PROJECTED BAYESIAN NEURAL NETWORKS
Avoiding Weight-space Pathologies via Latent Representations Learning

M. F. Pradier [1] W. Pan [1] J. Yao [1] S. Ghosh [2] F. Doshi-Velez [1]

[1]Harvard University [2]IBM Research

INTRODUCTION

Goal: Provide uncertainties for predictions of deep models.

Challenge: Characterizing uncertainty over parameters of modern neural networks in a Bayesian setting is difficult due to the ***high-dimensionality of the weight space*** and the ***complex patterns of dependencies*** among the weights.

Contribution: We propose a Bayesian neural network model, ***ProjBNN*** that encodes the uncertainty in the weights of a neural network via a low dimensional latent space as well as a framework for performing high-quality inference on this model.

LATENT PROJECTION BNN: MODEL

We posit that the neural network weights $\boldsymbol{w}$ are generated from a latent space or *manifold* of much smaller dimensionality. That is, we assume the following generative model:

$$\boldsymbol{z} \sim p(\boldsymbol{z}), \quad \boldsymbol{\phi} \sim p(\boldsymbol{\phi}), \quad \boldsymbol{w} = g_{\boldsymbol{\phi}}(\boldsymbol{z}), \quad \mathbf{y} \sim \mathcal{N}(f_{\boldsymbol{w}}(\mathbf{x}), \sigma_y^2) \tag{1}$$

where $\boldsymbol{w}$ lies in $\mathbb{R}^{D_w}$, the latent representation $\boldsymbol{z}$ lie in a lower dimensional space $\mathbb{R}^{D_z}$, and $\boldsymbol{\phi}$ parametrizes the arbitrary projection function $g_{\boldsymbol{\phi}} : \mathbb{R}^{D_z} \to \mathbb{R}^{D_w}$.

LATENT PROJECTION BNN: INFERENCE

Goal: Approximate posterior $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$.

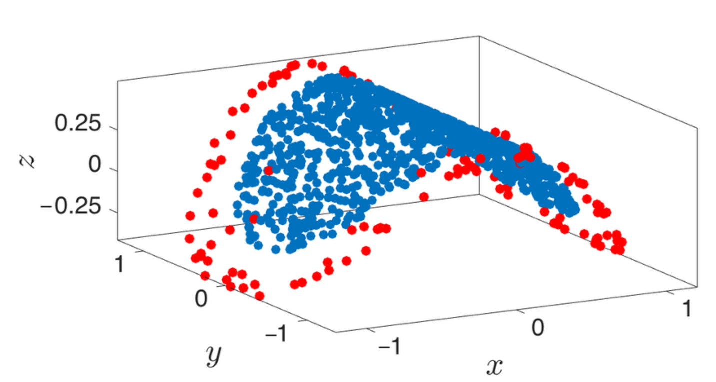
Variational distribution: We propose a variational distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) = q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}) q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi})$ such that:

$$\boldsymbol{z} \sim q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}),\ \boldsymbol{\phi} \sim q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi}),\ \boldsymbol{w} = g_{\boldsymbol{\phi}}(\boldsymbol{z}). \tag{2}$$

We use a mean-field approximation for each independent term.

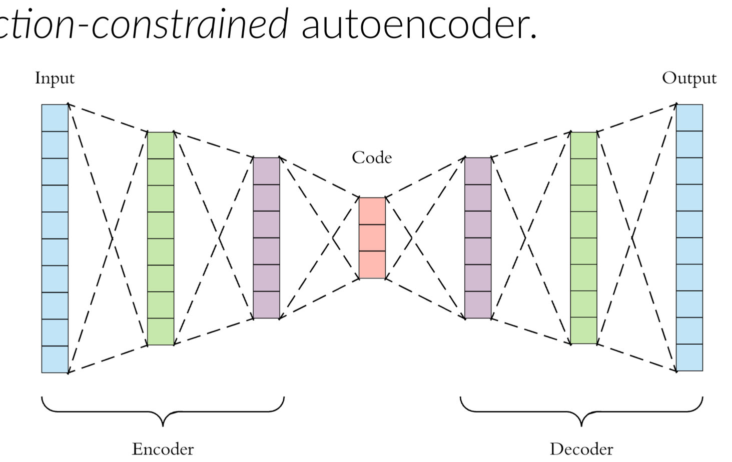
Inference Framework: We perform inference in three stages:

1. **Characterize the space of plausible weights.** Gather multiple sets of weights $\{\mathbf{w_c}^{(r)}\}_{r=1}^R$ by training an ensemble of R neural networks over random restarts.



2. **Learn a point-estimate for the projection function.** Train an autoencoder $h_{\boldsymbol{\theta},\boldsymbol{\phi}}$ using $\{\mathbf{w_c}^{(r)}\}_{r=1}^R$ as input data to:
   - minimize reconstruction loss
   - maximize predictiveness of the model $f_{\mathbf{w_c}^{(r)}}$

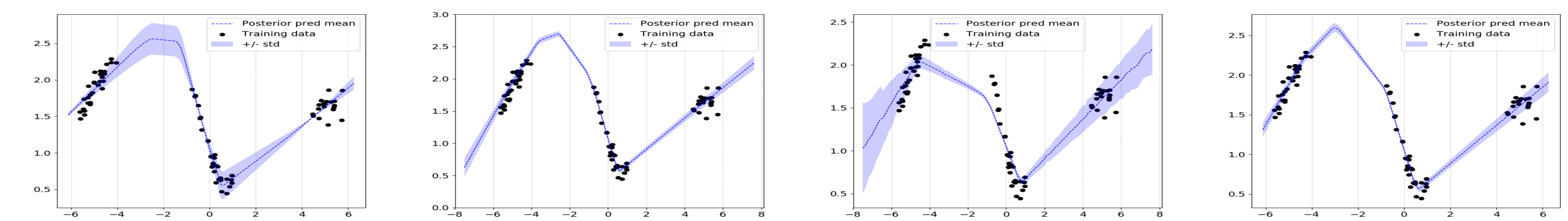
   We call this model a *prediction-constrained* autoencoder.



3. **Learn the approximate posterior** $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$. Perform BBVI in latent space to learn an approximate posterior distribution over latent representations $\boldsymbol{z}$ and projection parameters $\boldsymbol{\phi}$.

RESULTS: SYNTHETIC DATA

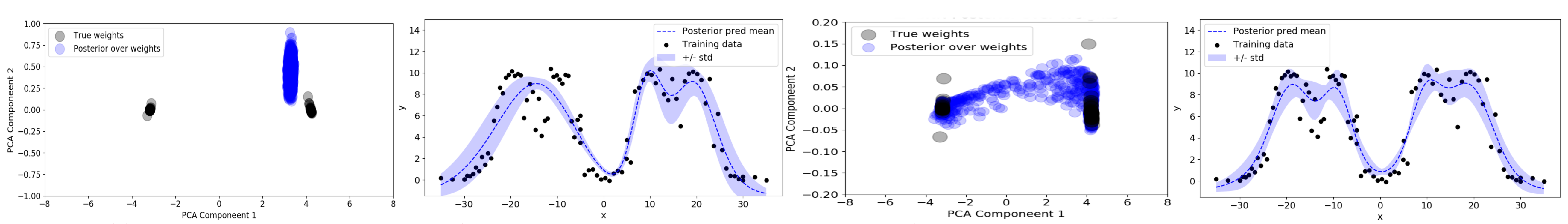
Take-away 1: Inference in latent space can provide better estimates of posterior predictive uncertainty.



(a) Proj-BNN ($D_z = 2$) (b) BBB (c) MNF (d) MVG

Figure: Inferred predictive posterior distribution for a toy data set drawn from a NN with 1-hidden layer, 20 hidden nodes and RBF activation functions. **LP-BNN is able to learn a plausible predictive mean and better capture predictive uncertainties.**

Take-away 2: Inference in latent space can improve posterior predictive quality by capturing complex geometries of the weight posterior.



(a) BNN (BBB) Posterior (b) BNN (BBB) Posterior Predictive (c) LP-BNN Posterior (d) LP-BNN Posterior Predictive

Figure: (a) shows the variational posterior over weights, $\boldsymbol{w}$, obtained by transforming the variational posterior over $\boldsymbol{z}$. Learning a variational posterior over $\boldsymbol{z}$ captures both modes in the weight space. (c) shows the variational posterior over weights learned by performing inference directly on $\boldsymbol{w}$, using Bayes by Back Prop (BBB). This posterior captures only one mode in the weight space. (b) shows the posterior predictive corresponding to the variational posterior over $\boldsymbol{z}$. The mean of the posterior predictive demonstrates four modes in the data. (d) shows the posterior predictive corresponding to the variational posterior over $\boldsymbol{w}$ using BBB. The mean of the posterior predictive demonstrates only three modes.

RESULTS: REAL DATA

Take-away 3: Inference in latent space can improve model generalization.

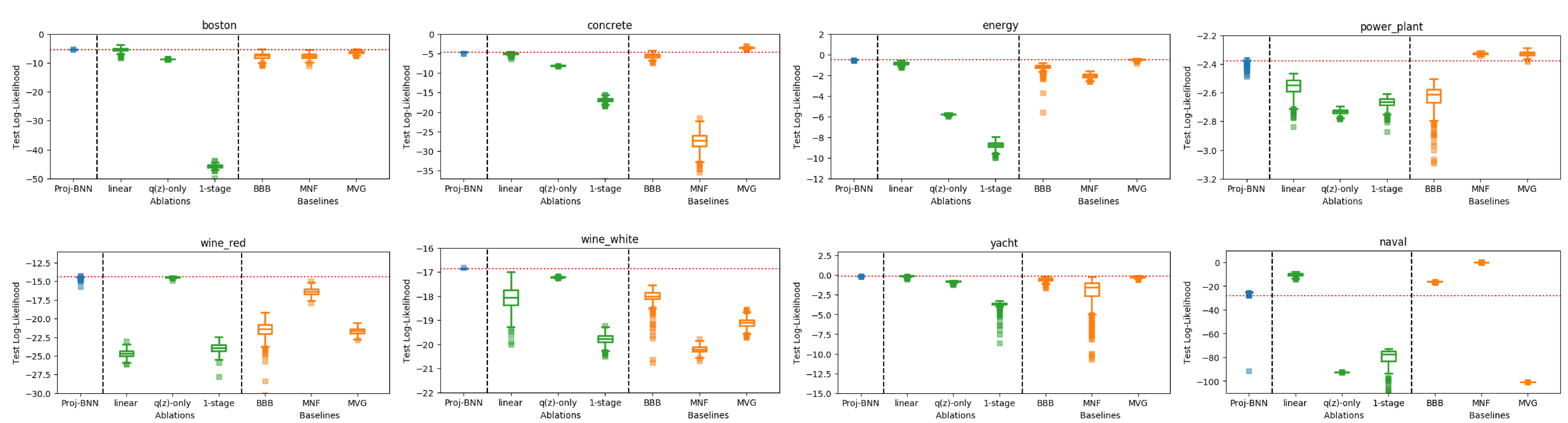


Figure: Test log-likelihood for UCI benchmark datasets for best dimensionality of z-space. Red dotted horizontal line corresponds to LP-BNN performance (our approach). Baselines methods are: 1) BBB: mean field (Blundell, et.al 2015); 2) MNF: multiplicative normalizing flow (Louizos et.al, 2017); 3) MVG: multivariate Gaussian prior BNN (Louizos et.al, 2016). Variants of LP-BNN are: LP-BNN, LP-BNN with linear projections (linear), LP-BNN without training the autoencoder, i.e., only stage 3 in inference framework (1-stage), LP-BNN modeling uncertainty only in $\boldsymbol{z}$ ($q(\boldsymbol{z})$-only). **In all but two cases LP-BNN performs better or as well as the benchmarks.**

RELATED WORK

- Nearly all other approaches perform inference directly on the weight space, for example (Sun et.al, 2017; Louizos et.al, 2017; Gal et.al, 2016) or works are based on hypernetworks, neural networks that outputs parameters of other networks (Krueger et.al, 2017; Pawlowski et.al, 2017). Instead, we perform **inference in a latent space of lower dimensionality**.
- (Louizos et.al, 2017) linearly project BNN weights layer-wise onto a latent space, on which they define a complex approximate posterior distribution via normalizing flows. Our approach learns a ***non-linear* projection of the *entire* network** onto a latent space, optimizing a tighter bound on the log evidence.
- We incorporate this **uncertainty explicitly in both our generative and variational models**. In this spirit, (Karaletsos, et.al. 2018) represents nodes in a neural network by latent variables via a *deterministic linear* projection, and drawing the weights conditioned on those representations.

DISCUSSION

- How to make it more scalable?
- How can we exploit information in latent space for meta-learning?
- Full arxiv version: `https://arxiv.org/abs/1811.07006`

ACKNOWLEDGEMENTS

We thank the Harvard Data Science Initiative, the Center for Research on Computation and Society, the Institute of Applied Computational Sciences, and IBM research.

REFERENCES

[1] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575--2583, 2015.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[3] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.