Hi-fi functional priors by learning activations

Marcin Sendera   Amin Sorkhei   Tomasz Kuśmierczyk

Mila, Université de Montréal   Jagiellonian University in Kraków

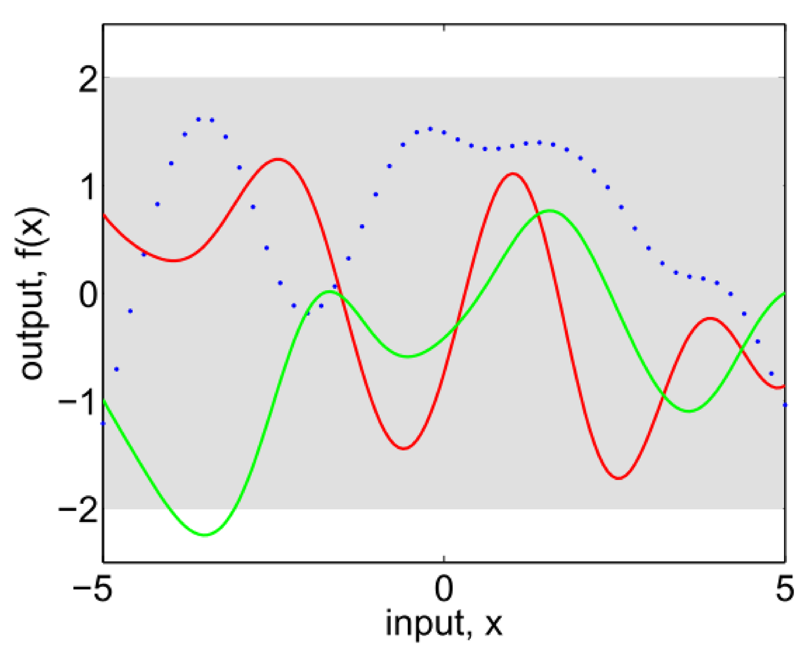
High-fidelity transfer of functional priors for wide Bayesian neural networks by learning activations

Problem: Real priors are not nice

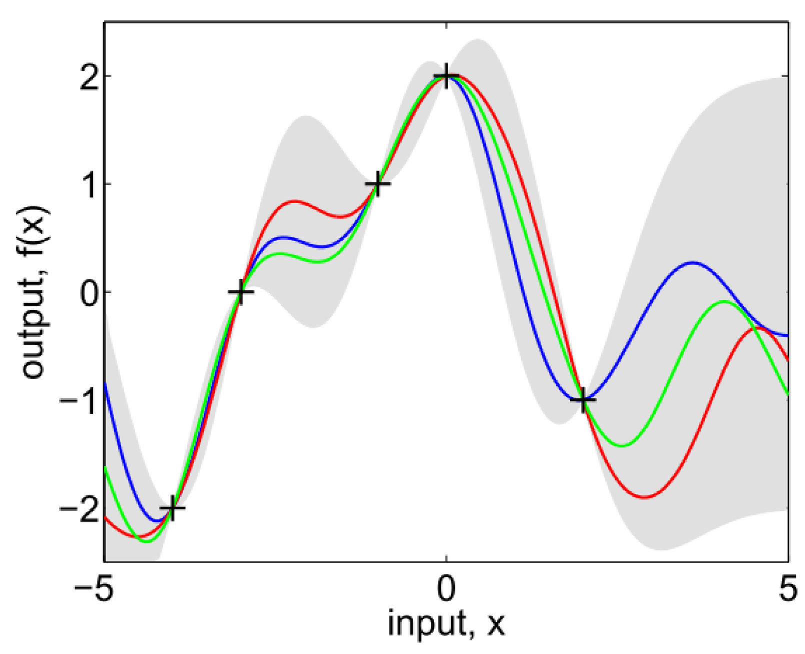
Recap:

prior $p(w, b)$ + likelihood $p(y|w, b, x)$ + data $\{x, y\}$
$\rightarrow$ posterior $p(w, b|\{x, y\})$

Expectations

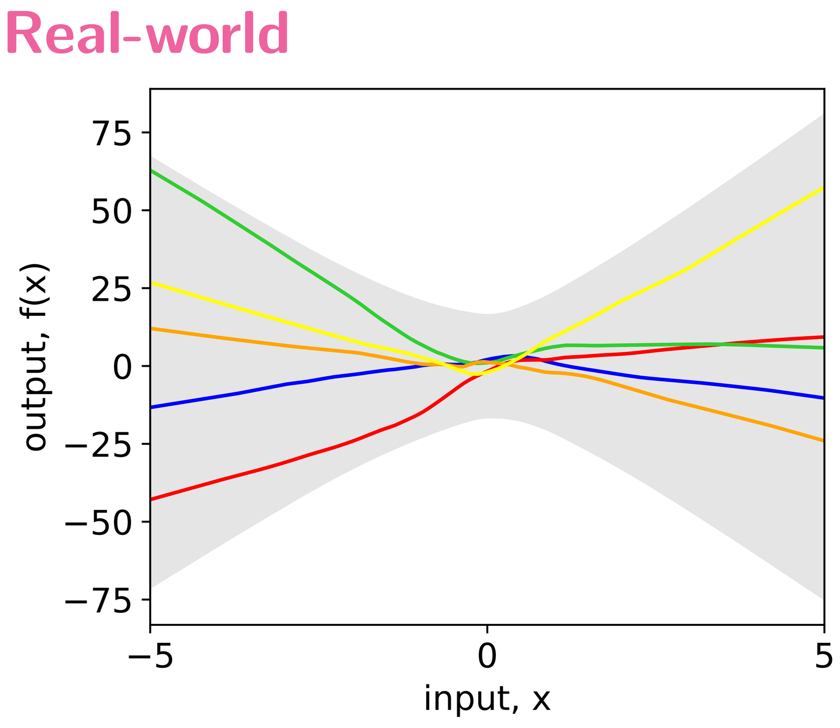


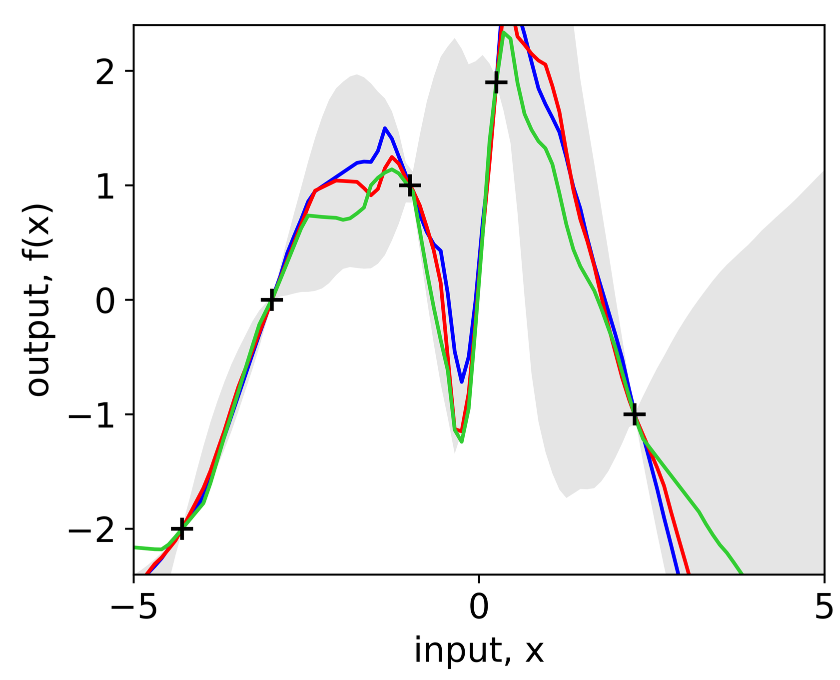
(a), prior



(b), posterior

Real-world





Adapted from *GPs for ML (Carl Edward Rasmussen, Christopher K. I. Williams, 2006)*

function-space view $\rightarrow$ smoothness and interpretability

prior $p(f^l)$ + likelihood $p(y|f^l, x)$ + data $\{x, y\}$

How to find such function-space priors?

Gaussian Processes:

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

where $m(x)$ is the mean function and $\kappa(x, x')$ is the kernel defining properties of $f(x)$. (**kernel specification**)

Wide BNNs are GPs!

$$p(f^l(x)) \xrightarrow{width \rightarrow \infty} \mathcal{N}(\mu(x), \sigma^2(x)),$$

$$\text{Cov}(f^l(x), f^l(x')) = \sigma_b^{l\,2} + \sigma_w^{l\,2} \mathbb{E}_{w_0, b}[\phi(w^0 x + b^0)\phi(w^0 x' + b^0)],$$

BNN corresponds to a GP$(\cdot, \kappa)$: $\kappa_f^l(x, x') = \text{Cov}(f^l(x), f^l(x'))$

- find κ_f^l given f^l: Easy
- identify f^l given κ_f^l: Hard

Enforcing function-space GP Priors in BNNs

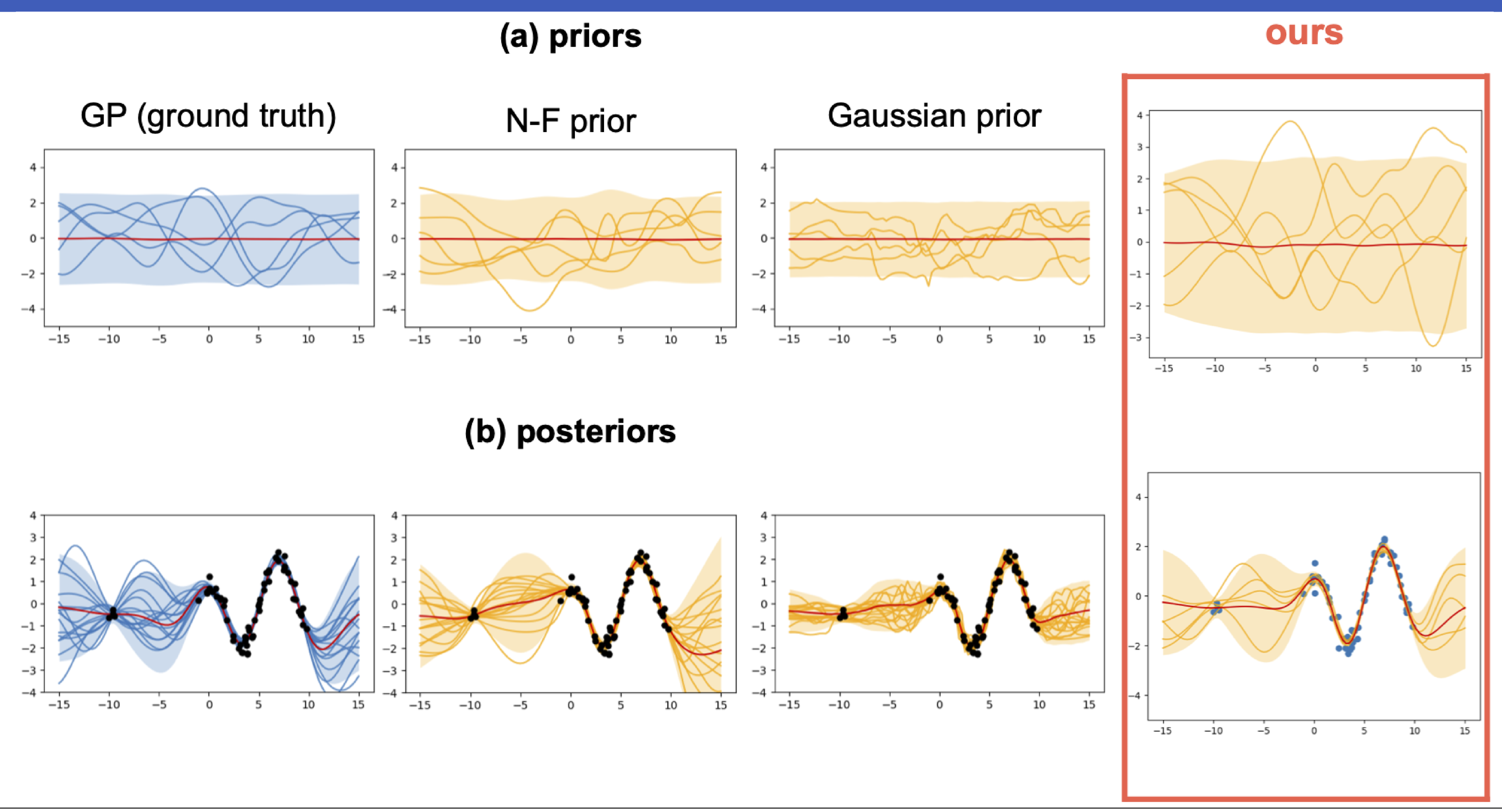
1. Reparameterize priors and activation:
$p(w, b|\lambda) = N(w|0, diag(\sigma_w))N(b|0, diag(\sigma_b))$, $\phi(\cdot|\eta)$

e.g. periodic activation for enforcing stationarity:
$\phi(x|\eta) = \Sigma_{i=1}^K A_i \cos(2\pi\psi_i x) + \Sigma_{j=1}^K A_j \sin(2\pi\psi_j x)$,
where $\eta = \{\psi_i, A_i, \psi_j, A_j\}$.

2. $\lambda^* = \text{argmin}_\lambda \frac{1}{S}\Sigma_{X \sim p_X} D(p_{nn}(f^l(X|\lambda)), p_{gp}(f^l(X)))$,
where $\lambda = \{\sigma_w, \sigma_b, \eta\}$

3. Closed-form 2-Wasserstein divergence between two Gaussians:
$D = ||\mu_1 - \mu_2||_2^2 + \text{Tr}\left[\Sigma_1 + \Sigma_2 - 2\sqrt{\sqrt{\Sigma_1}\Sigma_2\sqrt{\Sigma_1}}\right]$

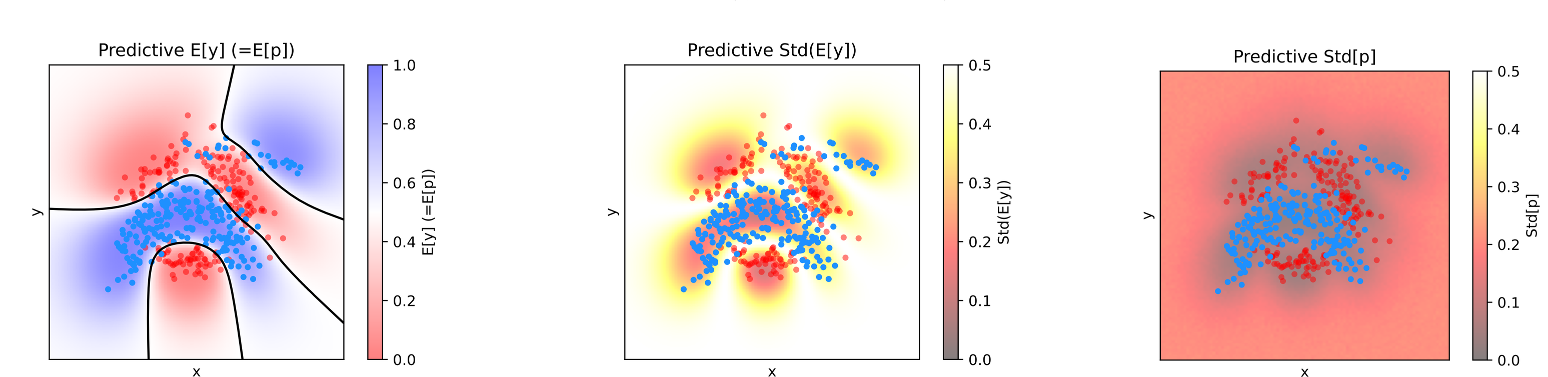
Regression



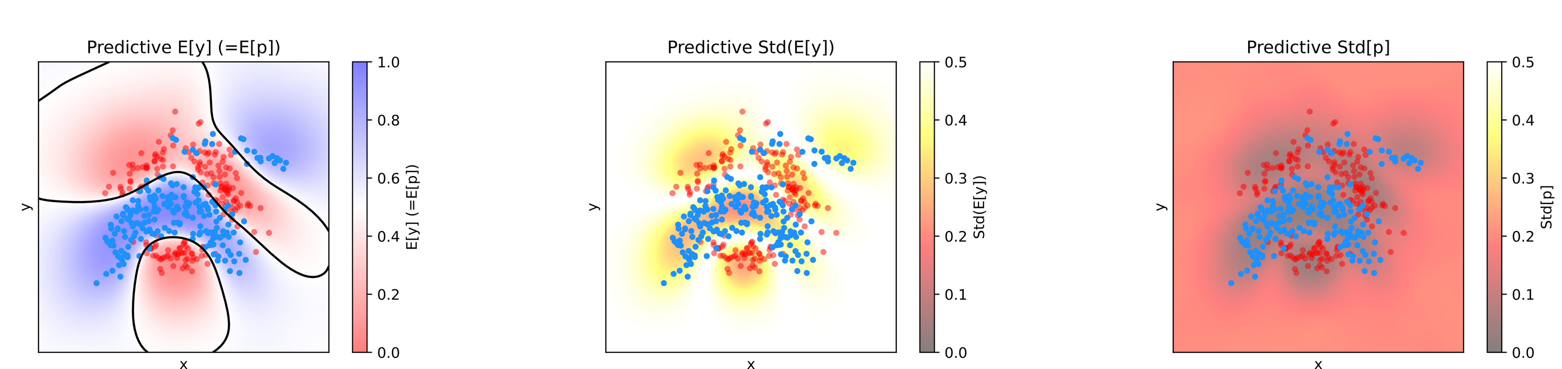
Method $\rightarrow$	our ($width = 128$, *periodic act.*)		our ($width = 128$)		our ($width = 1000$)		Baseline
Metric $\downarrow$	- regularization	+ regularization	- regularization	+ regularization	- regularization	+ regularization	—
RMSE	2.9067	2.8967	2.8643	**2.8348**	2.9189	3.0059	2.8402
NLL	2.5057	2.5072	2.4937	2.4862	2.5122	2.5971	**2.4778**

Classification

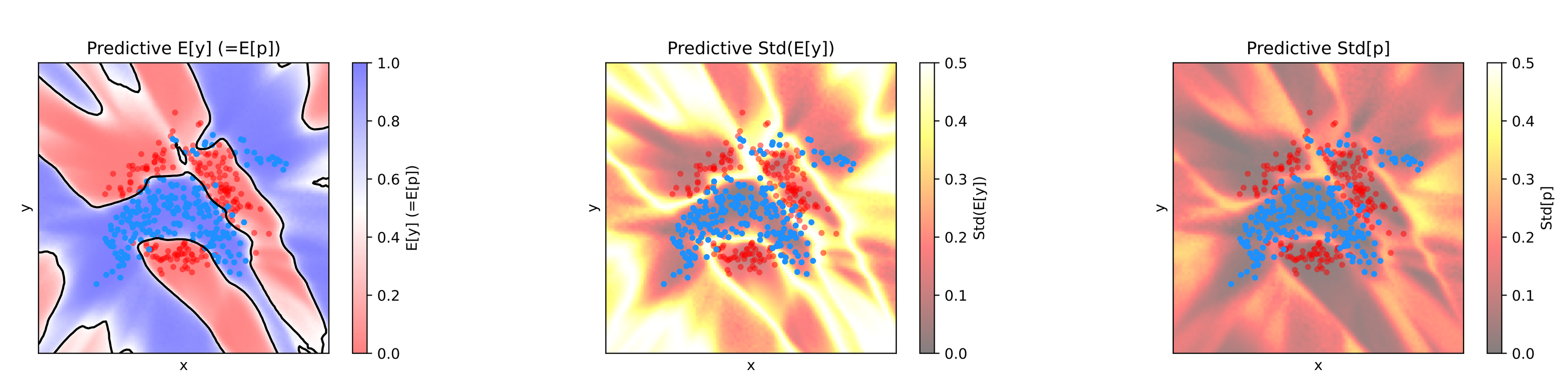
GP (ground truth)



Ours (trained parameters priors and activations)

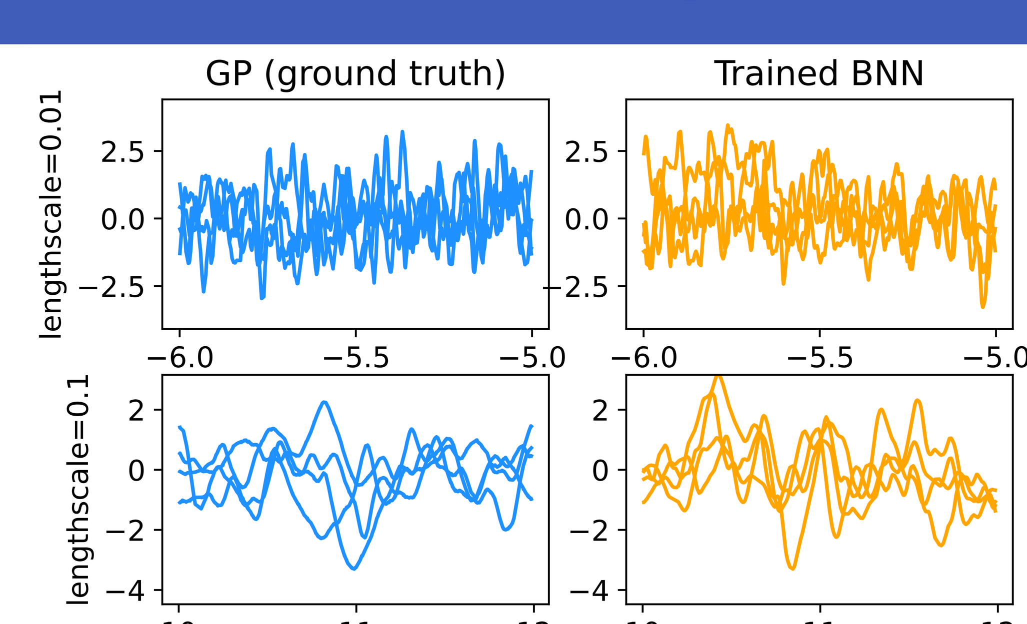


Meronen et al. (2020)



Stationarity outside training range

Conditioning



Future work

- Scaling to multi-dimensional outputs
- Creating plug-and-play GP-like layers for standard NNs
- Improving Bayesian deep learning approaches by enforcing GP-like baheviour (and posterior) on a specific layer of a NN