Revisiting the Equivalence of Bayesian Neural Networks and Gaussian Processes: On the Importance of Learning Activations

Marcin Sendera   Amin Sorkhei   Tomasz Kuśmierczyk

JAGIELLONIAN UNIVERSITY IN KRAKÓW

gmum group of machine learning research

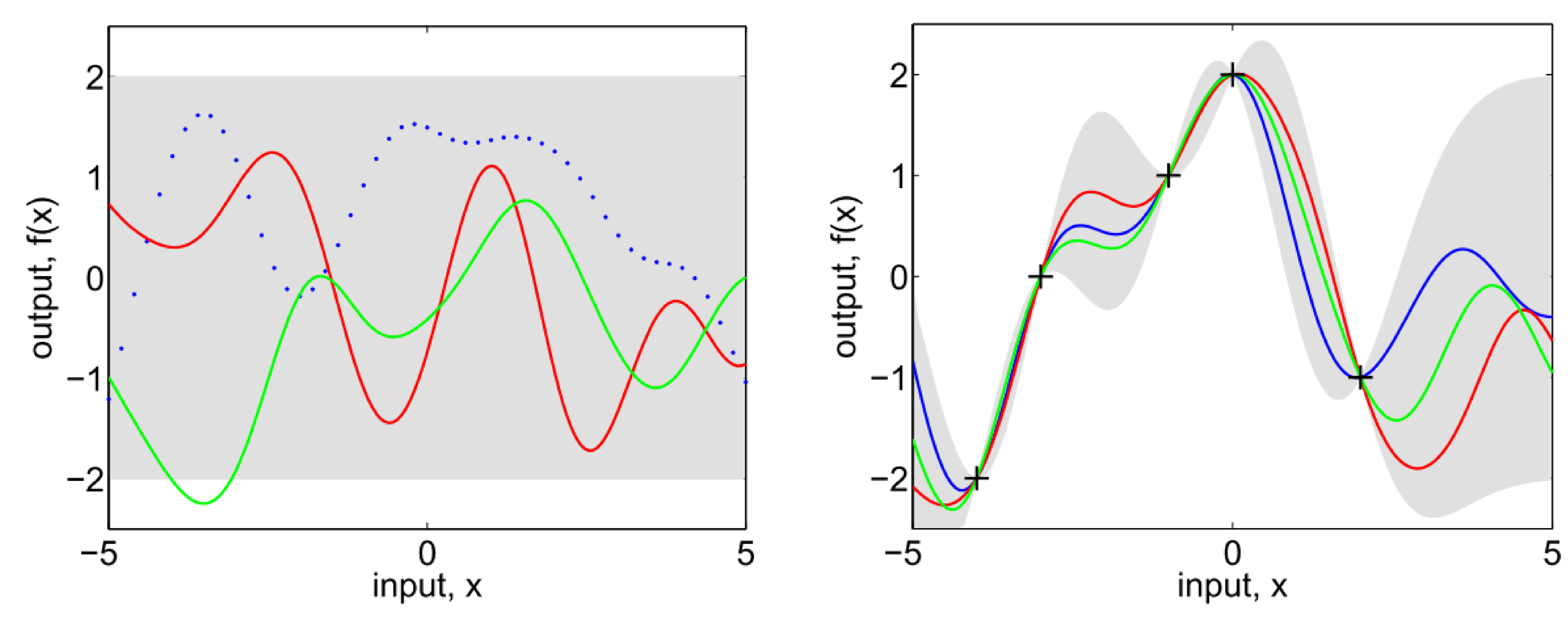
tl;dr: How to effectively transfer function-space priors into BNNs?

Problem: Real priors are not nice

Recap:

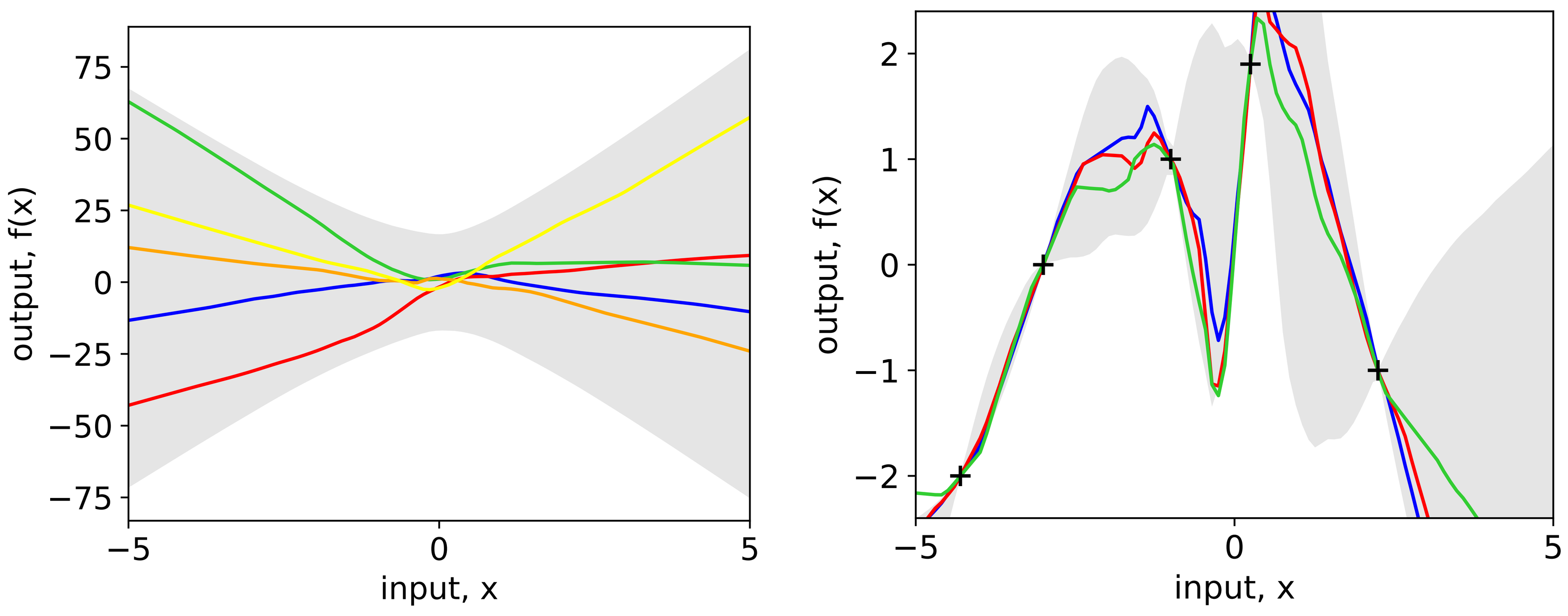
prior $p(w, b)$ + likelihood $p(y|w, b, x)$ + data $\{x, y\}$
$\rightarrow$ posterior $p(w, b|\{x, y\})$

Expectations



(a), prior   (b), posterior

Real-world BNNs



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 $\{x, y\}$
$\rightarrow$ posterior $p(f^l|\{x, y\})$

How to find such function-space priors?

Wide BNNs are GPs!

Gaussian Processes:

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

where the **kernel** $\kappa(x, x')$ **governs the properties of** $f(x)$

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

$$p(f^l(x)) \xrightarrow{width \to \infty} \mathcal{N}(\mu(x), \sigma^2(x)),$$
$$\text{Cov}(f^l(x), f^l(x')) = \sigma_b^{l\,2} + \sigma_w^{l\,2}\mathbb{E}_{w_0,b}[\phi(w^0x + b^0)\phi(w^0x' + b^0)],$$

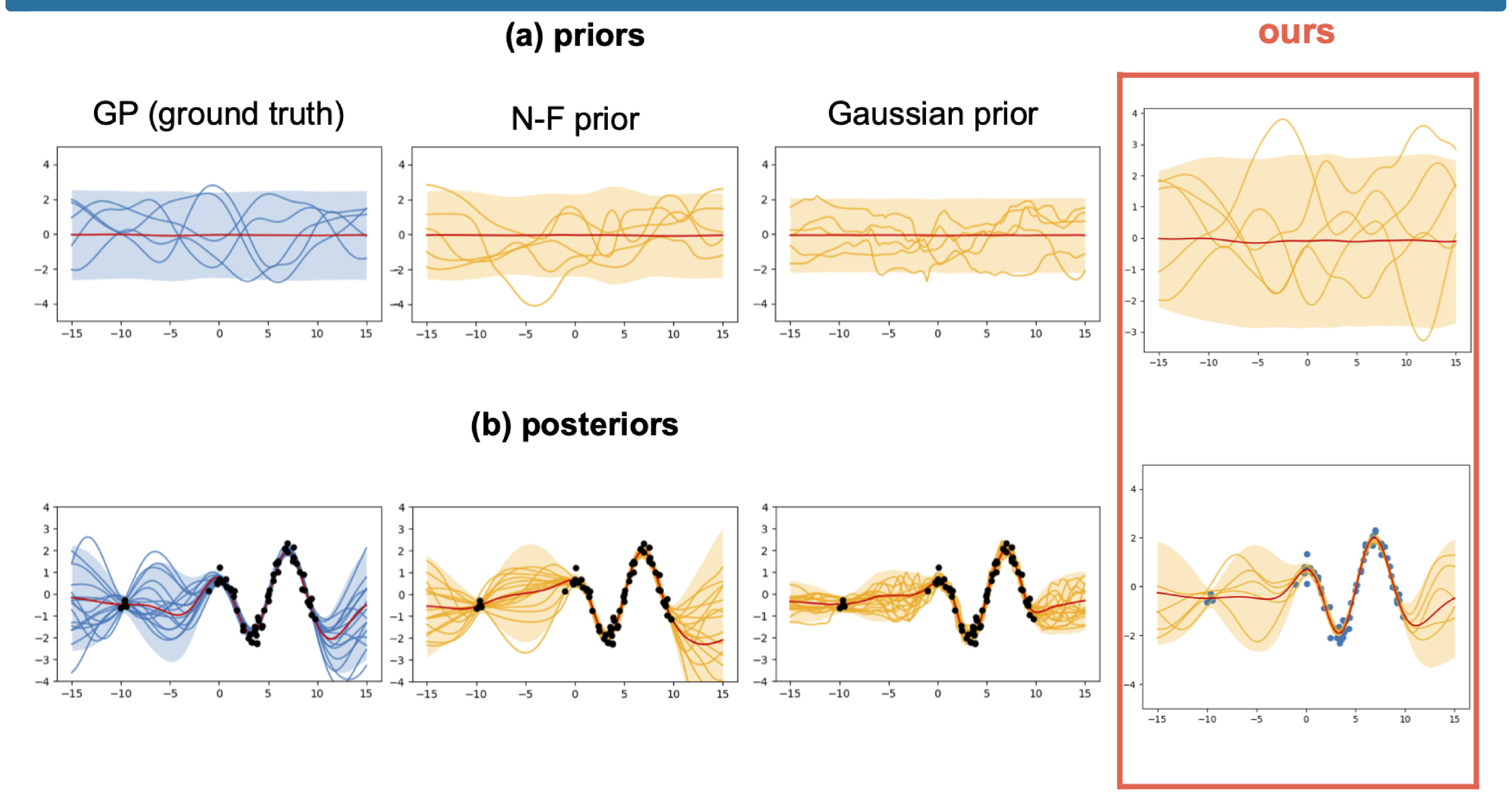
- **Easy**: BNN $\rightarrow$ GP (find κ_f^l given f^l)
- **Problem: GP $\rightarrow$ BNN (identify f^l given κ_f^l)**

Enforcing function-space GP Priors in BNNs

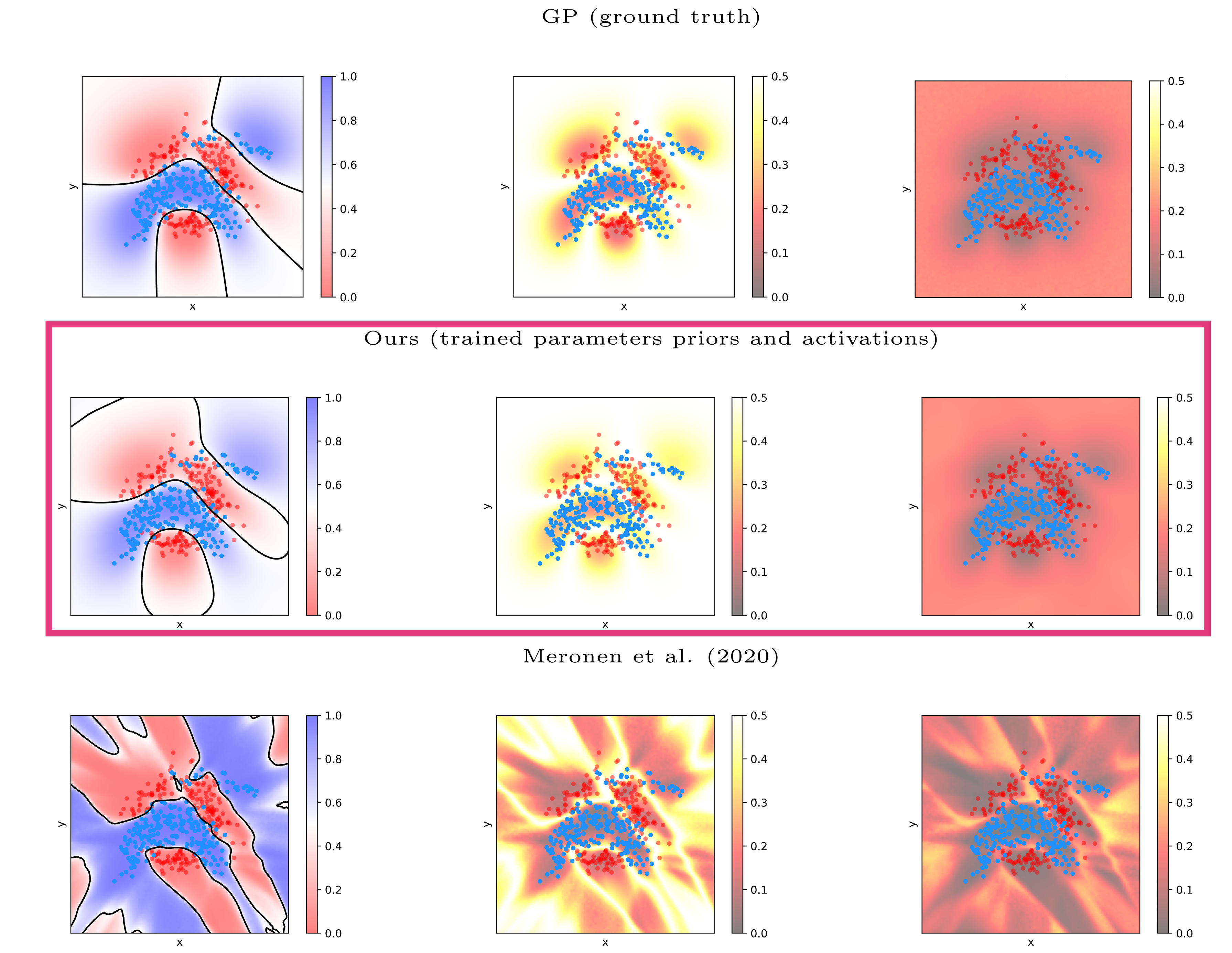
In three simple steps:

1. Reparameterize priors and activation:
$p(w, b|\lambda) = N(w|0, diag(\sigma_w))N(b|0, diag(\sigma_b))$, $\phi(\cdot|\eta)$
2. Optimize a distributional divergence between a GP and a BNN:
$\lambda^* = \text{argmin}_\lambda \frac{1}{S}\Sigma_{X \sim p_X} D(p_{nn}(f^l(X|\lambda)), p_{gp}(f^l(X)))$, where $\lambda = \{\sigma_w, \sigma_b, \eta\}$
3. Use 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

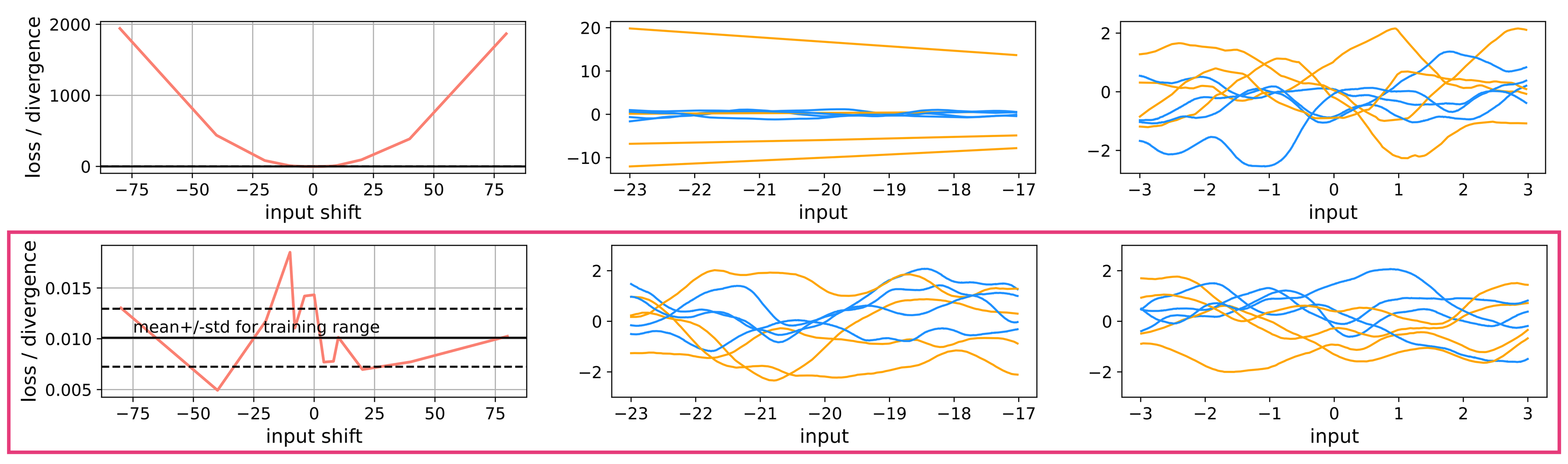


Classification



Stationarity outside training range

By default the BNNs are not preserving the stationarity of GPs.
However: appropriate activation (periodic) $\rightarrow$ stationarity!



$$\phi(x|\eta) = \Sigma_{i=1}^K A_i \cos(2\pi\psi_i x) + \Sigma_{j=1}^K A_j \sin(2\pi\psi_j x), \text{ where } \eta = \{\psi_i, A_i, \psi_j, A_j\}$$

Takeaway

- Function-space priors improve BNNs
- Learnable activations enable better fits
- Inductive biases (e.g. stationarity) can be enforced by tailored activations
- Model selection is enabled by conditioning activations
- Optimal transport with closed-form 2-W divergence helps optimization