Projected BNNs: Avoiding weight-space pathologies by projecting neural network weights

Melanie F. Pradier[1], **Weiwei Pan**[1], **Jiayu Yao**[1], **Soumya Ghosh**[2], **and Finale Doshi-Velez**[1]

1. School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138

2. Health Analytics Research Group
IBM Research
Cambridge, MA 02142

Abstract

While modern neural networks are making remarkable gains in terms of predictive accuracy, characterizing uncertainty over the parameters of these models (in a Bayesian setting) is challenging because of the high-dimensionality of the network parameter space and the correlations between these parameters. We introduce a novel inference framework that encodes complex distributions in high-dimensional parameter space with representations in a low-dimensional latent space. We show improvements in uncertainty characterization and generalization across a large array of synthetic and real-world datasets.

1 Introduction

Characterizing uncertainty over parameters of modern neural networks in a Bayesian setting is challenging due to the high-dimensionality of the weight space and complex patterns of dependencies among the weights. A recent body of work has attempted to improve the quality of inference for Bayesian neural networks via improved approximate inference methods [Graves, 2011, Blundell et al., 2015, Hernández-Lobato et al., 2016], or by improving the flexibility of the variational approximation for variational inference [Gershman et al., 2012, Ranganath et al., 2016, Louizos and Welling, 2017].

In this work, we introduce a novel approach for Bayesian neural network (BNN) inference in which we remove potential redundancies in neural network parameters by learning a non-linear projection of the weights onto a low-dimensional latent space. Our approach takes advantage of the following insight: learning (standard network) *parameters* is easier in the high-dimensional space, but characterizing (Bayesian) *uncertainty* is easier in the low-dimensional space. Low-dimensional spaces are generally easier to explore, especially if we have fewer correlations between dimensions, and can be better captured by standard variational approximations (e.g. mean field). At the same time, the non-linear transformation between latent space and weight space allows us to encode flexible approximating distributions for posteriors over weights.

We demonstrate, on synthetic datasets, the ability of our model to capture complex posterior distributions over weights by encoding them as distributions in latent space. We show that, as a result, our model is able to more accurately capture the uncertainty in the posterior predictive distribution. Finally, we demonstrate that across a wide range of real-world data sets, our approach produces accurate predictions on held-out data with highly compressed latent representation of weights.

2 Related Work

Recent works have attempted to move beyond fully-factorized approximations of Bayesian neural network posteriors by modeling structured correlations amongst BNN weights. While nearly all of these approaches perform inference directly on the weight space [Sun et al., 2017, Louizos

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

and Welling, 2016, Gal and Ghahramani, 2016], we perform inference in a latent space of lower dimensionality. Most comparable to our approach are works that build flexible approximating families of distributions by mixtures of simple distributions [Agakov and Barber, 2004, Maaløe et al., 2016, Ranganath et al., 2016, Salimans et al., 2013] or by non-linear transformations of simple distributionsn [Rezende and Mohamed, 2015, Kingma et al., 2016, Louizos and Welling, 2017].

In particular, 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. In contrast, rather than projecting the *nodes*, we find a latent representation for the *weights* directly. In terms of inference, we propose a multi-stage framework to enable learning both a non-linear projection and latent representations with uncertainties. On the other hand, Louizos and Welling [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. Furthermore, our generative model differs from the one in Louizos and Welling [2017] allowing us to optimize a tighter bound on the log evidence.

Finally, a number of recent works use the idea of hypernetworks, neural networks that outputs parameters of other networks, to represent neural network weights in a latent space [Krueger et al., 2017, Pawlowski et al., 2017]. However, these approaches again perform inference directly over weights, requiring one to use invertible latent projections or otherwise approximate implicit weight densities. Our approach of performing inference in the latent space avoids these challenges during inference. Furthermore, while uncertainty over the latent projection is not considered by Krueger et al. [2017], Pawlowski et al. [2017], we incorporate this uncertainty explicitly in both our generative and variational models.

3 Projected Bayesian Neural Network

Let $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots (\mathbf{x}_N, \mathbf{y}_N)\}$ be a dataset of N i.i.d observed input-output pairs. We model this data by $\mathbf{y} = f_{\boldsymbol{w}}(\mathbf{x}) + \epsilon$, where ϵ is a noise variable and $\boldsymbol{w}$ is the weights of a neural network. In the Bayesian setting, we often assume some prior over the weights $\boldsymbol{w} \sim p(\boldsymbol{w})$ and aim to infer the posterior distribution over weights $p(\boldsymbol{w}|\mathcal{D})$.

Generative Model In our approach, we posit that the neural network weights $\boldsymbol{w}$ lie in 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 $\boldsymbol{\phi}$ parametrizes the function $g_{\boldsymbol{\phi}}$. The posterior distributions $p(z|\mathcal{D})$, $p(\boldsymbol{w}|\mathcal{D})$ typically will not have analytic forms, due to the non-linearity of $g_{\boldsymbol{\phi}}$ and $f_{\boldsymbol{w}}$. Thus, we perform approximate them using variational inference. We refer to this model as *Projected BNN* (Proj-BNN).

Inference We propose the following variational model: $\boldsymbol{z} \sim q_{\boldsymbol{\lambda}_z}(\boldsymbol{z}),\ \boldsymbol{\phi} \sim q_{\boldsymbol{\lambda}_\phi}(\boldsymbol{\phi}),\ W = g_{\boldsymbol{\phi}}(\boldsymbol{z})$; and aim to learn a distribution over both the latent projection $g_{\boldsymbol{\phi}}$ and latent representation $\boldsymbol{z}$. We perform this inference in three distinct steps:

1. Characterize weight space. We collect multiple high-quality candidate solutions, $\{\mathbf{w_c}\}$, by training an ensemble of R neural networks over multiple restarts.

2. Learn projection into latent space. We train an autoencoder on $\{\mathbf{w_c}\}$ to learn a non-linear map between the space of BNN weights and the latent space. While we want to find latent projections that minimizes the reconstruction error of the weights, at the same time, we also encourage for projections that map into weights that yield high log likelihood values for our data. Denoting the autoencoder by $h_{\boldsymbol{\theta},\boldsymbol{\phi}}$, where $\boldsymbol{\theta}$, $\boldsymbol{\phi}$ parametrize the encoder and decoder respectively, our objective is to minimize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ over $\boldsymbol{\theta}, \boldsymbol{\phi}$:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{R}\sum_{r=1}^{R}\left(\mathbf{w_c}^{(r)} - h_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\mathbf{w_c}^{(r)}\right) + \gamma^{(r)}\right)^2 + \beta\,\mathbb{E}_{\mathcal{D}}\left[\frac{1}{R}\sum_{r=1}^{R}\log p\big(y|x, h_{\boldsymbol{\theta},\boldsymbol{\phi}}\left(\mathbf{w_c}^{(r)}\right)\right], \tag{2}$$

where $\mathbf{w}_{\boldsymbol{c}}^{(r)}$ denotes the set of weights gathered from the r-th network in the ensemble of R neural networks, and $\gamma^{(r)} \sim \mathcal{N}(0, 1)$ is an additional input noise term that makes training more robust. where γ refer to Gaussian noise to make We call this objective *prediction-constrained*.

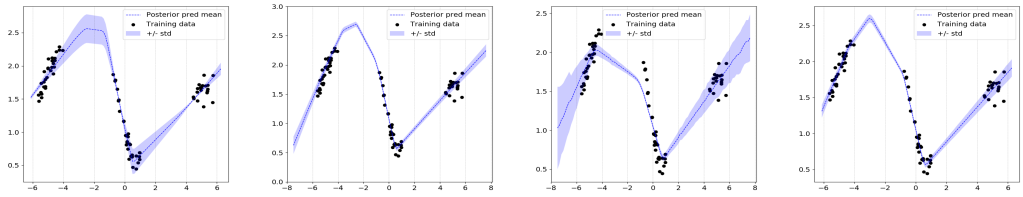
3. Infer posterior over z and ϕ. We optimize the variational parameters $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_z, \boldsymbol{\lambda}_\phi\}$ to minimize the KL-divergence $D_{\mathrm{KL}}\big(q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi}) || p(\boldsymbol{z}, \boldsymbol{\phi} | \mathbf{y}, \mathbf{x})\big)$. To facilitate this task, we first optimize the variational distribution in latent space $q_{\boldsymbol{\lambda}_z}(\boldsymbol{z})$ (fixing $\boldsymbol{\phi}$) via black-box variational inference (BBVI) [Ranganath et al., 2014] with the local reparametrization trick [Kingma et al., 2015], and then jointly fine-tune the uncertainty in both $\boldsymbol{z}$ and $\boldsymbol{\phi}$ via the distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{z}, \boldsymbol{\phi})$.

4 Results

Inference in latent space can provide better estimates of posterior predictive uncertainty (Synthetic data). In Figure 1, we compare the posterior predictive distributions on synthetic data obtained by Proj-BNN against those from three benchmark models: Bayes by Back Prop (BBB) [Blundell et al., 2015], Multiplicative Normalizing Flow (MNF) [Louizos and Welling, 2017], and Matrix Variate Gaussian Posteriors (MVG) [Louizos and Welling, 2016]. We observe that our method is able to obtain a mean posterior predictive that fits the data and is furthermore able to better capture complex patterns of uncertainty in the posterior predictive. For example, benchmark methods tend to underestimate predictive uncertainty, especially in places with few observations.

Inference in latent space can improve posterior predictive quality by capturing complex geometries of the weight posterior (Synthetic data). We argue that the reason for the observed improvement in the quality of posterior predictive obtained by our model is often due to the fact that it is difficult to approximate complex geometry of the solution set in the weight space. Mapping the solution set onto a simpler region in a lower dimensional latent space allows for more efficient approximations. In Figure 2 (Appendix A), we study a regression task where the solution set in the weight space for this model is naturally bimodal. Figure 3 (Appendix A) shows that direct variational approximation of the posterior over weights is only able to capture one of the modes in the solution set, while approximating the posterior over $\boldsymbol{z}$'s using our model captures both modes.

Inference in latent space can improve model generalization (Realworld data). On datasets where ground truth distributions are not available for comparison and the inferred distributions are not easily visualized, we argue that the higher quality posterior and posterior predictive potentially obtained from Proj-BNN can be observed through an improvement in the ability of our model to generalize. We compare the generalization performance, measured in terms of test likelihood, of Proj-BNN with the three benchmark models, BBB, MNF and MVG. In Figure 4 (Appendix A), we see that Proj-BNN performs competitively, if not better than benchmark models, on all but one dataset, with latent dimensions that are much smaller than the dimensions of the weight spaces.



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

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

5 Conclusion

We have presented a novel framework, the proj-BNN, for performing approximate inference for Bayesian Neural Networks that can avoid many of the optimization problems of traditional inference methods when the posterior over weights exhibits complex geometry. Empirically, the proj-BNN performs competitively if not better than current baselines on predictive likelihood, while working in latent dimensions that are much smaller than the dimensions of the weight space.

Further gains may be attained by improving each phase of our framework. One might implement sample efficient learning of the weight space [Zhang et al., 2015]; regularize the latent representation learned by the autoencoder to have desirable geometries [Hosseini-Asl et al., 2016]; or place complex priors on the latent space [Gershman et al., 2012].

References

Felix V Agakov and David Barber. An auxiliary variational method. In *International Conference on Neural Information Processing*, pages 561–566, 2004.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

Samuel Gershman, Matt Hoffman, and David Blei. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*, 2012.

Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang Bui, and Richard Eric Turner. Black-box α-divergence minimization. 2016.

Ehsan Hosseini-Asl, Jacek M Zurada, and Olfa Nasraoui. Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints. *IEEE transactions on neural networks and learning systems*, 27(12):2486–2498, 2016.

Theofanis Karaletsos, Peter Dayan, and Zoubin Ghahramani. Probabilistic Meta-Representations Of Neural Networks. October 2018. arXiv: 1810.00555.

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.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

David Krueger, Chin-Wei Huang, Riashat Islam, Ryan Turner, Alexandre Lacoste, and Aaron Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016.

Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. *arXiv preprint arXiv:1703.01961*, 2017.

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

Nick Pawlowski, Martin Rajchl, and Ben Glocker. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297*, 2017.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

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

Tim Salimans, David A Knowles, et al. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Shengyang Sun, Changyou Chen, and Lawrence Carin. Learning structured weight uncertainty in bayesian neural networks. In *Artificial Intelligence and Statistics*, pages 1283–1292, 2017.

Xiaohui Zhang, Daniel Povey, and Sanjeev Khudanpur. A diversity-penalizing ensemble training method for deep learning. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

A Figures

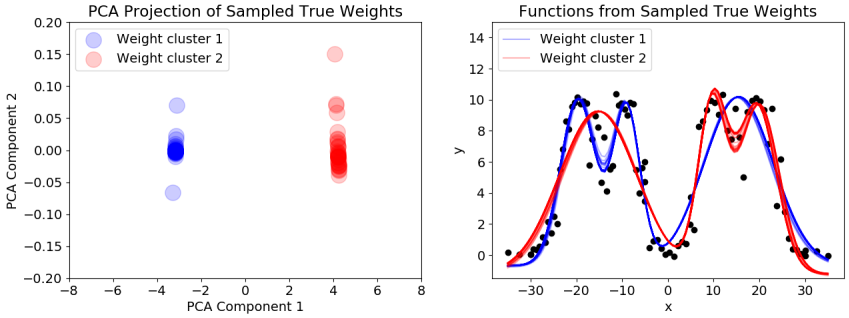
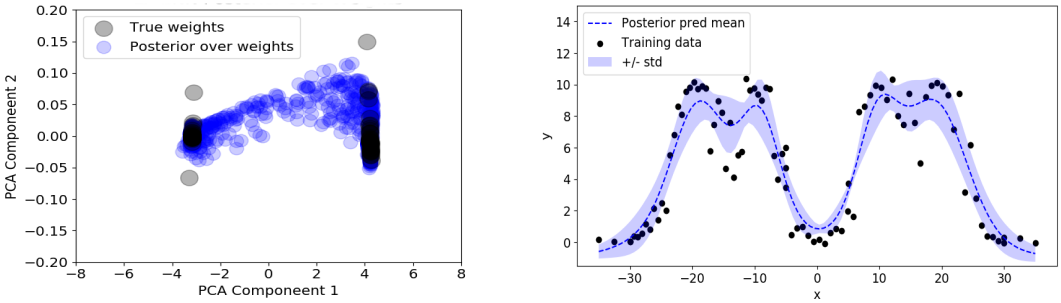
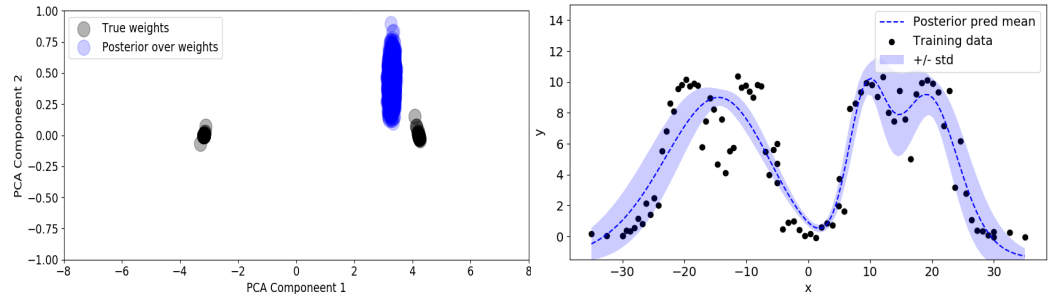


Figure 2: Toy data for regression is generated from a function with four modes. **(A)** visualizes the weights of feedforward network, with a single hidden layer with three nodes, obtained by fitting the data multiple times from random weight initializations. The distribution over 'good' weights is bimodal. **(B)** shows examples of functions corresponding to weights sampled from each weight cluster. Each cluster corresponds to fitting a different set of three of the four modes in the data.



(a) Proj-BNN posterior over weights

(b) Proj-BNN posterior predictive



(c) BNN posterior over weights (BBB)

(d) BNN posterior predictive (BBB)

Figure 3: **(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 $\boldsymbol{w}$. 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 $\boldsymbol{w}$. The mean of the posterior predictive demonstrates only three modes.

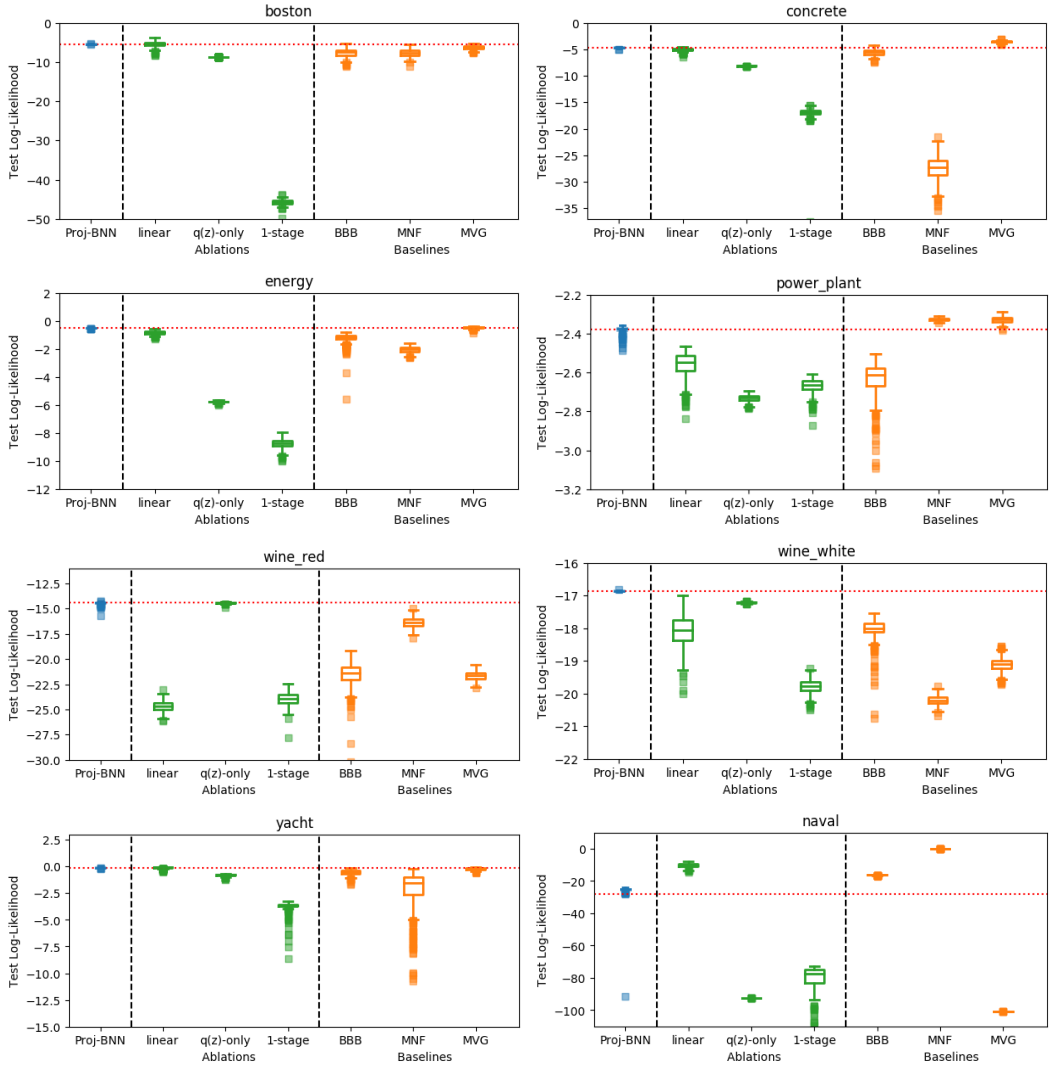


Figure 4: Test log-likelihood for UCI benchmark datasets for best dimensionality of z-space (see Figure 5 for performance across different dimensionality of the latent space D_z). Baselines methods are: 1) BBB: mean field (Blundell, et.al 2015); 2) MNF: multiplicative normalizing flow (Louizos et.al, 2017); 3) MVG: multivariate Gaussian prior BNN (Louizos et.al, 2016). Variants of Proj-BNN are: Proj-BNN, Proj-BNN with linear projections (linear), Proj-BNN without training the autoencoder (1-stage), Proj-BNN modeling uncertainty only in $\boldsymbol{z}$ ($q(z)$-only). **In all but two cases Proj-BNN performs better or as well as the benchmarks working with latent representations of much lower dimension.**

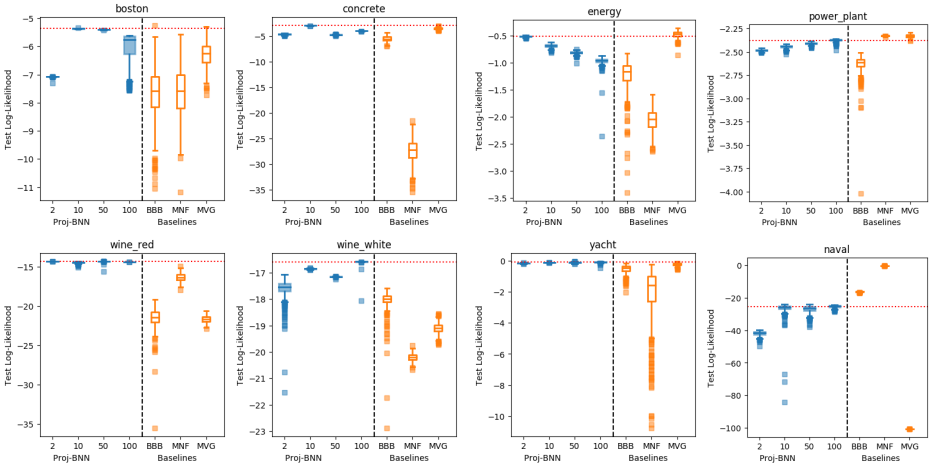


Figure 5: Test loglikelihood for UCI benchmark datasets for varying dimensionality of $\boldsymbol{z}$-space. Baselines: a) BBB: mean field (Blundell, et.al 2015); b) MNF: multiplicative normalizing flow (Louizos et.al, 2017); c) MVG: multivariate Gaussian prior BNN (Louizos et.al, 2016).