Scalable Training of Inference Networks for Gaussian-Process Models

Jiaxin Shi

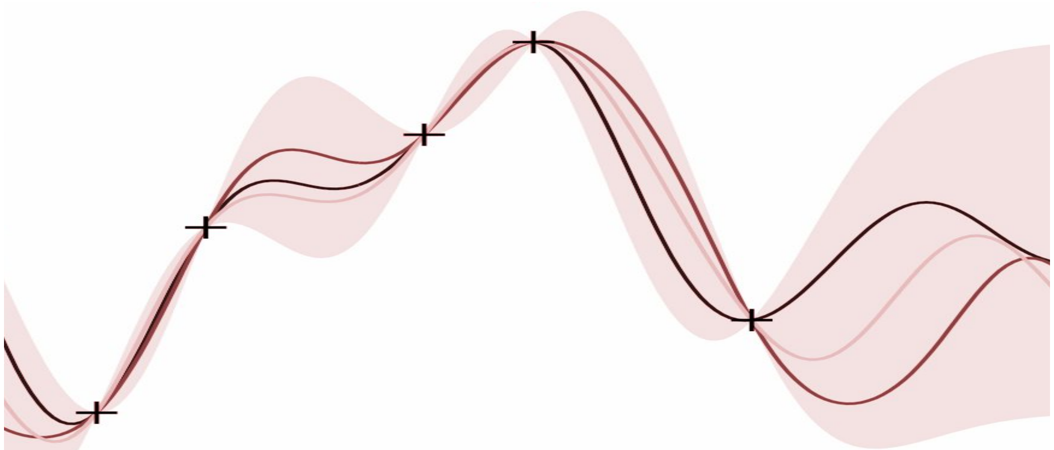
Tsinghua University

Joint work with

Mohammad Emtiyaz Khan

Jun Zhu

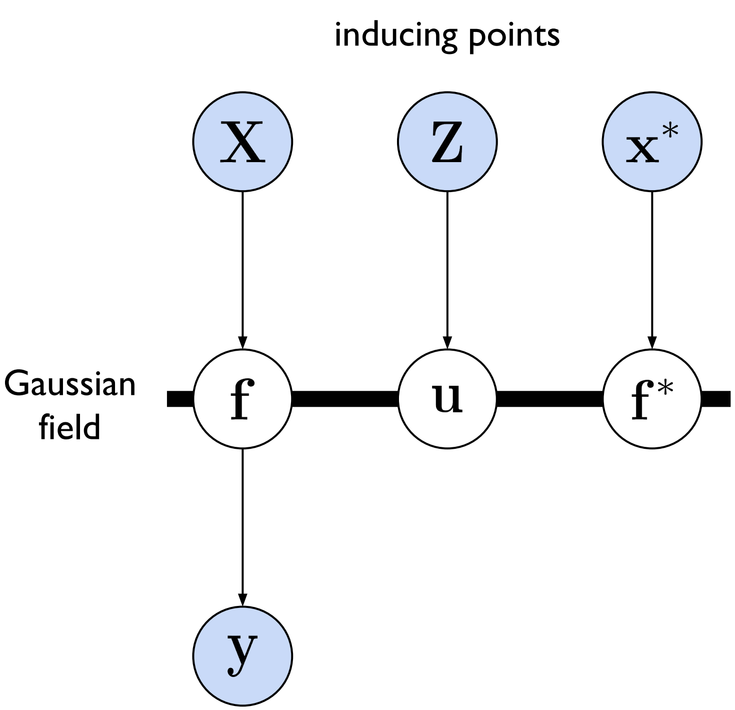
Gaussian Process



$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$$

mean function

covariance function / kernel



Posterior inference

$$p(\mathbf{f}, \mathbf{f}^*|\mathbf{y}) \propto p(\mathbf{f}, \mathbf{f}^*)p(\mathbf{y}|\mathbf{f})$$

$\mathcal{O}(N^3)$ complexity, conjugate likelihoods

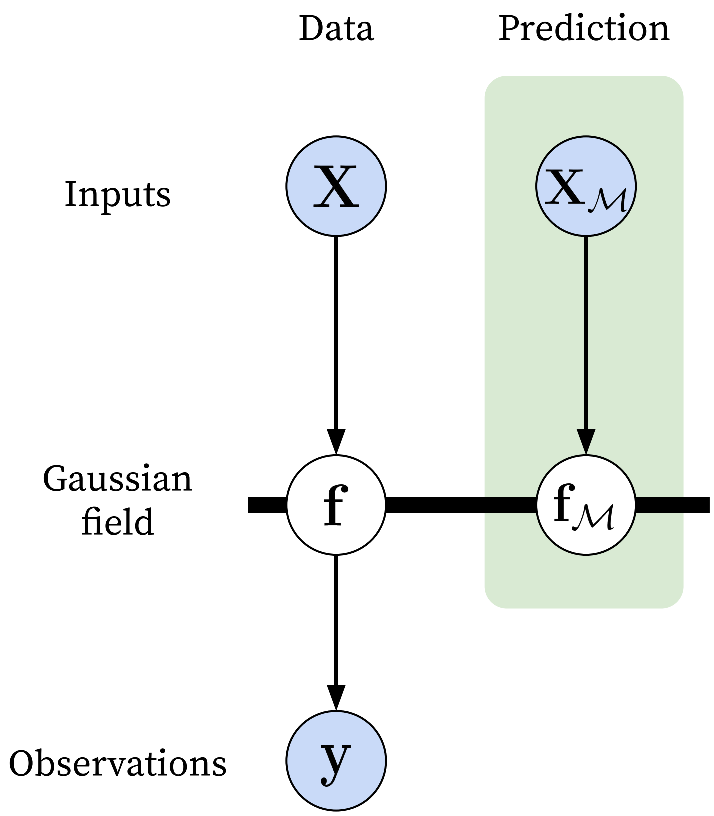
Sparse variational GP [Titsias, 09; Hensman et al., 13]

$$q(\mathbf{f}, \mathbf{f}^*, \mathbf{u}) := q(\mathbf{u})p(\mathbf{f}, \mathbf{f}^*|\mathbf{u})$$

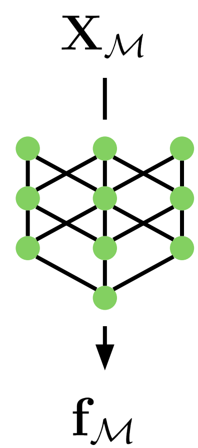
$$\mathcal{L}(q, \mathbf{Z}) := \mathbb{E}_{q(\mathbf{u})p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] - \mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})]$$

Inference Networks for GP Models

Remove sparse assumption



$\approx$

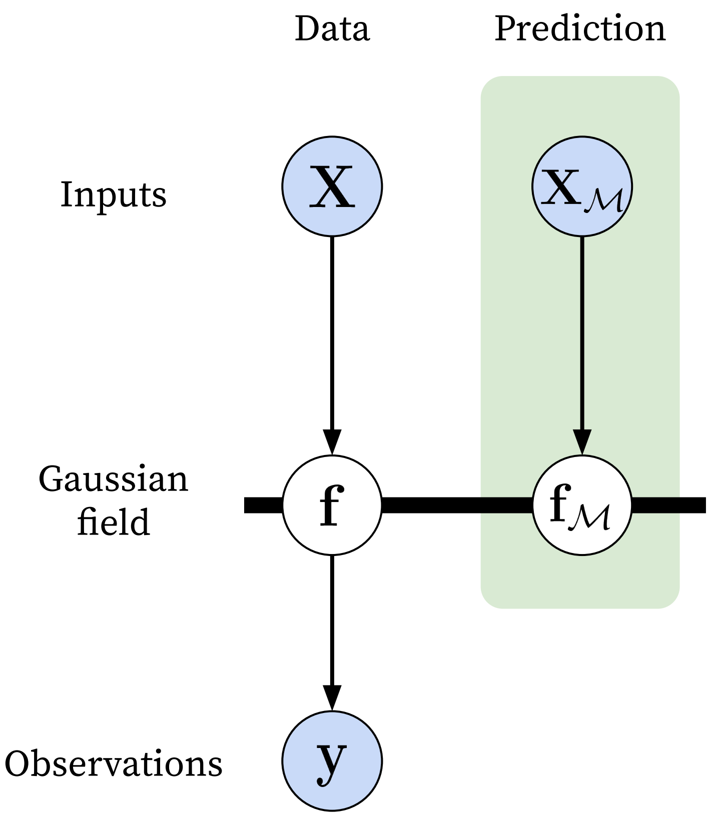


Inference network

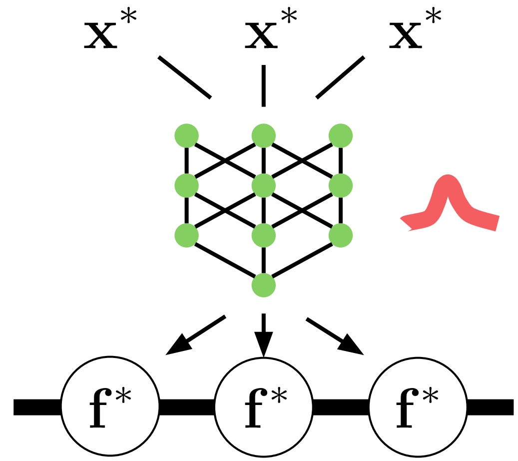
$$q(\mathbf{f}_\mathcal{M}) = \mathcal{N}(\mathbf{f}_\mathcal{M} | \mu_\mathcal{M}, \Sigma_\mathcal{M})$$

Inference Networks for GP Models

Remove sparse assumption



$\approx$



Inference network

$$q(\mathbf{f}_{\mathcal{M}}) = \mathcal{N}(\mathbf{f}_{\mathcal{M}} | \mu_{\mathcal{M}}, \Sigma_{\mathcal{M}})$$

Examples of Inference Networks

- Bayesian neural networks: [Sun et al., 19]
    - intractable output density

function space $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$

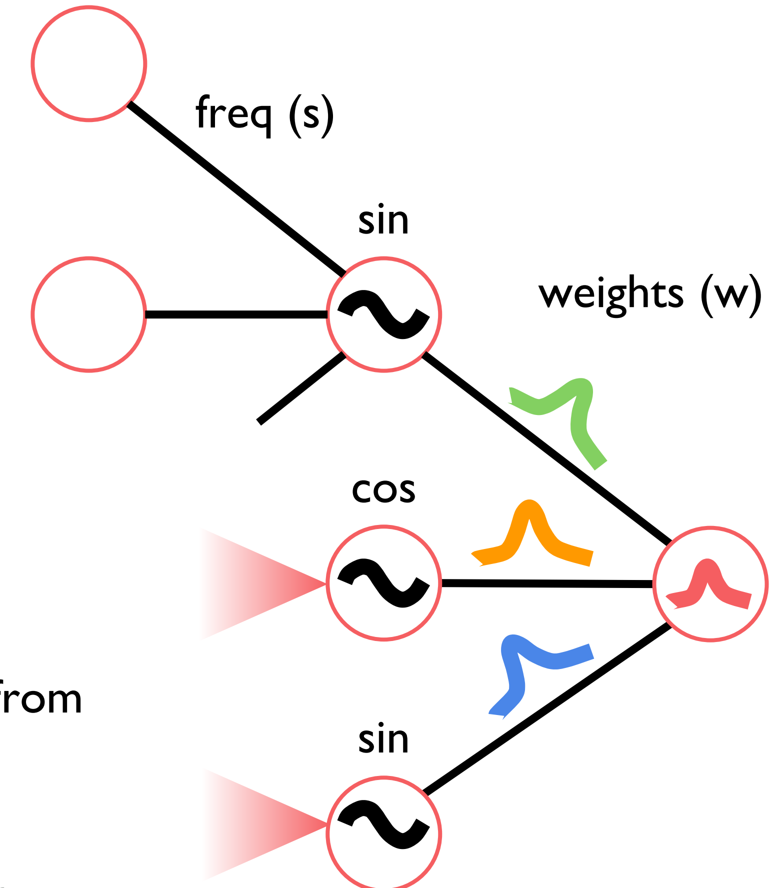
↕

weight space $f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}), \quad \mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

- Inference network architecture can be derived from the weight-space posterior

$$q(f) : f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + \xi_\theta(\mathbf{x}), \quad \mathbf{w} \sim \mathcal{N}(\mathbf{m}, \mathbf{V})$$

    - Random feature expansions [Cutajar, et al., 18]
    - Deep neural nets



Minibatch Training is Difficult

Functional Variational Bayesian Neural Networks (Sun et al., 19)

Measurement points

- Consider matching variational and true posterior processes at arbitrary $\mathbf{X}^{\mathcal{M}}$

$$\mathrm{KL}[q(\mathbf{f}^{\mathcal{M}})\|p(\mathbf{f}^{\mathcal{M}}|\mathbf{y})] \leq \mathrm{KL}[q_\phi(\mathbf{f}^{\mathcal{M}}, \mathbf{f})\|p(\mathbf{f}^{\mathcal{M}}, \mathbf{f}|\mathbf{y})]$$

- Full batch fELBO

$$\begin{aligned}\mathcal{L}_{\mathbf{X}^{\mathcal{M}},\mathbf{X}}(q) &= \log p(\mathcal{D}) - \mathrm{KL}[q(\mathbf{f}^{\mathcal{M}}, \mathbf{f})\|p(\mathbf{f}^{\mathcal{M}}, \mathbf{f}|\mathbf{y})]. \\ &= \sum_{(\mathbf{x},y)\in\mathcal{D}} \mathbb{E}_{q_\phi}\left[\log p(y|f(\mathbf{x}))\right] - \mathrm{KL}[q(\mathbf{f}^{\mathcal{M}}, \mathbf{f})\|p(\mathbf{f}^{\mathcal{M}}, \mathbf{f})]\end{aligned}$$

- Practical fELBO

$$\frac{1}{|\mathcal{D}_s|}\sum_{(\mathbf{x},y)\in\mathcal{D}_s} \mathrm{E}_{q_\phi}\left[\log p(y|f(\mathbf{x}))\right] - \lambda\mathrm{KL}[q(\mathbf{f}^{\mathcal{D}_s}, \mathbf{f}^{M})\|p(\mathbf{f}^{\mathcal{D}_s}, \mathbf{f}^{M})].$$

- This objective is doing improper minibatch for the KL divergence term

Scalable Training of Inference Networks for GP Models

Stochastic, functional mirror descent

- work with the functional density directly [Dai et al., 16; Cheng & Boots, 16]
  - natural gradient in the density space
  - minibatch approximation with stochastic functional gradient

$$q_{t+1} = \underset{q}{\operatorname{argmax}} \int \hat{\partial}\mathcal{L}(q_t) q(f) df - \frac{1}{\beta_t} \mathrm{KL}\left[q \| q_t\right]$$

- closed-form solution as an adaptive Bayesian filter

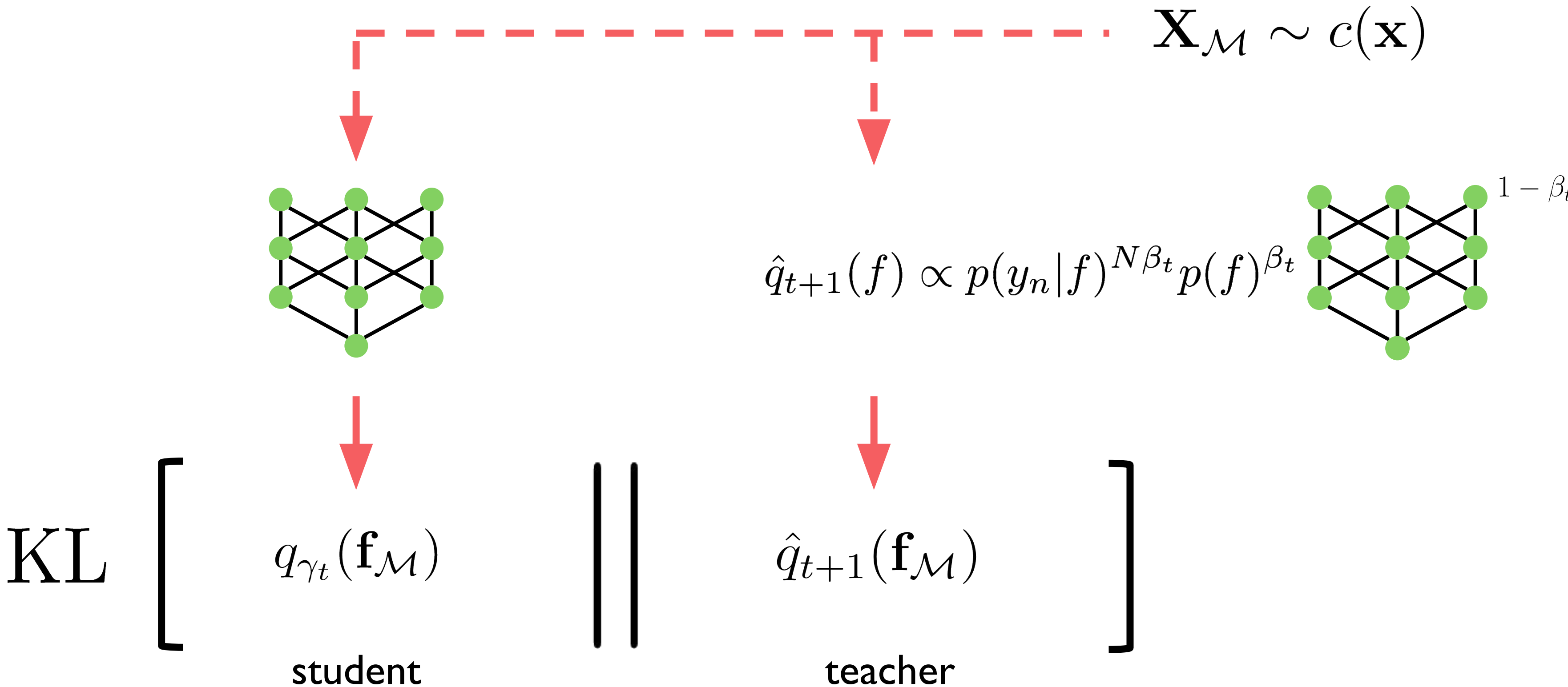
$$q_{t+1}(f) \propto \underbrace{p(y_n|f)^{N\beta_t}}_{\text{seeing next data point}} \underbrace{p(f)^{\beta_t} q_t(f)^{1-\beta_t}}_{\text{adapted prior}}$$

- sequentially applying Bayes' rule is the most natural gradient
  - in conjugate models: equivalent to natural gradient for exponential families

[Raskutti & Mukherjee, 13; Khan & Lin, 17]

Scalable Training of Inference Networks for GP Models

Minibatch training of inference networks



- an idea from filtering: bootstrap
    - similar idea: temporal difference (TD) learning with function approximation

Scalable Training of Inference Networks for GP Models

Minibatch training of inference networks

- (Gaussian likelihood case) closed-form marginals of $\hat{q}_{t+1}(f)$ at locations $\mathbf{X}_\mathcal{M}$
  - equivalent to GP regression

$$p(\mathbf{f}_\mathcal{M}, f_n)^{\beta_t} q_{\gamma_t}(\mathbf{f}_\mathcal{M}, f_n)^{1-\beta_t} := \mathcal{N}\left(\begin{bmatrix} \widetilde{\mathbf{m}}_\mathcal{M} \\ \widetilde{\mathbf{m}}_n \end{bmatrix}, \begin{bmatrix} \widetilde{\mathbf{K}}_{\mathcal{M},\mathcal{M}} & \widetilde{\mathbf{K}}_{\mathcal{M},n} \\ \widetilde{\mathbf{K}}_{n,\mathcal{M}} & \widetilde{\mathbf{K}}_{n,n} \end{bmatrix}\right)$$

$$\propto \quad \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathcal{M},\mathcal{M}} & \mathbf{K}_{\mathcal{M},n} \\ \mathbf{K}_{n,\mathcal{M}} & \mathbf{K}_{n,n} \end{bmatrix}\right)^{\beta_t} \times \mathcal{N}\left(\begin{bmatrix} \mu_\mathcal{M} \\ \mu_n \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}_{\mathcal{M},\mathcal{M}} & \mathbf{\Sigma}_{\mathcal{M},n} \\ \mathbf{\Sigma}_{n,\mathcal{M}} & \mathbf{\Sigma}_{n,n} \end{bmatrix}\right)^{(1-\beta_t)}$$

$$\hat{q}_{t+1}(\mathbf{f}_\mathcal{M}, f_n) \propto \mathcal{N}(y_n | f_n, \sigma^2/(N\beta_t)) \times \mathcal{N}\left(\begin{bmatrix} \widetilde{\mathbf{m}}_\mathcal{M} \\ \widetilde{\mathbf{m}}_n \end{bmatrix}, \begin{bmatrix} \widetilde{\mathbf{K}}_{\mathcal{M},\mathcal{M}} & \widetilde{\mathbf{K}}_{\mathcal{M},n} \\ \widetilde{\mathbf{K}}_{n,\mathcal{M}} & \widetilde{\mathbf{K}}_{n,n} \end{bmatrix}\right)$$

- (Nonconjugate case) optimize an upper bound of $\mathrm{KL}\left[q_\gamma(\mathbf{f}_\mathcal{M}) \| \hat{q}_{t+1}(\mathbf{f}_\mathcal{M})\right]$

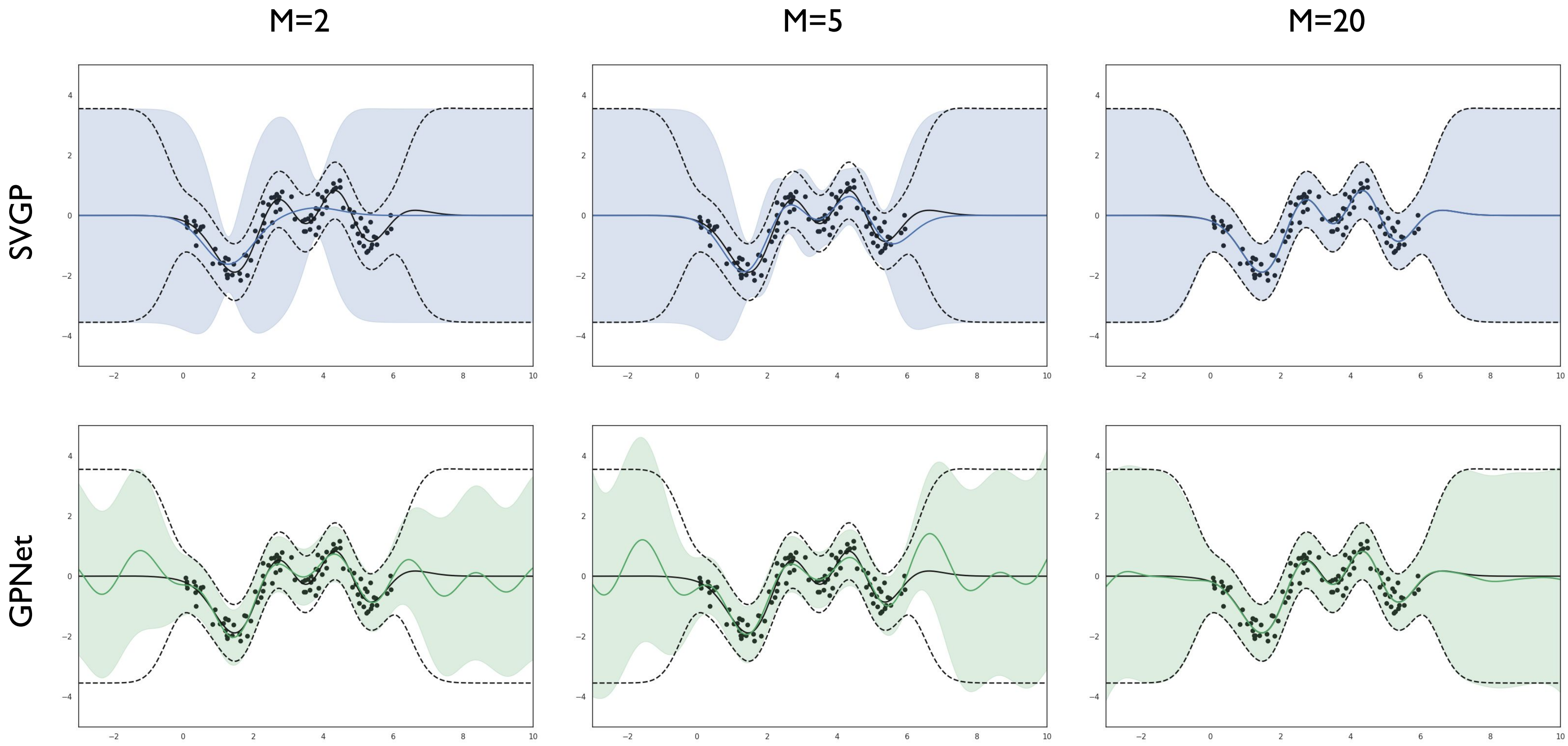
$$\min_\gamma \mathrm{KL}\left[q_\gamma(\mathbf{f}_\mathcal{M}, f_n) \| \hat{q}_{t+1}(\mathbf{f}_\mathcal{M}, f_n)\right] \Leftrightarrow \max_\gamma \mathcal{L}_t(q_\gamma; q_{\gamma_t}, \mathbf{X}_\mathcal{M})$$

$$\mathcal{L}_t(q_\gamma; q_{\gamma_t}, \mathbf{X}_\mathcal{M}) = \mathbb{E}_{q_\gamma(\mathbf{f}_\mathcal{M}, f_n)}\left[N\beta_t \log p(y_n | f_n) + \beta_t \log p(\mathbf{f}_\mathcal{M}, f_n) + \right.$$

$$\left. (1-\beta_t) \log q_{\gamma_t}(\mathbf{f}_\mathcal{M}, f_n) - \log q_\gamma(\mathbf{f}_\mathcal{M}, f_n)\right]$$

Scalable Training of Inference Networks for GP Models

Measurement points vs. inducing points



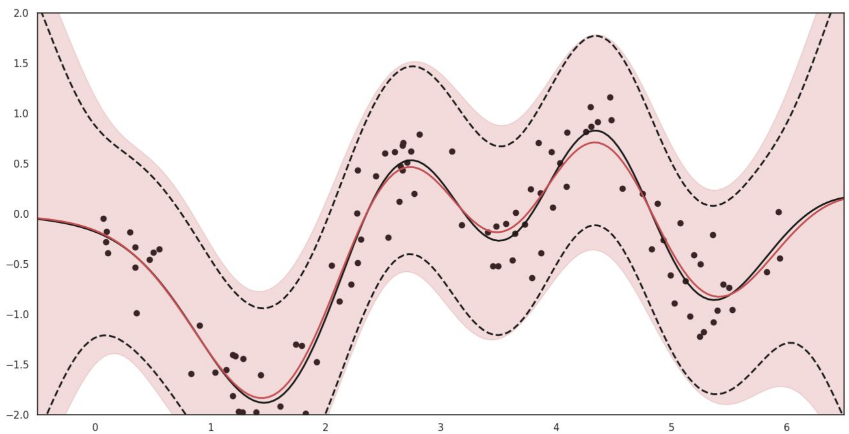
- inducing points - expressiveness of variational approximation
- measurement points - variance of training

Scalable Training of Inference Networks for GP Models

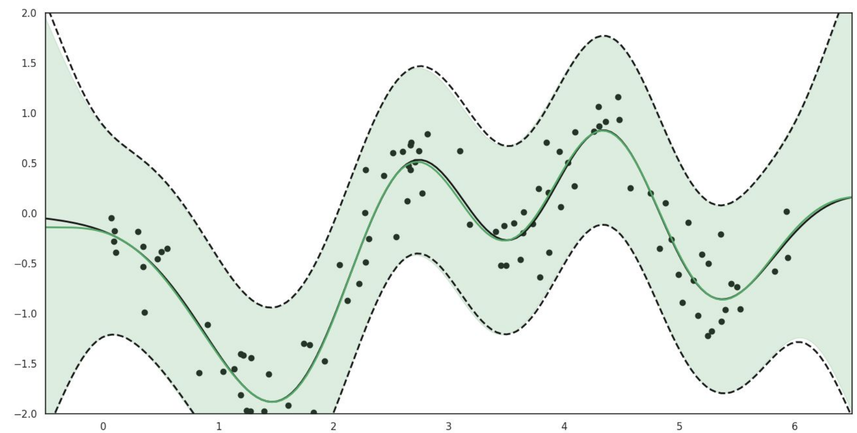
Effect of proper minibatch training

- Fix underfitting

N=100, batch size=20



FBNN, M=20



GPNet, M=20

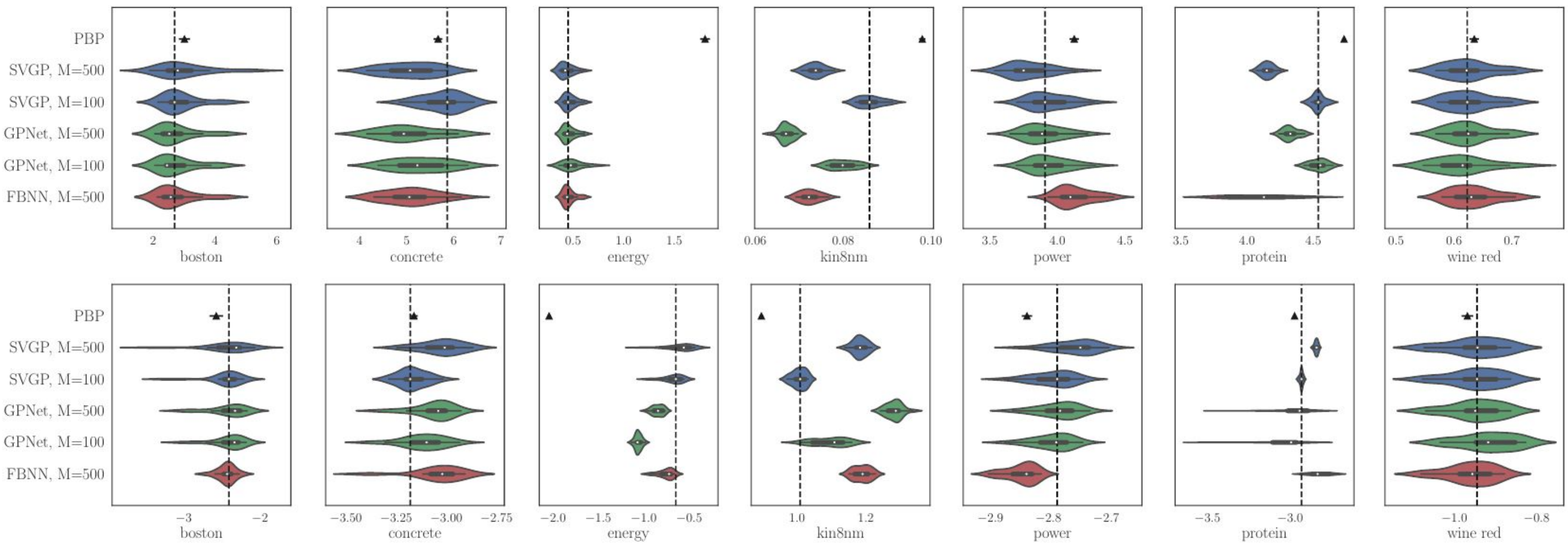
- Better performance with more measurement points

Airline Delay (700K)

METRIC	M=100			M=500		
	SVGP	GPNET	FBNN	SVGP	GPNET	FBNN
RMSE	24.261	24.055	23.801	23.698	**23.675**	24.114
Test LL	-4.618	-4.616	-4.586	-4.594	-4.601	**-4.582**

Scalable Training of Inference Networks for GP Models

Regression & Classification



Regression benchmarks

Methods	MNIST	CIFAR10
SVGP, RBF-ARD (Krauth et al., 2016)	1.55%	-
Conv GP (van der Wilk et al., 2017)	1.22%	35.4%
SVGP, CNN-GP (Garriga-Alonso et al., 2019)	2.4%	-
GPNet, CNN-GP	1.12%	**24.63%**
NN-GP (Lee et al., 2018)	1.21%	44.34%
CNN-GP (Garriga-Alonso et al., 2019)	0.96%	-
ResNet-GP (Garriga-Alonso et al., 2019)	**0.84%**	-
CNN-GP (Novak et al., 2019)	0.88%	32.86%

GP classification with a prior derived from infinite-width Bayesian ConvNets

Poster #227

Code: https://github.com/thjashin/gp-infer-net