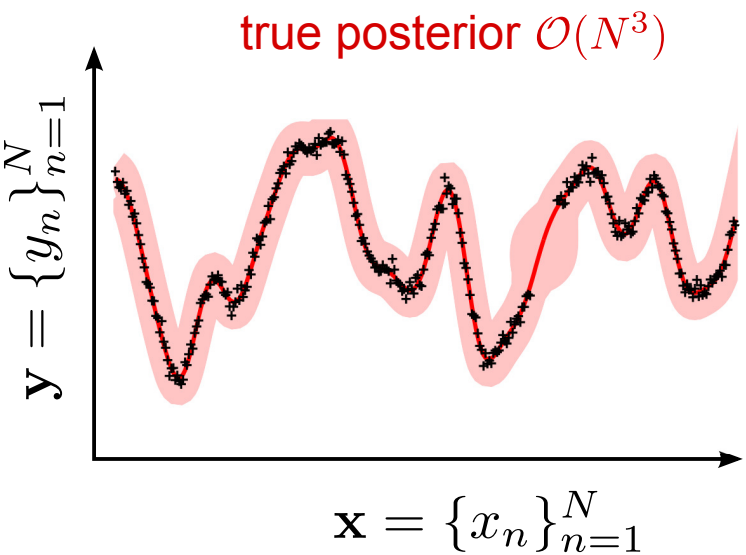
inference & learning

intractabilities
computational $\mathcal{O}(N^3)$
analytic



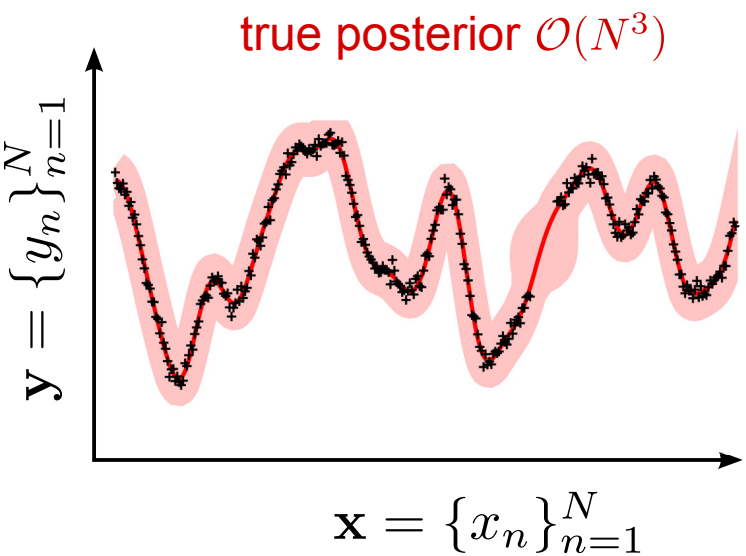
EP pseudo-point approximation

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



EP pseudo-point approximation

$$
\begin{aligned}
p^*(f) &= p(f, \mathbf{y}|\mathbf{x}, \theta) \\
&= p(f|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)
\end{aligned}
$$

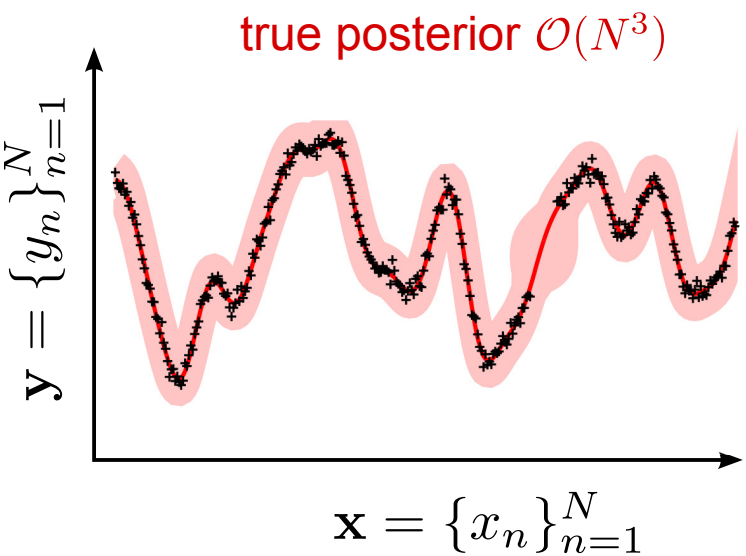


EP pseudo-point approximation

$$
\begin{aligned}
p^*(f) &= p(f, \mathbf{y}|\mathbf{x}, \theta) \\
&= p(f|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta) \\
&= p(\mathbf{y}|\mathbf{x}, \theta)\ p(f|\mathbf{y}, \mathbf{x}, \theta)
\end{aligned}
$$

marginal likelihood

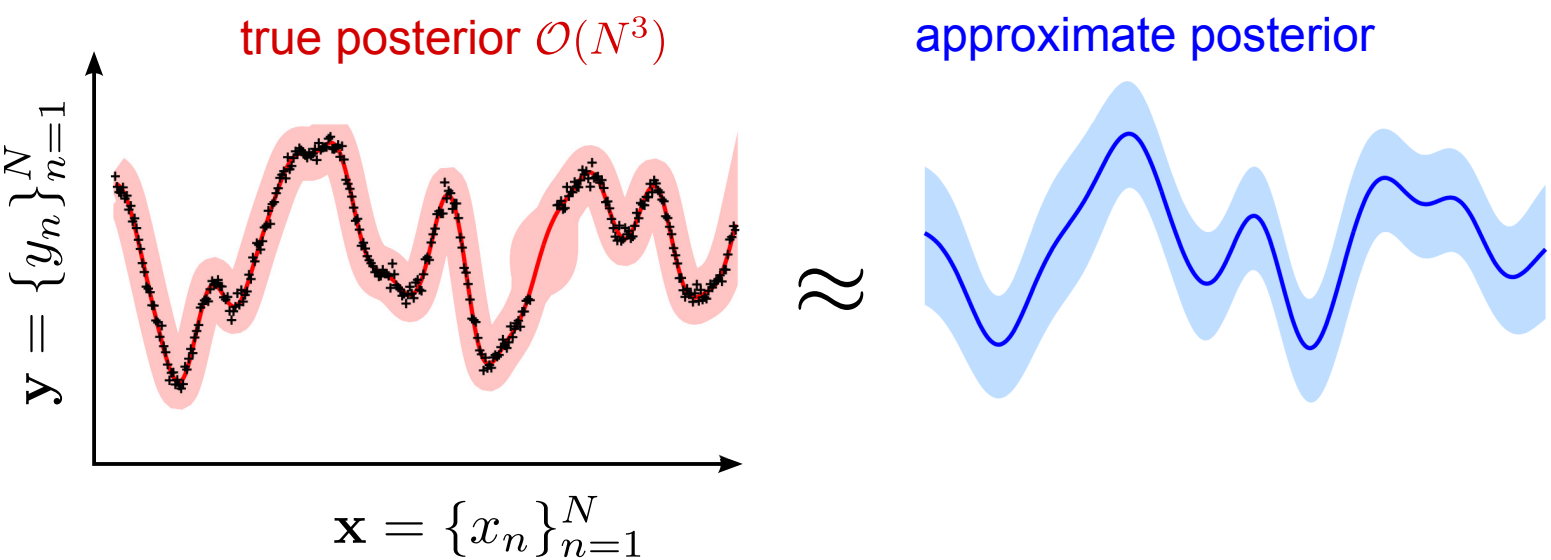
posterior



EP pseudo-point approximation

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta)$$
$$= p(f|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$
$$= \underbrace{p(\mathbf{y}|\mathbf{x}, \theta)}_{\text{marginal likelihood}} \underbrace{p(f|\mathbf{y}, \mathbf{x}, \theta)}_{\text{posterior}}$$

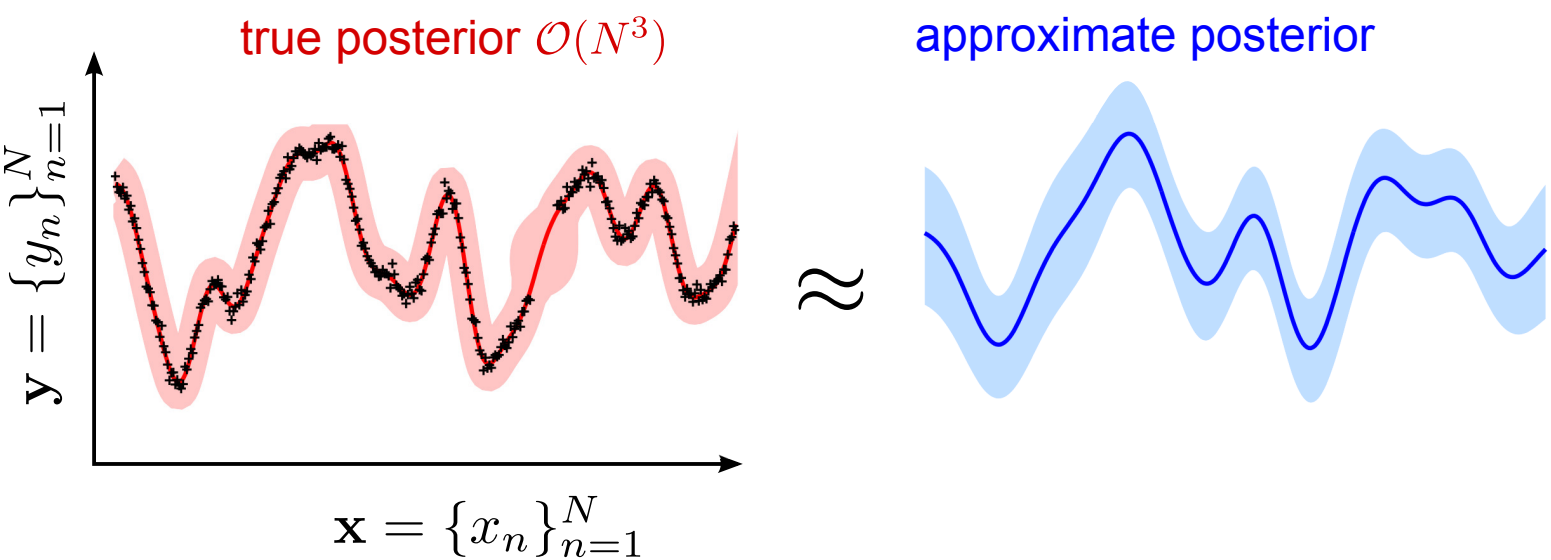
$$q^*(f) = p(f|\theta) \prod_{n=1}^{N} t_n(f)$$



EP pseudo-point approximation

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta)$$
$$= p(f|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$
$$= \underbrace{p(\mathbf{y}|\mathbf{x}, \theta)}_{\text{marginal likelihood}} \underbrace{p(f|\mathbf{y}, \mathbf{x}, \theta)}_{\text{posterior}}$$

$$q^*(f) = p(f|\theta) \prod_{n=1}^{N} t_n(f)$$
$$= Z_{\text{EP}}\, q(f)$$



EP pseudo-point approximation

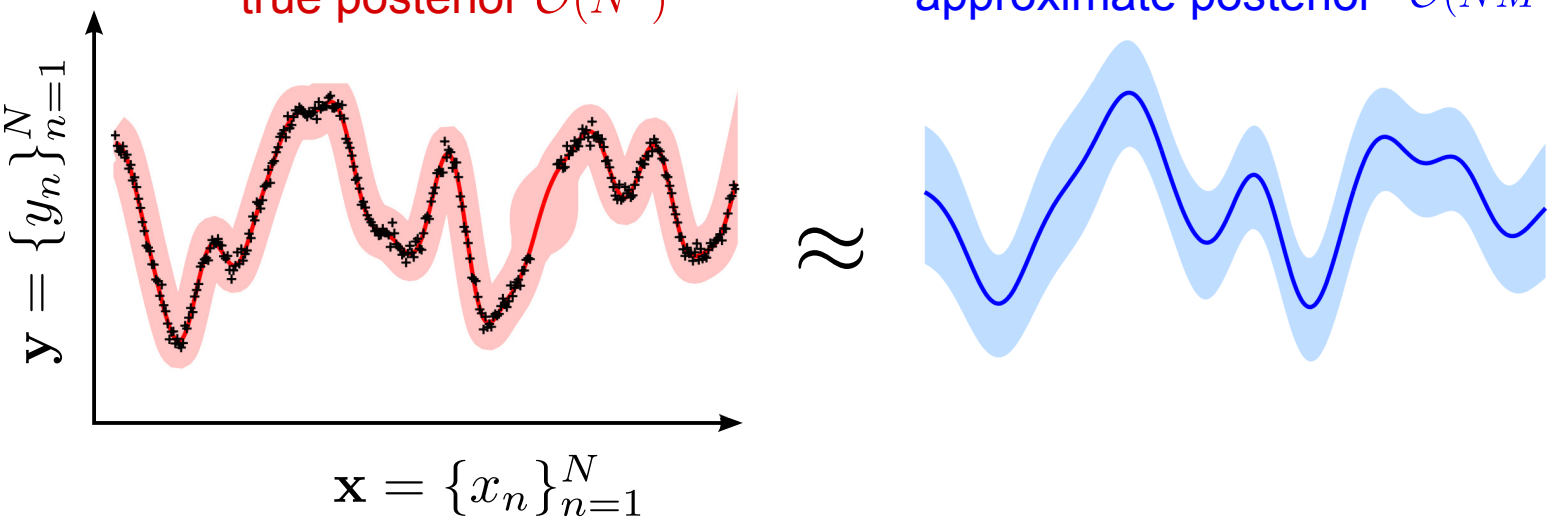
$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta)$$
$$= p(f|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$
$$= p(\mathbf{y}|\mathbf{x}, \theta)\ p(f|\mathbf{y}, \mathbf{x}, \theta)$$

marginal likelihood — posterior

$$q^*(f) = p(f|\theta) \prod_{n=1}^{N} t_n(f)$$
$$= Z_{\text{EP}}\ q(f)$$

$$t_n(f) = \mathcal{N}(\mathbf{u}; \mu_n, \Sigma_n)$$
$$\dim(\mathbf{u}) = M \quad f = \{\mathbf{u}, f_{\neq \mathbf{u}}\}$$

true posterior $\mathcal{O}(N^3)$ $\approx$ approximate posterior $\mathcal{O}(NM^2)$



EP pseudo-point approximation

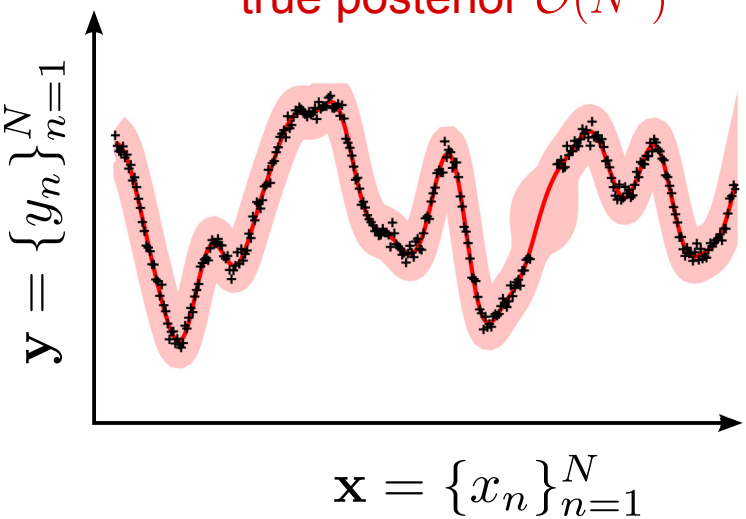
$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta)$$
$$= p(f|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$
$$= p(\mathbf{y}|\mathbf{x}, \theta)\ p(f|\mathbf{y}, \mathbf{x}, \theta)$$

marginal likelihood — posterior

$$q^*(f) = p(f|\theta)p(\tilde{\mathbf{y}}|\mathbf{u}, \tilde{\Sigma})$$
$$= p(f|\theta) \prod_{n=1}^{N} t_n(f)$$
$$= Z_{\text{EP}}\ q(f)$$

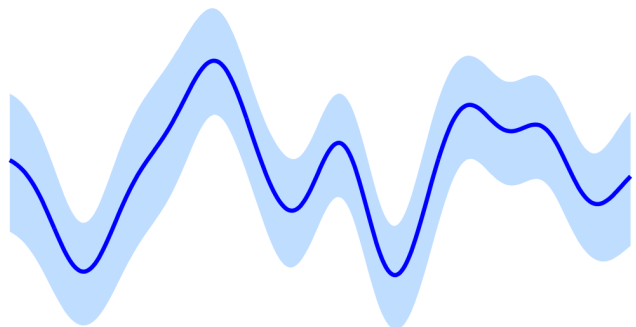
$$t_n(f) = \mathcal{N}(\mathbf{u}; \mu_n, \Sigma_n)$$
$$\dim(\mathbf{u}) = M \quad f = \{\mathbf{u}, f_{\neq \mathbf{u}}\}$$

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



$\approx$

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



EP pseudo-point approximation

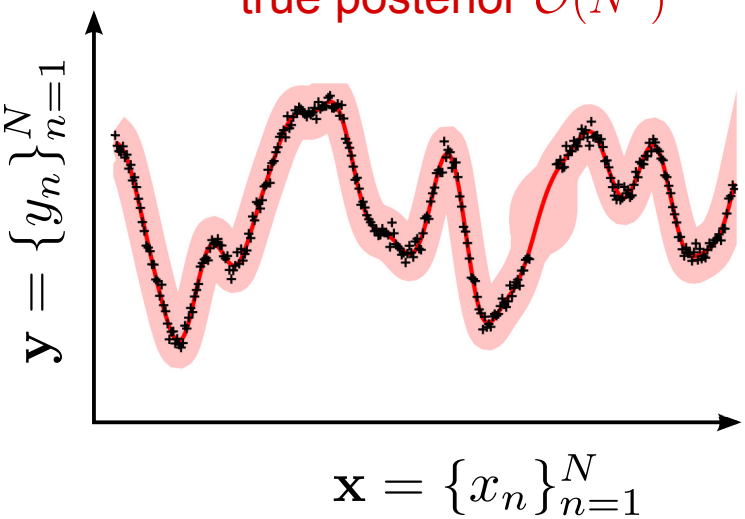
$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta)$$
$$= p(f|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$
$$= p(\mathbf{y}|\mathbf{x}, \theta)\ p(f|\mathbf{y}, \mathbf{x}, \theta)$$

marginal likelihood — posterior

$$q^*(f) = p(f|\theta)p(\tilde{\mathbf{y}}|\mathbf{u}, \tilde{\Sigma})$$
$$= p(f|\theta) \prod_{n=1}^{N} t_n(f)$$
$$= Z_{\text{EP}}\ q(f)$$

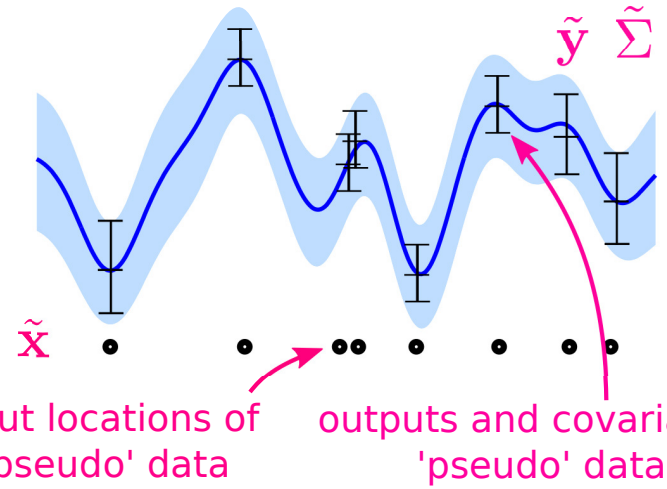
$$t_n(f) = \mathcal{N}(\mathbf{u}; \mu_n, \Sigma_n)$$
$$\dim(\mathbf{u}) = M \quad f = \{\mathbf{u}, f_{\neq \mathbf{u}}\}$$

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



$\approx$

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



EP algorithm

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one pseudo-observation likelihood

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one pseudo-observation likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$$

tilted

add in one true observation likelihood

EP algorithm

1. remove $q^{\backslash n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$ — take out one pseudo-observation likelihood

cavity

2. include $p_n^{\text{tilt}}(f) = q^{\backslash n}(f)p(y_n|f, x_n, \theta)$ — add in one true observation likelihood

tilted

3. project $q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \mathrm{KL}\left[p_n^{\text{tilt}}(f)||q^*(f)\right]$ — project onto approximating family

KL between unnormalised stochastic processes

EP algorithm

1. remove $q^{\backslash n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$ — take out one pseudo-observation likelihood

cavity

2. include $p_n^{\text{tilt}}(f) = q^{\backslash n}(f)p(y_n|f, x_n, \theta)$ — add in one true observation likelihood

tilted

3. project $q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \mathrm{KL}\left[p_n^{\text{tilt}}(f)||q^*(f)\right]$ — project onto approximating family

KL between unnormalised stochastic processes

4. update $t_n(\mathbf{u}) = \frac{q^*(f)}{q^{\backslash n}(f)}$ — update pseudo-observation likelihood

EP algorithm

1. remove $q^{\backslash n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$ — take out one pseudo-observation likelihood

cavity

2. include $p_n^{\text{tilt}}(f) = q^{\backslash n}(f)p(y_n|f, x_n, \theta)$ — add in one true observation likelihood

tilted

3. project $q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \mathrm{KL}\left[p_n^{\text{tilt}}(f)||q^*(f)\right]$ — project onto approximating family

KL between unnormalised stochastic processes

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$
2. Gaussian regression: matches moments everywhere

4. update $t_n(\mathbf{u}) = \frac{q^*(f)}{q^{\backslash n}(f)}$ — update pseudo-observation likelihood

EP algorithm

1. remove $q^{\backslash n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$ — take out one pseudo-observation likelihood

cavity

2. include $p_n^{\text{tilt}}(f) = q^{\backslash n}(f)p(y_n|f, x_n, \theta)$ — add in one true observation likelihood

tilted

3. project $q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \mathrm{KL}\left[p_n^{\text{tilt}}(f)||q^*(f)\right]$ — project onto approximating family

KL between unnormalised stochastic processes

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$
2. Gaussian regression: matches moments everywhere

4. update $t_n(\mathbf{u}) = \frac{q^*(f)}{q^{\backslash n}(f)} = z_n \mathcal{N}(\mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; g_n, v_n)$ — update pseudo-observation likelihood

rank 1

Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u})$$

suppressed θ & x_n

Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

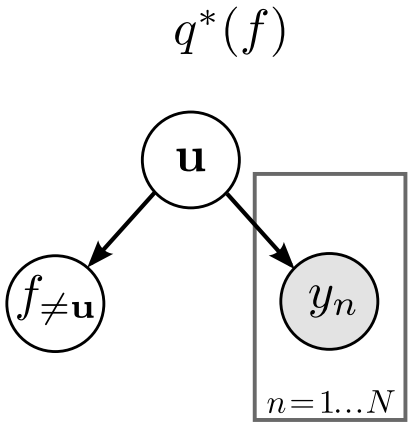
$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u})$$

suppressed θ & x_n



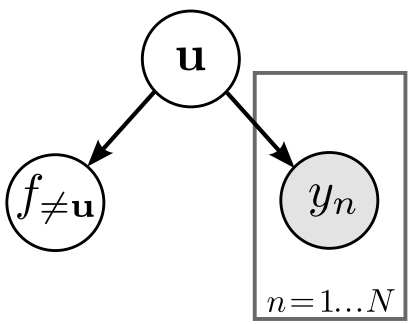
Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

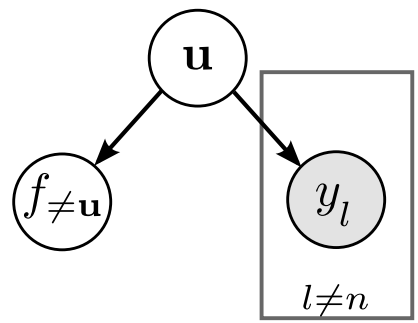
$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u})$$

suppressed θ & x_n

$q^*(f)$



$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

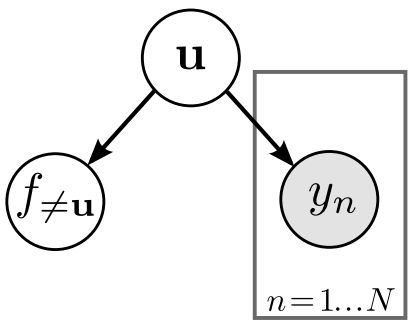


Fixed points of EP = FITC approximation

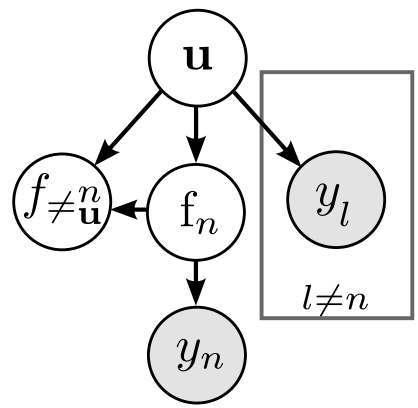
$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$q^*(f)$



$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$



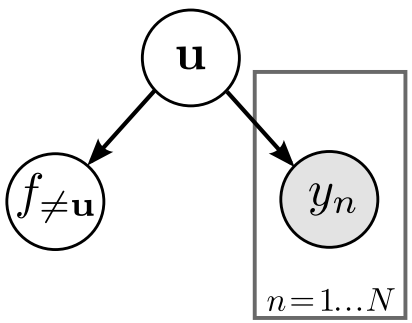
Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

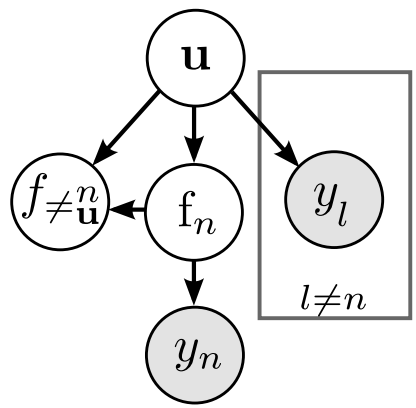
$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$q^*(f)$



$p_n^{\mathrm{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$



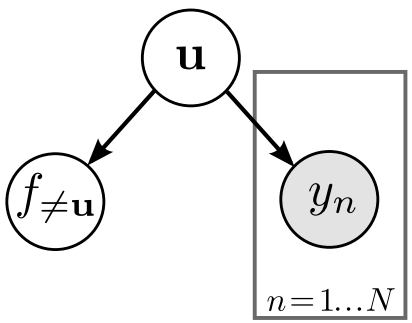
Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

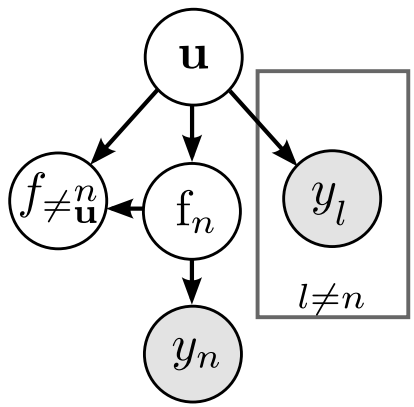
$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f)$



$\int \mathrm{d}f_{\neq\mathbf{u}}\; p_n^{\mathrm{tilt}}(f)$



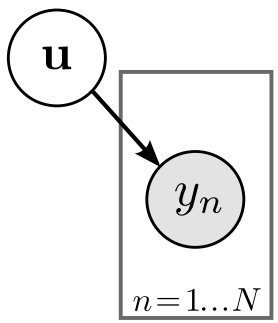
Fixed points of EP = FITC approximation

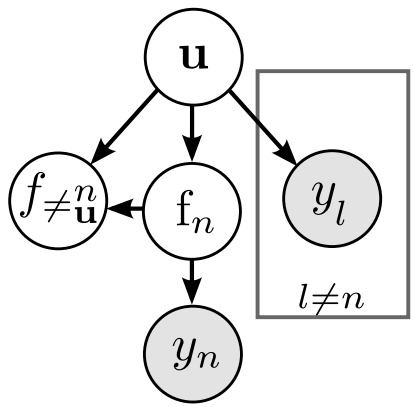
$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f) \qquad \int \mathrm{d}f_{\neq\mathbf{u}}\, p_n^{\mathrm{tilt}}(f)$$





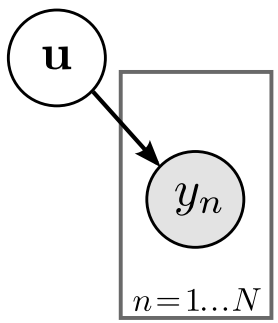
Fixed points of EP = FITC approximation

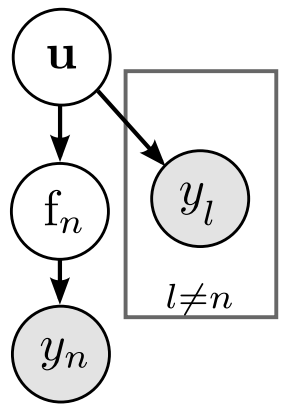
$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f) \qquad \int \mathrm{d}f_{\neq\mathbf{u}}\, p_n^{\mathrm{tilt}}(f)$$





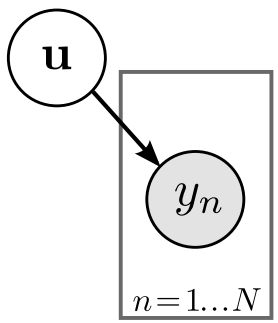
Fixed points of EP = FITC approximation

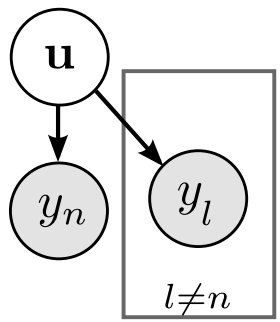
$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\text{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f) \qquad \int \mathrm{d}f_{\neq\mathbf{u}}\, p_n^{\text{tilt}}(f)$$





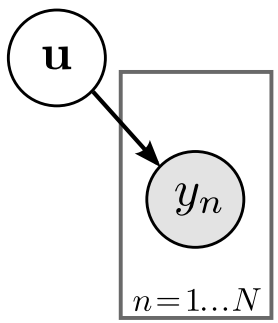
Fixed points of EP = FITC approximation

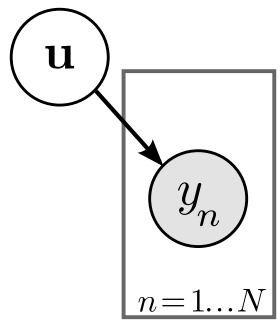
$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f) \qquad \int \mathrm{d}f_{\neq\mathbf{u}}\, p_n^{\mathrm{tilt}}(f)$$





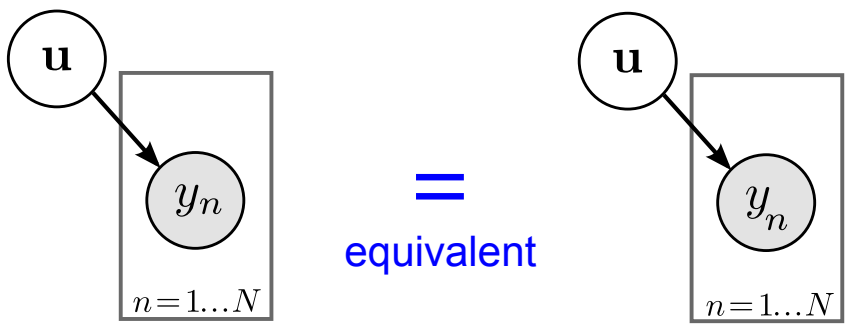
Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f) \qquad\qquad \int \mathrm{d}f_{\neq\mathbf{u}}\, p_n^{\mathrm{tilt}}(f)$$



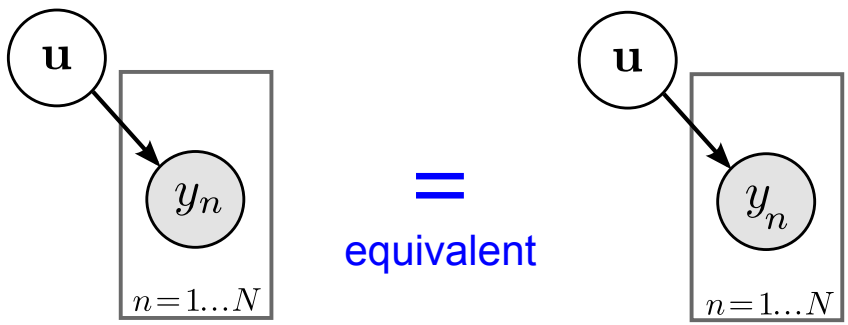
Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f) \qquad \int \mathrm{d}f_{\neq\mathbf{u}}\, p_n^{\mathrm{tilt}}(f)$$



Csato & Opper (2002)

Qi, Abdel-Gawad & Minka (2010)

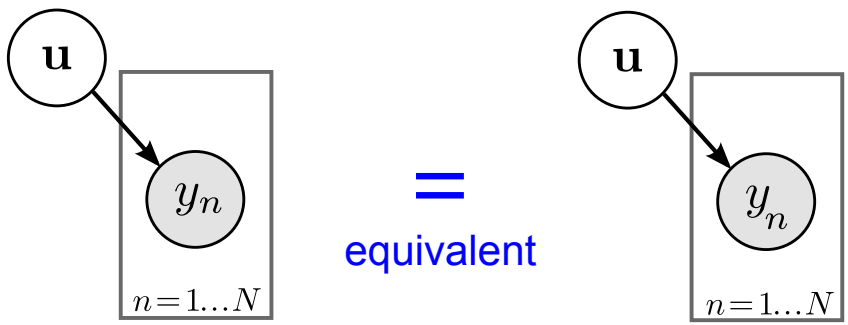
Fixed points of EP = FITC approximation

$$t_n(\mathbf{u}) = p(y_n|\mathbf{u}, x_n, \theta) = \mathcal{N}(y_n; \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathrm{K}_{f_nf_n} - \mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathrm{K}_{\mathbf{u}f_n} + \sigma_y^2)$$

$$q^*(f) = p(f)\prod_{n=1}^{N} t_n(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})\prod_{n=1}^{N} p(y_n|\mathbf{u}) \qquad \text{suppressed } \theta \text{ \& } x_n$$

$$p_n^{\mathrm{tilt}}(f) = p(f)p(y_n|f)\prod_{l\neq n} t_l(\mathbf{u}) = p(f_{\neq\mathbf{u}}|\mathbf{u})p(\mathbf{u})p(y_n|f)\prod_{l\neq n} p(y_l|\mathbf{u})$$

$$\int \mathrm{d}f_{\neq\mathbf{u}}\, q^*(f) \qquad \int \mathrm{d}f_{\neq\mathbf{u}}\, p_n^{\mathrm{tilt}}(f)$$



Csato & Opper (2002)

Qi, Abdel-Gawad & Minka (2010)

Interpretation resolves philosophical issues with FITC (increase M with N)
FITC known to overfit => EP over-estimates marginal likelihood

EP algorithm

1. remove $q^{\backslash n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$ — take out one pseudo-observation likelihood

cavity

2. include $p_n^{\text{tilt}}(f) = q^{\backslash n}(f)p(y_n|f, x_n, \theta)$ — add in one true observation likelihood

tilted

3. project $q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \mathrm{KL}\left[p_n^{\text{tilt}}(f)||q^*(f)\right]$ — project onto approximating family

KL between unnormalised stochastic processes

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$
2. Gaussian regression: matches moments everywhere

4. update $t_n(\mathbf{u}) = \frac{q^*(f)}{q^{\backslash n}(f)} = z_n \mathcal{N}(\mathrm{K}_{f_n\mathbf{u}}\mathrm{K}_{\mathbf{uu}}^{-1}\mathbf{u}; g_n, v_n)$ — update pseudo-observation likelihood

rank 1

Power EP algorithm (as tractable as EP)

1. remove $q^{\backslash n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})^\alpha}$ — take out fraction of pseudo-observation likelihood

cavity

2. include $p_n^{\text{tilt}}(f) = q^{\backslash n}(f) p(y_n|f, x_n, \theta)^\alpha$ — add in fraction of true observation likelihood

tilted

3. project $q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \mathrm{KL}\left[p_n^{\text{tilt}}(f) || q^*(f)\right]$ — project onto approximating family

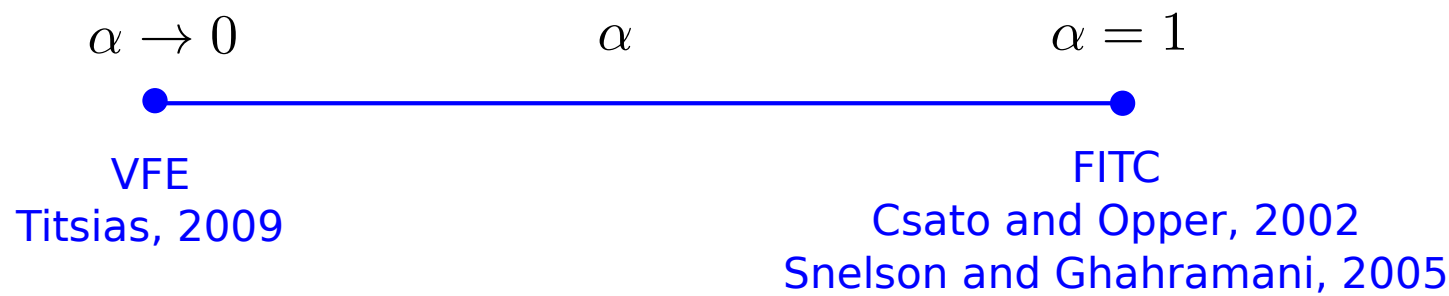
KL between unnormalised stochastic processes

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$
2. Gaussian regression: matches moments everywhere

4. update $t_n(\mathbf{u})^\alpha = \frac{q^*(f)}{q^{\backslash n}(f)}$ — update pseudo-observation likelihood

$t_n(\mathbf{u}) = z_n \mathcal{N}(\mathrm{K}_{f_n \mathbf{u}} \mathrm{K}_{\mathbf{uu}}^{-1} \mathbf{u}; g_n, v_n)$ — rank 1

Power EP: a unifying framework



$$t_n(\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; y_n, \alpha D_{\mathrm{f}_n\mathrm{f}_n} + \sigma_y^2)$$

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{K}_{\mathbf{fu}})$$

$$\log \mathcal{Z}_{\mathrm{PEP}} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^{\intercal}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y} + \frac{1-\alpha}{2\alpha}\sum_n \log\left(1 + \alpha D_{\mathrm{f}_n\mathrm{f}_n}/\sigma_y^2\right)$$

$$\overline{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \alpha\mathrm{diag}(\mathbf{D}_{\mathbf{ff}}) + \sigma_y^2\mathrm{I} \qquad \mathbf{D}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}$$

Power EP: a unifying framework

Approximate blocks of data: structured approximations

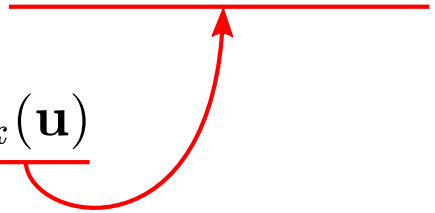
$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta) = p(f|\theta) \prod_{k=1}^{K} \prod_{n \in \mathcal{K}_n} p(y_n|f, x_n, \theta)$$

Power EP: a unifying framework

Approximate blocks of data: structured approximations

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta) = p(f|\theta) \prod_{k=1}^{K} \prod_{n \in \mathcal{K}_n} p(y_n|f, x_n, \theta)$$

$$q^*(f) = p(f|\theta) \prod_{k=1}^{K} t_k(\mathbf{u})$$



Power EP: a unifying framework

Approximate blocks of data: structured approximations

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta) = p(f|\theta) \prod_{k=1}^{K} \prod_{n \in \mathcal{K}_n} p(y_n|f, x_n, \theta)$$

$$q^*(f) = p(f|\theta) \prod_{k=1}^{K} t_k(\mathbf{u})$$

$\alpha = 1$

PITC / BCM
Schwaighofer & Tresp, 2002, Snelson 2006,

$\alpha \to 0$

VFE
Titsias, 2009

Power EP: a unifying framework

Approximate blocks of data: structured approximations

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta) = p(f|\theta) \prod_{k=1}^{K} \prod_{n \in \mathcal{K}_n} p(y_n|f, x_n, \theta)$$

$$q^*(f) = p(f|\theta) \prod_{k=1}^{K} t_k(\mathbf{u})$$

$\alpha = 1$

PITC / BCM
Schwaighofer & Tresp, 2002, Snelson 2006,

$\alpha \to 0$

VFE
Titsias, 2009

Place pseudo-data in different space: interdomain transformations

$$g(z) = \int w(z, z') f(z') \mathrm{d}z' \quad \text{(linear transform)}$$

Power EP: a unifying framework

Approximate blocks of data: structured approximations

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta) = p(f|\theta) \prod_{k=1}^{K} \prod_{n \in \mathcal{K}_n} p(y_n|f, x_n, \theta)$$

$$q^*(f) = p(f|\theta) \prod_{k=1}^{K} t_k(\mathbf{u})$$

$\alpha = 1$

PITC / BCM
Schwaighofer & Tresp, 2002, Snelson 2006,

$\alpha \to 0$

VFE
Titsias, 2009

Place pseudo-data in different space: interdomain transformations

$$g(z) = \int w(z, z') f(z') \mathrm{d}z' \quad \text{(linear transform)}$$

$$p^*(f, g) = p(f, g|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$

Power EP: a unifying framework

Approximate blocks of data: structured approximations

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta) = p(f|\theta) \prod_{k=1}^{K} \prod_{n \in \mathcal{K}_n} p(y_n|f, x_n, \theta)$$

$$q^*(f) = p(f|\theta) \prod_{k=1}^{K} t_k(\mathbf{u})$$

$\alpha = 1$

PITC / BCM
Schwaighofer & Tresp, 2002, Snelson 2006,

$\alpha \to 0$

VFE
Titsias, 2009

Place pseudo-data in different space: interdomain transformations

$$g(z) = \int w(z, z') f(z') \mathrm{d}z' \quad \text{(linear transform)}$$

$$p^*(f, g) = p(f, g|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$

$$q^*(f, g) = p(f, g|\theta) \prod_{n=1}^{N} t_n(\mathbf{u})$$

pseudo-data in new space

$g = \{\mathbf{u}, g_{\neq \mathbf{u}}\}$

Power EP: a unifying framework

Approximate blocks of data: structured approximations

$$p^*(f) = p(f, \mathbf{y}|\mathbf{x}, \theta) = p(f|\theta) \prod_{k=1}^{K} \prod_{n \in \mathcal{K}_n} p(y_n|f, x_n, \theta)$$

$$q^*(f) = p(f|\theta) \prod_{k=1}^{K} t_k(\mathbf{u})$$

$\alpha = 1$

PITC / BCM
Schwaighofer & Tresp, 2002, Snelson 2006,

$\alpha \to 0$

VFE
Titsias, 2009

Place pseudo-data in different space: interdomain transformations

$$g(z) = \int w(z, z') f(z') \mathrm{d}z' \quad \text{(linear transform)}$$

$$p^*(f, g) = p(f, g|\theta) \prod_{n=1}^{N} p(y_n|f, x_n, \theta)$$

$$q^*(f, g) = p(f, g|\theta) \prod_{n=1}^{N} t_n(\mathbf{u})$$

pseudo-data in new space

$g = \{\mathbf{u}, g_{\neq \mathbf{u}}\}$

$\alpha = 1$

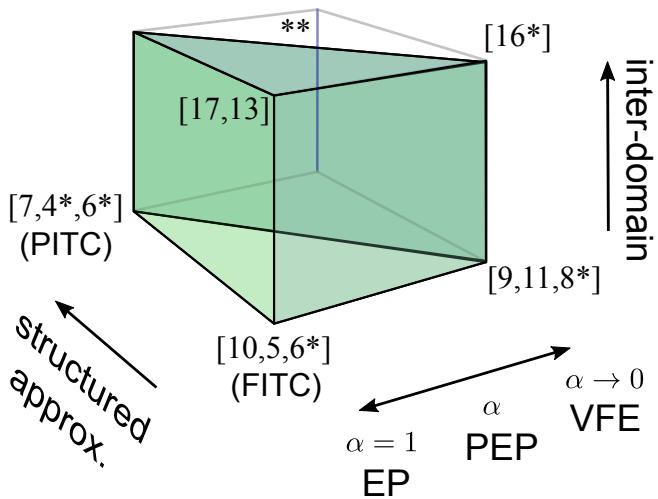
Figueiras-Vidal & Lázaro-Gredilla 2009

$\alpha \to 0$

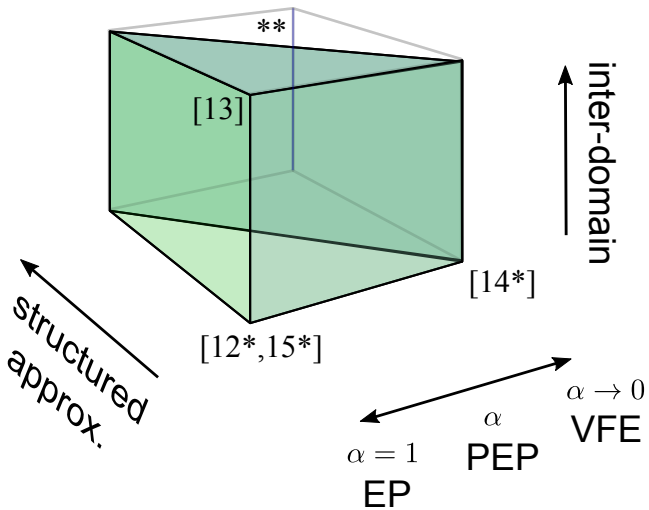
Tobar et al. 2015
Matthews et al, 2016

Power EP: a unifying framework

GP Regression



GP Classification

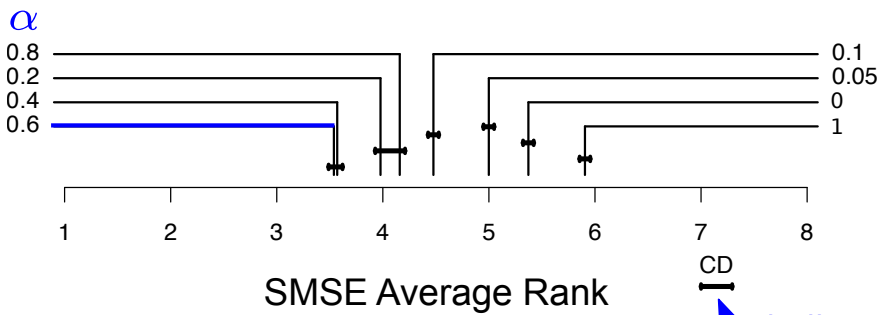


[4] Quiñonero-Candela et al. 2005
[5] Snelson et al., 2005
[6] Snelson, 2006
[7] Schwaighofer, 2002
[8] Titsias, 2009
[9] Csató, 2002
[10] Csató et al., 2002
[11] Seeger et al., 2003
[12] Naish-Guzman et al, 2007
[13] Qi et al., 2010
[14] Hensman et al., 2015
[15] Hernández-Lobato et al., 2016
[16] Matthews et al., 2016
[17] Figueiras-Vidal et al., 2009

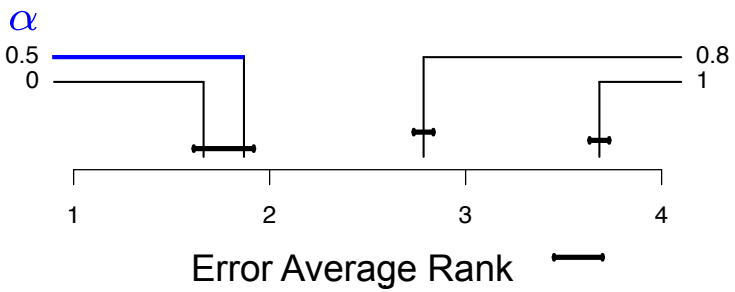
* = optimised pseudo-inputs
** = structured versions of VFE recover VFE

How should I set the power parameter α?

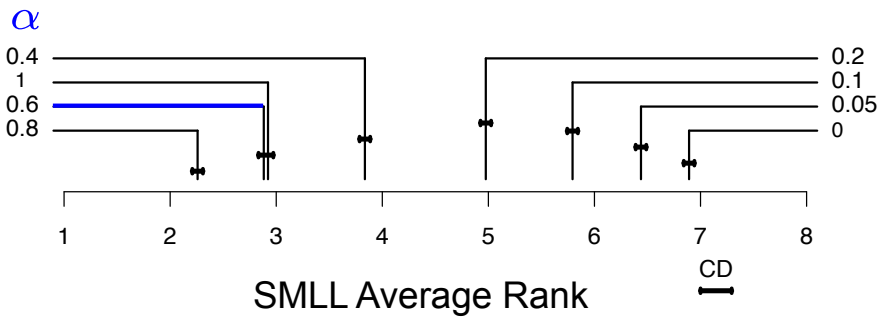
8 UCI regression datasets
20 random splits
M = 0 - 200
hypers and inducing inputs optimised

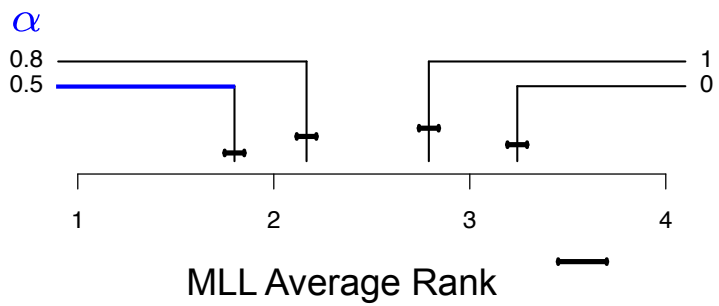


6 UCI classification datasets
20 random splits
M = 10, 50, 100
hypers and inducing inputs optimised



indicates significant difference

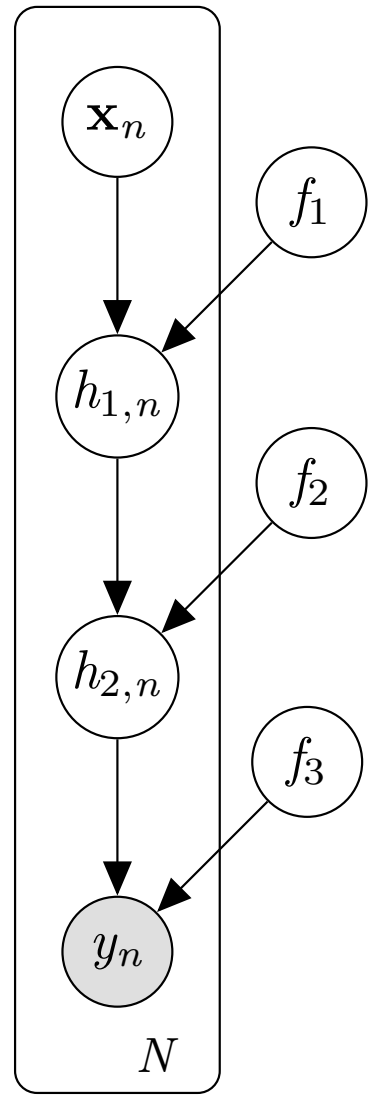




α = 0.5 does well over all

Deep Gaussian processes

$$f_l \sim \mathcal{GP}(0, k(.,.))$$
$$y_n = g(\mathbf{x}_n) = f_L(f_{L-1}(\cdots f_2(f_1(\mathbf{x}_n)))) + \epsilon_n$$
$$h_{L-1,n} \coloneqq f_{L-1}(\cdots f_1(\mathbf{x}_n)), y_n = f_L(h_{L-1,n}) + \epsilon_n$$



Deep GPs[a] are

- multi-layer generalisation of Gaussian processes,
- equivalent to deep neural networks with infinitely wide hidden layers

Questions:

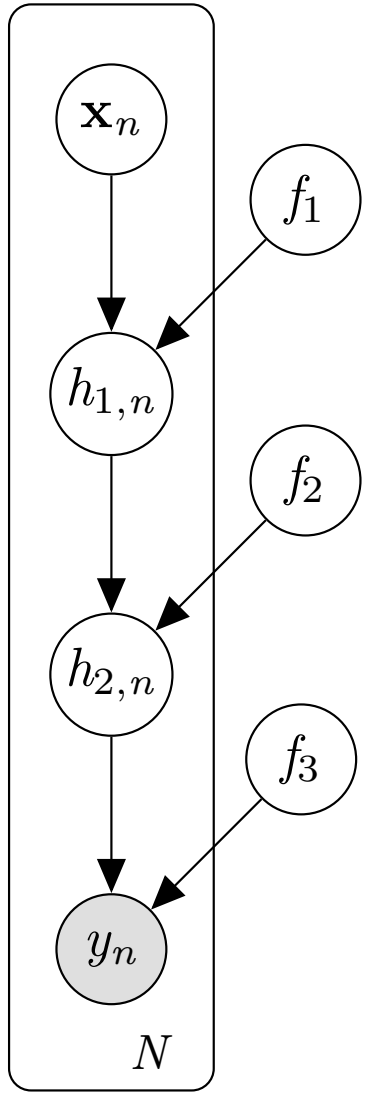
- How to perform inference and learning tractably?
- How Deep GPs compare to alternative, e.g. Bayesian neural networks?

[a] Damianou and Lawrence (2013) [unsupervised learning]

Pros and cons of Deep GPs

Why deep GPs? Because

- deep and nonparametric,
- discover useful input warping or dimensionality compression/expansion, i.e. automatic, nonparametric Bayesian kernel design,
- give a non-Gaussian functional mapping g,



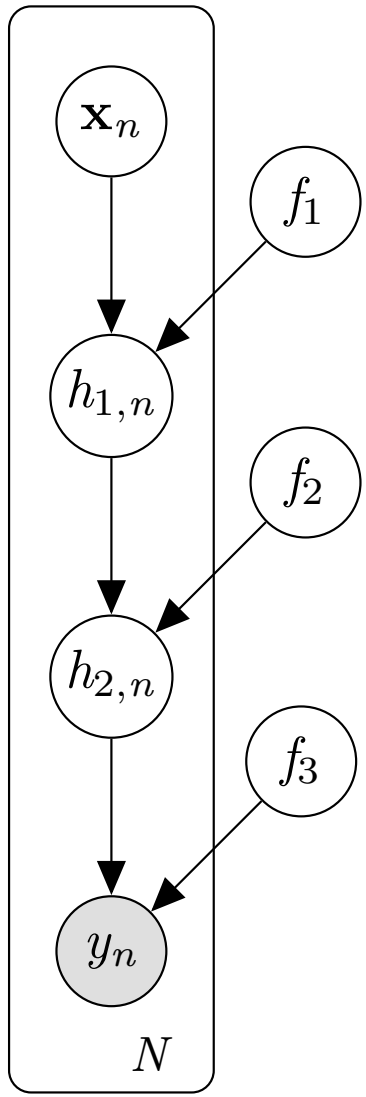
Pros and cons of Deep GPs

Why deep GPs? Because

- deep and nonparametric,
- discover useful input warping or dimensionality compression/expansion, i.e. automatic, nonparametric Bayesian kernel design,
- give a non-Gaussian functional mapping g,

Drawbacks:

- bottleneck in hierarchy? need medium/high dimensional hidden layers, skip links
- too flexible?
  - how to incorporate prior knowledge, e.g. invariance
  - learnability/identifiability



Power EP for Deep GPs

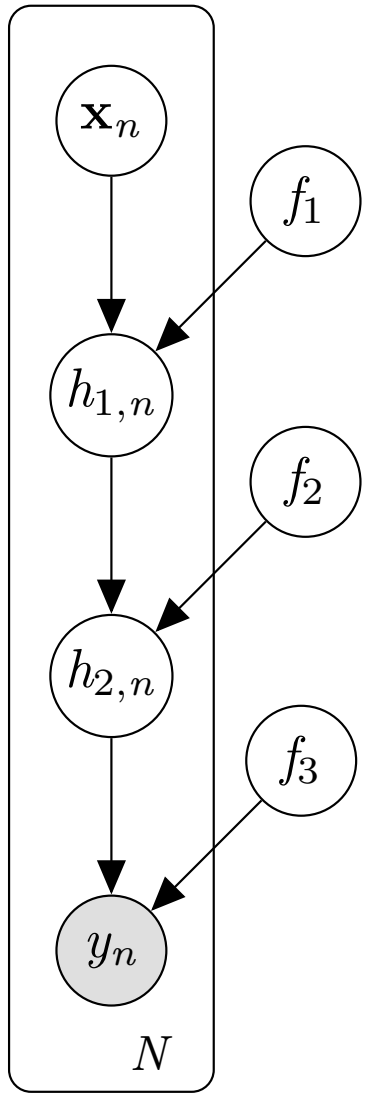
Joint distribution:

$$p(f_1)p(f_2)p(f_3)\prod_n p(y_n|h_{2,n},f_3)p(h_{2,n}|h_{1,n},f_2)p(h_{1,n}|f_1,x_n)$$

EP approximation:

$$p(f_1)p(f_2)p(f_3)\prod_n s_{3,n}(h_{2,n})t_{3,n}(\mathbf{u}_3)r_{2,n}(h_{2,n})s_{2,n}(h_{1,n})t_{2,n}(\mathbf{u}_2)$$

$$\times\, r_{1,n}(h_{1,n})t_{1,n}(\mathbf{u}_1)$$



Power EP for Deep GPs

Joint distribution:

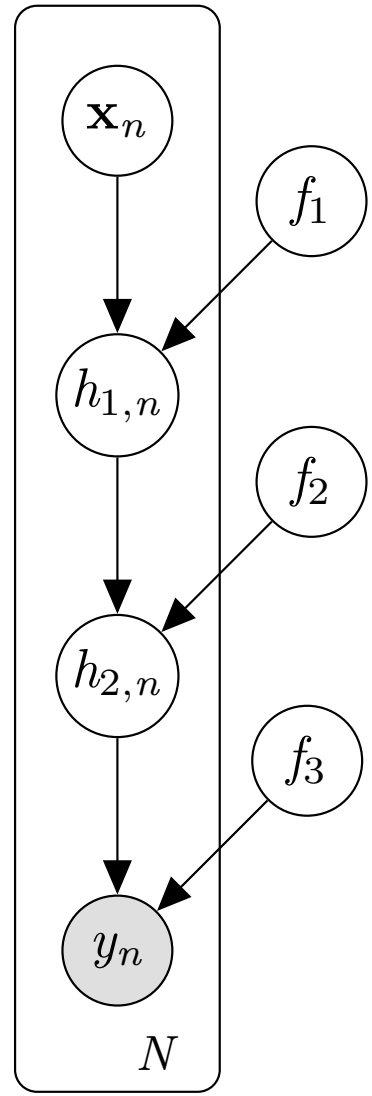
$$p(f_1)p(f_2)p(f_3)\prod_n p(y_n|h_{2,n},f_3)p(h_{2,n}|h_{1,n},f_2)p(h_{1,n}|f_1,x_n)$$

EP approximation:

$$p(f_1)p(f_2)p(f_3)\prod_n s_{3,n}(h_{2,n})t_{3,n}(\mathbf{u}_3)r_{2,n}(h_{2,n})s_{2,n}(h_{1,n})t_{2,n}(\mathbf{u}_2)$$
$$\times\, r_{1,n}(h_{1,n})t_{1,n}(\mathbf{u}_1)$$

Power EP:

- initialise with all approximate factors $= 1$
- incorporate $p(h_{1,n}|f_1,x_n)$, update $r_{1,n}(h_{1,n})t_{1,n}(\mathbf{u}_1)$
- incorporate $p(h_{2,n}|h_{1,n},f_2)$, update $r_{2,n}(h_{2,n})s_{2,n}(h_{1,n})t_{2,n}(\mathbf{u}_2)$
- incorporate $p(y_n|h_{2,n},f_3)$, update $s_{3,n}(h_{2,n})t_{3,n}(\mathbf{u}_3)$



Power EP for Deep GPs

Joint distribution:

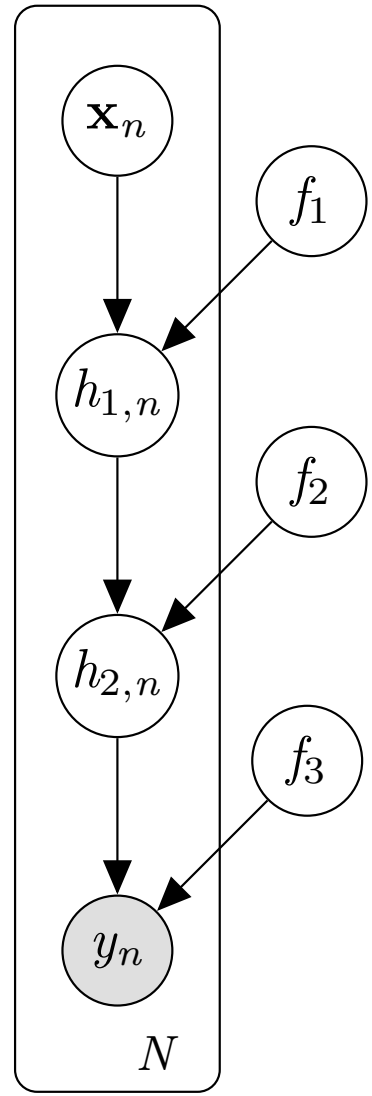
$$p(f_1)p(f_2)p(f_3)\prod_n p(y_n|h_{2,n},f_3)p(h_{2,n}|h_{1,n},f_2)p(h_{1,n}|f_1,x_n)$$

EP approximation:

$$p(f_1)p(f_2)p(f_3)\prod_n s_{3,n}(h_{2,n})t_{3,n}(\mathbf{u}_3)r_{2,n}(h_{2,n})s_{2,n}(h_{1,n})t_{2,n}(\mathbf{u}_2)$$
$$\times\, r_{1,n}(h_{1,n})t_{1,n}(\mathbf{u}_1)$$

Power EP:

- initialise with all approximate factors $= 1$
- incorporate $p(h_{1,n}|f_1,x_n)$, update $r_{1,n}(h_{1,n})t_{1,n}(\mathbf{u}_1)$
- incorporate $p(h_{2,n}|h_{1,n},f_2)$, update $r_{2,n}(h_{2,n})s_{2,n}(h_{1,n})t_{2,n}(\mathbf{u}_2)$
- incorporate $p(y_n|h_{2,n},f_3)$, update $s_{3,n}(h_{2,n})t_{3,n}(\mathbf{u}_3)$



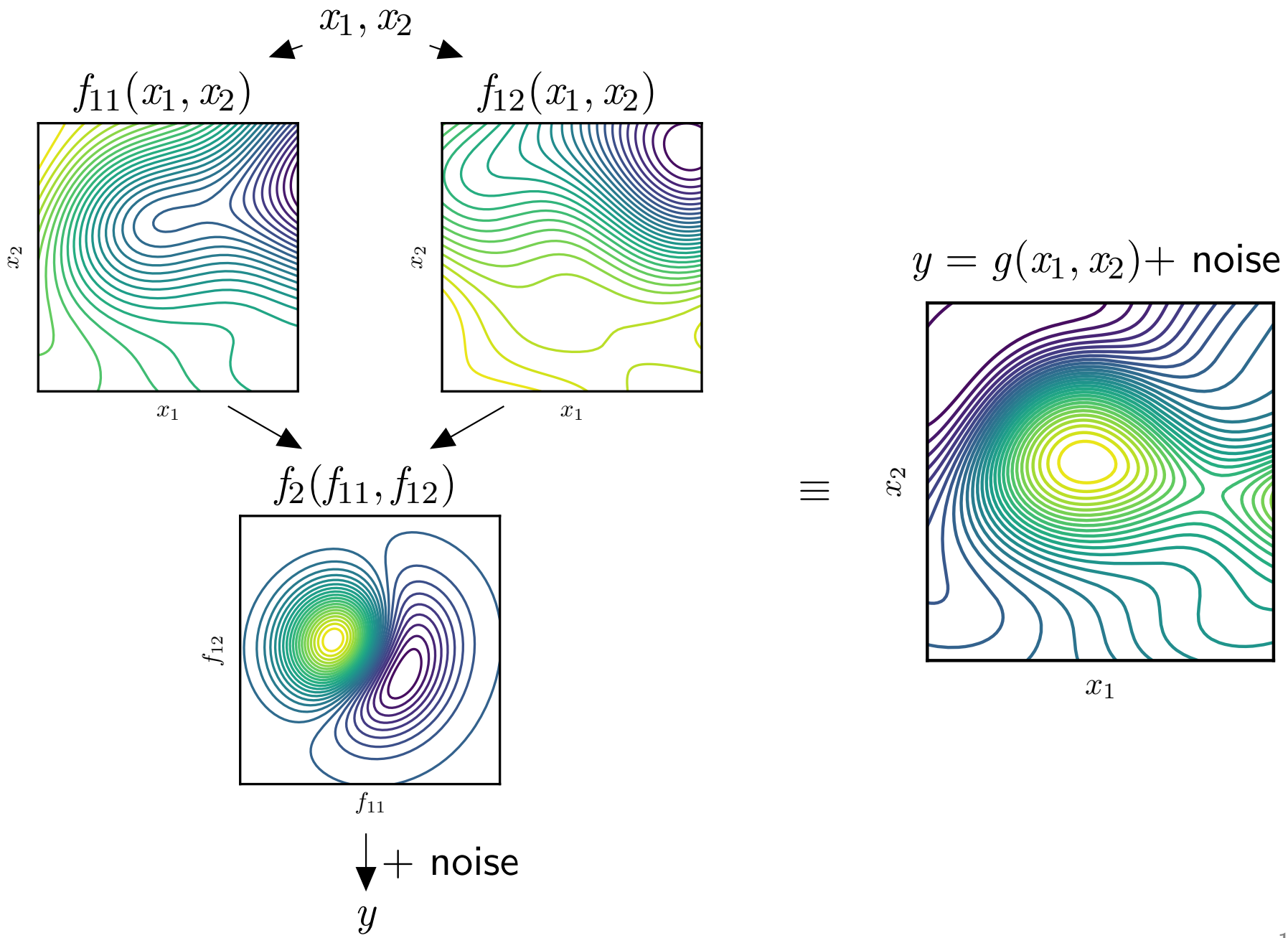
Once again: optimal Gaussian $t_{m,n}(\mathbf{u}_m)$ is rank 1

$\alpha \to 0$ recovers Damianou & Lawrence (2013)

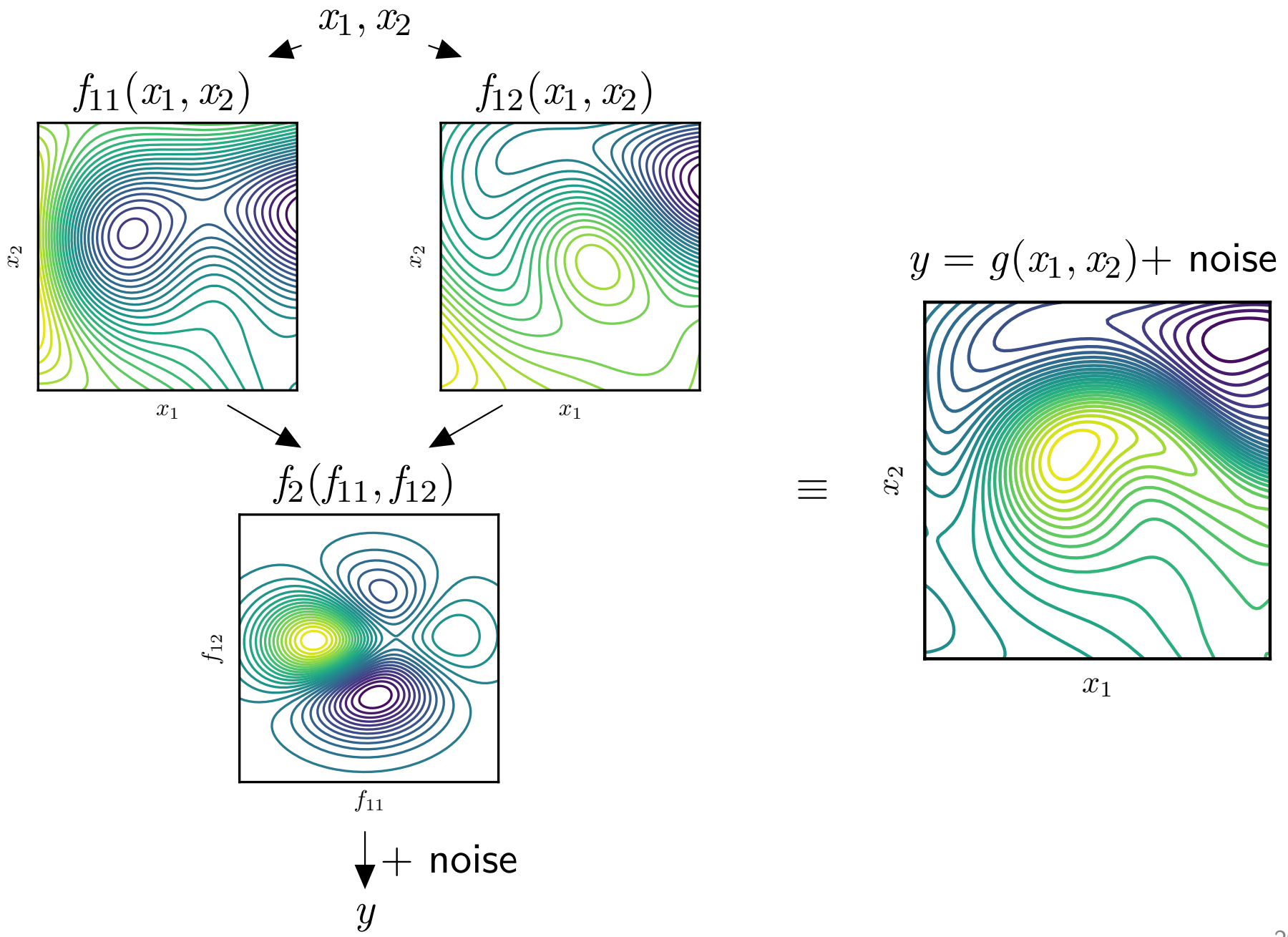
Power EP for Deep GPs: three key additional ideas

1. **Reduce memory overhead**: Tie factors $t_{m,n}(\mathbf{u}_m) = t_m(\mathbf{u}_m)$ (Stochastic Expectation Propagation for inducing variables)
2. **Reduce message passing overhead**: just pass them down from the inputs to the outputs (ADF for hidden unit activities)
3. **Improve hyper-parameter optimisation**: optimise the EP energy $\log Z_{EP}$ function directly using ADAM

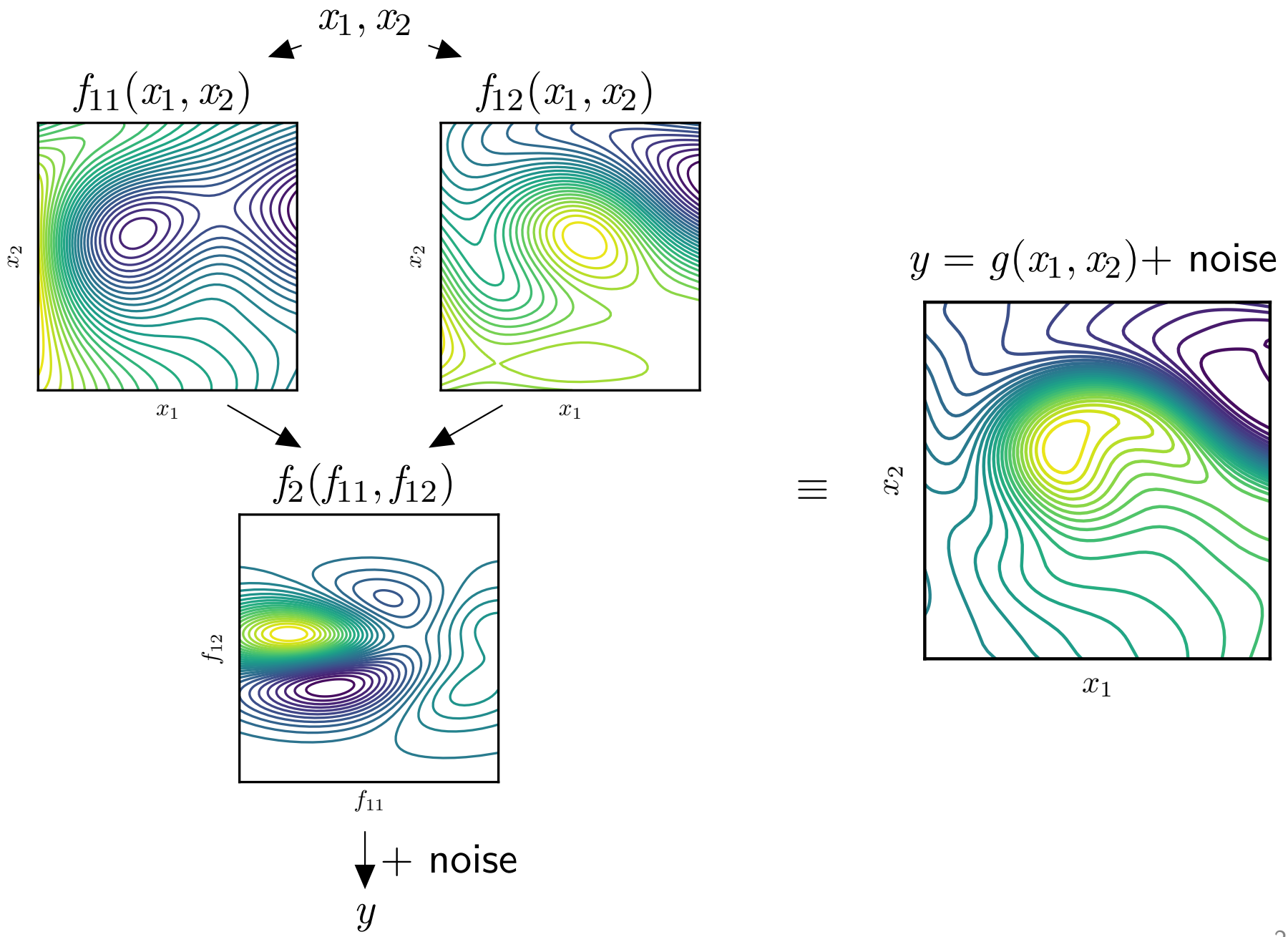
Training deep GPs



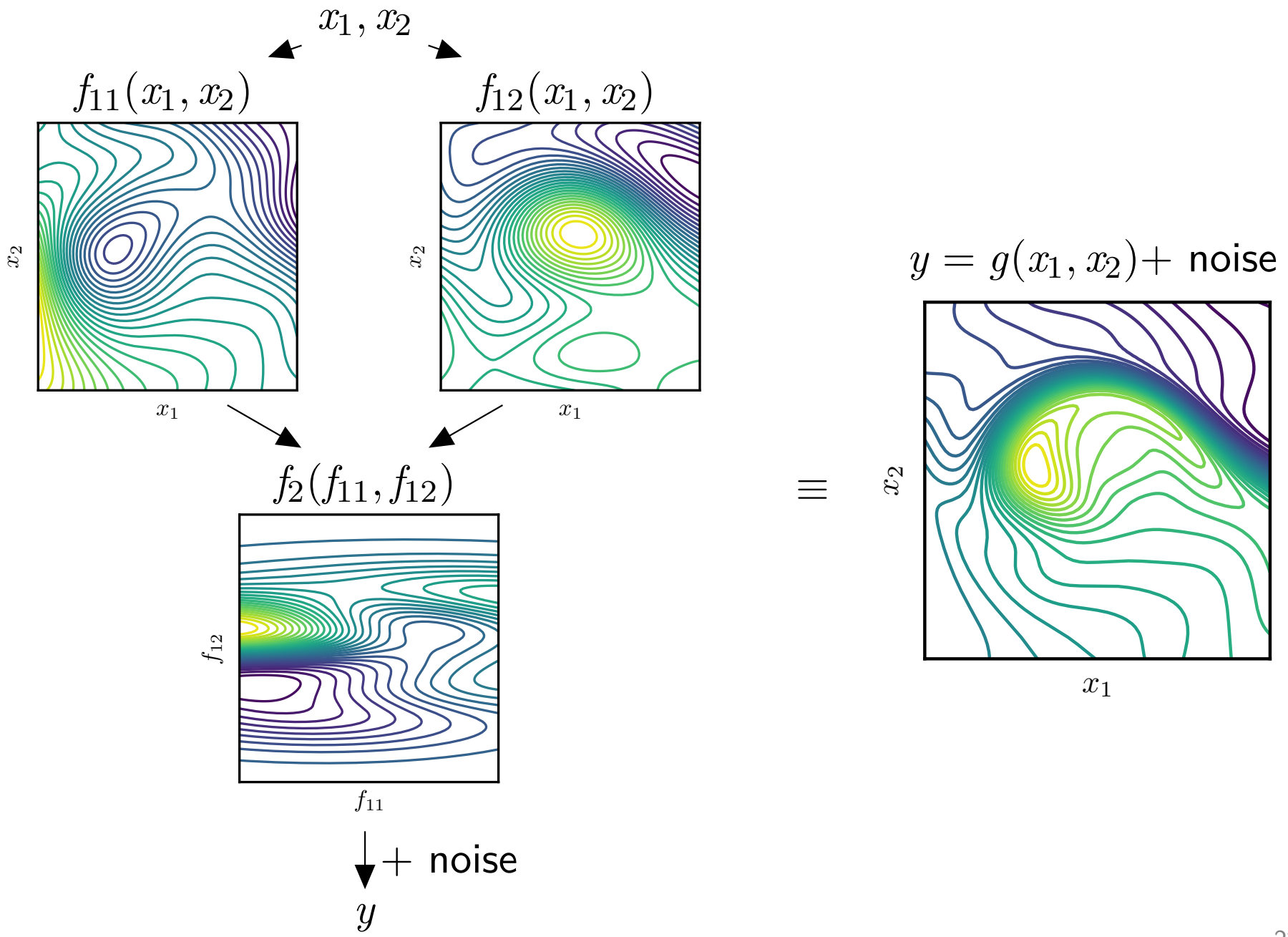
Training deep GPs



Training deep GPs

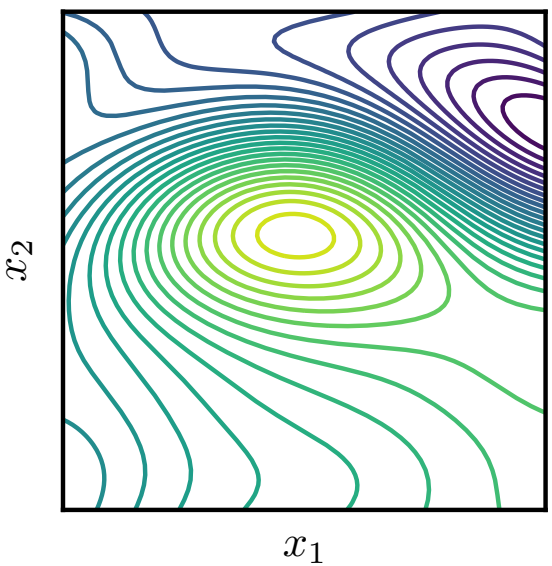


Training deep GPs

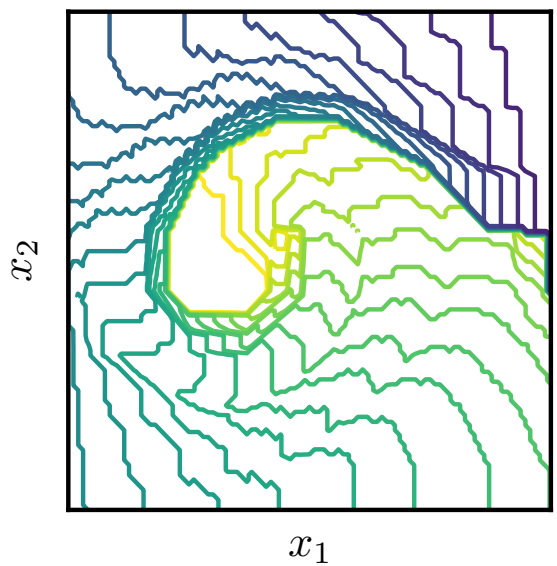


Experiment: Value function of the mountain car problem

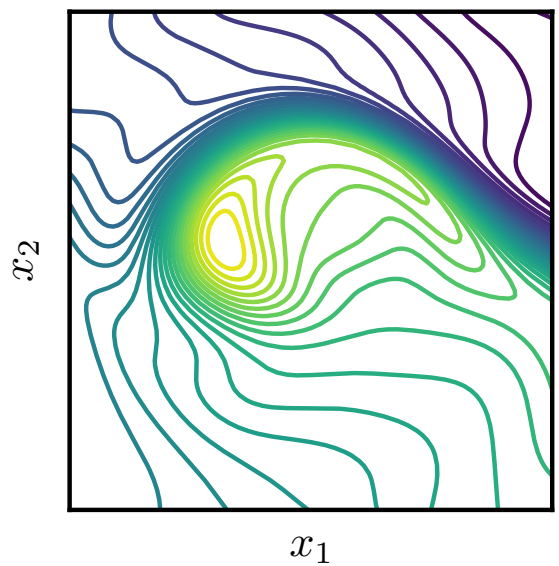
GP fit



Value function



DGP fit

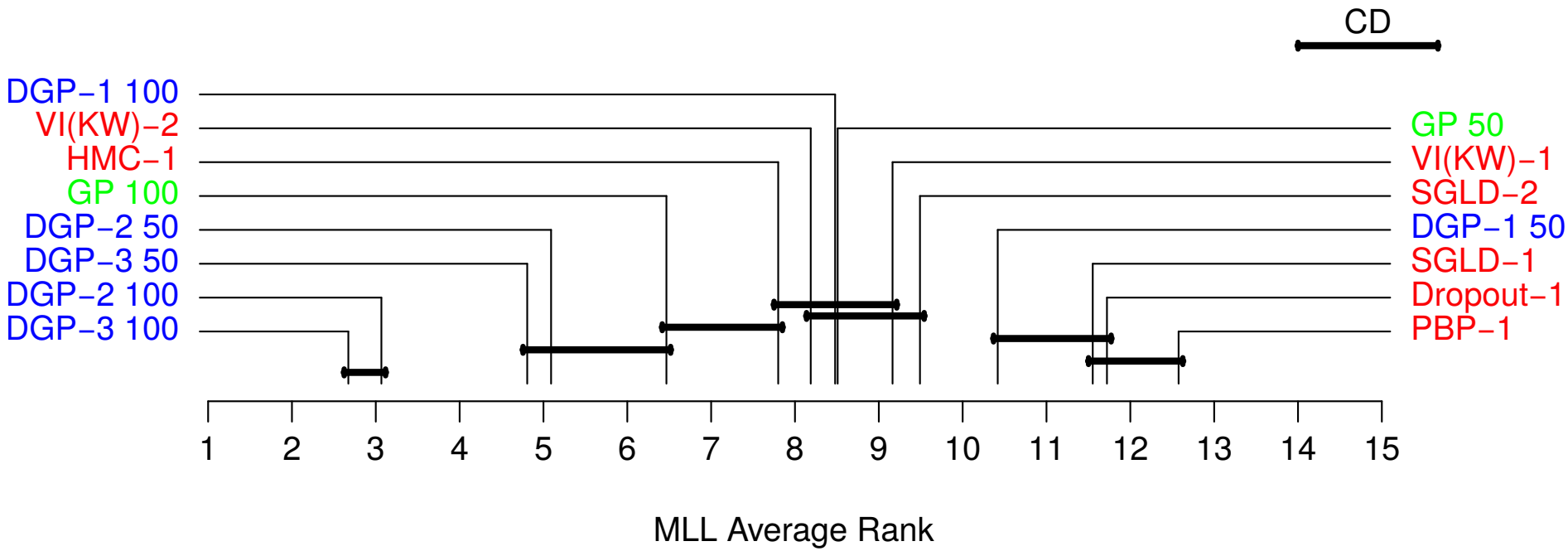


Experiment: Comparison to Bayesian neural networks

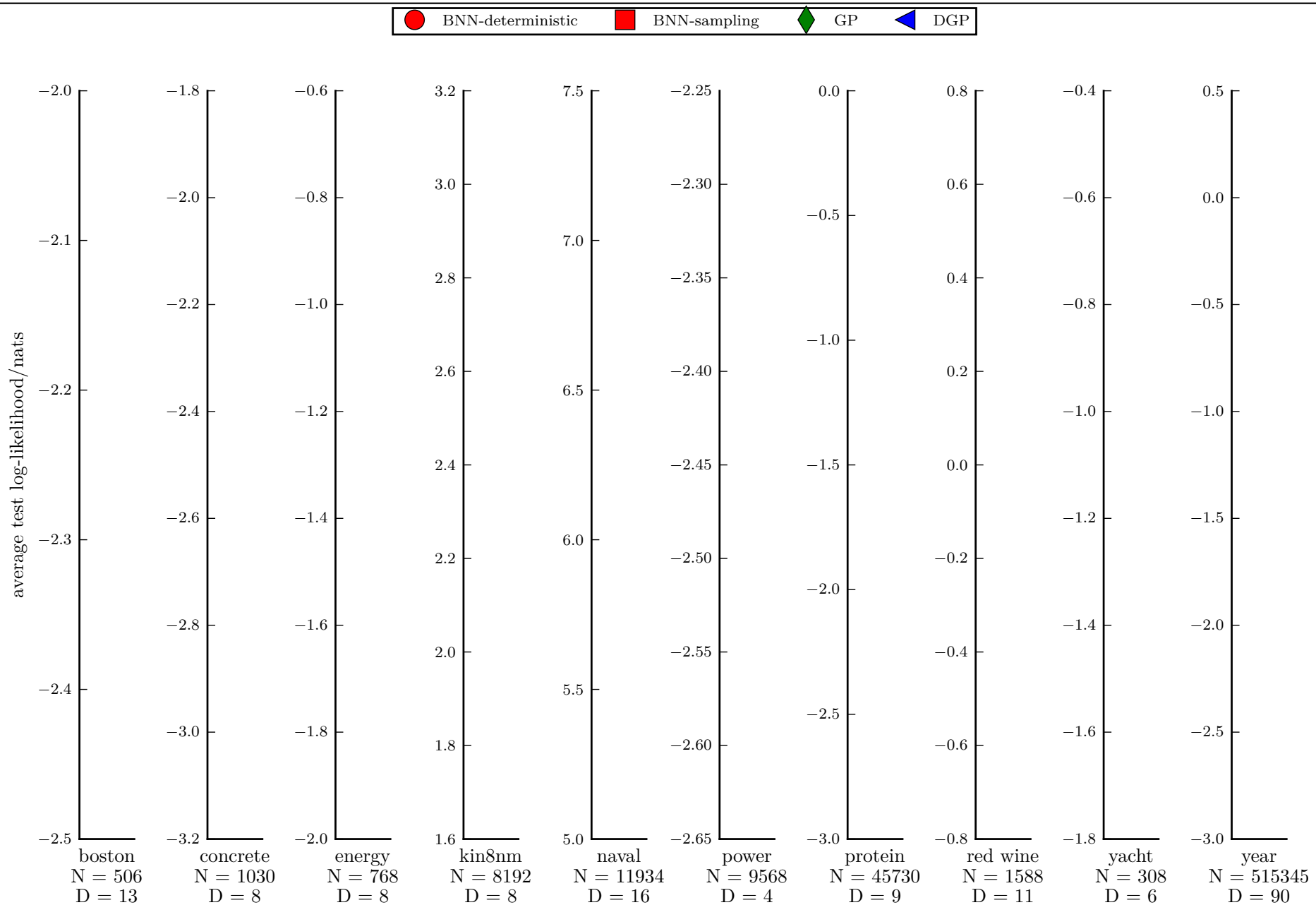
We compared **DGPs** with **GPs** and **Bayesian neural networks** with one and two hidden layers using:
VI(G): Graves' VI [diagonal Gaussian, without the reparam. trick]
VI(KW): Kingma and Welling's VI [with the reparam. trick]
PBP: ADF with Probablistic Backpropagation
Dropout: Combining dropout predictions at test time
SGLD: Stochastic gradient Langevin dynamics
HMC: Hamiltonian Monte Carlo [only for small networks]

Experiment: Comparison to Bayesian neural networks

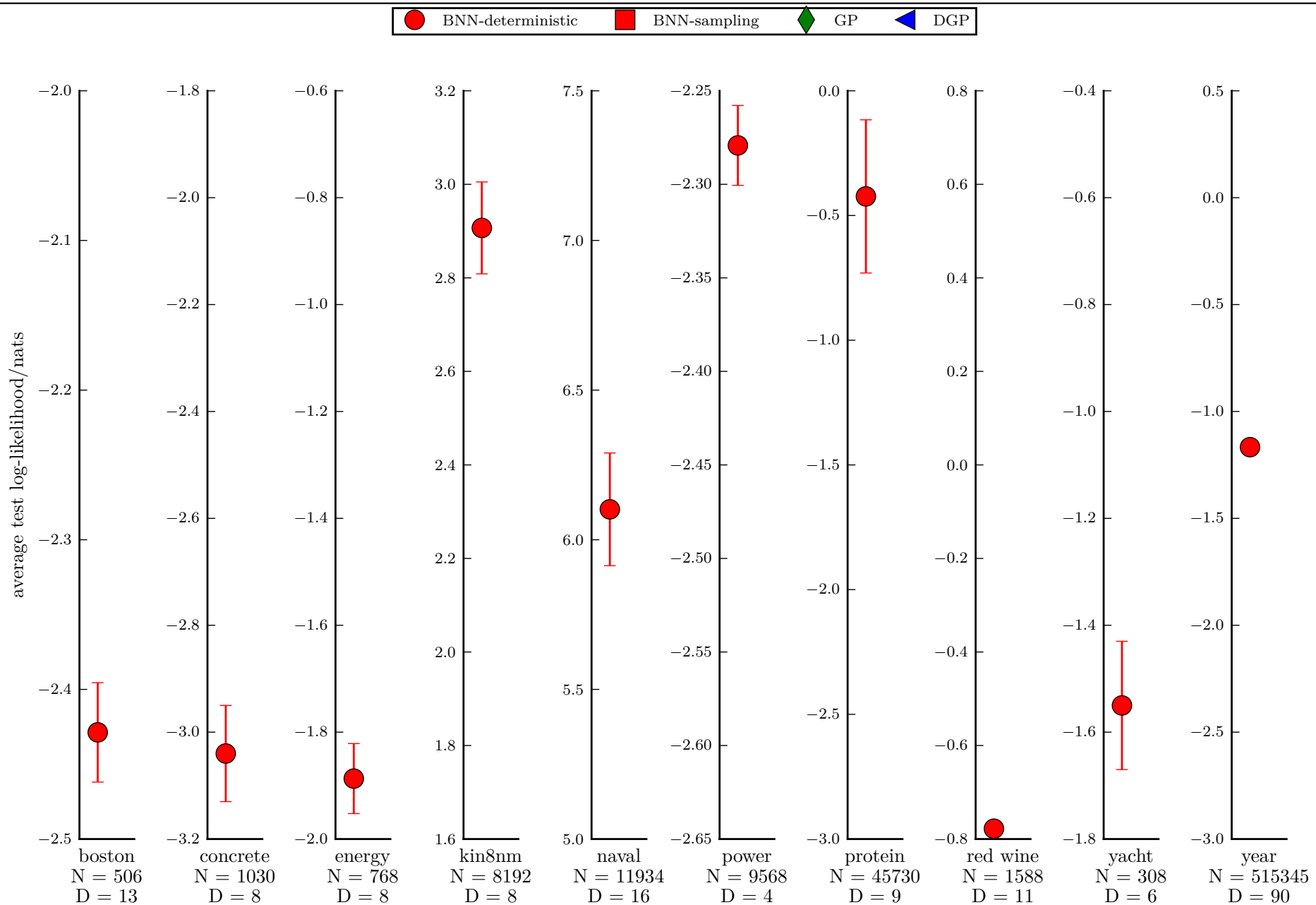
Rankings of all methods across all datasets



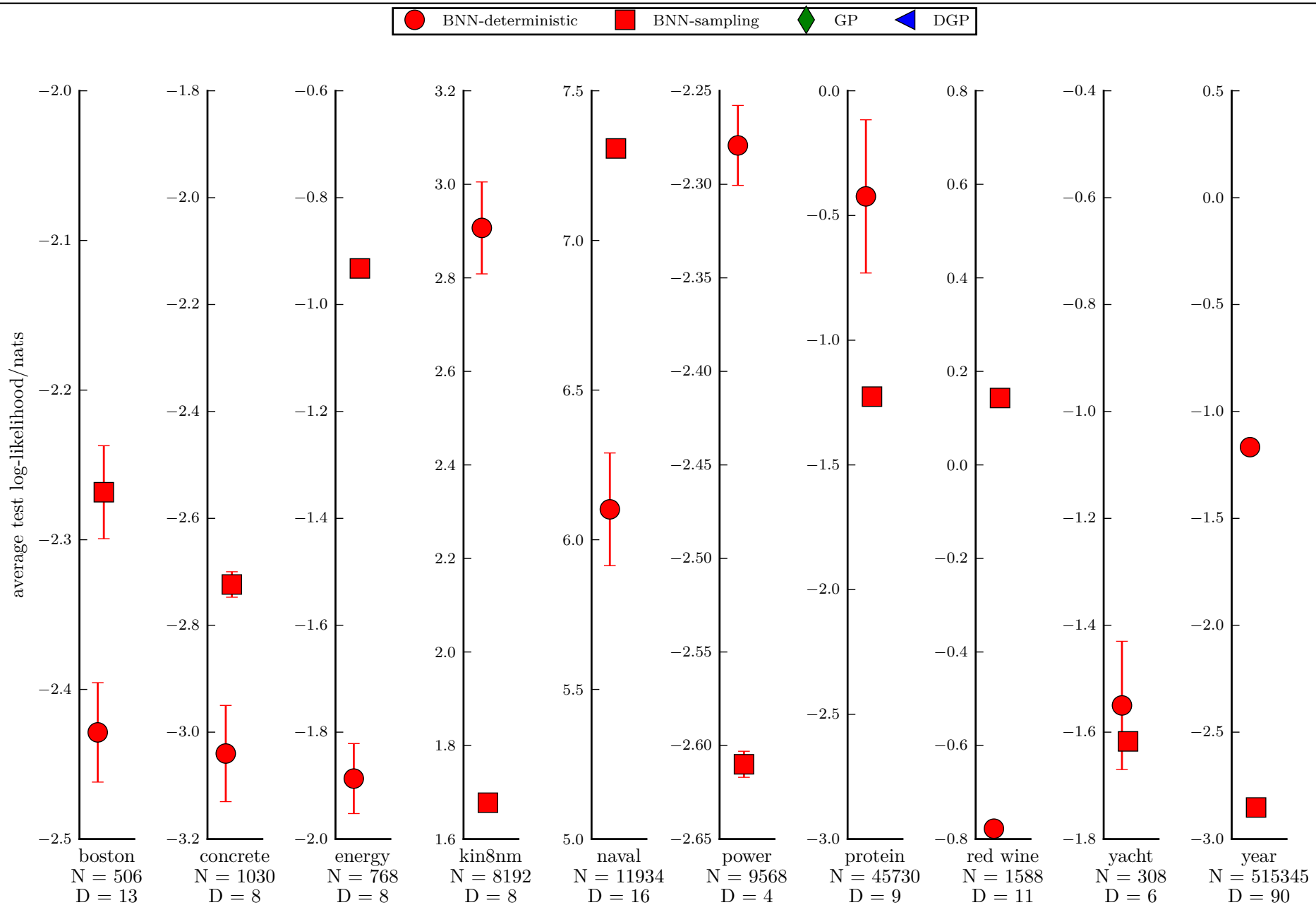
Experiment: Comparison to Bayesian neural networks [Best results]



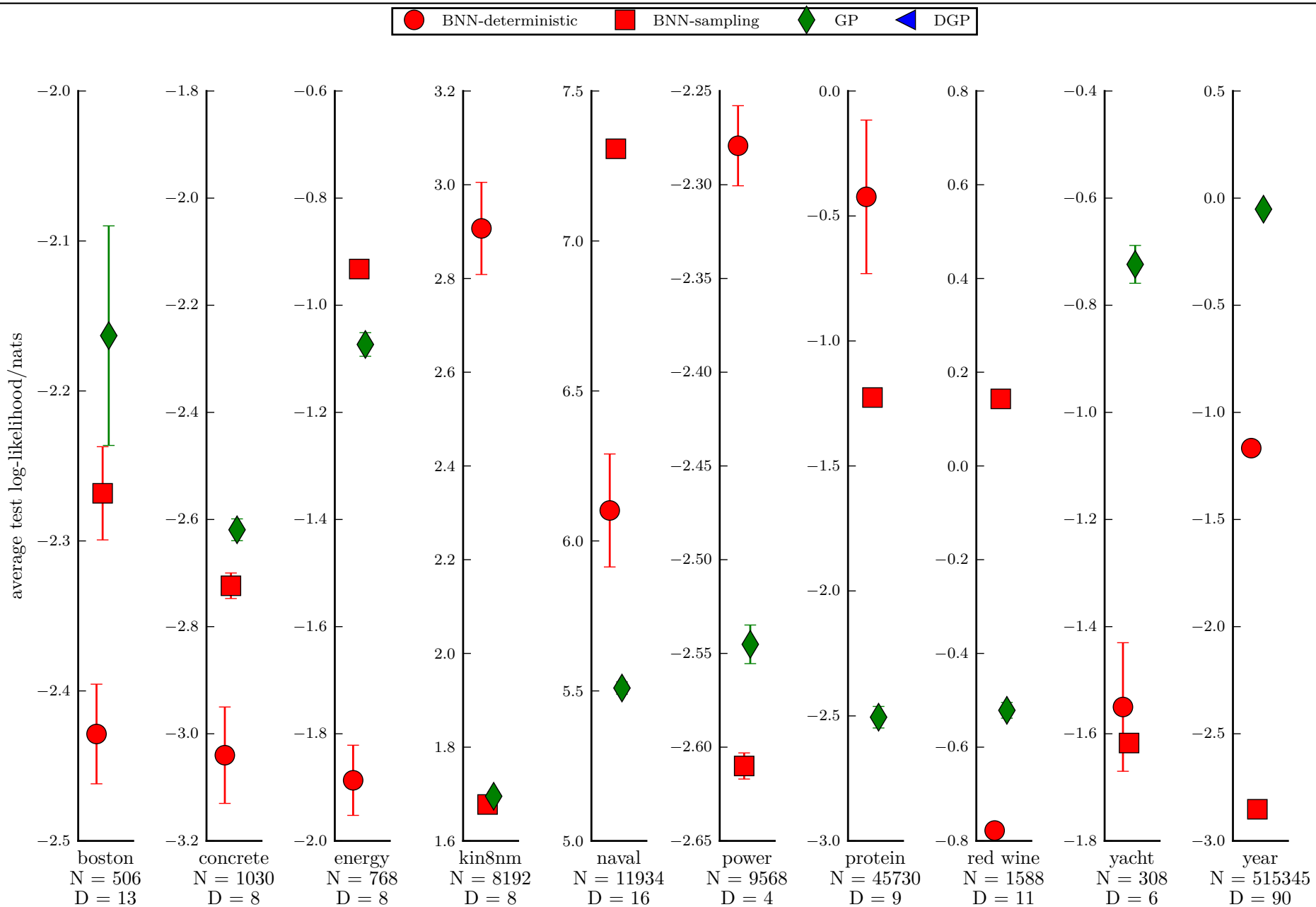
Experiment: Comparison to Bayesian neural networks [Best results]



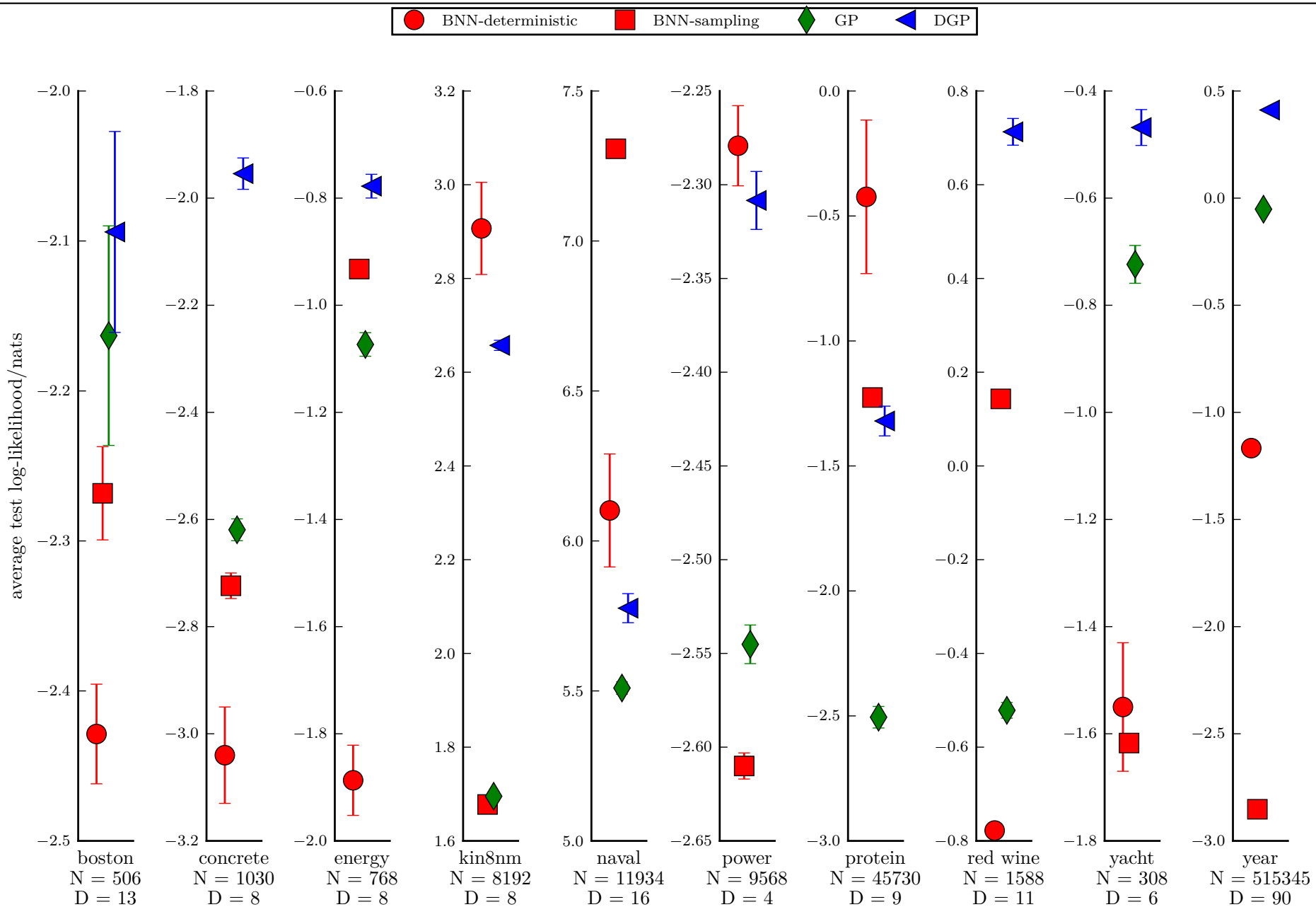
Experiment: Comparison to Bayesian neural networks [Best results]



Experiment: Comparison to Bayesian neural networks [Best results]

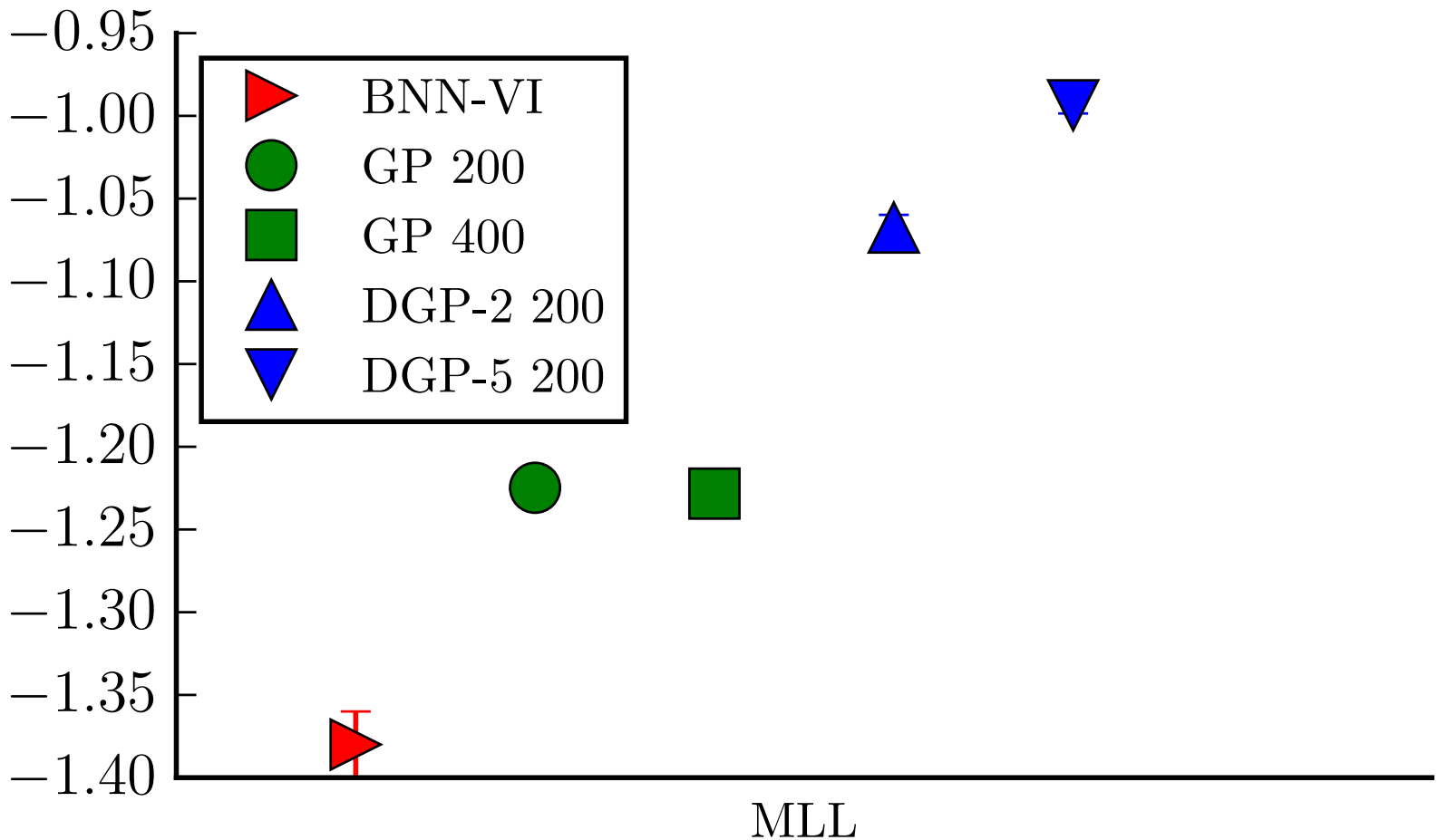


Experiment: Comparison to Bayesian neural networks [Best results]



Experiment: Efficiency of organic photovoltaic molecules

Dataset: 50k/10k training/test points, 512-dim. binary input features
Need error-bars for active learning or Bayesian optimisation



References (hyperlinked)

Core material:

- A Unifying Framework for Sparse Gaussian Process Approximation using Power Expectation Propagation, arXiv preprint
- Deep Gaussian Processes for Regression using Approximate Expectation Propagation, ICML 2016

Related papers:

- Stochastic Expectation Propagation, NIPS 20015
- Black-box α-divergence Minimization, ICML 2016